Cisco IOS Security Configuration Guide

Release 12.2SX

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Cisco IOS Security Configuration Guide

About Cisco IOS and Cisco IOS XE Software Documentation

Last updated: August 6, 2008

This document describes the objectives, audience, conventions, and organization used in Cisco IOS and Cisco IOS XE software documentation, collectively referred to in this document as Cisco IOS documentation. Also included are resources for obtaining technical assistance, additional documentation, and other information from Cisco. This document is organized into the following sections:

Documentation Objectives

Cisco IOS documentation describes the tasks and commands available to configure and maintain Cisco networking devices.

Audience

The Cisco IOS documentation set is i ntended for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the configuration and maintenance tasks, the relationship among tasks, or the Cisco IOS commands necessary to perform particular tasks. The Cisco IOS documentation set is also intended for those users experienced with Cisco IOS who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS release.

Documentation Conventions

In Cisco IOS documentation, the term *router* may be used to refer to various Cisco products; for example, routers, access servers, and switches. These and other networking devices that support Cisco IOS software are shown interchangeably in examples and are used only for illustrative purposes. An example that shows one product does not necessarily mean that other products are not supported.

This section includes the following topics:

Typographic Conventions

Cisco IOS documentation uses the following typographic conventions:

Convention	Description
^ or Ctrl	Both the ^ symbol and Ctrl represent the Control (Ctrl) key on a keyboard. For example, the key combination **^D** or **Ctrl-D** means that you hold down the Control key while you press the D key. (Keys are indicated in capital letters but are not case sensitive.)
string	A string is a nonquoted set of characters shown in italics. For example, when setting a Simple Network Management Protocol (SNMP) community string to *public*, do not use quotation marks around the string; otherwise, the string will include the quotation marks.

Command Syntax Conventions

Cisco IOS documentation uses the following command syntax conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter as shown.
italic	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional keyword or argument.
|	A vertical line, called a pipe, indicates a choice within a set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a pipe indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a pipe indicate a required choice.
[x {y | z}]	Braces and a pipe within square brackets indicate a required choice within an optional element.

Software Conventions

Cisco IOS uses the following program code conventions:

Convention	Description
`Courier font`	Courier font is used for information that is displayed on a PC or terminal screen.
`Bold Courier font`	Bold Courier font indicates text that the user must enter.
< >	Angle brackets enclose text that is not displayed, such as a password. Angle brackets also are used in contexts in which the italic font style is not supported; for example, ASCII text.
!	An exclamation point at the beginning of a line indicates that the text that follows is a comment, not a line of code. An exclamation point is also displayed by Cisco IOS software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Reader Alert Conventions

The Cisco IOS documentation set uses the following conventions for reader alerts:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Documentation Organization

This section describes the Cisco IOS documentation set, how it is organized, and how to access it on Cisco.com. Included are lists of configuration guides, command references, and supplementary references and resources that make up the documentation set. The following topics are included:

Cisco IOS Documentation Set

Cisco IOS documentation consists of the following:

- Release notes and caveats provide information about platform, technology, and feature support for a release and describe severity 1 (catastrophic), severity 2 (severe), and severity 3 (moderate) defects in released Cisco IOS code. Review release notes before other documents to learn whether or not updates have been made to a feature.
- Sets of configuration guides and command references organized by technology and published for each standard Cisco IOS release.
  - Configuration guides—Compilations of documents that provide informational and task-oriented descriptions of Cisco IOS features.
  - Command references—Compilations of command pages that provide detailed information about the commands used in the Cisco IOS features and processes that make up the related configuration guides. For each technology, there is a single command reference that covers all Cisco IOS releases and that is updated at each standard release.
- Lists of all the commands in a specific release and all commands that are new, modified, removed, or replaced in the release.
- Command reference book for **debug** commands. Command pages are listed in alphabetical order.
- Reference book for system messages for all Cisco IOS releases.

Cisco IOS Documentation on Cisco.com

The following sections describe the documentation organization and how to access various document types.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

New Features List

The New Features List for each release provides a list of all features in the release with hyperlinks to the feature guides in which they are documented.

Feature Guides

Cisco IOS features are documented in feature guides. Feature guides describe one feature or a group of related features that are supported on many different software releases and platforms. Your Cisco IOS software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release.

Configuration Guides

Configuration guides are provided by technology and release and comprise a set of individual feature guides relevant to the release and technology.

Command References

Command reference books describe Cisco IOS commands that are supported in many different software releases and on many different platforms. The books are provided by technology. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

Cisco IOS Supplementary Documents and Resources

Supplementary documents and resources are listed in Table 2 on page xi.

Configuration Guides, Command References, and Supplementary Resources

Table 1 lists, in alphabetical order, Cisco IOS and Cisco IOS XE software configuration guides and command references, including brief descriptions of the contents of the documents. The Cisco IOS command references are comprehensive, meaning that they include commands for both Cisco IOS software and Cisco IOS XE software, for all releases. The configuration guides and command references support many different software releases and platforms. Your Cisco IOS software release or platform may not support all these technologies.

For additional information about configuring and operating specific networking devices, go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

Table 2 lists documents and resources that supplement the Cisco IOS software configuration guides and command references. These supplementary resources include release notes and caveats; master command lists; new, modified, removed, and replaced command lists; system messages; and the debug command reference.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS AppleTalk Configuration Guide *Cisco IOS XE AppleTalk Configuration Guide* *Cisco IOS AppleTalk Command Reference*	AppleTalk protocol.
Cisco IOS Asynchronous Transfer Mode Configuration Guide *Cisco IOS Asynchronous Transfer Mode Command Reference*	LAN ATM, multiprotocol over ATM (MPoA), and WAN ATM.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Bridging and IBM Networking Configuration Guide *Cisco IOS Bridging Command Reference* *Cisco IOS IBM Networking Command Reference*	• Transparent and source-route transparent (SRT) bridging, source-route bridging (SRB), Token Ring Inter-Switch Link (TRISL), and token ring route switch module (TRRSM). • Data-link switching plus (DLSw+), serial tunnel (STUN), block serial tunnel (BSTUN); logical link control, type 2 (LLC2), synchronous data link control (SDLC); IBM Network Media Translation, including Synchronous Data Logical Link Control (SDLLC) and qualified LLC (QLLC); downstream physical unit (DSPU), Systems Network Architecture (SNA) service point, SNA frame relay access, advanced peer-to-peer networking (APPN), native client interface architecture (NCIA) client/server topologies, and IBM Channel Attach.
Cisco IOS Broadband and DSL Configuration Guide *Cisco IOS XE Broadband and DSL Configuration Guide* *Cisco IOS Broadband and DSL Command Reference*	Point-to-Point Protocol (PPP) over ATM (PPPoA) and PPP over Ethernet (PPPoE).
Cisco IOS Carrier Ethernet Configuration Guide *Cisco IOS Carrier Ethernet Command Reference*	Connectivity fault management (CFM), Ethernet Local Management Interface (ELMI), IEEE 802.3ad link bundling, Link Layer Discovery Protocol (LLDP), media endpoint discovery (MED), and operations, administration, and maintenance (OAM).
Cisco IOS Configuration Fundamentals Configuration Guide *Cisco IOS XE Configuration Fundamentals Configuration Guide* *Cisco IOS Configuration Fundamentals Command Reference*	Autoinstall, Setup, Cisco IOS command-line interface (CLI), Cisco IOS file system (IFS), Cisco IOS web browser user interface (UI), basic file transfer services, and file management.
Cisco IOS DECnet Configuration Guide *Cisco IOS XE DECnet Configuration Guide* *Cisco IOS DECnet Command Reference*	DECnet protocol.
Cisco IOS Dial Technologies Configuration Guide *Cisco IOS XE Dial Technologies Configuration Guide* *Cisco IOS Dial Technologies Command Reference*	Asynchronous communications, dial backup, dialer technology, dial-in terminal services and AppleTalk remote access (ARA), large scale dialout, dial-on-demand routing, dialout, modem and resource pooling, ISDN, multilink PPP (MLP), PPP, virtual private dialup network (VPDN).
Cisco IOS Flexible NetFlow Configuration Guide *Cisco IOS Flexible NetFlow Command Reference*	Flexible NetFlow.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS H.323 Configuration Guide	Gatekeeper enhancements for managed voice services, Gatekeeper Transaction Message Protocol, gateway codec order preservation and shutdown control, H.323 dual tone multifrequency relay, H.323 version 2 enhancements, Network Address Translation (NAT) support of H.323 v2 Registration, Admission, and Status (RAS) protocol, tokenless call authorization, and VoIP gateway trunk and carrier-based routing.
Cisco IOS High Availability Configuration Guide *Cisco IOS XE High Availability Configuration Guide* *Cisco IOS High Availability Command Reference*	A variety of High Availability (HA) features and technologies that are available for different network segments (from enterprise access to service provider core) to facilitate creation of end-to-end highly available networks. Cisco IOS HA features and technologies can be categorized in three key areas: system-level resiliency, network-level resiliency, and embedded management for resiliency.
Cisco IOS Integrated Session Border Controller Command Reference	A VoIP-enabled device that is deployed at the edge of networks. An SBC is a toolkit of functions, such as signaling interworking, network hiding, security, and quality of service (QoS).
Cisco IOS Intelligent Service Gateway Configuration Guide *Cisco IOS Intelligent Service Gateway Command Reference*	Subscriber identification, service and policy determination, session creation, session policy enforcement, session life-cycle management, accounting for access and service usage, session state monitoring.
Cisco IOS Interface and Hardware Component Configuration Guide *Cisco IOS XE Interface and Hardware Component Configuration Guide* *Cisco IOS Interface and Hardware Component Command Reference*	LAN interfaces, logical interfaces, serial interfaces, virtual interfaces, and interface configuration.
Cisco IOS IP Addressing Services Configuration Guide *Cisco IOS XE Addressing Services Configuration Guide* *Cisco IOS IP Addressing Services Command Reference*	Address Resolution Protocol (ARP), Network Address Translation (NAT), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Next Hop Address Resolution Protocol (NHRP).
Cisco IOS IP Application Services Configuration Guide *Cisco IOS XE IP Application Services Configuration Guide* *Cisco IOS IP Application Services Command Reference*	Enhanced Object Tracking (EOT), Gateway Load Balancing Protocol (GLBP), Hot Standby Router Protocol (HSRP), IP Services, Server Load Balancing (SLB), Stream Control Transmission Protocol (SCTP), TCP, Web Cache Communication Protocol (WCCP), User Datagram Protocol (UDP), and Virtual Router Redundancy Protocol (VRRP).
Cisco IOS IP Mobility Configuration Guide *Cisco IOS IP Mobility Command Reference*	Mobile ad hoc networks (MANet) and Cisco mobile networks.
Cisco IOS IP Multicast Configuration Guide *Cisco IOS XE IP Multicast Configuration Guide* *Cisco IOS IP Multicast Command Reference*	Protocol Independent Multicast (PIM) sparse mode (PIM-SM), bidirectional PIM (bidir-PIM), Source Specific Multicast (SSM), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), and Multicast VPN (MVPN).

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS IP Routing Protocols Configuration Guide *Cisco IOS XE IP Routing Protocols Configuration Guide* *Cisco IOS IP Routing Protocols Command Reference*	Border Gateway Protocol (BGP), multiprotocol BGP, multiprotocol BGP extensions for IP multicast, bidirectional forwarding detection (BFD), Enhanced Interior Gateway Routing Protocol (EIGRP), Interior Gateway Routing Protocol (IGRP), Intermediate System-to-Intermediate System (IS-IS), on-demand routing (ODR), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).
Cisco IOS IP SLAs Configuration Guide *Cisco IOS XE IP SLAs Configuration Guide* *Cisco IOS IP SLAs Command Reference*	Cisco IOS IP Service Level Agreements (IP SLAs).
Cisco IOS IP Switching Configuration Guide *Cisco IOS XE IP Switching Configuration Guide* *Cisco IOS IP Switching Command Reference*	Cisco Express Forwarding, fast switching, and Multicast Distributed Switching (MDS).
Cisco IOS IPv6 Configuration Guide *Cisco IOS XE IPv6 Configuration Guide* *Cisco IOS IPv6 Command Reference*	For IPv6 features, protocols, and technologies, go to the IPv6 "Start Here" document at the following URL: http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-roadmap.html
Cisco IOS ISO CLNS Configuration Guide *Cisco IOS XE ISO CLNS Configuration Guide* *Cisco IOS ISO CLNS Command Reference*	ISO connectionless network service (CLNS).
Cisco IOS LAN Switching Configuration Guide *Cisco IOS XE LAN Switching Configuration Guide* *Cisco IOS LAN Switching Command Reference*	VLANs, Inter-Switch Link (ISL) encapsulation, IEEE 802.10 encapsulation, IEEE 802.1Q encapsulation, and multilayer switching (MLS).
Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide *Cisco IOS Mobile Wireless Gateway GPRS Support Node Command Reference*	Cisco IOS Gateway GPRS Support Node (GGSN) in a 2.5-generation general packet radio service (GPRS) and 3-generation universal mobile telecommunication system (UMTS) network.
Cisco IOS Mobile Wireless Home Agent Configuration Guide *Cisco IOS Mobile Wireless Home Agent Command Reference*	Cisco Mobile Wireless Home Agent, an anchor point for mobile terminals for which mobile IP or proxy mobile IP services are provided.
Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide *Cisco IOS Mobile Wireless Packet Data Serving Node Command Reference*	Cisco Packet Data Serving Node (PDSN), a wireless gateway that is between the mobile infrastructure and standard IP networks and that enables packet data services in a code division multiple access (CDMA) environment.
Cisco IOS Mobile Wireless Radio Access Networking Configuration Guide *Cisco IOS Mobile Wireless Radio Access Networking Command Reference*	Cisco IOS radio access network products.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Multiprotocol Label Switching Configuration Guide *Cisco IOS XE Multiprotocol Label Switching Configuration Guide* *Cisco IOS Multiprotocol Label Switching Command Reference*	MPLS Label Distribution Protocol (LDP), MPLS Layer 2 VPNs, MPLS Layer 3 VPNs, MPLS Traffic Engineering (TE), and MPLS Embedded Management (EM) and MIBs.
Cisco IOS Multi-Topology Routing Configuration Guide *Cisco IOS Multi-Topology Routing Command Reference*	Unicast and multicast topology configurations, traffic classification, routing protocol support, and network management support.
Cisco IOS NetFlow Configuration Guide *Cisco IOS XE NetFlow Configuration Guide* *Cisco IOS NetFlow Command Reference*	Network traffic data analysis, aggregation caches, export features.
Cisco IOS Network Management Configuration Guide *Cisco IOS XE Network Management Configuration Guide* *Cisco IOS Network Management Command Reference*	Basic system management; system monitoring and logging; troubleshooting, logging, and fault management; Cisco Discovery Protocol; Cisco IOS Scripting with Tool Control Language (Tcl); Cisco networking services (CNS); DistributedDirector; Embedded Event Manager (EEM); Embedded Resource Manager (ERM); Embedded Syslog Manager (ESM); HTTP; Remote Monitoring (RMON); SNMP; and VPN Device Manager Client for Cisco IOS Software (XSM Configuration).
Cisco IOS Novell IPX Configuration Guide *Cisco IOS XE Novell IPX Configuration Guide* *Cisco IOS Novell IPX Command Reference*	Novell Internetwork Packet Exchange (IPX) protocol.
Cisco IOS Optimized Edge Routing Configuration Guide *Cisco IOS Optimized Edge Routing Command Reference*	Optimized edge routing (OER) monitoring, policy configuration, routing control, logging and reporting, and VPN IPsec/generic routing encapsulation (GRE) tunnel interface optimization.
Cisco IOS Quality of Service Solutions Configuration Guide *Cisco IOS XE Quality of Service Solutions Configuration Guide* *Cisco IOS Quality of Service Solutions Command Reference*	Class-based weighted fair queuing (CBWFQ), custom queuing, distributed traffic shaping (DTS), generic traffic shaping (GTS), IP- to-ATM class of service (CoS), low latency queuing (LLQ), modular QoS CLI (MQC), Network-Based Application Recognition (NBAR), priority queuing, Security Device Manager (SDM), Multilink PPP (MLPPP) for QoS, header compression, AutoQoS, QoS features for voice, Resource Reservation Protocol (RSVP), weighted fair queuing (WFQ), and weighted random early detection (WRED).
Cisco IOS Security Configuration Guide *Cisco IOS XE Security Configuration Guide* *Cisco IOS Security Command Reference*	Access control lists (ACLs), authentication, authorization, and accounting (AAA), firewalls, IP security and encryption, neighbor router authentication, network access security, network data encryption with router authentication, public key infrastructure (PKI), RADIUS, TACACS+, terminal access security, and traffic filters.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Service Selection Gateway Configuration Guide *Cisco IOS Service Selection Gateway Command Reference*	Subscriber authentication, service access, and accounting.
Cisco IOS Software Activation Configuration Guide *Cisco IOS Software Activation Command Reference*	An orchestrated collection of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.
Cisco IOS Software Modularity Installation and Configuration Guide *Cisco IOS Software Modularity Command Reference*	Installation and basic configuration of software modularity images, including installations on single and dual route processors, installation rollbacks, software modularity binding, software modularity processes and patches.
Cisco IOS Terminal Services Configuration Guide *Cisco IOS Terminal Services Command Reference* *Cisco IOS XE Terminal Services Command Reference*	DEC, local-area transport (LAT), and X.25 packet assembler/disassembler (PAD).
Cisco IOS Virtual Switch Command Reference	Virtual switch redundancy, high availability, and packet handling; converting between standalone and virtual switch modes; virtual switch link (VSL); Virtual Switch Link Protocol (VSLP). **Note** For information about virtual switch configuration, refer to the product-specific software configuration information for the Cisco Catalyst 6500 series switch or for the Metro Ethernet 6500 series switch.
Cisco IOS Voice Configuration Library *Cisco IOS Voice Command Reference*	Cisco IOS support for voice call control protocols, interoperability, physical and virtual interface management, and troubleshooting. The library includes documentation for IP telephony applications.
Cisco IOS VPDN Configuration Guide *Cisco IOS XE VPDN Configuration Guide* *Cisco IOS VPDN Command Reference*	Layer 2 Tunneling Protocol (L2TP) dial-out load balancing and redundancy, L2TP extended failover, L2TP security VPDN, multihop by Dialed Number Identification Service (DNIS), timer and retry enhancements for L2TP and Layer 2 Forwarding (L2F), RADIUS Attribute 82: tunnel assignment ID, shell-based authentication of VPDN users, tunnel authentication via RADIUS on tunnel terminator.
Cisco IOS Wide-Area Networking Configuration Guide *Cisco IOS XE Wide-Area Networking Configuration Guide* *Cisco IOS Wide-Area Networking Command Reference*	Frame Relay, Layer 2 Tunneling Protocol Version 3 (L2TPv3), Link Access Procedure, Balanced (LAPB), Switched Multimegabit Data Service (SMDS), and X.25.
Cisco IOS Wireless LAN Configuration Guide *Cisco IOS Wireless LAN Command Reference*	Broadcast key rotation, IEEE 802.11x support, IEEE 802.1x authenticator, IEEE 802.1x local authentication service for Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling (EAP-FAST), Multiple Basic Service Set ID (BSSID), Wi-Fi Multimedia (WMM) required elements, and Wi-Fi Protected Access (WPA).

Table 2 Cisco IOS Supplementary Documents and Resources

Document Title	Description
Cisco IOS Master Command List, All Releases	Alphabetical list of all the commands documented in all Cisco IOS releases.
Cisco IOS New, Modified, Removed, and Replaced Commands	List of all the new, modified, removed, and replaced commands for a Cisco IOS release.
Cisco IOS Software System Messages	List of Cisco IOS system messages and descriptions. System messages may indicate problems with your system; be informational only; or may help diagnose problems with communications lines, internal hardware, or the system software.
Cisco IOS Debug Command Reference	Alphabetical list of **debug** commands including brief descriptions of use, command syntax, and usage guidelines.
Release Notes and Caveats	Information about new and changed features, system requirements, and other useful information about specific software releases; information about defects in specific Cisco IOS software releases.
MIBs	Files used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator at the following URL: http://www.cisco.com/go/mibs
RFCs	Standards documents maintained by the Internet Engineering Task Force (IETF) that Cisco IOS documentation references where applicable. The full text of referenced RFCs may be obtained at the following URL: http://www.rfc-editor.org/

Additional Resources and Documentation Feedback

What's New in Cisco Product Documentation is published monthly and describes all new and revised Cisco technical documentation. The *What's New in Cisco Product Documentation* publication also provides information about obtaining the following resources:

- Technical documentation
- Cisco product security overview
- Product alerts and field notices
- Technical assistance

Cisco IOS technical documentation includes embedded feedback forms where you can rate documents and provide suggestions for improvement. Your feedback helps us improve our documentation.

Using the Command-Line Interface in Cisco IOS and Cisco IOS XE Software

Last updated: August 6, 2008

This document provides basic information about the command-line interface (CLI) in Cisco IOS and Cisco IOS XE software and how you can use some of the CLI features. This document contains the following sections:

For more information about using the CLI, see the "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information about the software documentation set, see the "About Cisco IOS and Cisco IOS XE Software Documentation" document.

Initially Configuring a Device

Initially configuring a device varies by platform. For information about performing an initial configuration, see the hardware installation documentation that is provided with the original packaging of the product or go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

After you have performed the initial configuration and connected the device to your network, you can configure the device by using the console port or a remote access method, such as Telnet or Secure Shell (SSH), to access the CLI or by using the configuration method provided on the device, such as Security Device Manager.

Changing the Default Settings for a Console or AUX Port

There are only two changes that you can make to a console port and an AUX port:

- Change the port speed with the **config-register 0x** command. Changing the port speed is not recommended. The well-known default speed is 9600.
- Change the behavior of the port; for example, by adding a password or changing the timeout value.

Note The AUX port on the Route Processor (RP) installed in a Cisco ASR1000 series router does not serve any useful customer purpose and should be accessed only under the advisement of a customer support representative.

Using the CLI

This section describes the following topics:

Understanding Command Modes

The CLI command mode structure is hierarchical, and each mode supports a set of specific commands. This section describes the most common of the many modes that exist.

Table 1 lists common command modes with associated CLI prompts, access and exit methods, and a brief description of how each mode is used.

Table 1 CLI Command Modes

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
User EXEC	Log in.	`Router>`	Issue the **logout** or **exit** command.	• Change terminal settings. • Perform basic tests. • Display device status.
Privileged EXEC	From user EXEC mode, issue the **enable** command.	`Router#`	Issue the **disable** command or the **exit** command to return to user EXEC mode.	• Issue **show** and **debug** commands. • Copy images to the device. • Reload the device. • Manage device configuration files. • Manage device file systems.
Global configuration	From privileged EXEC mode, issue the **configure terminal** command.	`Router(config)#`	Issue the **exit** command or the **end** command to return to privileged EXEC mode.	Configure the device.
Interface configuration	From global configuration mode, issue the **interface** command.	`Router(config-if)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual interfaces.
Line configuration	From global configuration mode, issue the **line vty** or **line console** command.	`Router(config-line)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual terminal lines.

Table 1 CLI Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
ROM monitor	From privileged EXEC mode, issue the **reload** command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon # >` The # symbol represents the line number and increments at each prompt.	Issue the **continue** command.	• Run as the default operating mode when a valid image cannot be loaded. • Access the fall-back procedure for loading an image when the device lacks a valid image and cannot be booted. • Perform password recovery when a CTRL-Break sequence is issued within 60 seconds of a power-on or reload event.
Diagnostic (available only on the Cisco ASR1000 series router)	The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS process or processes fail, in most scenarios the router will reload. • A user-configured access policy was configured using the **transport-map** command, which directed the user into diagnostic mode. • The router was accessed using an RP auxiliary port. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) was entered, and the router was configured to enter diagnostic mode when the break signal was received.	`Router(diag)#`	If a Cisco IOS process failure is the reason for entering diagnostic mode, the failure must be resolved and the router must be rebooted to exit diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or using a method that is configured to connect to the Cisco IOS CLI. If the RP auxiliary port was used to access the router, use another port for access. Accessing the router through the auxiliary port is not useful for customer purposes.	• Inspect various states on the router, including the Cisco IOS state. • Replace or roll back the configuration. • Provide methods of restarting the Cisco IOS software or other processes. • Reboot hardware, such as the entire router, an RP, an ESP, a SIP, a SPA, or possibly other hardware components. • Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

EXEC commands are not saved when the software reboots. Commands that you issue in a configuration mode can be saved to the startup configuration. If you save the running configuration to the startup configuration, these commands will execute when the software is rebooted. Global configuration mode is the highest level of configuration mode. From global configuration mode, you can enter a variety of other configuration modes, including protocol-specific modes.

ROM monitor mode is a separate mode that is used when the software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode. Use the question symbol (?) to view the commands that you can use while the device is in ROM monitor mode.

```
rommon 1 > ?
alias               set and display aliases command
boot                boot up an external process
confreg             configuration register utility
cont                continue executing a downloaded image
context             display the context of a loaded image
cookie              display contents of cookie PROM in hex
.
.
.
rommon 2 >
```

The following example shows how the command prompt changes to indicate a different command mode:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 1/1
Router(config-if)# ethernet
Router(config-line)# exit
Router(config)# end
Router#
```

Note A keyboard alternative to the **end** command is Ctrl-Z.

Using the Interactive Help Feature

The CLI includes an interactive Help feature. Table 2 describes how to use the Help feature.

Table 2 CLI Interactive Help Commands

Command	Purpose
help	Provides a brief description of the help feature in any command mode.
?	Lists all commands available for a particular command mode.
*partial command***?**	Provides a list of commands that begin with the character string (no space between the command and the question mark).
partial command<**Tab**>	Completes a partial command name (no space between the command and <Tab>).
command **?**	Lists the keywords, arguments, or both associated with the command (space between the command and the question mark).
command keyword **?**	Lists the arguments that are associated with the keyword (space between the keyword and the question mark).

The following examples show how to use the help commands:

help

```
Router> help

Help may be requested at any point in a command by entering a question mark '?'.  If
nothing matches, the help list will be empty and you must backup until entering a '?'
shows the available options.

Two styles of help are provided:

1. Full help is available when you are ready to enter a command argument (e.g. 'show ?')
and describes each possible argument.

2. Partial help is provided when an abbreviated argument is entered and you want to know
what arguments match the input (e.g. 'show pr?'.)
```

?

```
Router# ?
Exec commands:
  access-enable        Create a temporary access-List entry
  access-profile       Apply user-profile to interface
  access-template      Create a temporary access-List entry
  alps                 ALPS exec commands
  archive              manage archive files
<snip>
```

***partial command*?**

```
Router(config)# zo?
zone  zone-pair
```

***partial command*<Tab>**

```
Router(config)# we<Tab> webvpn
```

***command* ?**

```
Router(config-if)# pppoe ?
  enable        Enable pppoe
  max-sessions  Maximum PPPOE sessions
```

***command keyword* ?**

```
Router(config-if)# pppoe enable ?
  group  attach a BBA group
  <cr>
```

Understanding Command Syntax

Command syntax is the format in which a command should be entered in the CLI. Commands include the name of the command, keywords, and arguments. Keywords are alphanumeric strings that are used literally. Arguments are placeholders for values that a user must supply. Keywords and arguments may be required or optional.

Specific conventions convey information about syntax and command elements. Table 3 describes these conventions.

Table 3 CLI Syntax Conventions

Symbol/Text	Function	Notes
< > (angle brackets)	Indicate that the option is an argument.	Sometimes arguments are displayed without angle brackets.
A.B.C.D.	Indicates that you must enter a dotted decimal IP address.	Angle brackets (< >) are not always used to indicate that an IP address is an argument.
WORD (all capital letters)	Indicates that you must enter one word.	Angle brackets (< >) are not always used to indicate that a WORD is an argument.
LINE (all capital letters)	Indicates that you must enter more than one word.	Angle brackets (< >) are not always used to indicate that a LINE is an argument.
<cr> (carriage return)	Indicates the end of the list of available keywords and arguments, and also indicates when keywords and arguments are optional. When <cr> is the only option, you have reached the end of the branch or the end of the command if the command has only one branch.	—

The following examples show syntax conventions:

```
Router(config)# ethernet cfm domain ?
  WORD  domain name
Router(config)# ethernet cfm domain dname ?
  level
Router(config)# ethernet cfm domain dname level ?
  <0-7>  maintenance level number
Router(config)# ethernet cfm domain dname level 7 ?
  <cr>
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
Router(config)# logging host ?
  Hostname or A.B.C.D  IP address of the syslog server
  ipv6                 Configure IPv6 syslog server
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
```

Understanding Enable and Enable Secret Passwords

Some privileged EXEC commands are used for actions that impact the system, and it is recommended that you set a password for these commands to prevent unauthorized use. Two types of passwords, enable (not encrypted) and enable secret (encrypted), can be set. The following commands set these passwords and are issued in global configuration mode:

- **enable** *password*
- **enable secret** *password*

Using an enable secret password is recommended because it is encrypted and more secure than the enable password. When you use an enable secret password, text is encrypted (unreadable) before it is written to the config.text file. When you use an enable password, the text is written as entered (readable) to the config.text file.

Each type of password is case sensitive, can contain from 1 to 25 uppercase and lowercase alphanumeric characters, and can start with a number. Spaces are also valid password characters; for example, "two words" is a valid password. Leading spaces are ignored, but trailing spaces are recognized.

Note Both password commands have numeric keywords that are single integer values. If you choose a number for the first character of your password followed by a space, the system will read the number as if it were the numeric keyword and not as part of your password.

When both passwords are set, the enable secret password takes precedence over the enable password.

To remove a password, use the **no** form of the commands: **no enable** *password* or **no enable secret** *password*.

For more information about password recovery procedures for Cisco products, see http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_tech_note09186a00801746e6.shtml.

Using the Command History Feature

The CLI command history feature saves the commands you enter during a session in a command history buffer. The default number of commands saved is 10, but the number is configurable within the range of 0 to 256. This command history feature is particularly useful for recalling long or complex commands.

To change the number of commands saved in the history buffer for a terminal session, issue the **terminal history size** command:

```
Router# terminal history size num
```

A command history buffer is also available in line configuration mode with the same default and configuration options. To set the command history buffer size for a terminal session in line configuration mode, issue the **history** command:

```
Router(config-line)# history [size num]
```

To recall commands from the history buffer, use the following methods:

- Press Ctrl-P or the up arrow key—Recalls commands beginning with the most recent command. Repeat the key sequence to recall successively older commands.

- Press Ctrl-N or the down arrow key—Recalls the most recent commands in the history buffer after they have been recalled using Ctrl-P or the up arrow key. Repeat the key sequence to recall successively more recent commands.

Note The arrow keys function only on ANSI-compatible terminals such as the VT100.

- Issue the **show history** command in user EXEC or privileged EXEC mode—Lists the most recent commands that you entered. The number of commands that are displayed is determined by the setting of the **terminal history size** and **history** commands.

  The CLI command history feature is enabled by default. To disable this feature for a terminal session, issue the **terminal no history** command in user EXEC or privileged EXEC mode or the **no history** command in line configuration mode.

Abbreviating Commands

Typing a complete command name is not always required for the command to execute. The CLI recognizes an abbreviated command when the abbreviation contains enough characters to uniquely identify the command. For example, the **show version** command can be abbreviated as **sh ver**. It cannot be abbreviated as **s ver** because **s** could mean **show**, **set**, or **systat**. The **sh v** abbreviation also is not valid because the **show** command has **vrrp** as a keyword in addition to **version**. (Command and keyword examples from Cisco IOS Release 12.4(13)T.)

Using Aliases for CLI Commands

To save time and the repetition of entering the same command multiple times, you can use a command alias. An alias can be configured to do anything that can be done at the command line, but an alias cannot move between modes, type in passwords, or perform any interactive functions.

Table 4 shows the default command aliases.

Table 4 Default Command Aliases

Command Alias	Original Command
h	help
lo	logout
p	ping
s	show
u or **un**	undebug
w	where

To create a command alias, issue the **alias** command in global configuration mode. The syntax of the command is **alias** *mode command-alias original-command*. Following are some examples:

- Router(config)# **alias exec prt partition**—privileged EXEC mode
- Router(config)# **alias configure sb source-bridge**—global configuration mode
- Router(config)# **alias interface rl rate-limit**—interface configuration mode

To view both default and user-created aliases, issue the **show alias** command.

For more information about the **alias** command, see http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html.

Using the no and default Forms of Commands

Most configuration commands have a **no** form that is used to reset a command to its default value or disable a feature or function. For example, the **ip routing** command is enabled by default. To disable this command, you would issue the **no ip routing** command. To re-enable IP routing, you would issue the **ip routing** command.

Configuration commands may also have a **default** form, which returns the command settings to their default values. For commands that are disabled by default, using the **default** form has the same effect as using the **no** form of the command. For commands that are enabled by default and have default settings, the **default** form enables the command and returns the settings to their default values.

The **no** and **default** forms of commands are described in the command pages of command references.

Using the debug Command

A **debug** command produces extensive output that helps you troubleshoot problems in your network. These commands are available for many features and functions within Cisco IOS and Cisco IOS XE software. Some **debug** commands are **debug all**, **debug aaa accounting**, and **debug mpls packets**. To use **debug** commands during a Telnet session with a device, you must first enter the **terminal monitor** command. To turn off debugging completely, you must enter the **undebug all** command.

For more information about **debug** commands, see the *Cisco IOS Debug Command Reference* at http://www.cisco.com/en/US/docs/ios/debug/command/reference/db_book.html.

Caution Debugging is a high priority and high CPU utilization process that can render your device unusable. Use **debug** commands only to troubleshoot specific problems. The best times to run debugging are during periods of low network traffic and when few users are interacting with the network. Debugging during these periods decreases the likelihood that the **debug** command processing overhead will affect network performance or user access or response times.

Filtering Output Using Output Modifiers

Many commands produce lengthy output that may use several screens to display. Using output modifiers, you can filter this output to show only the information that you want to see.

Three output modifiers are available and are described as follows:

- **begin** *regular expression*—Displays the first line in which a match of the regular expression is found and all lines that follow.
- **include** *regular expression*—Displays all lines in which a match of the regular expression is found.
- **exclude** *regular expression*—Displays all lines except those in which a match of the regular expression is found.

To use one of these output modifiers, type the command followed by the pipe symbol (|), the modifier, and the regular expression that you want to search for or filter. A regular expression is a case-sensitive alphanumeric pattern. It can be a single character or number, a phrase, or a more complex string.

The following example illustrates how to filter output of the **show interface** command to display only lines that include the expression "protocol."

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

Understanding CLI Error Messages

You may encounter some error messages while using the CLI. Table 5 shows the common CLI error messages.

Table 5 Common CLI Error Messages

Error Message	Meaning	How to Get Help
% Ambiguous command: "show con"	You did not enter enough characters for the command to be recognized.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Incomplete command.	You did not enter all the keywords or values required by the command.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Invalid input detected at "^" marker.	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all the commands that are available in this command mode. The keywords that you are allowed to enter for the command appear.

For more system error messages, see the following documents:

- *Cisco IOS Release 12.2SR System Message Guide*
- *Cisco IOS System Messages, Volume 1 of 2* (Cisco IOS Release 12.4)
- *Cisco IOS System Messages, Volume 2 of 2* (Cisco IOS Release 12.4)

Saving Changes to a Configuration

To save changes that you made to the configuration of a device, you must issue the **copy running-config startup-config** command or the **copy system:running-config nvram:startup-config** command. When you issue these commands, the configuration changes that you made are saved to the startup configuration and saved when the software reloads or power to the device is turned off or interrupted. The following example shows the syntax of the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
```

You press Enter to accept the startup-config filename (the default), or type a new filename and then press Enter to accept that name. The following output is displayed indicating that the configuration was saved:

```
Building configuration...
[OK]
Router#
```

On most platforms, the configuration is saved to NVRAM. On platforms with a Class A flash file system, the configuration is saved to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Additional Information

- "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*:

  http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/cf_cli-basics.html

  or

  "Using Cisco IOS XE Software" chapter of the *Cisco ASR1000 Series Aggregation Services Routers Software Configuration Guide*:

  http://www.cisco.com/en/US/docs/routers/asr1000/configuration/guide/chassis/using_cli.html
- Cisco Product Support Resources

  http://www.cisco.com/web/psa/products/index.html
- Support area on Cisco.com (also search for documentation by task or product)

  http://www.cisco.com/en/US/support/index.html
- *White Paper: Cisco IOS Reference Guide*

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1828/products_white_paper09186a008018305e.shtml
- Software Download Center (downloads; tools; licensing, registration, advisory, and general information) (requires Cisco.com User ID and password)

  http://www.cisco.com/kobayashi/sw-center/
- Error Message Decoder, a tool to help you research and resolve error messages for Cisco IOS software

  http://www.cisco.com/pcgi-bin/Support/Errordecoder/index.cgi

- Command Lookup Tool, a tool to help you find detailed descriptions of Cisco IOS commands (requires Cisco.com user ID and password)

  http://tools.cisco.com/Support/CLILookup

- Output Interpreter, a troubleshooting tool that analyzes command output of supported **show** commands

  https://www.cisco.com/pcgi-bin/Support/OutputInterpreter/home.pl\

Security Overview

This chapter contains the following sections:

- About This Guide

  Preview the topics in this guide.

- Creating Effective Security Policies

  Learn tips and hints for creating a security policy for your organization. A security policy should be finalized and up to date *before* you configure any security features.

- Identifying Security Risks and Cisco IOS Solutions

  Identify common security risks that might be present in your network, and find the right Cisco IOS security feature to prevent security break-ins.

About This Guide

The *Cisco IOS Security Configuration Guide* describes how to configure Cisco IOS security features for your Cisco networking devices. These security features can protect your network against degradation or failure and also against data loss or compromise resulting from intentional attacks and from unintended but damaging mistakes by well-meaning network users.

This guide is divided into seven parts:

- Authentication, Authorization, and Accounting (AAA)
- Security Server Protocols
- Traffic Filtering, Firewalls, and Virus Detection
- IP Security (IPSec) and Internet Key Exchange (IKE)
- Public Key Infrastructure (PKI)
- Other Security Features
- Cisco IOS Secure Infrastructure

Appendixes follow the seven main divisions.

The following sections briefly describe each of these sections and the appendixes.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Authentication, Authorization, and Accounting (AAA)

This part describes how to configure Cisco's authentication, authorization, and accounting (AAA) paradigm. AAA is an architectural framework for configuring a set of three independent security functions in a consistent, modular manner.

- Authentication—Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods and then applying that list to various interfaces.
- Authorization—Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.

  Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared with the information contained in a database for a given user, and the result is returned to AAA to determine the user's actual capabilities and restrictions.
- Accounting—Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes. Accounting enables you to track the services users are accessing, as well as the amount of network resources they are consuming.

Note You can configure authentication outside of AAA. However, you must configure AAA if you want to use RADIUS, TACACS+, or Kerberos or if you want to configure a backup authentication method.

Security Server Protocols

In many circumstances, AAA uses security protocols to administer its security functions. If your router or access server is acting as a network access server, AAA is the means through which you establish communication between your network access server and your RADIUS, TACACS+, or Kerberos security server.

The chapters in this part describe how to configure the following security server protocols:

- RADIUS—A distributed client/server system implemented through AAA that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.
- TACACS+—A security application implemented through AAA that provides centralized validation of users attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. TACACS+ provides for separate and modular authentication, authorization, and accounting facilities.
- Kerberos—A secret-key network authentication protocol implemented through AAA that uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication. Kerberos was designed to authenticate requests for network resources. Kerberos is based on the concept of a trusted third party that performs secure verification of users and services. The primary use of Kerberos is to verify that users and the network services they use are really who and what

they claim to be. To accomplish this, a trusted Kerberos server issues tickets to users. These tickets, which have a limited lifespan, are stored in a user's credential cache and can be used in place of the standard username-and-password authentication mechanism.

Traffic Filtering, Firewalls, and Virus Detection

This part describes how to configure your networking devices to filter traffic, function as a firewall, or detect potential viruses.

- Cisco implements traffic filters with access control lists (also called access lists). Access lists determine what traffic is blocked and what traffic is forwarded at router interfaces. Cisco provides both basic and advanced access list capabilities.
  - Basic access lists

    An overview of basic access lists is in the chapter "Access Control Lists: Overview and Guidelines." This chapter describes tips, cautions, considerations, recommendations, and general guidelines for configuring access lists for the various network protocols. You should configure basic access lists for all network protocols that will be routed through your networking device, such as IP, IPX, AppleTalk, and so forth.
  - Advanced access lists

    The advanced access list capabilities and configuration are described in the remaining chapters in the "Traffic Filtering, Firewalls, and Virus Detection" part of this document. The advanced access lists provide sophisticated and dynamic traffic filtering capabilities for stronger, more flexible network security.
- Cisco IOS Firewall provides an extensive set of security features, allowing you to configure a simple or elaborate firewall, according to your particular requirements. The following features are key components of Cisco IOS Firewall:
  - Context-based Access Control (CBAC)

    CBAC intelligently filters TCP and UDP packets based on application-layer protocol session information. You can configure CBAC to permit specified TCP and UDP traffic through a firewall only when the connection is initiated from within the network you want to protect. CBAC can inspect traffic for sessions that originate from either side of the firewall, and CBAC can be used for intranet, extranet, and Internet perimeters of your network.
  - Cisco IOS Intrusion Prevention System (IPS)

    Cisco IOS IPS acts as an in-line intrusion detection sensor, "watching" packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When Cisco IOS IPS detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or Security Device Event Exchange (SDEE).

    Customers can download the Cisco IOS IPS (via a signature detection file [SDF]) to their router from Cisco.com via the VPN and Security Management Solution (VMS) IDS Management Console (MC) 2.3 network management device or via the Cisco Router and Security Device Manager (SDM). Thus VMS IDS MC or SDM can immediately begin scanning for new signatures.
  - Authentication Proxy

    The Cisco IOS Firewall authentication proxy feature allows network administrators to apply specific security policies on a per-user basis. Previously, user identity and related authorized access were associated with a user's IP address, or a single security policy had to be applied to

an entire user group or sub network. Now, users can be identified and authorized on the basis of their per-user policy, and access privileges tailored on an individual basis are possible, as opposed to general policy applied across multiple users.

– Port to Application Mapping (PAM)

Port to Application Mapping (PAM) is a feature of Cisco Secure Integrated Software. PAM allows you to customize TCP or UDP port numbers for network services or applications. PAM uses this information to support network environments that run services using ports that are different from the registered or well-known ports associated with an application. For example, the information in the PAM table enables Context-based Access Control (CBAC) supported services to run on non-standard ports.

Firewalls are discussed in the chapters “Cisco IOS Firewall Overview” and “Configuring Context-Based Access Control.”

- Cisco addresses the increased threat and impact of worms and viruses to networked businesses with Cisco Network Admission Control (NAC). NAC enables Cisco routers to enforce access privileges when an endpoint attempts to connect to a network. This access decision is made on the basis of information about the endpoint device, such as its current antivirus state. The antivirus state includes information such as the version of antivirus software, virus definitions, and version of the scan engine.

  NAC systems allow noncompliant devices to be denied access, placed in a quarantined area, or given restricted access to computing resources, thus keeping insecure nodes from infecting the network.

IP Security (IPSec) and Internet Key Exchange (IKE)

This section describes how to configure security for VPNs via IPSec and IKE:

- Configuring Security for VPNs with IPSec

  This module describes how to configure IPSec. IPSec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec provides data authentication and anti-replay services in addition to data confidentiality services.

- Configuring Internet Key Exchange for IPSec VPNs

  This module describes how to configure IKE for use with IPSec VPNs. IKE is a key management protocol standard that is used in conjunction with the IPSec standard. IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard.

Public Key Infrastructure (PKI)

This section describes how to implement and manage a Cisco IOS PKI, which provides certificate management to support security protocols such as IPSec, secure shell (SSH), and secure socket layer (SSL). This section is divided into the following modules:

- Cisco IOS PKI Overview: Understanding and Planning a PKI

  This module identifies general concepts necessary to understand how a PKI functions.

- Deploying RSA Keys Within a PKI

This module explains how to set up and deploy Rivest, Shamir, and Adelman (RSA) keys within a PKI. An RSA key pair is required before you can obtain a certificate for your router; that is, the end host must generate a pair of RSA keys and exchange the public key with the certification authority (CA) to obtain a certificate and enroll in a PKI.

- Configuring Revocation and Authorization of Certificates in a PKI

  This module describes how to configure revocation and authorization of certificates in a PKI. After a certificate is validated as a properly signed certificate, it is authorized (via methods such as, certificate maps, PKI-AAA, or a certificate-based ACL) and the revocation status is checked by the issuing CA to ensure that the certificate has not been revoked.

- Configuring Certificate Enrollment for a PKI

  Certificate enrollment, which is the process of obtaining a certificate from a CA, occurs between the end host requesting the certificate and the CA. Each peer that participates in the PKI must enroll with a CA. This module describes the different methods available for certificate enrollment and describes how to set up each method for a participating PKI peer.

- Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI

  This module describes how to use SDP in a PKI. SDP is a web-based certificate enrollment interface that can be used to easily deploy PKI between two end devices, such as a Cisco IOS client and a Cisco IOS certificate server. SDP provides a solution for users deploying a large number of peer devices, including certificates and configurations.

- Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment

  This module describes how to set up and manage a Cisco IOS Certificate Server (CS) for PKI deployment. A CS embeds a simple certificate server, with limited CA functionality, into the Cisco IOS software.

- Storing PKI Credentials External to the Router

  This module explains how to store RSA keys on device external to the router via a USB eToken. eTokens provide secure configuration distribution and allow users to store PKI credentials, such as RSA keys, for deployment.

Other Security Features

This section describes six security features in the following chapters:

- Neighbor Router Authentication: Overview and Guidelines

  This chapter briefly describes the security benefits and operation of neighbor router authentication.

  When neighbor authentication is configured on a router, the router authenticates its neighbor router before accepting any route updates from that neighbor. This ensures that a router always receives reliable routing update information from a trusted source.

- Configuring IP Security Options

  This chapter describes how to configure IP Security Options (IPSO) as described in RFC 1108. IPSO is generally used to comply with the security policy of the U.S. government's Department of Defense.

- Configuring Unicast Reverse Path Forwarding

  This chapter describes the Unicast Reverse Path Forwarding (Unicast RPF) feature, which helps mitigate problems caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf and Tribe Flood

Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers (ISPs) that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.

- Configuring Secure Shell

  This chapter describes the Secure Shell (SSH) feature. SSH is an application and a protocol that provides a secure replacement to a suite of Unix r-commands such as rsh, rlogin and rcp. (Cisco IOS supports rlogin.) The protocol secures the sessions using standard cryptographic mechanisms, and the application can be used similarly to the Berkeley rexec and rsh tools. There are currently two versions of SSH available: SSH Version 1 and SSH Version 2.

- Configuring 802.1x Authentication Services

  This section describes how to configure local authentication and VPN access via the Institute of Electrical and Electronics Engineers (IEEE) 802.1X protocol framework.

- WebVPN

  WebVPN provides end users with unrestricted, secure remote access to enterprise sites without having VPN installed on their end devices. Users can access the enterprise sites from anywhere on the Internet and can access enterprise applications such as e-mail and web browsing.

Cisco IOS Secure Infrastructure

- This section contains features that help users secure their network infrastructure. Some of the available features are as follows: Autosecure (which simplifies the security configuration of a router and hardens the router configuration); Image Verification (which enables routers to automatically detect when the integrity of an image is accidentally corrupted as a result of transmission errors or disk corruption); Role-Based CLI Access (which allows network administrators to exercise better control over access to Cisco networking devices), and “Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices” (which is a guide to implementing a baseline level of security for your networking devices).

Appendixes

The appendixes describe the supported RADIUS attributes and TACACS+ attribute-value pairs as follows:

- RADIUS Attributes

  RADIUS attributes are used to define specific AAA elements in a user profile, which is stored on the RADIUS daemon. This appendix lists the RADIUS attributes currently supported.

- TACACS+ Attribute-Value Pairs

  TACACS+ attribute-value pairs are used to define specific AAA elements in a user profile, which is stored on the TACACS+ daemon. This appendix lists the TACACS+ attribute-value pairs currently supported.

Creating Effective Security Policies

An effective security policy works to ensure that your organization's network assets are protected from sabotage and from inappropriate access—both intentional and accidental.

All network security features should be configured in compliance with your organization's security policy. If you do not have a security policy, or if your policy is out of date, you should ensure that the policy is created or updated before you decide how to configure security on your Cisco device.

The following sections provide guidelines to help you create an effective security policy:

The Nature of Security Policies

You should recognize these aspects of security policies:

- Security policies represent trade-offs.

  With all security policies, there is some trade-off between user productivity and security measures that can be restrictive and time consuming. The goal of any security design is to provide maximum security with minimum impact on user access and productivity. Some security measures, such as network data encryption, do not restrict access and productivity. On the other hand, cumbersome or unnecessarily redundant verification and authorization systems can frustrate users and even prevent access to critical network resources.

- Security policies should be determined by business needs.

  Business needs should dictate the security policy; a security policy should not determine how a business operates.

- Security policies are living documents.

  Because organizations are constantly subject to change, security policies must be systematically updated to reflect new business directions, technological changes, and resource allocations.

Two Levels of Security Policies

You can think of a security policy as having two levels: a requirements level and an implementation level.

- At the requirements level, a policy defines the degree to which your network assets must be protected against intrusion or destruction and also estimates the cost (consequences) of a security breach. For example, the policy could state that only human resources personnel should be able to access personnel records, or that only IS personnel should be able to configure the backbone routers. The policy could also address the consequences of a network outage (due to sabotage), and the consequences of inadvertently making sensitive information public.
- At the implementation level, a policy defines guidelines to implement the requirements-level policy, using specific technology in a predefined way. For example, the implementation-level policy could require access lists to be configured so that only traffic from human resources host computers can access the server containing personnel records.

When creating a policy, define security requirements before defining security implementations so that you do not end up merely justifying particular technical solutions that might not actually be required.

Tips for Developing an Effective Security Policy

To develop an effective security policy, consider the recommendations in the following sections:

Identifying Your Network Assets to Protect

The first step to developing a security policy is to understand and identify your organization's network assets. Network assets include the following:

- Networked hosts (such as PCs; includes the hosts' operating systems, applications, and data)
- Networking devices (such as routers)
- Network data (data that travels across the network)

You must both identify your network's assets and determine the degree to which each of these assets must be protected. For example, one subnetwork of hosts might contain extremely sensitive data that should be protected at all costs, while a different subnetwork of hosts might require only modest protection against security risks because there is less cost involved if the subnetwork is compromised.

Determining Points of Risk

You must understand how potential intruders can enter your organization's network or sabotage network operation. Special areas of consideration are network connections, dial-up access points, and misconfigured hosts. Misconfigured hosts, frequently overlooked as points of network entry, can be systems with unprotected login accounts (guest accounts), employ extensive trust in remote commands (such as rlogin and rsh), have illegal modems attached to them, and use easy-to-break passwords.

Limiting the Scope of Access

Organizations can create multiple barriers within networks, so that unlawful entry to one part of the system does not automatically grant entry to the entire infrastructure. Although maintaining a high level of security for the entire network can be prohibitively expensive (in terms of systems and equipment as well as productivity), you can often provide higher levels of security to the more sensitive areas of your network.

Identifying Assumptions

Every security system has underlying assumptions. For example, an organization might assume that its network is not tapped, that intruders are not very knowledgeable, that intruders are using standard software, or that a locked room is safe. It is important to identify, examine, and justify your assumptions: any hidden assumption is a potential security hole.

Determining the Cost of Security Measures

In general, providing security comes at a cost. This cost can be measured in terms of increased connection times or inconveniences to legitimate users accessing the assets, or in terms of increased network management requirements, and sometimes in terms of actual dollars spent on equipment or software upgrades.

Some security measures inevitably inconvenience some sophisticated users. Security can delay work, create expensive administrative and educational overhead, use significant computing resources, and require dedicated hardware.

When you decide which security measures to implement, you must understand their costs and weigh these against potential benefits. If the security costs are out of proportion to the actual dangers, it is a disservice to the organization to implement them.

Considering Human Factors

If security measures interfere with essential uses of the system, users resist these measures and sometimes even circumvent them. Many security procedures fail because their designers do not take this fact into account. For example, because automatically generated “nonsense” passwords can be difficult to remember, users often write them on the undersides of keyboards. A “secure” door that leads to a system’s only tape drive is sometimes propped open. For convenience, unauthorized modems are often connected to a network to avoid cumbersome dial-in security procedures. To ensure compliance with your security measures, users must be able to get their work done as well as understand and accept the need for security.

Any user can compromise system security to some degree. For example, an intruder might learn passwords by simply calling legitimate users on the telephone claiming to be a system administrator and asking for them. If users understand security issues and understand the reasons for them, they are far less likely to compromise security in this way.

Defining such human factors and any corresponding policies needs to be included as a formal part of your complete security policy.

At a minimum, users must be taught never to release passwords or other secrets over unsecured telephone lines (especially through cordless or cellular telephones) or electronic mail. They should be wary of questions asked by people who call them on the telephone. Some companies have implemented formalized network security training for their employees in which employees are not allowed access to the network until they have completed a formal training program.

Keeping a Limited Number of Secrets

Most security is based on secrets; for example, passwords and encryption keys are secrets. But the more secrets there are, the harder it is to keep all of them. It is prudent, therefore, to design a security policy that relies on a limited number of secrets. Ultimately, the most important secret an organization has is the information that can help someone circumvent its security.

Implementing Pervasive and Scalable Security

Use a systematic approach to security that includes multiple, overlapping security methods.

Almost any change that is made to a system can affect security. This is especially true when new services are created. System administrators, programmers, and users need to consider the security implications of every change they make. Understanding the security implications of a change takes practice; it requires lateral thinking and a willingness to explore every way that a service could potentially be manipulated. The goal of any security policy is to create an environment that is not susceptible to every minor change.

Understanding Typical Network Functions

Understand how your network system normally functions, know what is expected and unexpected behavior, and be familiar with how devices are usually used. This kind of awareness helps the organization detect security problems. Noticing unusual events can help catch intruders before they can damage the system. Software auditing tools can help detect, log, and track unusual events. In addition, an organization should know exactly what software it relies on to provide auditing trails, and a security system should not operate on the assumption that all software is bug free.

Remembering Physical Security

The physical security of your network devices and hosts cannot be neglected. For example, many facilities implement physical security by using security guards, closed circuit television, card-key entry systems, or other means to control physical access to network devices and hosts. Physical access to a computer or router usually gives a sophisticated user complete control over that device. Physical access to a network link usually allows a person to tap into that link, jam it, or inject traffic into it. Software security measures can often be circumvented when access to the hardware is not controlled.

Identifying Security Risks and Cisco IOS Solutions

Cisco IOS software provides a comprehensive set of security features to guard against specific security risks. This section describes a few common security risks that might be present in your network, and describes how to use Cisco IOS software to protect against each of these risks:

- Preventing Unauthorized Access into Networking Devices
- Preventing Unauthorized Access into Networks
- Preventing Network Data Interception
- Preventing Fraudulent Route Updates

Preventing Unauthorized Access into Networking Devices

If someone were to gain console or terminal access into a networking device, such as a router, switch, or network access server, that person could do significant damage to your network—perhaps by reconfiguring the device, or even by simply viewing the device's configuration information.

Typically, you want administrators to have access to your networking device; you do not want other users on your local-area network or those dialing in to the network to have access to the router.

Users can access Cisco networking devices by dialing in from outside the network through an asynchronous port, connecting from outside the network through a serial port, or connecting via a terminal or workstation from within the local network.

To prevent unauthorized access into a networking device, you should configure one or more of the following security features:

- At a minimum, you should configure passwords and privileges at each networking device for all device lines and ports, as described in the module "Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices." These passwords are stored on the networking device. When users attempt to access the device through a particular line or port, they must enter the password applied to the line or port before they can access the device.
- For an additional layer of security, you can also configure username/password pairs, stored in a database on the networking device, as described in the module "Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices." These pairs are assigned to lines or interfaces and authenticate each user before that user can access the device. If you have defined privilege levels, you can also assign a specific privilege level (with associated rights and privileges) to each username/password pair.
- If you want to use username/password pairs, but you want to store them centrally instead of locally on each individual networking device, you can store them in a database on a security server. Multiple networking devices can then use the same database to obtain user authentication (and, if necessary, authorization) information. Cisco supports a variety of security server protocols, such as RADIUS, TACACS+, and Kerberos. If you decide to use the database on a security server to store login username/password pairs, you must configure your router or access server to support the applicable protocol; in addition, because most supported security protocols must be administered through the AAA security services, you will probably need to enable AAA. For more information about security protocols and AAA, refer to the chapters in the "Authentication, Authorization, and Accounting (AAA)" part of this document.

  Note Cisco recommends that, whenever possible, AAA be used to implement authentication.

- If you want to authorize individual users for specific rights and privileges, you can implement AAA's authorization feature, using a security protocol such as TACACS+ or RADIUS. For more information about security protocol features and AAA, refer to the chapters in the "Authentication, Authorization, and Accounting (AAA)" part of this document.
- If you want to have a backup authentication method, you must configure AAA. AAA allows you to specify the primary method for authenticating users (for example, a username/password database stored on a TACACS+ server) and then specify backup methods (for example, a locally stored username/password database.) The backup method is used if the primary method's database cannot be accessed by the networking device. To configure AAA, refer to the chapters in the "Authentication, Authorization, and Accounting (AAA)" part of this document. You can configure up to four sequential backup methods.

  Note If you do not have backup methods configured, you will be denied access to the device if the username/password database cannot be accessed for any reason.

- If you want to keep an audit trail of user access, configure AAA accounting as described in the chapter "Configuring Accounting."

Preventing Unauthorized Access into Networks

If someone were to gain unauthorized access to your organization's internal network, that person could cause damage in many ways, perhaps by accessing sensitive files from a host, by planting a virus, or by hindering network performance by flooding your network with illegitimate packets.

This risk can also apply to a person within your network attempting to access another internal network such as a Research and Development subnetwork with sensitive and critical data. That person could intentionally or inadvertently cause damage; for example, that person might access confidential files or tie up a time-critical printer.

To prevent unauthorized access through a networking device into a network, you should configure one or more of these security features:

- Traffic Filtering

  Cisco uses access lists to filter traffic at networking devices. Basic access lists allow only specified traffic through the device; other traffic is simply dropped. You can specify individual hosts or subnets that should be allowed into the network, and you can specify what type of traffic should be allowed into the network. Basic access lists generally filter traffic based on source and destination addresses, and protocol type of each packet.

  Advanced traffic filtering is also available, providing additional filtering capabilities; for example, the Lock-and-Key Security feature requires each user to be authenticated via a username/password before that user's traffic is allowed onto the network.

  All the Cisco IOS traffic filtering capabilities are described in the chapters in the "Traffic Filtering, Firewalls, and Virus Detection" part of this document.

- Authentication

  You can require users to be authenticated before they gain access into a network. When users attempt to access a service or host (such as a web site or file server) within the protected network, they must first enter certain data such as a username and password, and possibly additional information such as their date of birth or mother's maiden name. After successful authentication (depending on the method of authentication), users will be assigned specific privileges, allowing them to access specific network assets. In most cases, this type of authentication would be facilitated by using CHAP or PAP over a serial PPP connection in conjunction with a specific security protocol, such as TACACS+ or RADIUS.

  Just as in preventing unauthorized access to specific network devices, you need to decide whether or not you want the authentication database to reside locally or on a separate security server. In this case, a local security database is useful if you have very few routers providing network access. A local security database does not require a separate (and costly) security server. A remote, centralized security database is convenient when you have a large number of routers providing network access because it prevents you from having to update each router with new or changed username authentication and authorization information for potentially hundreds of thousands of dial-in users. A centralized security database also helps establish consistent remote access policies throughout a corporation.

  Cisco IOS software supports a variety of authentication methods. Although AAA is the primary (and recommended) method for access control, Cisco IOS software provides additional features for simple access control that are outside the scope of AAA. For more information, refer to the chapter "Configuring Authentication."

Preventing Network Data Interception

When packets travel across a network, they are susceptible to being read, altered, or "hijacked." (Hijacking occurs when a hostile party intercepts a network traffic session and poses as one of the session endpoints.)

If the data is traveling across an unsecured network such as the Internet, the data is exposed to a fairly significant risk. Sensitive or confidential data could be exposed, critical data could be modified, and communications could be interrupted if data is altered.

To protect data as it travels across a network, configure network data encryption, as described in the chapter "Configuring IPSec Network Security."

IPSec provides the following network security services. These services are optional. In general, local security policy will dictate the use of one or more of the following services:

- Data Confidentiality—The IPSec sender can encrypt packets before transmitting them across a network.
- Data Integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.
- Data Origin Authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service is dependent upon the data integrity service.
- Anti-Replay—The IPSec receiver can detect and reject replayed packets.

  Cisco IPSec prevents routed traffic from being examined or tampered with while it travels across a network. This feature causes IP packets to be encrypted at a Cisco router, routed across a network as encrypted information, and decrypted at the destination Cisco router. In between the two routers, the packets are in encrypted form and therefore the packets' contents cannot be read or altered. You define what traffic should be encrypted between the two routers, according to what data is more sensitive or critical.

  If you want to protect traffic for protocols other than IP, you can encapsulate those other protocols into IP packets using GRE encapsulation, and then encrypt the IP packets.

  Typically, you do not use IPSec for traffic that is routed through networks that you consider secure. Consider using IPSec for traffic that is routed across unsecured networks, such as the Internet, if your organization could be damaged if the traffic is examined or tampered with by unauthorized individuals.

Preventing Fraudulent Route Updates

All routing devices determine where to route individual packets by using information stored in route tables. This route table information is created using route updates obtained from neighboring routers.

If a router receives a fraudulent update, the router could be tricked into forwarding traffic to the wrong destination. This could cause sensitive data to be exposed, or could cause network communications to be interrupted.

To ensure that route updates are received only from known, trusted neighbor routers, configure neighbor router authentication as described in the chapter "Neighbor Router Authentication: Overview and Guidelines."

Authentication, Authorization, and Accounting (AAA)

AAA Overview

Access control is the way you control who is allowed access to the network server and what services they are allowed to use once they have access. Authentication, authorization, and accounting (AAA) network security services provide the primary framework through which you set up access control on your router or access server.

In This Chapter

This chapter includes the following sections:

About AAA Security Services

AAA is an architectural framework for configuring a set of three independent security functions in a consistent manner. AAA provides a modular way of performing the following services:

- Authentication—Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol you select, encryption.

  Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods, and then applying that list to various interfaces. The method list defines the types of authentication to be performed and the sequence in which they will be performed; it must be applied to a specific interface before any of the defined authentication methods will be performed. The only exception is the default method list (which is named “default”). The default method list is automatically applied to all interfaces if no other method list is defined. A defined method list overrides the default method list.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

All authentication methods, except for local, line password, and enable authentication, must be defined through AAA. For information about configuring all authentication methods, including those implemented outside of the AAA security services, refer to the chapter "Configuring Authentication."

- Authorization—Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.

  AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared to the information contained in a database for a given user and the result is returned to AAA to determine the user's actual capabilities and restrictions. The database can be located locally on the access server or router or it can be hosted remotely on a RADIUS or TACACS+ security server. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights with the appropriate user. All authorization methods must be defined through AAA.

  As with authentication, you configure AAA authorization by defining a named list of authorization methods, and then applying that list to various interfaces. For information about configuring authorization using AAA, refer to the chapter "Configuring Authorization."

- Accounting—Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

  Accounting enables you to track the services users are accessing as well as the amount of network resources they are consuming. When AAA accounting is activated, the network access server reports user activity to the RADIUS or TACACS+ security server (depending on which security method you have implemented) in the form of accounting records. Each accounting record is comprised of accounting AV pairs and is stored on the access control server. This data can then be analyzed for network management, client billing, and/or auditing. All accounting methods must be defined through AAA. As with authentication and authorization, you configure AAA accounting by defining a named list of accounting methods, and then applying that list to various interfaces. For information about configuring accounting using AAA, refer to the chapter "Configuring Accounting."

In many circumstances, AAA uses protocols such as RADIUS, TACACS+, or Kerberos to administer its security functions. If your router or access server is acting as a network access server, AAA is the means through which you establish communication between your network access server and your RADIUS, TACACS+, or Kerberos security server.

Although AAA is the primary (and recommended) method for access control, Cisco IOS software provides additional features for simple access control that are outside the scope of AAA, such as local username authentication, line password authentication, and enable password authentication. However, these features do not provide the same degree of access control that is possible by using AAA.

This section includes the following sections:

Benefits of Using AAA

AAA provides the following benefits:

- Increased flexibility and control of access configuration

- Scalability
- Standardized authentication methods, such as RADIUS, TACACS+, and Kerberos
- Multiple backup systems

Note The deprecated protocols, TACACS and extended TACACS, are not compatible with AAA; if you select these security protocols, you will not be able to take advantage of the AAA security services.

AAA Philosophy

AAA is designed to enable you to dynamically configure the type of authentication and authorization you want on a per-line (per-user) or per-service (for example, IP, IPX, or VPDN) basis. You define the type of authentication and authorization you want by creating method lists, then applying those method lists to specific services or interfaces.

For information about applications that use AAA, such as per-user configuration and virtual profiles, refer to the chapters “Configuring Per-User Configuration” and “Configuring Virtual Profiles” in the *Cisco IOS Dial Technologies Configuration Guide,* Release 12.2.

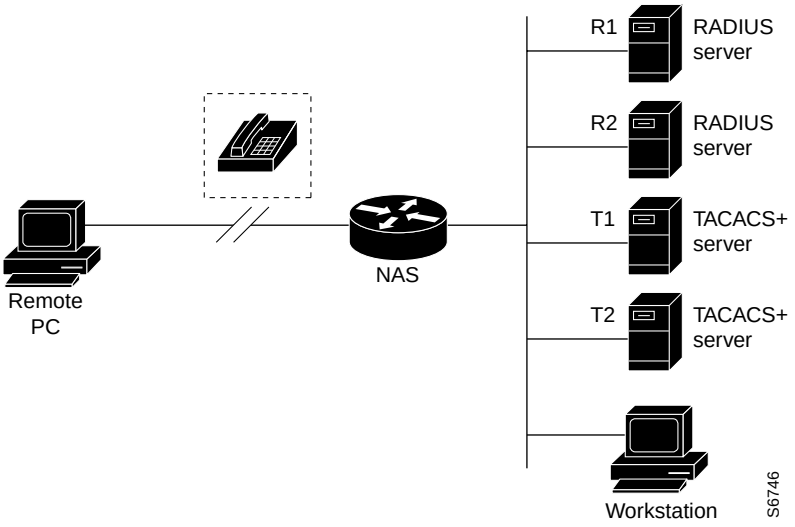
Method Lists

A method list is a sequential list that defines the authentication methods used to authenticate a user. Method lists enable you to designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. Cisco IOS software uses the first method listed to authenticate users; if that method does not respond, Cisco IOS software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or the authentication method list is exhausted, in which case authentication fails.

Note Cisco IOS software attempts authentication with the next listed authentication method only when there is no response from the previous method. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops and no other authentication methods are attempted.

Figure 1 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers.

Figure 1 Typical AAA Network Configuration



Suppose the system administrator has defined a method list where R1 will be contacted first for authentication information, then R2, T1, T2, and finally the local username database on the access server itself. When a remote user attempts to dial in to the network, the network access server first queries R1 for authentication information. If R1 authenticates the user, it issues a PASS response to the network access server and the user is allowed to access the network. If R1 returns a FAIL response, the user is denied access and the session is terminated. If R1 does not respond, then the network access server processes that as an ERROR and queries R2 for authentication information. This pattern continues through the remaining designated methods until the user is either authenticated or rejected, or until the session is terminated. If all of the authentication methods return errors, the network access server will process the session as a failure, and the session will be terminated.

Note A FAIL response is significantly different from an ERROR. A FAIL means that the user has not met the criteria contained in the applicable authentication database to be successfully authenticated. Authentication ends with a FAIL response. An ERROR means that the security server has not responded to an authentication query. Because of this, no authentication has been attempted. Only when an ERROR is detected will AAA select the next authentication method defined in the authentication method list.

Where to Begin

You must first decide what kind of security solution you want to implement. You need to assess the security risks in your particular network and decide on the appropriate means to prevent unauthorized entry and attack. For more information about assessing your security risks and possible security solutions, refer to the chapter “Security Overview.” Cisco recommends that you use AAA, no matter how minor your security needs might be.

This section includes the following subsections:

- Overview of the AAA Configuration Process
- Enabling AAA
- Disabling AAA

Overview of the AAA Configuration Process

Configuring AAA is relatively simple after you understand the basic process involved. To configure security on a Cisco router or access server using AAA, follow this process:

1. Enable AAA by using the **aaa new-model** global configuration command.
2. If you decide to use a separate security server, configure security protocol parameters, such as RADIUS, TACACS+, or Kerberos.
3. Define the method lists for authentication by using an AAA authentication command.
4. Apply the method lists to a particular interface or line, if required.
5. (Optional) Configure authorization using the **aaa authorization** command.
6. (Optional) Configure accounting using the **aaa accounting** command.

For a complete description of the commands used in this chapter, refer to the chapter "Authentication Commands" of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Enabling AAA

Before you can use any of the services AAA network security services provide, you must enable AAA.

Note When you enable AAA, you can no longer access the commands to configure the older protocols, TACACS or extended TACACS. If you decided to use TACACS or extended TACACS in your security solution, do not enable AAA.

To enable AAA, use the following command in global configuration mode:

Command	Purpose
`Router (config)# aaa new-model`	Enables AAA.

Disabling AAA

You can disable AAA functionality with a single command if you decide that your security needs cannot be met by AAA but can be met by using TACACS, extended TACACS, or a line security method that can be implemented without AAA. To disable AAA, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no aaa new-model`	Disables AAA.

What to Do Next

Once you have enabled AAA, you are ready to configure the other elements relating to your selected security solution. Table 3 describes AAA configuration tasks and where to find more information.

Table 3 AAA Access Control Security Solutions Methods

Task	Chapter in the *Cisco IOS Security Configuration Guide*
Configuring local login authentication	"Configuring Authentication"
Controlling login using security server authentication	"Configuring Authentication"
Defining method lists for authentication	"Configuring Authentication"
Applying method lists to a particular interface or line	"Configuring Authentication"
Configuring RADIUS security protocol parameters	"Configuring RADIUS"
Configuring TACACS+ security protocol parameters	"Configuring TACACS+"
Configuring Kerberos security protocol parameters	"Configuring Kerberos"
Enabling TACACS+ authorization	"Configuring Authorization"
Enabling RADIUS authorization	"Configuring Authorization"
Viewing supported IETF RADIUS attributes	"RADIUS Attributes" (Appendix)
Viewing supported vendor-specific RADIUS attributes	"RADIUS Attributes" (Appendix)
Viewing supported TACACS+ AV pairs	"TACACS+ AV Pairs" (Appendix)
Enabling accounting	"Configuring Accounting"

If you have elected not to use the AAA security services, see the "Configuring Authentication" chapter for the non-AAA configuration task "Configuring Login Authentication."

Authentication

Configuring Authentication

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Authentication verifies users before they are allowed access to the network and network services. The Cisco IOS software implementation of authentication is divided into two main categories:

- AAA Authentication Methods Configuration Task List
- Non-AAA Authentication Methods

Authentication, for the most part, is implemented through the AAA security services. Cisco recommends that, whenever possible, AAA be used to implement authentication.

This chapter describes both AAA and non-AAA authentication methods. For authentication configuration examples, refer to the "Authentication Examples" section at the end of this chapter. For a complete description of the AAA commands used in this chapter, refer to the "Authentication, Authorization, and Accounting (AAA)" part of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature, or refer to the software release notes for a specific release. For more information, see the section "Identifying Supported Platforms" in the chapter "Using Cisco IOS Software."

In This Chapter

This chapter contains the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Named Method Lists for Authentication

To configure AAA authentication, you must first define a named list of authentication methods, and then apply that list to various interfaces. The method list defines the types of authentication to be performed and the sequence in which they will be performed; it must be applied to a specific interface before any of the defined authentication methods will be performed. The only exception is the default method list (which is named "default"). The default method list is automatically applied to all interfaces except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list is a sequential list describing the authentication methods to be queried in order to authenticate a user. Method lists enable you to designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. Cisco IOS software uses the first listed method to authenticate users. If that method fails to respond, the Cisco IOS software selects the next authentication method listed in the method list. This process continues until there is successful communication with a listed authentication method, or all methods defined in the method list are exhausted.

It is important to note that the Cisco IOS software attempts authentication with the next listed authentication method only when there is no response from the previous method. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops and no other authentication methods are attempted.

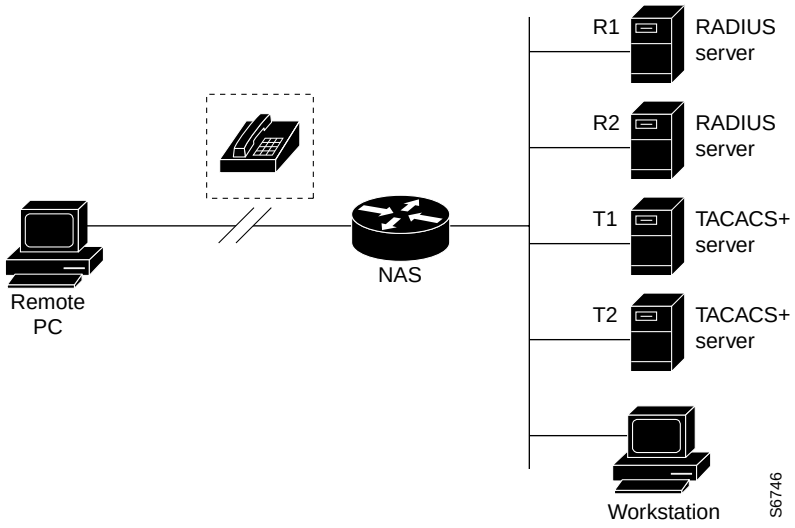
Note Effective with Cisco IOS Release 12.3, the number of AAA method lists that can be configured is 250.

This section contains the following subsections:

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. Figure 2 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers and T1 and T2 are TACACS+ servers. R1 and R2 make up the group of RADIUS servers. T1 and T2 make up the group of TACACS+ servers.

Figure 2 Typical AAA Network Configuration



Using server groups, you can specify a subset of the configured server hosts and use them for a particular service. For example, server groups allow you to define R1 and R2 as a server group, and define T1 and T2 as a separate server group. For example, you can specify R1 and T1 in the method list for authentication login, while specifying R2 and T2 in the method list for PPP authentication.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, authentication—the second host entry configured acts as failover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order in which they are configured.)

For more information about configuring server groups and about configuring server groups based on Dialed Number Identification Service (DNIS) numbers, refer to the “Configuring RADIUS” or “Configuring TACACS+” chapter.

Method List Examples

Suppose the system administrator has decided on a security solution where all interfaces will use the same authentication methods to authenticate PPP connections. In the RADIUS group, R1 is contacted first for authentication information, then if there is no response, R2 is contacted. If R2 does not respond, T1 in the TACACS+ group is contacted; if T1 does not respond, T2 is contacted. If all designated servers fail to respond, authentication falls to the local username database on the access server itself. To implement this solution, the system administrator would create a default method list by entering the following command:

```
aaa authentication ppp default group radius group tacacs+ local
```

In this example, “default” is the name of the method list. The protocols included in this method list are listed after the name, in the order they are to be queried. The default list is automatically applied to all interfaces.

When a remote user attempts to dial in to the network, the network access server first queries R1 for authentication information. If R1 authenticates the user, it issues a PASS response to the network access server and the user is allowed to access the network. If R1 returns a FAIL response, the user is denied access and the session is terminated. If R1 does not respond, then the network access server processes that as an ERROR and queries R2 for authentication information. This pattern would continue through the remaining designated methods until the user is either authenticated or rejected, or until the session is terminated.

It is important to remember that a FAIL response is significantly different from an ERROR. A FAIL means that the user has not met the criteria contained in the applicable authentication database to be successfully authenticated. Authentication ends with a FAIL response. An ERROR means that the security server has not responded to an authentication query. Because of this, no authentication has been attempted. Only when an ERROR is detected will AAA select the next authentication method defined in the authentication method list.

Suppose the system administrator wants to apply a method list only to a particular interface or set of interfaces. In this case, the system administrator creates a named method list and then applies this named list to the applicable interfaces. The following example shows how the system administrator can implement an authentication method that will be applied only to interface 3:

```
aaa authentication ppp default group radius group tacacs+ local
aaa authentication ppp apple group radius group tacacs+ local none
 interface async 3
 ppp authentication chap apple
```

In this example, “apple” is the name of the method list, and the protocols included in this method list are listed after the name in the order in which they are to be performed. After the method list has been created, it is applied to the appropriate interface. Note that the method list name (apple) in both the AAA and PPP authentication commands must match.

In the following example, the system administrator uses server groups to specify that only R2 and T2 are valid servers for PPP authentication. To do this, the administrator must define specific server groups whose members are R2 (172.16.2.7) and T2 (172.16.2.77), respectively. In this example, the RADIUS server group “rad2only” is defined as follows using the **aaa group server** command:

```
aaa group server radius rad2only
 server 172.16.2.7
```

The TACACS+ server group “tac2only” is defined as follows using the **aaa group server** command:

```
aaa group server tacacs+ tac2only
 server 172.16.2.77
```

The administrator then applies PPP authentication using the server groups. In this example, the default methods list for PPP authentication follows this order: **group rad2only**, **group tac2only**, and **local**:

```
aaa authentication ppp default group rad2only group tac2only local
```

AAA Authentication General Configuration Procedure

To configure AAA authentication, perform the following tasks:

1. Enable AAA by using the **aaa new-model** global configuration command. For more information about configuring AAA, refer to the chapter “AAA Overview”.
2. Configure security protocol parameters, such as RADIUS, TACACS+, or Kerberos if you are using a security server. For more information about RADIUS, refer to the chapter “Configuring RADIUS”. For more information about TACACS+, refer to the chapter “Configuring TACACS+”. For more information about Kerberos, refer to the chapter “Configuring Kerberos”.

3. Define the method lists for authentication by using an AAA authentication command.
4. Apply the method lists to a particular interface or line, if required.

AAA Authentication Methods Configuration Task List

This section discusses the following AAA authentication methods:

- Configuring Login Authentication Using AAA
- Configuring PPP Authentication Using AAA
- Configuring AAA Scalability for PPP Requests
- Configuring ARAP Authentication Using AAA
- Configuring NASI Authentication Using AAA
- Specifying the Amount of Time for Login Input
- Enabling Password Protection at the Privileged Level
- Changing the Text Displayed at the Password Prompt
- Configuring Message Banners for AAA Authentication
- Configuring AAA Packet of Disconnect
- Enabling Double Authentication
- Enabling Automated Double Authentication

Note AAA features are not available for use until you enable AAA globally by issuing the **aaa new-model** command. For more information about enabling AAA, refer to the "AAA Overview" chapter.

For authentication configuration examples using the commands in this chapter, refer to the section "Authentication Examples" at the end of the this chapter.

Configuring Login Authentication Using AAA

The AAA security services facilitate a variety of login authentication methods. Use the **aaa authentication login** command to enable AAA authentication no matter which of the supported login authentication methods you decide to use. With the **aaa authentication login** command, you create one or more lists of authentication methods that are tried at login. These lists are applied using the **login authentication** line configuration command.

To configure login authentication by using AAA, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA globally.
Step 2	`Router(config)# aaa authentication login {default | list-name} method1 [method2...]`	Creates a local authentication list.

	Command	Purpose
Step 3	`Router(config)# line [aux | console | tty | vty] line-number [ending-line-number]`	Enters line configuration mode for the lines to which you want to apply the authentication list.
Step 4	`Router(config-line)# login authentication {default | list-name}`	Applies the authentication list to a line or set of lines.

The *list-name* is a character string used to name the list you are creating. The method argument refers to the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

For example, to specify that authentication should succeed even if (in this example) the TACACS+ server returns an error, enter the following command:

```
aaa authentication login default group tacacs+ none
```

Note Because the **none** keyword enables *any* user logging in to successfully authenticate, it should be used only as a backup method of authentication.

To create a default list that is used when a named list is *not* specified in the **login authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces.

For example, to specify RADIUS as the default method for user authentication during login, enter the following command:

```
aaa authentication login default group radius
```

Table 4 lists the supported login authentication methods.

Table 4 AAA Authentication Login Methods

Keyword	Description
enable	Uses the enable password for authentication.
krb5	Uses Kerberos 5 for authentication.
krb5-telnet	Uses Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router. If selected, this keyword must be listed as the first method in the method list.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

Note The **login** command only changes username and privilege level but does not execute a shell; therefore autocommands will not be executed. To execute autocommands under this circumstance, you need to establish a Telnet session back into the router (loop-back). Make sure that the router has been configured for secure Telnet sessions if you choose to implement autocommands this way.

This section includes the following sections:

Login Authentication Using Enable Password

Use the **aaa authentication login** command with the **enable** *method* keyword to specify the enable password as the login authentication method. For example, to specify the enable password as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default enable
```

Before you can use the enable password as the login authentication method, you need to define the enable password. For more information about defining enable passwords, refer to the chapter "Configuring Passwords and Privileges."

Login Authentication Using Kerberos

Authentication via Kerberos is different from most other authentication methods: the user's password is never sent to the remote access server. Remote users logging in to the network are prompted for a username. If the key distribution center (KDC) has an entry for that user, it creates an encrypted ticket granting ticket (TGT) with the password for that user and sends it back to the router. The user is then prompted for a password, and the router attempts to decrypt the TGT with that password. If it succeeds, the user is authenticated and the TGT is stored in the user's credential cache on the router.

While **krb5** does use the KINIT program, a user does not need to run the KINIT program to get a TGT to authenticate to the router. This is because KINIT has been integrated into the login procedure in the Cisco IOS implementation of Kerberos.

Use the **aaa authentication login** command with the **krb5** *method* keyword to specify Kerberos as the login authentication method. For example, to specify Kerberos as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default krb5
```

Before you can use Kerberos as the login authentication method, you need to enable communication with the Kerberos security server. For more information about establishing communication with a Kerberos server, refer to the chapter "Configuring Kerberos."

Login Authentication Using Line Password

Use the **aaa authentication login** command with the **line** *method* keyword to specify the line password as the login authentication method. For example, to specify the line password as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default line
```

Before you can use a line password as the login authentication method, you need to define a line password. For more information about defining line passwords, refer to the section "Configuring Line Password Protection" in this chapter.

Login Authentication Using Local Password

Use the **aaa authentication login** command with the **local** *method* keyword to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default local
```

For information about adding users into the local username database, refer to the section "Establishing Username Authentication" in this chapter.

Login Authentication Using Group RADIUS

Use the **aaa authentication login** command with the **group radius** *method* to specify RADIUS as the login authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group radius
```

Before you can use RADIUS as the login authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS."

Configuring RADIUS Attribute 8 in Access Requests

Once you have used the **aaa authentication login** command to specify RADIUS and your login host has been configured to request its IP address from the NAS, you can send attribute 8 (Framed-IP-Address) in access-request packets by using the **radius-server attribute 8 include-in-access-req** command in global configuration mode. This command makes it possible for a NAS to provide the RADIUS server with a hint of the user IP address in advance of user authentication. For more information about attribute 8, refer to the appendix "RADIUS Attributes" at the end of the book.

Login Authentication Using Group TACACS+

Use the **aaa authentication login** command with the **group tacacs+** *method* to specify TACACS+ as the login authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group tacacs+
```

Before you can use TACACS+ as the login authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Login Authentication Using group group-name

Use the **aaa authentication login** command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the login authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group loginrad**:

```
aaa group server radius loginrad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *loginrad*.

To specify **group loginrad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group loginrad
```

Before you can use a group name as the login authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS." For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Configuring PPP Authentication Using AAA

Many users access network access servers through dialup via async or ISDN. Dialup via async or ISDN bypasses the CLI completely; instead, a network protocol (such as PPP or ARA) starts as soon as the connection is established.

The AAA security services facilitate a variety of authentication methods for use on serial interfaces running PPP. Use the **aaa authentication ppp** command to enable AAA authentication no matter which of the supported PPP authentication methods you decide to use.

To configure AAA authentication methods for serial lines using PPP, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA globally.
Step 2	`Router(config)# aaa authentication ppp {default | list-name} method1 [method2...]`	Creates a local authentication list.

	Command	Purpose
Step 3	`Router(config)# interface interface-type interface-number`	Enters interface configuration mode for the interface to which you want to apply the authentication list.
Step 4	`Router(config-if)# ppp authentication {protocol1 [protocol2...]} [if-needed] {default | list-name} [callin] [one-time][optional]`	Applies the authentication list to a line or set of lines. In this command, *protocol1* and *protocol2* represent the following protocols: CHAP, MS-CHAP, and PAP. PPP authentication is attempted first using the first authentication method, specified by *protocol1*. If *protocol1* is unable to establish authentication, the next configured protocol is used to negotiate authentication.

With the **aaa authentication ppp** command, you create one or more lists of authentication methods that are tried when a user tries to authenticate via PPP. These lists are applied using the **ppp authentication** line configuration command.

To create a default list that is used when a named list is *not* specified in the **ppp authentication** command, use the **default** keyword followed by the methods you want used in default situations.

For example, to specify the local username database as the default method for user authentication, enter the following command:

```
aaa authentication ppp default local
```

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

For example, to specify that authentication should succeed even if (in this example) the TACACS+ server returns an error, enter the following command:

```
aaa authentication ppp default group tacacs+ none
```

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

Table 5 lists the supported login authentication methods.

Table 5 AAA Authentication PPP Methods

Keyword	Description
if-needed	Does not authenticate if user has already been authenticated on a TTY line.
krb5	Uses Kerberos 5 for authentication (can only be used for PAP authentication).
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.

Table 5 AAA Authentication PPP Methods (continued)

Keyword	Description
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

This section includes the following sections:

- PPP Authentication Using Kerberos
- PPP Authentication Using Local Password
- PPP Authentication Using Group RADIUS
- PPP Authentication Using Group TACACS+
- PPP Authentication Using group group-name

PPP Authentication Using Kerberos

Use the **aaa authentication ppp** command with the **krb5** *method* keyword to specify Kerberos as the authentication method for use on interfaces running PPP. For example, to specify Kerberos as the method of user authentication when no other method list has been defined, enter the following command:

```
aaa authentication ppp default krb5
```

Before you can use Kerberos as the PPP authentication method, you need to enable communication with the Kerberos security server. For more information about establishing communication with a Kerberos server, refer to the chapter “Configuring Kerberos”.

Note Kerberos login authentication works only with PPP PAP authentication.

PPP Authentication Using Local Password

Use the **aaa authentication ppp** command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of authentication for use on lines running PPP when no other method list has been defined, enter the following command:

```
aaa authentication ppp default local
```

For information about adding users into the local username database, refer to the section “Establishing Username Authentication” in this chapter.

PPP Authentication Using Group RADIUS

Use the **aaa authentication ppp** command with the **group radius** *method* to specify RADIUS as the login authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group radius
```

Before you can use RADIUS as the PPP authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS."

Configuring RADIUS Attribute 44 in Access Requests

Once you have used the **aaa authentication ppp** command with the **group radius** *method* to specify RADIUS as the login authentication method, you can configure your router to send attribute 44 (Acct-Seccion-ID) in access-request packets by using the **radius-server attribute 44 include-in-access-req** command in global configuration mode. This command allows the RADIUS daemon to track a call from the beginning of the call to the end of the call. For more information on attribute 44, refer to the appendix "RADIUS Attributes" at the end of the book.

PPP Authentication Using Group TACACS+

Use the **aaa authentication ppp** command with the **group tacacs+** *method* to specify TACACS+ as the login authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group tacacs+
```

Before you can use TACACS+ as the PPP authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

PPP Authentication Using group group-name

Use the **aaa authentication ppp** command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the login authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group ppprad**:

```
aaa group server radius ppprad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *ppprad*.

To specify **group ppprad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group ppprad
```

Before you can use a group name as the PPP authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS". For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Configuring AAA Scalability for PPP Requests

You can configure and monitor the number of background processes allocated by the PPP manager in the network access server (NAS) to deal with AAA authentication and authorization requests. In previous Cisco IOS releases, only one background process was allocated to handle all AAA requests for PPP. This meant that parallelism in AAA servers could not be fully exploited. The AAA Scalability feature enables you to configure the number of processes used to handle AAA requests for PPP, thus increasing the number of users that can be simultaneously authenticated or authorized.

To allocate a specific number of background processes to handle AAA requests for PPP, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa processes number`	Allocates a specific number of background processes to handle AAA authentication and authorization requests for PPP.

The argument *number* defines the number of background processes earmarked to process AAA authentication and authorization requests for PPP and can be configured for any value from 1 to 2147483647. Because of the way the PPP manager handles requests for PPP, this argument also defines the number of new users that can be simultaneously authenticated. This argument can be increased or decreased at any time.

Note Allocating additional background processes can be expensive. You should configure the minimum number of background processes capable of handling the AAA requests for PPP.

Configuring ARAP Authentication Using AAA

With the **aaa authentication arap** command, you create one or more lists of authentication methods that are tried when AppleTalk Remote Access Protocol (ARAP) users attempt to log in to the router. These lists are used with the **arap authentication** line configuration command.

Use the following commands starting in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA globally.
Step 2	`Router(config)# aaa authentication arap {default | list-name} method1 [method2...]`	Enables authentication for ARAP users.
Step 3	`Router(config)# line number`	(Optional) Changes to line configuration mode.
Step 4	`Router(config-line)# autoselect arap`	(Optional) Enables autoselection of ARAP.
Step 5	`Router(config-line)# autoselect during-login`	(Optional) Starts the ARAP session automatically at user login.
Step 6	`Router(config-line)# arap authentication list-name`	(Optional—not needed if **default** is used in the **aaa authentication arap** command) Enables TACACS+ authentication for ARAP on a line.

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered.

To create a default list that is used when a named list is *not* specified in the **arap authentication** command, use the **default** keyword followed by the methods you want to be used in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

Table 6 lists the supported login authentication methods.

Table 6 AAA Authentication ARAP Methods

Keyword	Description
auth-guest	Allows guest logins only if the user has already logged in to EXEC.
guest	Allows guest logins.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

For example, to create a default AAA authentication method list used with ARAP, enter the following command:

```
aaa authentication arap default if-needed none
```

To create the same authentication method list for ARAP but name the list *MIS-access*, enter the following command:

```
aaa authentication arap MIS-access if-needed none
```

This section includes the following sections:

- ARAP Authentication Allowing Authorized Guest Logins
- ARAP Authentication Allowing Guest Logins
- ARAP Authentication Using Line Password
- ARAP Authentication Using Local Password
- ARAP Authentication Using Group RADIUS
- ARAP Authentication Using Group TACACS+
- ARAP Authentication Using Group group-name

ARAP Authentication Allowing Authorized Guest Logins

Use the **aaa authentication arap** command with the **auth-guest** keyword to allow guest logins only if the user has already successfully logged in to the EXEC. This method must be the first listed in the ARAP authentication method list but it can be followed by other methods if it does not succeed. For example, to allow all authorized guest logins—meaning logins by users who have already successfully logged in to the EXEC—as the default method of authentication, using RADIUS only if that method fails, enter the following command:

```
aaa authentication arap default auth-guest group radius
```

For more information about ARAP authorized guest logins, refer to the chapter "Configuring AppleTalk" in the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

Note By default, guest logins through ARAP are disabled when you initialize AAA. To allow guest logins, you must use the **aaa authentication arap** command with either the **guest** or the **auth-guest** keyword.

ARAP Authentication Allowing Guest Logins

Use the **aaa authentication arap** command with the **guest** keyword to allow guest logins. This method must be the first listed in the ARAP authentication method list but it can be followed by other methods if it does not succeed. For example, to allow all guest logins as the default method of authentication, using RADIUS only if that method fails, enter the following command:

```
aaa authentication arap default guest group radius
```

For more information about ARAP guest logins, refer to the chapter "Configuring AppleTalk" in the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

ARAP Authentication Using Line Password

Use the **aaa authentication arap** command with the *method* keyword **line** to specify the line password as the authentication method. For example, to specify the line password as the method of ARAP user authentication when no other method list has been defined, enter the following command:

```
aaa authentication arap default line
```

Before you can use a line password as the ARAP authentication method, you need to define a line password. For more information about defining line passwords, refer to the section "Configuring Line Password Protection" in this chapter.

ARAP Authentication Using Local Password

Use the **aaa authentication arap** command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of ARAP user authentication when no other method list has been defined, enter the following command:

```
aaa authentication arap default local
```

For information about adding users to the local username database, refer to the section "Establishing Username Authentication" in this chapter.

ARAP Authentication Using Group RADIUS

Use the **aaa authentication arap** command with the **group radius** *method* to specify RADIUS as the ARAP authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group radius
```

Before you can use RADIUS as the ARAP authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS."

ARAP Authentication Using Group TACACS+

Use the **aaa authentication arap** command with the **group tacacs+** *method* to specify TACACS+ as the ARAP authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group tacacs+
```

Before you can use TACACS+ as the ARAP authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

ARAP Authentication Using Group group-name

Use the **aaa authentication arap** command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the ARAP authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group araprad**:

```
aaa group server radius araprad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *araprad*.

To specify **group araprad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group araprad
```

Before you can use a group name as the ARAP authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS." For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Configuring NASI Authentication Using AAA

With the **aaa authentication nasi** command, you create one or more lists of authentication methods that are tried when NetWare Asynchronous Services Interface (NASI) users attempt to log in to the router. These lists are used with the **nasi authentication** line configuration command.

To configure NASI authentication using AAA, use the following commands starting in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA globally.
Step 2	`Router(config)# aaa authentication nasi {default | list-name} method1 [method2...]`	Enables authentication for NASI users.
Step 3	`Router(config)# line number`	(Optional—not needed if **default** is used in the **aaa authentication nasi** command) Enters line configuration mode.
Step 4	`Router(config-line)# nasi authentication list-name`	(Optional—not needed if **default** is used in the **aaa authentication nasi** command) Enables authentication for NASI on a line.

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered.

To create a default list that is used when a named list is *not* specified in the **aaa authentication nasi** command, use the **default** keyword followed by the methods you want to be used in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

Table 7 lists the supported NASI authentication methods.

Table 7 AAA Authentication NASI Methods

Keyword	Description
enable	Uses the enable password for authentication.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

This section includes the following sections:

- NASI Authentication Using Enable Password
- NASI Authentication Using Line Password
- NASI Authentication Using Local Password

- NASI Authentication Using Group RADIUS
- NASI Authentication Using Group TACACS+
- NASI Authentication Using group group-name

NASI Authentication Using Enable Password

Use the **aaa authentication nasi** command with the *method* keyword **enable** to specify the enable password as the authentication method. For example, to specify the enable password as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default enable
```

Before you can use the enable password as the authentication method, you need to define the enable password. For more information about defining enable passwords, refer to the chapter "Configuring Passwords and Privileges."

NASI Authentication Using Line Password

Use the **aaa authentication nasi** command with the *method* keyword **line** to specify the line password as the authentication method. For example, to specify the line password as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default line
```

Before you can use a line password as the NASI authentication method, you need to define a line password. For more information about defining line passwords, refer to the section "Configuring Line Password Protection" in this chapter.

NASI Authentication Using Local Password

Use the **aaa authentication nasi** command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication information. For example, to specify the local username database as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default local
```

For information about adding users to the local username database, refer to the section "Establishing Username Authentication" in this chapter.

NASI Authentication Using Group RADIUS

Use the **aaa authentication nasi** command with the **group radius** *method* to specify RADIUS as the NASI authentication method. For example, to specify RADIUS as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group radius
```

Before you can use RADIUS as the NASI authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS."

NASI Authentication Using Group TACACS+

Use the **aaa authentication nasi** command with the **group tacacs+** *method* keyword to specify TACACS+ as the NASI authentication method. For example, to specify TACACS+ as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group tacacs+
```

Before you can use TACACS+ as the authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

NASI Authentication Using group group-name

Use the **aaa authentication nasi** command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the NASI authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group nasirad**:

```
aaa group server radius nasirad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *nasirad*.

To specify **group nasirad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group nasirad
```

Before you can use a group name as the NASI authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS". For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Specifying the Amount of Time for Login Input

The **timeout login response** command allows you to specify how long the system will wait for login input (such as username and password) before timing out. The default login value is 30 seconds; with the **timeout login response** command, you can specify a timeout value from 1 to 300 seconds. To change the login timeout value from the default of 30 seconds, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# timeout login response` *seconds*	Specifies how long the system will wait for login information before timing out.

Enabling Password Protection at the Privileged Level

Use the **aaa authentication enable default** command to create a series of authentication methods that are used to determine whether a user can access the privileged EXEC command level. You can specify up to four authentication methods. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authentication enable default` *`method1`* `[`*`method2...`*`]`	Enables user ID and password checking for users requesting privileged EXEC level. **Note** All **aaa authentication enable default** requests sent by the router to a RADIUS server include the username "$enab15$." Requests sent to a TACACS+ server will include the username that is entered for login authentication.

The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered. Table 8 lists the supported enable authentication methods.

Table 8 AAA Authentication Enable Default Methods

Keyword	Description
enable	Uses the enable password for authentication.
line	Uses the line password for authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS hosts for authentication. **Note** The RADIUS method does not work on a per-username basis.
group tacacs+	Uses the list of all TACACS+ hosts for authentication.
group *`group-name`*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

Changing the Text Displayed at the Password Prompt

Use the **aaa authentication password-prompt** command to change the default text that the Cisco IOS software displays when prompting a user to enter a password. This command changes the password prompt for the enable password as well as for login passwords that are not supplied by remote security servers. The **no** form of this command returns the password prompt to the following default value:

```
Password:
```

The **aaa authentication password-prompt** command does not change any dialog that is supplied by a remote TACACS+ or RADIUS server.

The **aaa authentication password-prompt** command works when RADIUS is used as the login method. You will be able to see the password prompt defined in the command shown even when the RADIUS server is unreachable. The **aaa authentication password-prompt** command does not work with

TACACS+. TACACS+ supplies the NAS with the password prompt to display to the users. If the TACACS+ server is reachable, the NAS gets the password prompt from the server and uses that prompt instead of the one defined in the **aaa authentication password-prompt** command. If the TACACS+ server is not reachable, the password prompt defined in the **aaa authentication password-prompt** command may be used.

Use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authentication password-prompt text-string`	Changes the default text displayed when a user is prompted to enter a password.

Configuring Message Banners for AAA Authentication

AAA supports the use of configurable, personalized login and failed-login banners. You can configure message banners that will be displayed when a user logs in to the system to be authenticated using AAA and when, for whatever reason, authentication fails.

This section includes the following sections:

- Configuring a Login Banner
- Configuring a Failed-Login Banner

Configuring a Login Banner

To create a login banner, you need to configure a delimiting character, which notifies the system that the following text string is to be displayed as the banner, and then the text string itself. The delimiting character is repeated at the end of the text string to signify the end of the banner. The delimiting character can be any single character in the extended ASCII character set, but once defined as the delimiter, that character cannot be used in the text string making up the banner.

To configure a banner that will be displayed whenever a user logs in (replacing the default message for login), use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA.
Step 2	`Router(config)# aaa authentication banner delimiter string delimiter`	Creates a personalized login banner.

The maximum number of characters that can be displayed in the login banner is 2996 characters.

Configuring a Failed-Login Banner

To create a failed-login banner, you need to configure a delimiting character, which notifies the system that the following text string is to be displayed as the banner, and then the text string itself. The delimiting character is repeated at the end of the text string to signify the end of the failed-login banner. The delimiting character can be any single character in the extended ASCII character set, but once defined as the delimiter, that character cannot be used in the text string making up the banner.

To configure a message that will be displayed whenever a user fails login (replacing the default message for failed login), use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA.
Step 2	`Router(config)# aaa authentication fail-message` *`delimiter string delimiter`*	Creates a message to be displayed when a user fails login.

The maximum number of characters that can be displayed in the failed-login banner is 2996 characters.

Configuring AAA Packet of Disconnect

Packet of disconnect (POD) terminates connections on the network access server (NAS) when particular session attributes are identified. By using session information obtained from AAA, the POD client residing on a UNIX workstation sends disconnect packets to the POD server running on the network access server. The NAS terminates any inbound user session with one or more matching key attributes. It rejects requests when required fields are missing or when an exact match is not found.

To configure POD, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa accounting network default start-stop radius`	Enables AAA accounting records.
Step 2	`Router(config)# aaa accounting delay-start`	(Optional) Delays generation of the start accounting record until the Framed-IP-Address is assigned, allowing its use in the POD packet.
Step 3	`Router(config)# aaa pod server server-key` *`string`*	Enables POD reception.
Step 4	`Router(config)# radius-server host` *`IP address`* `non-standard`	Declares a RADIUS host that uses a vendor-proprietary version of RADIUS.

Enabling Double Authentication

Previously, PPP sessions could only be authenticated by using a single authentication method: either PAP or CHAP. Double authentication requires remote users to pass a second stage of authentication—after CHAP or PAP authentication—before gaining network access.

This second (“double”) authentication requires a password that is known to the user but *not* stored on the user’s remote host. Therefore, the second authentication is specific to a user, not to a host. This provides an additional level of security that will be effective even if information from the remote host is stolen. In addition, this also provides greater flexibility by allowing customized network privileges for each user.

The second stage authentication can use one-time passwords such as token card passwords, which are not supported by CHAP. If one-time passwords are used, a stolen user password is of no use to the perpetrator.

This section includes the following subsections:

- How Double Authentication Works

- Configuring Double Authentication
- Accessing the User Profile After Double Authentication

How Double Authentication Works

With double authentication, there are two authentication/authorization stages. These two stages occur after a remote user dials in and a PPP session is initiated.

In the first stage, the user logs in using the remote host name; CHAP (or PAP) authenticates the remote host, and then PPP negotiates with AAA to authorize the remote host. In this process, the network access privileges associated with the remote host are assigned to the user.

Note We suggest that the network administrator restrict authorization at this first stage to allow only Telnet connections to the local host.

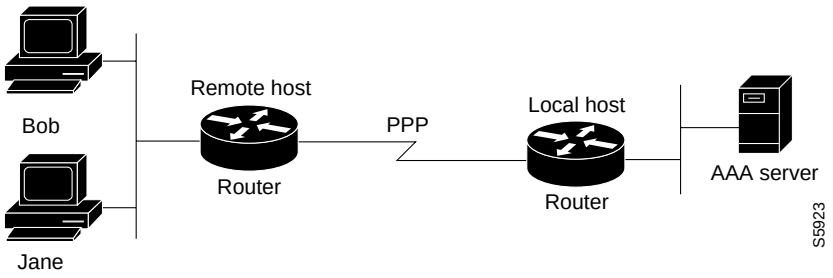
In the second stage, the remote user must Telnet to the network access server to be authenticated. When the remote user logs in, the user must be authenticated with AAA login authentication. The user then must enter the **access-profile** command to be reauthorized using AAA. When this authorization is complete, the user has been double authenticated, and can access the network according to per-user network privileges.

The system administrator determines what network privileges remote users will have after each stage of authentication by configuring appropriate parameters on a security server. To use double authentication, the user must activate it by issuing the **access-profile** command.

Caution Double authentication can cause certain undesirable events if multiple hosts share a PPP connection to a network access server, as shown in Figure 3.

First, if a user, Bob, initiates a PPP session and activates double authentication at the network access server (per Figure 3), any other user will automatically have the same network privileges as Bob until Bob's PPP session expires. This happens because Bob's authorization profile is applied to the network access server's interface during the PPP session and any PPP traffic from other users will use the PPP session Bob established.
Second, if Bob initiates a PPP session and activates double authentication, and then—before Bob's PPP session has expired—another user, Jane, executes the **access-profile** command (or, if Jane Telnets to the network access server and **autocommand access-profile** is executed), a reauthorization will occur and Jane's authorization profile will be applied to the interface—replacing Bob's profile. This can disrupt or halt Bob's PPP traffic, or grant Bob additional authorization privileges Bob should not have.

Figure 3 Possibly Risky Topology: Multiple Hosts Share a PPP Connection to a Network Access Server



Configuring Double Authentication

To configure double authentication, you must complete the following steps:

1. Enable AAA by using the **aaa-new model** global configuration command. For more information about enabling AAA, refer to the chapter "AAA Overview."
2. Use the **aaa authentication** command to configure your network access server to use login and PPP authentication method lists, then apply those method lists to the appropriate lines or interfaces.
3. Use the **aaa authorization** command to configure AAA network authorization at login. For more information about configuring network authorization, refer to the "Configuring Authorization" chapter.
4. Configure security protocol parameters (for example, RADIUS or TACACS+). For more information about RADIUS, refer to the chapter "Configuring RADIUS". For more information about TACACS+, refer to the chapter "Configuring TACACS+."
5. Use access control list AV pairs on the security server that the user can connect to the local host only by establishing a Telnet connection.
6. (Optional) Configure the **access-profile** command as an autocommand. If you configure the autocommand, remote users will not have to manually enter the **access-profile** command to access authorized rights associated with their personal user profile. To learn about configuring autocommands, refer to the **autocommand** command in the *Cisco IOS Dial Technologies Command Reference: Network Services.*

Note If the **access-profile** command is configured as an autocommand, users will still have to Telnet to the local host and log in to complete double authentication.

Follow these rules when creating the user-specific authorization statements (These rules relate to the default behavior of the **access-profile** command):

- Use valid AV pairs when configuring access control list AV pairs on the security server. For a list of valid AV pairs, refer to the chapter "Authentication Commands" in the *Cisco IOS Security Command Reference*.
- If you want remote users to use the interface's existing authorization (that which existed prior to the second stage authentication/authorization), but you want them to have different access control lists (ACLs), you should specify *only* ACL AV pairs in the user-specific authorization definition. This might be desirable if you set up a default authorization profile to apply to the remote host, but want to apply specific ACLs to specific users.

- When these user-specific authorization statements are later applied to the interface, they can either be *added to* the existing interface configuration or they can *replace* the existing interface configuration—depending on which form of the **access-profile** command is used to authorize the user. You should understand how the **access-profile** command works before configuring the authorization statements.
- If you will be using ISDN or Multilink PPP, you must also configure virtual templates at the local host.

To troubleshoot double authentication, use the **debug aaa per-user** debug command. For more information about this command, refer to the *Cisco IOS Debug Command Reference*.

Accessing the User Profile After Double Authentication

In double authentication, when a remote user establishes a PPP link to the local host using the local host name, the remote host is CHAP (or PAP) authenticated. After CHAP (or PAP) authentication, PPP negotiates with AAA to assign network access privileges associated with the remote host to the user. (We suggest that privileges at this stage be restricted to allow the user to connect to the local host only by establishing a Telnet connection.)

When the user needs to initiate the second phase of double authentication, establishing a Telnet connection to the local host, the user enters a personal username and password (different from the CHAP or PAP username and password). This action causes AAA reauthentication to occur according to the personal username/password. The initial rights associated with the local host, though, are still in place. By using the **access-profile** command, the rights associated with the local host are replaced by or merged with those defined for the user in the user's profile.

To access the user profile after double authentication, use the following command in EXEC configuration mode:

Command	Purpose
`Router> access-profile [merge | replace] [ignore-sanity-checks]`	Accesses the rights associated for the user after double authentication.

If you configured the **access-profile** command to be executed as an autocommand, it will be executed automatically after the remote user logs in.

Enabling Automated Double Authentication

You can make the double authentication process easier for users by implementing automated double authentication. Automated double authentication provides all of the security benefits of double authentication, but offers a simpler, more user-friendly interface for remote users. With double authentication, a second level of user authentication is achieved when the user Telnets to the network access server or router and enters a username and password. With automated double authentication, the user does not have to Telnet to the network access server; instead the user responds to a dialog box that requests a username and password or personal identification number (PIN). To use the automated double authentication feature, the remote user hosts must be running a companion client application. As of Cisco IOS Release 12.0, the only client application software available is the Glacier Bay application server software for PCs.

Note Automated double authentication, like the existing double authentication feature, is for Multilink PPP ISDN connections only. Automated double authentication cannot be used with other protocols such as X.25 or SLIP.

Automated double authentication is an enhancement to the existing double authentication feature. To configure automated double authentication, you must first configure double authentication by completing the following steps:

1. Enable AAA by using the **aaa-new model** global configuration command. For more information about enabling AAA, refer to the chapter "AAA Overview."
2. Use the **aaa authentication** command to configure your network access server to use login and PPP authentication method lists, then apply those method lists to the appropriate lines or interfaces.
3. Use the **aaa authorization** command to configure AAA network authorization at login. For more information about configuring network authorization, refer to the chapter "Configuring Authorization."
4. Configure security protocol parameters (for example, RADIUS or TACACS+). For more information about RADIUS, refer to the chapter "Configuring RADIUS". For more information about TACACS+, refer to the chapter "Configuring TACACS+."
5. Use access control list AV pairs on the security server that the user can connect to the local host only by establishing a Telnet connection.
6. Configure the **access-profile** command as an autocommand. If you configure the autocommand, remote users will not have to manually enter the **access-profile** command to access authorized rights associated with their personal user profile. To learn about configuring autocommands, refer to the **autocommand** command in the *Cisco IOS Dial Technologies Command Reference*, Release 12.2.

Note If the **access-profile** command is configured as an autocommand, users will still have to Telnet to the local host and log in to complete double authentication.

Follow these rules when creating the user-specific authorization statements (These rules relate to the default behavior of the **access-profile** command):

- Use valid AV pairs when configuring access control list AV pairs on the security server. For a list of valid AV pairs, refer to the chapter "Authentication Commands" in the *Cisco IOS Security Command Reference*.
- If you want remote users to use the interface's existing authorization (that which existed prior to the second stage authentication/authorization), but you want them to have different access control lists (ACLs), you should specify *only* ACL AV pairs in the user-specific authorization definition. This might be desirable if you set up a default authorization profile to apply to the remote host, but want to apply specific ACLs to specific users.
- When these user-specific authorization statements are later applied to the interface, they can either be *added to* the existing interface configuration, or *replace* the existing interface configuration—depending on which form of the **access-profile** command is used to authorize the user. You should understand how the **access-profile** command works before configuring the authorization statements.
- If you will be using ISDN or Multilink PPP, you must also configure virtual templates at the local host.

To troubleshoot double authentication, use the **debug aaa per-user** debug command. For more information about this command, refer to the *Cisco IOS Debug Command Reference*.

After you have configured double authentication, you are ready to configure the automation enhancement.

To configure automated double authentication, use the following commands, starting in global configuration mode.

:

	Command	Purpose
Step 1	**Router(config)# ip trigger-authentication** [**timeout** *seconds*] [**port** *number*]	Enables automation of double authentication.
Step 2	**Router(config)# interface bri** *number* or **Router(config)# interface serial** *number*:**23**	Selects an ISDN BRI or ISDN PRI interface and enter the interface configuration mode.
Step 3	**Router(config-if)# ip trigger-authentication**	Applies automated double authentication to the interface.

To troubleshoot automated double authentication, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	**Router# show ip trigger-authentication**	Displays the list of remote hosts for which automated double authentication has been attempted (successfully or unsuccessfully).
Step 2	**Router# clear ip trigger-authentication**	Clears the list of remote hosts for which automated double authentication has been attempted. (This clears the table displayed by the **show ip trigger-authentication** command.)
Step 3	**Router# debug ip trigger-authentication**	Displays **debug** output related to automated double authentication.

Non-AAA Authentication Methods

This section discusses the following non-AAA authentication tasks:

- Configuring Line Password Protection
- Establishing Username Authentication
- Enabling CHAP or PAP Authentication
- Using MS-CHAP

Configuring Line Password Protection

You can provide access control on a terminal line by entering the password and establishing password checking. To do so, use the following commands in line configuration mode:

	Command	Purpose
Step 1	`Router(config-line)# password` *password*	Assigns a password to a terminal or other device on a line.
Step 2	`Router(config-line)# login`	Enables password checking at login.

The password checker is case sensitive and can include spaces; for example, the password "Secret" is different from the password "secret," and "two words" is an acceptable password.

You can disable line password verification by disabling password checking. To do so, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# no login`	Disables password checking or allow access to a line without password verification.

If you configure line password protection and then configure TACACS or extended TACACS, the TACACS username and password take precedence over line passwords. If you have not yet implemented a security policy, we recommend that you use AAA.

Note The **login** command only changes username and privilege level but it does not execute a shell; therefore autocommands will not be executed. To execute autocommands under this circumstance, you need to establish a Telnet session back into the router (loop-back). Make sure that the router has been configured for secure Telnet sessions if you choose to implement autocommands this way.

Establishing Username Authentication

You can create a username-based authentication system, which is useful in the following situations:

- To provide a TACACS-like username and encrypted password-authentication system for networks that cannot support TACACS
- To provide special-case logins: for example, access list verification, no password verification, autocommand execution at login, and "no escape" situations

To establish username authentication, use the following commands in global configuration mode as needed for your system configuration:

	Command	Purpose
Step 1	`Router(config)# username` *name* [`nopassword` | `password` *password* `| password` *encryption-type encrypted password*]	Establishes username authentication with encrypted passwords.
	or	or
	`Router(config)# username` *name* [`access-class` *number*]	(Optional) Establishes username authentication by access list.
Step 2	`Router(config)# username` *name* [`privilege` *level*]	(Optional) Sets the privilege level for the user.

	Command	Purpose
Step 3	`Router(config)# username` *name* [`autocommand` *command*]	(Optional) Specifies a command to be executed automatically.
Step 4	`Router(config)# username` *name* [`noescape`] [`nohangup`]	(Optional) Sets a "no escape" login environment.

The keyword **noescape** prevents users from using escape characters on the hosts to which they are connected. The **nohangup** feature does not disconnect after using the autocommand.

Caution Passwords will be displayed in clear text in your configuration unless you enable the **service password-encryption** command. For more information about the **service password-encryption** command, refer to the chapter "Passwords and Privileges Commands" in the *Cisco IOS Security Command Reference*.

Enabling CHAP or PAP Authentication

One of the most common transport protocols used in Internet service providers' (ISPs') dial solutions is the Point-to-Point Protocol (PPP). Traditionally, remote users dial in to an access server to initiate a PPP session. After PPP has been negotiated, remote users are connected to the ISP network and to the Internet.

Because ISPs want only customers to connect to their access servers, remote users are required to authenticate to the access server before they can start up a PPP session. Normally, a remote user authenticates by typing in a username and password when prompted by the access server. Although this is a workable solution, it is difficult to administer and awkward for the remote user.

A better solution is to use the authentication protocols built into PPP. In this case, the remote user dials in to the access server and starts up a minimal subset of PPP with the access server. This does not give the remote user access to the ISP's network—it merely allows the access server to talk to the remote device.

PPP currently supports two authentication protocols: Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP). Both are specified in RFC 1334 and are supported on synchronous and asynchronous interfaces. Authentication via PAP or CHAP is equivalent to typing in a username and password when prompted by the server. CHAP is considered to be more secure because the remote user's password is never sent across the connection.

PPP (with or without PAP or CHAP authentication) is also supported in dialout solutions. An access server utilizes a dialout feature when it initiates a call to a remote device and attempts to start up a transport protocol such as PPP.

See the chapter "Configuring Interfaces" in the *Cisco IOS Configuration Fundamentals Configuration Guide* for more information about CHAP and PAP.

Note To use CHAP or PAP, you must be running PPP encapsulation.

When CHAP is enabled on an interface and a remote device attempts to connect to it, the access server sends a CHAP packet to the remote device. The CHAP packet requests or "challenges" the remote device to respond. The challenge packet consists of an ID, a random number, and the host name of the local router.

When the remote device receives the challenge packet, it concatenates the ID, the remote device's password, and the random number, and then encrypts all of it using the remote device's password. The remote device sends the results back to the access server, along with the name associated with the password used in the encryption process.

When the access server receives the response, it uses the name it received to retrieve a password stored in its user database. The retrieved password should be the same password the remote device used in its encryption process. The access server then encrypts the concatenated information with the newly retrieved password—if the result matches the result sent in the response packet, authentication succeeds.

The benefit of using CHAP authentication is that the remote device's password is never transmitted in clear text. This prevents other devices from stealing it and gaining illegal access to the ISP's network.

CHAP transactions occur only at the time a link is established. The access server does not request a password during the rest of the call. (The local device can, however, respond to such requests from other devices during a call.)

When PAP is enabled, the remote router attempting to connect to the access server is required to send an authentication request. If the username and password specified in the authentication request are accepted, the Cisco IOS software sends an authentication acknowledgment.

After you have enabled CHAP or PAP, the access server will require authentication from remote devices dialing in to the access server. If the remote device does not support the enabled protocol, the call will be dropped.

To use CHAP or PAP, you must perform the following tasks:

1. Enable PPP encapsulation.
2. Enable CHAP or PAP on the interface.
3. For CHAP, configure host name authentication and the secret or password for each remote system with which authentication is required.

This section includes the following sections:

Enabling PPP Encapsulation

To enable PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# encapsulation ppp`	Enables PPP on an interface.

Enabling PAP or CHAP

To enable CHAP or PAP authentication on an interface configured for PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# ppp authentication {*protocol1* [*protocol2...*]} [**if-needed**] {**default** | *list-name*} [**callin**] [**one-time**]	Defines the authentication protocols supported and the order in which they are used. In this command, *protocol1, protocol2* represent the following protocols: CHAP, MS-CHAP, and PAP. PPP authentication is attempted first using the first authentication method, which is *protocol1*. If *protocol1* is unable to establish authentication, the next configured protocol is used to negotiate authentication.

If you configure **ppp authentication chap** on an interface, all incoming calls on that interface that initiate a PPP connection will have to be authenticated using CHAP; likewise, if you configure **ppp authentication pap**, all incoming calls that start a PPP connection will have to be authenticated via PAP. If you configure **ppp authentication chap pap**, the access server will attempt to authenticate all incoming calls that start a PPP session with CHAP. If the remote device does not support CHAP, the access server will try to authenticate the call using PAP. If the remote device does not support either CHAP or PAP, authentication will fail and the call will be dropped. If you configure **ppp authentication pap chap**, the access server will attempt to authenticate all incoming calls that start a PPP session with PAP. If the remote device does not support PAP, the access server will try to authenticate the call using CHAP. If the remote device does not support either protocol, authentication will fail and the call will be dropped. If you configure the **ppp authentication** command with the **callin** keyword, the access server will only authenticate the remote device if the remote device initiated the call.

Authentication method lists and the **one-time** keyword are only available if you have enabled AAA—they will not be available if you are using TACACS or extended TACACS. If you specify the name of an authentication method list with the **ppp authentication** command, PPP will attempt to authenticate the connection using the methods defined in the specified method list. If AAA is enabled and no method list is defined by name, PPP will attempt to authenticate the connection using the methods defined as the default. The **ppp authentication** command with the **one-time** keyword enables support for one-time passwords during authentication.

The **if-needed** keyword is only available if you are using TACACS or extended TACACS. The **ppp authentication** command with the **if-needed** keyword means that PPP will only authenticate the remote device via PAP or CHAP if they have not yet authenticated during the life of the current call. If the remote device authenticated via a standard login procedure and initiated PPP from the EXEC prompt, PPP will not authenticate via CHAP if **ppp authentication chap if-needed** is configured on the interface.

Caution If you use a *list-name* that has not been configured with the **aaa authentication ppp** command, you disable PPP on the line.

For information about adding a **username** entry for each remote system from which the local router or access server requires authentication, see the section "Establishing Username Authentication."

Inbound and Outbound Authentication

PPP supports two-way authentication. Normally, when a remote device dials in to an access server, the access server requests that the remote device prove that it is allowed access. This is known as inbound authentication. At the same time, the remote device can also request that the access server prove that it is who it says it is. This is known as outbound authentication. An access server also does outbound authentication when it initiates a call to a remote device.

Enabling Outbound PAP Authentication

To enable outbound PAP authentication, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp pap sent-username` *username* `password` *password*	Enables outbound PAP authentication.

The access server uses the username and password specified by the **ppp pap sent-username** command to authenticate itself whenever it initiates a call to a remote device or when it has to respond to a remote device's request for outbound authentication.

Refusing PAP Authentication Requests

To refuse PAP authentication from peers requesting it, meaning that PAP authentication is disabled for all calls, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp pap refuse`	Refuses PAP authentication from peers requesting PAP authentication.

If the **refuse** keyword is not used, the router will not refuse any PAP authentication challenges received from the peer.

Creating a Common CHAP Password

For remote CHAP authentication only, you can configure your router to create a common CHAP secret password to use in response to challenges from an unknown peer; for example, if your router calls a rotary of routers (either from another vendor, or running an older version of the Cisco IOS software) to which a new (that is, unknown) router has been added. The **ppp chap password** command allows you to replace several username and password configuration commands with a single copy of this command on any dialer interface or asynchronous group interface.

To enable a router calling a collection of routers to configure a common CHAP secret password, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp chap password` *secret*	Enables a router calling a collection of routers to configure a common CHAP secret password.

Refusing CHAP Authentication Requests

To refuse CHAP authentication from peers requesting it, meaning that CHAP authentication is disabled for all calls, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp chap refuse [callin]`	Refuses CHAP authentication from peers requesting CHAP authentication.

If the **callin** keyword is used, the router will refuse to answer CHAP authentication challenges received from the peer, but will still require the peer to answer any CHAP challenges the router sends.

If outbound PAP has been enabled (using the **ppp pap sent-username** command), PAP will be suggested as the authentication method in the refusal packet.

Delaying CHAP Authentication Until Peer Authenticates

To specify that the router will not authenticate to a peer requesting CHAP authentication until after the peer has authenticated itself to the router, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp chap wait` *secret*	Configures the router to delay CHAP authentication until after the peer has authenticated itself to the router.

This command (which is the default) specifies that the router will not authenticate to a peer requesting CHAP authentication until the peer has authenticated itself to the router. The **no ppp chap wait** command specifies that the router will respond immediately to an authentication challenge.

Using MS-CHAP

Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) is the Microsoft version of CHAP and is an extension of RFC 1994. Like the standard version of CHAP, MS-CHAP is used for PPP authentication; in this case, authentication occurs between a PC using Microsoft Windows NT or Microsoft Windows 95 and a Cisco router or access server acting as a network access server.

MS-CHAP differs from the standard CHAP as follows:

- MS-CHAP is enabled by negotiating CHAP Algorithm 0x80 in LCP option 3, Authentication Protocol.
- The MS-CHAP Response packet is in a format designed to be compatible with Microsoft Windows NT 3.5 and 3.51, Microsoft Windows 95, and Microsoft LAN Manager 2.*x*. This format does not require the authenticator to store a clear or reversibly encrypted password.
- MS-CHAP provides an authenticator-controlled authentication retry mechanism.
- MS-CHAP provides an authenticator-controlled change password mechanism.
- MS-CHAP defines a set of “reason-for failure” codes returned in the Failure packet message field.

Depending on the security protocols you have implemented, PPP authentication using MS-CHAP can be used with or without AAA security services. If you have enabled AAA, PPP authentication using MS-CHAP can be used in conjunction with both TACACS+ and RADIUS. Table 9 lists the vendor-specific RADIUS attributes (IETF Attribute 26) that enable RADIUS to support MS-CHAP.

Table 9 Vendor-Specific RADIUS Attributes for MS-CHAP

Vendor-ID Number	Vendor-Type Number	Vendor-Proprietary Attribute	Description
311	11	MSCHAP-Challenge	Contains the challenge sent by a network access server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets.
211	11	MSCHAP-Response	Contains the response value provided by a PPP MS-CHAP user in response to the challenge. It is only used in Access-Request packets. This attribute is identical to the PPP CHAP Identifier.

To define PPP authentication using MS-CHAP, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 2	`Router(config-if)# ppp authentication ms-chap [if-needed] [list-name | default] [callin] [one-time]`	Defines PPP authentication using MS-CHAP.

If you configure **ppp authentication ms-chap** on an interface, all incoming calls on that interface that initiate a PPP connection will have to be authenticated using MS-CHAP. If you configure the **ppp authentication** command with the **callin** keyword, the access server will only authenticate the remote device if the remote device initiated the call.

Authentication method lists and the **one-time** keyword are only available if you have enabled AAA—they will not be available if you are using TACACS or extended TACACS. If you specify the name of an authentication method list with the **ppp authentication** command, PPP will attempt to authenticate the connection using the methods defined in the specified method list. If AAA is enabled and no method list is defined by name, PPP will attempt to authenticate the connection using the methods defined as the default. The **ppp authentication** command with the **one-time** keyword enables support for one-time passwords during authentication.

The **if-needed** keyword is only available if you are using TACACS or extended TACACS. The **ppp authentication** command with the **if-needed** keyword means that PPP will only authenticate the remote device via MS-CHAP if that device has not yet authenticated during the life of the current call. If the remote device authenticated through a standard login procedure and initiated PPP from the EXEC prompt, PPP will not authenticate through MS-CHAP if **ppp authentication chap if-needed** is configured.

Note If PPP authentication using MS-CHAP is used with username authentication, you must include the MS-CHAP secret in the local username/password database. For more information about username authentication, refer to the "Establish Username Authentication" section.

Authentication Examples

The following sections provide authentication configuration examples:

- RADIUS Authentication Examples
- TACACS+ Authentication Examples
- Kerberos Authentication Examples
- AAA Scalability Example
- Login and Failed Banner Examples
- AAA Packet of Disconnect Server Key Example
- Double Authentication Examples
- Automated Double Authentication Example
- MS-CHAP Example

RADIUS Authentication Examples

This section provides two sample configurations using RADIUS.

The following example shows how to configure the router to authenticate and authorize using RADIUS:

```
aaa authentication login radius-login group radius local
aaa authentication ppp radius-ppp if-needed group radius
aaa authorization exec default group radius if-authenticated
aaa authorization network default group radius
line 3
login authentication radius-login
interface serial 0
ppp authentication radius-ppp
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login radius-login group radius local** command configures the router to use RADIUS for authentication at the login prompt. If RADIUS returns an error, the user is authenticated using the local database.
- The **aaa authentication ppp radius-ppp if-needed group radius** command configures the Cisco IOS software to use PPP authentication using CHAP or PAP if the user has not already logged in. If the EXEC facility has authenticated the user, PPP authentication is not performed.
- The **aaa authorization exec default group radius if-authenticated** command queries the RADIUS database for information that is used during EXEC authorization, such as autocommands and privilege levels, but only provides authorization if the user has successfully authenticated.
- The **aaa authorization network default group radius** command queries RADIUS for network authorization, address assignment, and other access lists.
- The **login authentication radius-login** command enables the radius-login method list for line 3.

- The **ppp authentication radius-ppp** command enables the radius-ppp method list for serial interface 0.

The following example shows how to configure the router to prompt for and verify a username and password, authorize the user's EXEC level, and specify it as the method of authorization for privilege level 2. In this example, if a local username is entered at the username prompt, that username is used for authentication.

If the user is authenticated using the local database, EXEC authorization using RADIUS will fail because no data is saved from the RADIUS authentication. The method list also uses the local database to find an autocommand. If there is no autocommand, the user becomes the EXEC user. If the user then attempts to issue commands that are set at privilege level 2, TACACS+ is used to attempt to authorize the command.

```
aaa authentication login default group radius local
aaa authorization exec default group radius local
aaa authorization command 2 default group tacacs+ if-authenticated
radius-server host 172.16.71.146 auth-port 1645 acct-port 1646
radius-server attribute 44 include-in-access-req
radius-server attribute 8 include-in-access-req
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login default group radius local** command specifies that the username and password are verified by RADIUS or, if RADIUS is not responding, by the router's local user database.
- The **aaa authorization exec default group radius local** command specifies that RADIUS authentication information be used to set the user's EXEC level if the user authenticates with RADIUS. If no RADIUS information is used, this command specifies that the local user database be used for EXEC authorization.
- The **aaa authorization command 2 default group tacacs+ if-authenticated** command specifies TACACS+ authorization for commands set at privilege level 2, if the user has already successfully authenticated.
- The **radius-server host 172.16.71.146 auth-port 1645 acct-port 1646** command specifies the IP address of the RADIUS server host, the UDP destination port for authentication requests, and the UDP destination port for accounting requests.
- The **radius-server attribute 44 include-in-access-req** command sends RADIUS attribute 44 (Acct-Seccion-ID) in access-request packets.
- The **radius-server attribute 8 include-in-access-req** command sends RADIUS attribute 8 (Framed-IP-Address) in access-request packets.

TACACS+ Authentication Examples

The following example shows how to configure TACACS+ as the security protocol to be used for PPP authentication:

```
aaa new-model
aaa authentication ppp test group tacacs+ local
interface serial 0
ppp authentication chap pap test
tacacs-server host 10.1.2.3
tacacs-server key goaway
```

The lines in this sample TACACS+ authentication configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "test," to be used on serial interfaces running PPP. The keywords **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **interface** command selects the line.
- The **ppp authentication** command applies the test method list to this line.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3.
- The **tacacs-server key** command defines the shared encryption key to be "goaway."

The following example shows how to configure AAA authentication for PPP:

```
aaa authentication ppp default if-needed group tacacs+ local
```

In this example, the keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP is not necessary and can be skipped. If authentication is needed, the keywords **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.

The following example shows how to create the same authentication algorithm for PAP, but it calls the method list "MIS-access" instead of "default":

```
aaa authentication ppp MIS-access if-needed group tacacs+ local
interface serial 0
ppp authentication pap MIS-access
```

In this example, because the list does not apply to any interfaces (unlike the default list, which applies automatically to all interfaces), the administrator must select interfaces to which this authentication scheme should apply by using the **interface** command. The administrator must then apply this method list to those interfaces by using the **ppp authentication** command.

Kerberos Authentication Examples

To specify Kerberos as the login authentication method, use the following command:

```
aaa authentication login default krb5
```

To specify Kerberos authentication for PPP, use the following command:

```
aaa authentication ppp default krb5
```

AAA Scalability Example

The following example shows a general security configuration using AAA with RADIUS as the security protocol. In this example, the network access server is configured to allocate 16 background processes to handle AAA requests for PPP.

```
aaa new-model
radius-server host alcatraz
radius-server key myRaDiUSpassWoRd
radius-server configure-nas
username root password ALongPassword
```

```
aaa authentication ppp dialins group radius local
aaa authentication login admins local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa processes 16
line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication pap dialins
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **radius-server configure-nas** command defines that the Cisco router or access server will query the RADIUS server for static routes and IP pool definitions when the device first starts up.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list "dialins," which specifies that RADIUS authentication, then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authentication login admins local** command defines another method list, "admins," for login authentication.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **aaa processes** command allocates 16 background processes to handle AAA requests for PPP.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the "admins" method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.

- The **ppp authentication pap dialins** command applies the “dialins” method list to the specified interfaces.

Login and Failed Banner Examples

The following example shows how to configure a login banner (in this case, the phrase “Unauthorized Access Prohibited”) that will be displayed when a user logs in to the system. The asterisk (*) is used as the delimiting character. (RADIUS is specified as the default login authentication method.)

```
aaa new-model
aaa authentication banner *Unauthorized Access Prohibited*
aaa authentication login default group radius
```

This configuration produces the following login banner:

```
Unauthorized Access Prohibited
Username:
```

The following example shows how to additionally configure a failed login banner (in this case, the phrase “Failed login. Try again.”) that will be displayed when a user tries to log in to the system and fails. The asterisk (*) is used as the delimiting character. (RADIUS is specified as the default login authentication method.)

```
aaa new-model
aaa authentication banner *Unauthorized Access Prohibited*
aaa authentication fail-message *Failed login. Try again.*
aaa authentication login default group radius
```

This configuration produces the following login and failed login banner:

```
Unauthorized Access Prohibited
Username:
Password:
Failed login. Try again.
```

AAA Packet of Disconnect Server Key Example

The following example shows how to configure POD (packet of disconnect), which terminates connections on the network access server (NAS) when particular session attributes are identified.

```
aaa new-model
aaa authentication ppp default radius
aaa accounting network default start-stop radius
aaa accounting delay-start
aaa pod server server-key xyz123
radius-server host 172.16.0.0 non-standard
radius-server key rad123
```

Double Authentication Examples

The examples in this section illustrate possible configurations to be used with double authentication. Your configurations could differ significantly, depending on your network and security requirements.

This section includes the following examples:

- Configuration of the Local Host for AAA with Double Authentication Examples
- Configuration of the AAA Server for First-Stage (PPP) Authentication and Authorization Example

- Configuration of the AAA Server for Second-Stage (Per-User) Authentication and Authorization Examples
- Complete Configuration with TACACS+ Example

Note These configuration examples include specific IP addresses and other specific information. This information is for illustration purposes only: your configuration will use different IP addresses, different usernames and passwords, and different authorization statements.

Configuration of the Local Host for AAA with Double Authentication Examples

These two examples show how to configure a local host to use AAA for PPP and login authentication, and for network and EXEC authorization. One example is shown for RADIUS and one example for TACACS+.

In both examples, the first three lines configure AAA, with a specific server as the AAA server. The next two lines configure AAA for PPP and login authentication, and the last two lines configure network and EXEC authorization. The last line is necessary only if the **access-profile** command will be executed as an autocommand.

The following example shows router configuration with a RADIUS AAA server:

```
aaa new-model
radius-server host secureserver
radius-server key myradiuskey
aaa authentication ppp default group radius
aaa authentication login default group radius
aaa authorization network default group radius
aaa authorization exec default group radius
```

The following example shows router configuration with a TACACS+ server:

```
aaa new-model
tacacs-server host security
tacacs-server key mytacacskey
aaa authentication ppp default group tacacs+
aaa authentication login default group tacacs+
aaa authorization network default group tacacs+
aaa authorization exec default group tacacs+
```

Configuration of the AAA Server for First-Stage (PPP) Authentication and Authorization Example

This example shows a configuration on the AAA server. A partial sample AAA configuration is shown for RADIUS.

TACACS+ servers can be configured similarly. (See the section "Complete Configuration with TACACS+ Example" later in this chapter.)

This example defines authentication/authorization for a remote host named "hostx" that will be authenticated by CHAP in the first stage of double authentication. Note that the ACL AV pair limits the remote host to Telnet connections to the local host. The local host has the IP address 10.0.0.2.

The following example shows a partial AAA server configuration for RADIUS:

```
hostx   Password = "welcome"
        User-Service-Type = Framed-User,
        Framed-Protocol = PPP,
        cisco-avpair = "lcp:interface-config=ip unnumbered ethernet 0",
        cisco-avpair = "ip:inacl#3=permit tcp any 172.21.114.0 0.0.0.255 eq telnet",
```

```
cisco-avpair = “ip:inacl#4=deny icmp any any”,
cisco-avpair = “ip:route#5=55.0.0.0 255.0.0.0”,
cisco-avpair = “ip:route#6=66.0.0.0 255.0.0.0”,
cisco-avpair = “ipx:inacl#3=deny any”
```

Configuration of the AAA Server for Second-Stage (Per-User) Authentication and Authorization Examples

This section contains partial sample AAA configurations on a RADIUS server. These configurations define authentication and authorization for a user (Pat) with the username “patuser,” who will be user-authenticated in the second stage of double authentication.

TACACS+ servers can be configured similarly. (See the section “Complete Configuration with TACACS+ Example” later in this chapter.)

Three examples show sample RADIUS AAA configurations that could be used with each of the three forms of the **access-profile** command.

The first example shows a partial sample AAA configuration that works with the default form (no keywords) of the **access-profile** command. Note that only ACL AV pairs are defined. This example also sets up the **access-profile** command as an autocommand.

```
patuser   Password = “welcome”
          User-Service-Type = Shell-User,
          cisco-avpair = “shell:autocmd=access-profile”
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = “ip:inacl#3=permit tcp any host 10.0.0.2 eq telnet”,
          cisco-avpair = “ip:inacl#4=deny icmp any any”
```

The second example shows a partial sample AAA configuration that works with the **access-profile merge** form of the **access-profile** command. This example also sets up the **access-profile merge** command as an autocommand.

```
patuser   Password = “welcome”
          User-Service-Type = Shell-User,
          cisco-avpair = “shell:autocmd=access-profile merge”
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = “ip:inacl#3=permit tcp any any”
          cisco-avpair = “ip:route=10.0.0.0 255.255.0.0",
          cisco-avpair = “ip:route=10.1.0.0 255.255.0.0",
          cisco-avpair = “ip:route=10.2.0.0 255.255.0.0"
```

The third example shows a partial sample AAA configuration that works with the **access-profile replace** form of the **access-profile** command. This example also sets up the **access-profile replace** command as an autocommand.

```
patuser   Password = “welcome”
          User-Service-Type = Shell-User,
          cisco-avpair = “shell:autocmd=access-profile replace”
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = “ip:inacl#3=permit tcp any any”,
          cisco-avpair = “ip:inacl#4=permit icmp any any”,
          cisco-avpair = “ip:route=10.10.0.0 255.255.0.0",
          cisco-avpair = “ip:route=10.11.0.0 255.255.0.0",
          cisco-avpair = “ip:route=10.12.0.0 255.255.0.0"
```

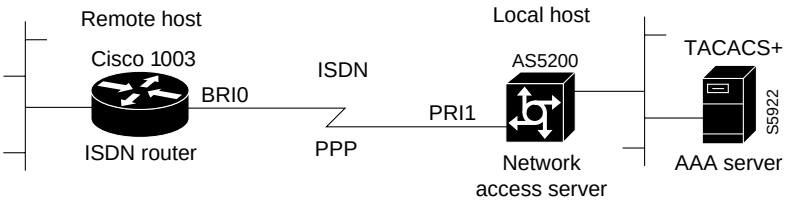
Complete Configuration with TACACS+ Example

This example shows TACACS+ authorization profile configurations both for the remote host (used in the first stage of double authentication) and for specific users (used in the second stage of double authentication). This TACACS+ example contains approximately the same configuration information as shown in the previous RADIUS examples.

This sample configuration shows authentication/authorization profiles on the TACACS+ server for the remote host “hostx” and for three users, with the usernames “pat_default,” “pat_merge,” and “pat_replace.” The configurations for these three usernames illustrate different configurations that correspond to the three different forms of the **access-profile** command. The three user configurations also illustrate setting up the autocommand for each form of the **access-profile** command.

Figure 4 shows the topology. The example that follows the figure shows a TACACS+ configuration file.

Figure 4 Example Topology for Double Authentication



This sample configuration shows authentication/authorization profiles on the TACACS+ server for the remote host “hostx” and for three users, with the usernames “pat_default,” “pat_merge,” and “pat_replace.”

```
key = “mytacacskey”

default authorization = permit


#-----------------------------Remote Host (BRI)-------------------------
#
# This allows the remote host to be authenticated by the local host
# during fist-stage authentication, and provides the remote host
# authorization profile.
#
#-----------------------------------------------------------------------

user = hostx
{
    login = cleartext “welcome”
    chap = cleartext “welcome”


    service = ppp protocol = lcp {
                interface-config=”ip unnumbered ethernet 0"
    }

    service = ppp protocol = ip {
            # It is important to have the hash sign and some string after
            # it. This indicates to the NAS that you have a per-user
            # config.

            inacl#3=”permit tcp any 172.21.114.0 0.0.0.255 eq telnet”
            inacl#4=”deny icmp any any”
```

```
            route#5=”55.0.0.0 255.0.0.0"
            route#6=”66.0.0.0 255.0.0.0"
    }

    service = ppp protocol = ipx {
            # see previous comment about the hash sign and string, in protocol = ip
            inacl#3=”deny any”
    }


}


#------------------- “access-profile” default user “only acls” ------------------
#
# Without arguments, access-profile removes any access-lists it can find
# in the old configuration (both per-user and per-interface), and makes sure
# that the new profile contains ONLY access-list definitions.
#
#-------------------------------------------------------------------------------

user = pat_default
{
        login = cleartext “welcome”
        chap = cleartext “welcome”

        service = exec

        {
                # This is the autocommand that executes when pat_default logs in.
                autocmd = “access-profile”
        }

        service = ppp protocol = ip {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!

                inacl#3=”permit tcp any host 10.0.0.2 eq telnet”
                inacl#4=”deny icmp any any”
        }


        service = ppp protocol = ipx {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }


}


#--------------------- “access-profile merge” user --------------------------
#
# With the 'merge' option, first all old access-lists are removed (as before),
# but then (almost) all AV pairs are uploaded and installed. This will allow
# for uploading any custom static routes, sap-filters, and so on, that the user
# may need in his or her profile. This needs to be used with care, as it leaves
# open the possibility of conflicting configurations.
```

```
#
#-------------------------------------------------------------------------

user = pat_merge
{
        login = cleartext “welcome”
        chap = cleartext “welcome”


        service = exec
        {
                # This is the autocommand that executes when pat_merge logs in.
                autocmd = “access-profile merge”
        }

        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!

                inacl#3=”permit tcp any any”
                route#2=”10.0.0.0 255.255.0.0"
                route#3=”10.1.0.0 255.255.0.0"
                route#4=”10.2.0.0 255.255.0.0"

        }

        service = ppp protocol = ipx
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!

        }

}


#--------------------- “access-profile replace” user ----------------------------
#
# With the 'replace' option, ALL old configuration is removed and ALL new
# configuration is installed.
#
# One caveat: access-profile checks the new configuration for address-pool and
# address AV pairs. As addresses cannot be renegotiated at this point, the
# command will fail (and complain) when it encounters such an AV pair.
# Such AV pairs are considered to be “invalid” for this context.
#-------------------------------------------------------------------------


user = pat_replace
{
        login = cleartext “welcome”
        chap = cleartext “welcome”


        service = exec
        {
                # This is the autocommand that executes when pat_replace logs in.
```

```
                autocmd = “access-profile replace”
        }

        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!

                inacl#3=”permit tcp any any”
                inacl#4=”permit icmp any any”


                route#2=”10.10.0.0 255.255.0.0"
                route#3=”10.11.0.0 255.255.0.0"
                route#4=”10.12.0.0 255.255.0.0"
        }

        service = ppp protocol = ipx
        {
                # put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }

}
```

Automated Double Authentication Example

This example shows a complete configuration file for a Cisco 2509 router with automated double authentication configured. The configuration commands that apply to automated double authentication are preceded by descriptions with a double asterisk (**).

```
Current configuration:
!
version 11.3
no service password-encryption
!
hostname myrouter
!
!
! **The following AAA commands are used to configure double authentication:
!
! **The following command enables AAA:
aaa new-model
! **The following command enables user authentication via the TACACS+ AAA server:
aaa authentication login default group tacacs+
aaa authentication login console none
! **The following command enables device authentication via the TACACS+ AAA server:
aaa authentication ppp default group tacacs+
! **The following command causes the remote user’s authorization profile to be
! downloaded from the AAA server to the Cisco 2509 router when required:
aaa authorization exec default group tacacs+
! **The following command causes the remote device’s authorization profile to be
! downloaded from the AAA server to the Cisco 2509 router when required:
aaa authorization network default group tacacs+
enable password mypassword
```

```
!
ip host blue 172.21.127.226
ip host green 172.21.127.218
ip host red 172.21.127.114
ip domain-name example.com
ip name-server 171.69.2.75
! **The following command globally enables automated double authentication:
ip trigger-authentication timeout 60 port 7500
isdn switch-type basic-5ess
!
!
interface Ethernet0
 ip address 172.21.127.186 255.255.255.248
 no ip route-cache
 no ip mroute-cache
 no keepalive
 ntp disable
 no cdp enable
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 no ip route-cache
 no ip mroute-cache
!
interface Serial0
 ip address 172.21.127.105 255.255.255.248
 encapsulation ppp
 no ip mroute-cache
 no keepalive
 shutdown
 clockrate 2000000
 no cdp enable
!
interface Serial1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no cdp enable
!
! **Automated double authentication occurs via the ISDN BRI interface BRI0:
interface BRI0
 ip unnumbered Ethernet0
! **The following command turns on automated double authentication at this interface:
 ip trigger-authentication
! **PPP encapsulation is required:
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer idle-timeout 500
 dialer map ip 172.21.127.113 name myrouter 60074
 dialer-group 1
 no cdp enable
```

```
! **The following command specifies that device authentication occurs via PPP CHAP:
 ppp authentication chap
!
router eigrp 109
 network 172.21.0.0
 no auto-summary
!
ip default-gateway 172.21.127.185
no ip classless
ip route 172.21.127.114 255.255.255.255 172.21.127.113
! **Virtual profiles are required for double authentication to work:
virtual-profile virtual-template 1
dialer-list 1 protocol ip permit
no cdp run
! **The following command defines where the TACACS+ AAA server is:
tacacs-server host 171.69.57.35 port 1049
tacacs-server timeout 90
! **The following command defines the key to use with TACACS+ traffic (required):
tacacs-server key mytacacskey
snmp-server community public RO
!
line con 0
 exec-timeout 0 0
 login authentication console
line aux 0
 transport input all
line vty 0 4
 exec-timeout 0 0
 password lab
!
end
```

MS-CHAP Example

The following example shows how to configure a Cisco AS5200 Universal Access Server (enabled for AAA and communication with a RADIUS security server) for PPP authentication using MS-CHAP:

```
aaa new-model
aaa authentication login admins local
aaa authentication ppp dialins group radius local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius

username root password ALongPassword

radius-server host alcatraz
radius-server key myRaDiUSpassWoRd

interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication ms-chap dialins

line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines another method list, “admins”, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins,” which specifies that RADIUS authentication then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication ms-chap dialins** command selects MS-CHAP as the method of PPP authentication and applies the “dialins” method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the “admins” method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

Configuring Authorization

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

AAA authorization enables you to limit the services available to a user. When AAA authorization is enabled, the network access server uses information retrieved from the user's profile, which is located either in the local user database or on the security server, to configure the user's session. Once this is done, the user will be granted access to a requested service only if the information in the user profile allows it.

For a complete description of the authorization commands used in this chapter, refer to the chapter "Authorization Commands" in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the chapter "Identifying Supported Platforms" section in the "Using Cisco IOS Software."

In This Chapter

This chapter contains the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Named Method Lists for Authorization

Method lists for authorization define the ways that authorization will be performed and the sequence in which these methods will be performed. A method list is simply a named list describing the authorization methods to be queried (such as RADIUS or TACACS+), in sequence. Method lists enable you to designate one or more security protocols to be used for authorization, thus ensuring a backup system in case the initial method fails. Cisco IOS software uses the first method listed to authorize users for specific network services; if that method fails to respond, the Cisco IOS software selects the next method listed in the method list. This process continues until there is successful communication with a listed authorization method, or all methods defined are exhausted.

Note The Cisco IOS software attempts authorization with the next listed method only when there is no response from the previous method. If authorization fails at any point in this cycle—meaning that the security server or local username database responds by denying the user services—the authorization process stops and no other authorization methods are attempted.

Method lists are specific to the authorization type requested:

- **Auth-proxy**—Applies specific security policies on a per-user basis. For detailed information on the authentication proxy feature, refer to the chapter “Configuring Authentication Proxy” in the “Traffic Filtering and Firewalls” part of this book.
- **Commands**—Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- **EXEC**—Applies to the attributes associated with a user EXEC terminal session.
- **Network**—Applies to network connections. This can include a PPP, SLIP, or ARAP connection.
- **Reverse Access**—Applies to reverse Telnet sessions.

When you create a named method list, you are defining a particular list of authorization methods for the indicated authorization type.

Once defined, method lists must be applied to specific lines or interfaces before any of the defined methods will be performed. The only exception is the default method list (which is named “default”). If the **aaa authorization** command for a particular authorization type is issued without a named method list specified, the default method list is automatically applied to all interfaces or lines except those that have a named method list explicitly defined. (A defined method list overrides the default method list.) If no default method list is defined, local authorization takes place by default.

AAA Authorization Methods

AAA supports five different methods of authorization:

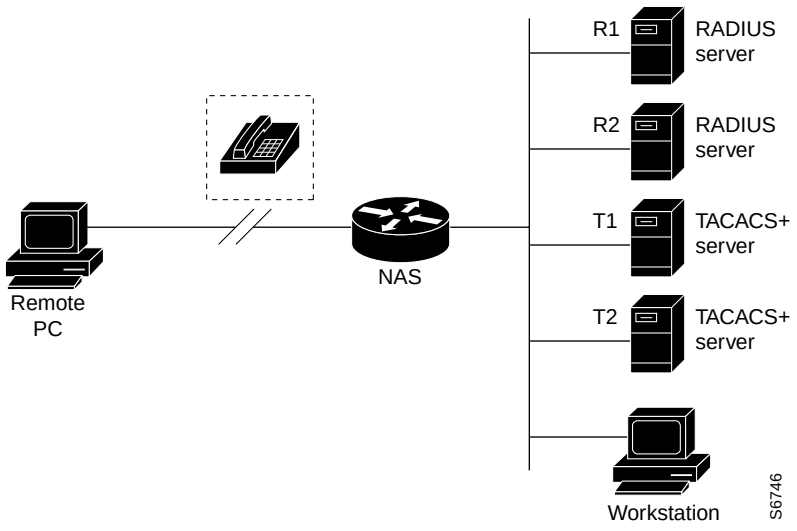
- TACACS+—The network access server exchanges authorization information with the TACACS+ security daemon. TACACS+ authorization defines specific rights for users by associating attribute-value pairs, which are stored in a database on the TACACS+ security server, with the appropriate user.

- If-Authenticated—The user is allowed to access the requested function provided the user has been authenticated successfully.
- **None**—The network access server does not request authorization information; authorization is not performed over this line/interface.
- Local—The router or access server consults its local database, as defined by the **username** command, for example, to authorize specific rights for users. Only a limited set of functions can be controlled via the local database.
- RADIUS—The network access server requests authorization information from the RADIUS security server. RADIUS authorization defines specific rights for users by associating attributes, which are stored in a database on the RADIUS server, with the appropriate user.

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. Figure 5 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers. R1 and R2 make up the group of RADIUS servers. T1 and T2 make up the group of TACACS+ servers.

Figure 5 Typical AAA Network Configuration



Using server groups, you can specify a subset of the configured server hosts and use them for a particular service. For example, server groups allow you to define R1 and R2 as separate server groups, and T1 and T2 as separate server groups. This means you can specify either R1 and T1 in the method list or R2 and T2 in the method list, which provides more flexibility in the way that you assign RADIUS and TACACS+ resources.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, authorization—the second host entry configured acts as fail-over

backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order they are configured.)

For more information about configuring server groups and about configuring server groups based on DNIS numbers, refer to the chapter “Configuring RADIUS” or the chapter “Configuring TACACS+”

AAA Authorization Types

Cisco IOS software supports five different types of authorization:

- **Auth-proxy**—Applies specific security policies on a per-user basis. For detailed information on the authentication proxy feature, refer to the “Configuring Authentication Proxy” chapter in the “Traffic Filtering and Firewalls” section of this book.
- **Commands**—Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- **EXEC**—Applies to the attributes associated with a user EXEC terminal session.
- **Network**—Applies to network connections. This can include a PPP, SLIP, or ARAP connection.
- **Reverse Access**—Applies to reverse Telnet sessions.
- Configuration—Applies to downloading configurations from the AAA server.
- IP Mobile—Applies to authorization for IP mobile services.

AAA Authorization Prerequisites

Before configuring authorization using named method lists, you must first perform the following tasks:

- Enable AAA on your network access server. For more information about enabling AAA on your Cisco router or access server, refer to the “AAA Overview” chapter.
- Configure AAA authentication. Authorization generally takes place after authentication and relies on authentication to work properly. For more information about AAA authentication, refer to the “Configuring Authentication” chapter.
- Define the characteristics of your RADIUS or TACACS+ security server if you are issuing RADIUS or TACACS+ authorization. For more information about configuring your Cisco network access server to communicate with your RADIUS security server, refer to the chapter “Configuring RADIUS”. For more information about configuring your Cisco network access server to communicate with your TACACS+ security server, refer to the chapter “Configuring TACACS+”.
- Define the rights associated with specific users by using the **username** command if you are issuing local authorization. For more information about the **username** command, refer to the *Cisco IOS Security Command Reference*.

AAA Authorization Configuration Task List

This section describes the following configuration tasks:

- Configuring AAA Authorization Using Named Method Lists

- Disabling Authorization for Global Configuration Commands
- Configuring Authorization for Reverse Telnet

For authorization configuration examples using the commands in this chapter, refer to the section "Authorization Configuration Examples" at the end of the this chapter.

Configuring AAA Authorization Using Named Method Lists

To configure AAA authorization using named method lists, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa authorization {auth-proxy | network | exec | commands level | reverse-access | configuration | ipmobile} {default | list-name} [method1 [method2...]]`	Creates an authorization method list for a particular authorization type and enable authorization.
Step 2	`Router(config)# line [aux | console | tty | vty] line-number [ending-line-number]` or Router(config)# **interface** *interface-type interface-number*	Enters the line configuration mode for the lines to which you want to apply the authorization method list. Alternately, enters the interface configuration mode for the interfaces to which you want to apply the authorization method list.
Step 3	`Router(config-line)# authorization {arap | commands level | exec | reverse-access} {default | list-name}` or `Router(config-line)# ppp authorization {default | list-name}`	Applies the authorization list to a line or set of lines. Alternately, applies the authorization list to an interface or set of interfaces.

This section includes the following sections:

- Authorization Types
- Authorization Methods

Authorization Types

Named authorization method lists are specific to the indicated type of authorization.

To create a method list to enable authorization that applies specific security policies on a per-user basis, use the auth-proxy keyword. For detailed information on the authentication proxy feature, refer to the chapter "Configuring Authentication Proxy" in the "Traffic Filtering and Firewalls" part of this book.

To create a method list to enable authorization for all network-related service requests (including SLIP, PPP, PPP NCPs, and ARAP), use the **network** keyword.

To create a method list to enable authorization to determine if a user is allowed to run an EXEC shell, use the **exec** keyword.

To create a method list to enable authorization for specific, individual EXEC commands associated with a specific privilege level, use the **commands** keyword. (This allows you to authorize all commands associated with a specified command level from 0 to 15.)

To create a method list to enable authorization for reverse Telnet functions, use the **reverse-access** keyword.

For information about the types of authorization supported by the Cisco IOS software, refer to the "AAA Authorization Types" section of this chapter.

Authorization Methods

To have the network access server request authorization information via a TACACS+ security server, use the **aaa authorization** command with the **group tacacs+** *method* keyword. For more specific information about configuring authorization using a TACACS+ security server, refer to the chapter "Configuring TACACS+." For an example of how to enable a TACACS+ server to authorize the use of network services, including PPP and ARA, see the section "TACACS+ Authorization Examples" at the end of this chapter.

To allow users to have access to the functions they request as long as they have been authenticated, use the **aaa authorization** command with the **if-authenticated** *method* keyword. If you select this method, all requested functions are automatically granted to authenticated users.

There may be times when you do not want to run authorization from a particular interface or line. To stop authorization activities on designated lines or interfaces, use the **none** *method* keyword. If you select this method, authorization is disabled for all actions.

To select local authorization, which means that the router or access server consults its local user database to determine the functions a user is permitted to use, use the **aaa authorization** command with the **local** *method* keyword. The functions associated with local authorization are defined by using the **username** global configuration command. For a list of permitted functions, refer to the chapter "Configuring Authentication."

To have the network access server request authorization via a RADIUS security server, use the **radius** *method* keyword. For more specific information about configuring authorization using a RADIUS security server, refer to the chapter "Configuring RADIUS."

To have the network access server request authorization via a RADIUS security server, use the **aaa authorization** command with the **group radius** *method* keyword. For more specific information about configuring authorization using a RADIUS security server, refer to the chapter "Configuring RADIUS". For an example of how to enable a RADIUS server to authorize services, see the "RADIUS Authorization Example" section at the end of this chapter.

Note Authorization method lists for SLIP follow whatever is configured for PPP on the relevant interface. If no lists are defined and applied to a particular interface (or no PPP settings are configured), the default setting for authorization applies.

Disabling Authorization for Global Configuration Commands

The **aaa authorization** command with the keyword **commands** attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level. Because there are configuration commands that are identical to some EXEC-level commands, there can be some confusion in the authorization process. Using **no aaa authorization config-commands** stops the network access server from attempting configuration command authorization.

To disable AAA authorization for all global configuration commands, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no aaa authorization config-commands`	Disables authorization for all global configuration commands.

Configuring Authorization for Reverse Telnet

Telnet is a standard terminal emulation protocol used for remote terminal connection. Normally, you log in to a network access server (typically through a dialup connection) and then use Telnet to access other network devices from that network access server. There are times, however, when it is necessary to establish a reverse Telnet session. In reverse Telnet sessions, the Telnet connection is established in the opposite direction—from inside a network to a network access server on the network periphery to gain access to modems or other devices connected to that network access server. Reverse Telnet is used to provide users with dialout capability by allowing them to Telnet to modem ports attached to a network access server.

It is important to control access to ports accessible through reverse Telnet. Failure to do so could, for example, allow unauthorized users free access to modems where they can trap and divert incoming calls or make outgoing calls to unauthorized destinations.

Authentication during reverse Telnet is performed through the standard AAA login procedure for Telnet. Typically the user has to provide a username and password to establish either a Telnet or reverse Telnet session. Reverse Telnet authorization provides an additional (optional) level of security by requiring authorization in addition to authentication. When enabled, reverse Telnet authorization can use RADIUS or TACACS+ to authorize whether or not this user is allowed reverse Telnet access to specific asynchronous ports, after the user successfully authenticates through the standard Telnet login procedure.

Reverse Telnet authorization offers the following benefits:

- An additional level of protection by ensuring that users engaged in reverse Telnet activities are indeed authorized to access a specific asynchronous port using reverse Telnet.
- An alternative method (other than access lists) to manage reverse Telnet authorization.

To configure a network access server to request authorization information from a TACACS+ or RADIUS server before allowing a user to establish a reverse Telnet session, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authorization reverse-access` *`method1`* `[`*`method2 ...`*`]`	Configures the network access server to request authorization information before allowing a user to establish a reverse Telnet session.

This feature enables the network access server to request reverse Telnet authorization information from the security server, whether RADIUS or TACACS+. You must configure the specific reverse Telnet privileges for the user on the security server itself.

Authorization Attribute-Value Pairs

RADIUS and TACACS+ authorization both define specific rights for users by processing attributes, which are stored in a database on the security server. For both RADIUS and TACACS+, attributes are defined on the security server, associated with the user, and sent to the network access server where they are applied to the user's connection.

For a list of supported RADIUS attributes, refer to the appendix "RADIUS Attributes". For a list of supported TACACS+ AV pairs, refer to the appendix "TACACS+ Attribute-Value Pairs."

Authorization Configuration Examples

The following sections provide authorization configuration examples:

Named Method List Configuration Example

The following example shows how to configure a Cisco AS5300 (enabled for AAA and communication with a RADIUS security server) for AAA services to be provided by the RADIUS server. If the RADIUS server fails to respond, then the local database will be queried for authentication and authorization information, and accounting services will be handled by a TACACS+ server.

```
aaa new-model
aaa authentication login admins local
aaa authentication ppp dialins group radius local
aaa authorization network scoobee group radius local
aaa accounting network charley start-stop group radius

username root password ALongPassword

radius-server host alcatraz
radius-server key myRaDiUSpassWoRd

interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication chap dialins
 ppp authorization scoobee
 ppp accounting charley

line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines a method list, admins, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins,” which specifies that RADIUS authentication then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authorization network scoobee group radius local** command defines the network authorization method list named scoobee, which specifies that RADIUS authorization will be used on serial lines using PPP. If the RADIUS server fails to respond, then local network authorization will be performed.

- The **aaa accounting network charley start-stop group radius** command defines the network accounting method list named charley, which specifies that RADIUS accounting services (in this case, start and stop records for specific events) will be used on serial lines using PPP.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication chap dialins** command selects Challenge Handshake Authentication Protocol (CHAP) as the method of PPP authentication and applies the "dialins" method list to the specified interfaces.
- The **ppp authorization scoobee** command applies the scoobee network authorization method list to the specified interfaces.
- The **ppp accounting charley** command applies the charley network accounting method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the admins method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

TACACS+ Authorization Examples

The following examples show how to use a TACACS+ server to authorize the use of network services, including PPP and ARA. If the TACACS+ server is not available or an error occurs during the authorization process, the fallback method (none) is to grant all authorization requests:

```
aaa authorization network default group tacacs+ none
```

The following example shows how to allow network authorization using TACACS+:

```
aaa authorization network default group tacacs+
```

The following example shows how to provide the same authorization, but it also creates address pools called "*mci*" and "*att*":

```
aaa authorization network default group tacacs+
ip address-pool local
ip local-pool mci 172.16.0.1 172.16.0.255
ip local-pool att 172.17.0.1 172.17.0.255
```

These address pools can then be selected by the TACACS daemon. A sample configuration of the daemon follows:

```
user = mci_customer1 {
    login = cleartext “some password”
    service = ppp protocol = ip {
        addr-pool=mci
    }
}

user = att_customer1 {
    login = cleartext “some other password”
    service = ppp protocol = ip {
        addr-pool=att
     }
```

RADIUS Authorization Example

The following example shows how to configure the router to authorize using RADIUS:

```
aaa new-model
aaa authorization exec default group radius if-authenticated
aaa authorization network default group radius
radius-server host ip
radius-server key
```

The lines in this sample RADIUS authorization configuration are defined as follows:

- The **aaa authorization exec default group radius if-authenticated** command configures the network access server to contact the RADIUS server to determine if users are permitted to start an EXEC shell when they log in. If an error occurs when the network access server contacts the RADIUS server, the fallback method is to permit the CLI to start, provided the user has been properly authenticated.

  The RADIUS information returned may be used to specify an autocommand or a connection access list be applied to this connection.
- The **aaa authorization network default group radius** command configures network authorization via RADIUS. This can be used to govern address assignment, the application of access lists, and various other per-user quantities.

Note Because no fallback method is specified in this example, authorization will fail if, for any reason, there is no response from the RADIUS server.

Reverse Telnet Authorization Examples

The following examples show how to cause the network access server to request authorization information from a TACACS+ security server before allowing a user to establish a reverse Telnet session:

```
aaa new-model
aaa authentication login default group tacacs+
aaa authorization reverse-access default group tacacs+
!
tacacs-server host 172.31.255.0
tacacs-server timeout 90
tacacs-server key goaway
```

The lines in this sample TACACS+ reverse Telnet authorization configuration are defined as follows:

- The **aaa new-model** command enables AAA.
- The **aaa authentication login default group tacacs+** command specifies TACACS+ as the default method for user authentication during login.
- The **aaa authorization reverse-access default group tacacs+** command specifies TACACS+ as the method for user authorization when trying to establish a reverse Telnet session.
- The **tacacs-server host** command identifies the TACACS+ server.
- The **tacacs-server timeout** command sets the interval of time that the network access server waits for the TACACS+ server to reply.
- The **tacacs-server key** command defines the encryption key used for all TACACS+ communications between the network access server and the TACACS+ daemon.

The following example shows how to configure a generic TACACS+ server to grant a user, pat, reverse Telnet access to port tty2 on the network access server named “maple” and to port tty5 on the network access server named “oak”:

```
user = pat
  login = cleartext lab
  service = raccess {
    port#1 = maple/tty2
    port#2 = oak/tty5
```

Note In this example, “maple” and “oak” are the configured host names of network access servers, not DNS names or alias.

The following example shows how to configure the TACACS+ server (CiscoSecure) to grant a user named pat reverse Telnet access:

```
user = pat
profile_id = 90
profile_cycle = 1
member = Tacacs_Users
service=shell {
default cmd=permit
}
service=raccess {
allow “c2511e0” “tty1” “.*”
refuse “.*” “.*” “.*”
password = clear “goaway”
```

Note CiscoSecure only supports reverse Telnet using the command line interface in versions 2.1(*x*) through version 2.2(1).

An empty “service=raccess {}” clause permits a user to have unconditional access to network access server ports for reverse Telnet. If no “service=raccess” clause exists, the user is denied access to any port for reverse Telnet.

For more information about configuring TACACS+, refer to the chapter “Configuring TACACS+.” For more information about configuring CiscoSecure, refer to the *CiscoSecure Access Control Server User Guide*, version 2.1(2) or greater.

The following example shows how to cause the network access server to request authorization from a RADIUS security server before allowing a user to establish a reverse Telnet session:

```
aaa new-model
```

```
aaa authentication login default group radius
aaa authorization reverse-access default group radius
!
radius-server host 172.31.255.0
radius-server key go away
auth-port 1645 acct-port 1646
```

The lines in this sample RADIUS reverse Telnet authorization configuration are defined as follows:

- The **aaa new-model** command enables AAA.
- The **aaa authentication login default group radius** command specifies RADIUS as the default method for user authentication during login.
- The **aaa authorization reverse-access default group radius** command specifies RADIUS as the method for user authorization when trying to establish a reverse Telnet session.
- The **radius-server host** command identifies the RADIUS server.
- The **radius-server key** command defines the encryption key used for all RADIUS communications between the network access server and the RADIUS daemon.

The following example shows how to send a request to the RADIUS server to grant a user named “pat” reverse Telnet access at port tty2 on the network access server named “maple”:

```
Username = “pat”
Password = “goaway”
User-Service-Type = Shell-User
cisco-avpair = “raccess:port#1=maple/tty2”
```

The syntax "raccess:port=any/any" permits a user to have unconditional access to network access server ports for reverse Telnet. If no "raccess:port={*nasname*}/{*tty number*}" clause exists in the user profile, the user is denied access to reverse Telnet on all ports.

For more information about configuring RADIUS, refer to the chapter “Configuring RADIUS.”

Configuring Accounting

The AAA accounting feature allows the services that users are accessing and the amount of network resources that users are consuming to be tracked. When AAA accounting is enabled, the network access server reports user activity to the TACACS+ or RADIUS security server (depending on which security method is implemented) in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. This data can then be analyzed for network management, client billing, and auditing.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Configuring Accounting" section on page 28.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring Accounting

The following tasks must be performed before configuring accounting using named method lists:

- Enable AAA on the network access server. For more information about enabling AAA on a Cisco router or access server, see the chapter "AAA Overview" in the in the *Cisco IOS Security Configuration Guide*.
- Define the characteristics of the RADIUS or TACACS+ security server if RADIUS or TACACS+ authorization is issued. For more information about configuring the Cisco network access server to communicate with the RADIUS security server, see the chapter "Configuring RADIUS." For more information about configuring the Cisco network access server to communicate with the TACACS+ security server, see the chapter "Configuring TACACS+."

Restrictions for Configuring Accounting

The AAA Accounting feature has the following restrictions:

- Accounting information can be sent simultaneously to a maximum of four AAA servers.
- SSG Restriction—For SSG systems, the **aaa accounting network broadcast** command broadcasts only **start-stop** accounting records. If interim accounting records are configured using the **ssg accounting interval** command, the interim accounting records are sent only to the configured default RADIUS server.

Information About Configuring Accounting

The following sections discuss how Accounting feature:

Named Method Lists for Accounting

Like authentication and authorization method lists, method lists for accounting define the way accounting is performed and the sequence in which these methods are performed.

Named accounting method lists allow particular security protocol to be designated and used on specific lines or interfaces for accounting services. The only exception is the default method list (which, by coincidence, is named "default"). The default method list is automatically applied to all interfaces except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list is simply a named list describing the accounting methods to be queried (such as RADIUS or TACACS+), in sequence. Method lists allow one or more security protocols to be designated and used for accounting, thus ensuring a backup system for accounting in case the initial method fails. Cisco IOS software uses the first method listed to support accounting; if that method fails to respond, the Cisco IOS software selects the next accounting method listed in the method list. This process continues until there is successful communication with a listed accounting method, or all methods defined are exhausted.

Note The Cisco IOS software attempts accounting with the next listed accounting method only when there is no response from the previous method. If accounting fails at any point in this cycle—meaning that the security server responds by denying the user access—the accounting process stops and no other accounting methods are attempted.

Accounting method lists are specific to the type of accounting being requested. AAA supports six different types of accounting:

- **Network**—Provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.
- **EXEC**—Provides information about user EXEC terminal sessions of the network access server.
- **Commands**—Provides information about the EXEC mode commands that a user issues. Command accounting generates accounting records for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- **Connection**—Provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembler/disassembler (PAD), and rlogin.
- **System**—Provides information about system-level events.
- **Resource**—Provides "start" and "stop" records for calls that have passed user authentication, and provides "stop" records for calls that fail to authenticate.

Note System accounting does not use named accounting lists; only the default list for system accounting can be defined.

Once again, when a named method list is created, a particular list of accounting methods for the indicated accounting type are defined.

Accounting method lists must be applied to specific lines or interfaces before any of the defined methods are performed. The only exception is the default method list (which is named "default"). If the **aaa accounting** command for a particular accounting type is issued without a named method list specified, the default method list is automatically applied to all interfaces or lines except those that have a named method list explicitly defined. (A defined method list overrides the default method list.) If no default method list is defined, then no accounting takes place.

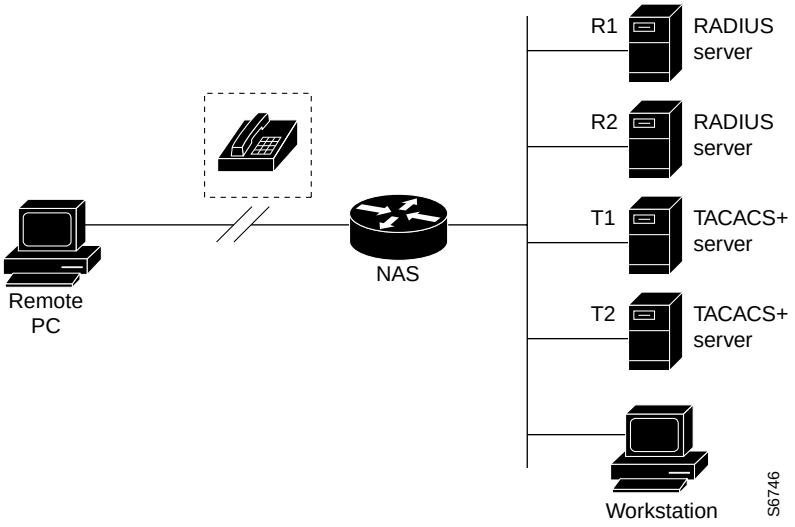
This section includes the following subsections:

- Method Lists and Server Groups, page 4
- AAA Accounting Methods, page 5

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. Figure 1 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers. R1 and R2 comprise the group of RADIUS servers. T1 and T2 comprise the group of TACACS+ servers.

Figure 1 Typical AAA Network Configuration



In Cisco IOS software, RADIUS and TACACS+ server configurations are global. A subset of the configured server hosts can be specified using server groups. These server groups can be used for a particular service. For example, server groups allow R1 and R2 to be defined as separate server groups (SG1 and SG2), and T1 and T2 as separate server groups (SG3 and SG4). This means either R1 and T1 (SG1 and SG3) can be specified in the method list or R2 and T2 (SG2 and SG4) in the method list, which provides more flexibility in the way that RADIUS and TACACS+ resources are assigned.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as failover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order in which they are configured.)

For more information about configuring server groups and about configuring server groups based on DNIS numbers, see “Configuring RADIUS” or “Configuring TACACS+” in the *Cisco IOS Security Configuration Guide*.

AAA Accounting Methods

Cisco IOS supports the following two methods for accounting:

- TACACS+—The network access server reports user activity to the TACACS+ security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server.
- RADIUS—The network access server reports user activity to the RADIUS security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server.

AAA Accounting Types

AAA supports six different accounting types:

- Network Accounting
- Connection Accounting
- EXEC Accounting
- System Accounting
- Command Accounting
- Resource Accounting

Network Accounting

Network accounting provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.

The following example shows the information contained in a RADIUS network accounting record for a PPP user who comes in through an EXEC session:

```
Wed Jun 27 04:44:45 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 5
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “0000000D”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:45:00 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 5
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = “0000000E”
        Framed-IP-Address = “10.1.1.2”
```

```
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:47:46 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 5
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = “0000000E”
        Framed-IP-Address = “10.1.1.2”
        Framed-Protocol = PPP
        Acct-Input-Octets = 3075
        Acct-Output-Octets = 167
        Acct-Input-Packets = 39
        Acct-Output-Packets = 9
        Acct-Session-Time = 171
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:48:45 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 5
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “408”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “0000000D”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

The following example shows the information contained in a TACACS+ network accounting record for a PPP user who first started an EXEC session:

```
Wed Jun 27 04:00:35 2001 172.16.25.15    username1   tty4    562/4327528
starttask_id=28      service=shell
Wed Jun 27 04:00:46 2001 172.16.25.15   username1   tty4 562/4327528    starttask_id=30
addr=10.1.1.1   service=ppp
Wed Jun 27 04:00:49 2001 172.16.25.15    username1   tty4    408/4327528     update
task_id=30      addr=10.1.1.1   service=ppp     protocol=ip     addr=10.1.1.1
Wed Jun 27 04:01:31 2001 172.16.25.15    username1   tty4    562/4327528
stoptask_id=30       addr=10.1.1.1   service=ppp     protocol=ip     addr=10.1.1.1
bytes_in=2844        bytes_out=1682  paks_in=36      paks_out=24     elapsed_time=51
Wed Jun 27 04:01:32 2001 172.16.25.15    username1   tty4    562/4327528
stoptask_id=28       service=shell   elapsed_time=57
```

Note The precise format of accounting packets records may vary depending on the security server daemon.

The following example shows the information contained in a RADIUS network accounting record for a PPP user who comes in through autoselect:

```
Wed Jun 27 04:30:52 2001
```

```
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 3
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = “0000000B”
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:36:49 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 3
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = “0000000B”
        Framed-Protocol = PPP
        Framed-IP-Address = “10.1.1.1”
        Acct-Input-Octets = 8630
        Acct-Output-Octets = 5722
        Acct-Input-Packets = 94
        Acct-Output-Packets = 64
        Acct-Session-Time = 357
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

The following example shows the information contained in a TACACS+ network accounting record for a PPP user who comes in through autoselect:

```
Wed Jun 27 04:02:19 2001 172.16.25.15    username1   Async5  562/4327528
starttask_id=35      service=ppp
Wed Jun 27 04:02:25 2001 172.16.25.15    username1   Async5  562/4327528     update
task_id=35      service=ppp     protocol=ip     addr=10.1.1.2
Wed Jun 27 04:05:03 2001 172.16.25.15    username1   Async5  562/4327528
stoptask_id=35       service=ppp     protocol=ip     addr=10.1.1.2   bytes_in=3366
bytes_out=2149       paks_in=42      paks_out=28     elapsed_time=164
```

Connection Accounting

Connection accounting provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembler/disassembler (PAD), and rlogin.

The following example shows the information contained in a RADIUS connection accounting record for an outbound Telnet connection:

```
Wed Jun 27 04:28:00 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 2
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “5622329477”
        Acct-Status-Type = Start
```

```
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = “00000008”
        Login-Service = Telnet
        Login-IP-Host = “10.68.202.158”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:28:39 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 2
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “5622329477”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = “00000008”
        Login-Service = Telnet
        Login-IP-Host = “10.68.202.158”
        Acct-Input-Octets = 10774
        Acct-Output-Octets = 112
        Acct-Input-Packets = 91
        Acct-Output-Packets = 99
        Acct-Session-Time = 39
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound Telnet connection:

```
Wed Jun 27 03:47:43 2001        172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=10      service=connection      protocol=telnet addr=10.68.202.158
cmd=telnet username1-sun
Wed Jun 27 03:48:38 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=10      service=connection      protocol=telnet addr=10.68.202.158
cmd=telnet username1-sun     bytes_in=4467   bytes_out=96    paks_in=61      paks_out=72
elapsed_time=55
```

The following example shows the information contained in a RADIUS connection accounting record for an outbound rlogin connection:

```
Wed Jun 27 04:29:48 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 2
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “5622329477”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = “0000000A”
        Login-Service = Rlogin
        Login-IP-Host = “10.68.202.158”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:30:09 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 2
        User-Name = “username1”
```

```
        Client-Port-DNIS = “4327528”
        Caller-ID = “5622329477”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = “0000000A”
        Login-Service = Rlogin
        Login-IP-Host = “10.68.202.158”
        Acct-Input-Octets = 18686
        Acct-Output-Octets = 86
        Acct-Input-Packets = 90
        Acct-Output-Packets = 68
        Acct-Session-Time = 22
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound rlogin connection:

```
Wed Jun 27 03:48:46 2001        172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=12      service=connection      protocol=rlogin addr=10.68.202.158
cmd=rlogin username1-sun /user username1
Wed Jun 27 03:51:37 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=12      service=connection      protocol=rlogin addr=10.68.202.158
cmd=rlogin username1-sun /user username1 bytes_in=659926 bytes_out=138   paks_in=2378
paks_
out=1251        elapsed_time=171
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound LAT connection:

```
Wed Jun 27 03:53:06 2001        172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=18      service=connection      protocol=lat    addr=VAX        cmd=lat
VAX
Wed Jun 27 03:54:15 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=18      service=connection      protocol=lat    addr=VAX        cmd=lat
VAX  bytes_in=0      bytes_out=0     paks_in=0      paks_out=0      elapsed_time=6
```

EXEC Accounting

EXEC accounting provides information about user EXEC terminal sessions (user shells) on the network access server, including username, date, start and stop times, the access server IP address, and (for dial-in users) the telephone number the call originated from.

The following example shows the information contained in a RADIUS EXEC accounting record for a dial-in user:

```
Wed Jun 27 04:26:23 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 1
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “5622329483”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “00000006”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

```
Wed Jun 27 04:27:25 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 1
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “5622329483”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “00000006”
        Acct-Session-Time = 62
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

The following example shows the information contained in a TACACS+ EXEC accounting record for a dial-in user:

```
Wed Jun 27 03:46:21 2001        172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=2       service=shell
Wed Jun 27 04:08:55 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=2       service=shell   elapsed_time=1354
```

The following example shows the information contained in a RADIUS EXEC accounting record for a Telnet user:

```
Wed Jun 27 04:48:32 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 26
        User-Name = “username1”
        Caller-ID = “10.68.202.158”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “00000010”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:48:46 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 26
        User-Name = “username1”
        Caller-ID = “10.68.202.158”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “00000010”
        Acct-Session-Time = 14
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
```

The following example shows the information contained in a TACACS+ EXEC accounting record for a Telnet user:

```
Wed Jun 27 04:06:53 2001        172.16.25.15    username1   tty26   10.68.202.158
starttask_id=41      service=shell
Wed Jun 27 04:07:02 2001        172.16.25.15    username1   tty26   10.68.202.158
stoptask_id=41       service=shell   elapsed_time=9
```

System Accounting

System accounting provides information about all system-level events (for example, when the system reboots or when accounting is turned on or off).

The following accounting record shows a typical TACACS+ system accounting record server indicating that AAA accounting has been turned off:

```
Wed Jun 27 03:55:32 2001       172.16.25.15    unknown unknown unknown start   task_id=25
service=system  event=sys_acct  reason=reconfigure
```

Note The precise format of accounting packets records may vary depending on the TACACS+ daemon.

The following accounting record shows a TACACS+ system accounting record indicating that AAA accounting has been turned on:

```
Wed Jun 27 03:55:22 2001       172.16.25.15    unknown unknown unknown stop    task_id=23
service=system  event=sys_acct  reason=reconfigure
```

Additional tasks for measuring system resources are covered in the Cisco IOS software configuration guides. For example, IP accounting tasks are described in the chapter"Configuring IP Services" in the *Cisco IOS Application Services Configuration Guide*.

Command Accounting

Command accounting provides information about the EXEC shell commands for a specified privilege level that are being executed on a network access server. Each command accounting record includes a list of the commands executed for that privilege level, as well as the date and time each command was executed, and the user who executed it.

The following example shows the information contained in a TACACS+ command accounting record for privilege level 1:

```
Wed Jun 27 03:46:47 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=3       service=shell   priv-lvl=1      cmd=show version <cr>
Wed Jun 27 03:46:58 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=4       service=shell   priv-lvl=1      cmd=show interfaces Ethernet 0
<cr>
Wed Jun 27 03:47:03 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=5       service=shell   priv-lvl=1      cmd=show ip route <cr>
```

The following example shows the information contained in a TACACS+ command accounting record for privilege level 15:

```
Wed Jun 27 03:47:17 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=6       service=shell   priv-lvl=15     cmd=configure terminal <cr>
Wed Jun 27 03:47:21 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=7       service=shell   priv-lvl=15     cmd=interface Serial 0 <cr>
Wed Jun 27 03:47:29 2001        172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=8       service=shell   priv-lvl=15     cmd=ip address 10.1.1.1
255.255.255.0 <cr>
```

Note The Cisco Systems implementation of RADIUS does not support command accounting.

Resource Accounting

The Cisco implementation of AAA accounting provides “start” and “stop” record support for calls that have passed user authentication. The additional feature of generating “stop” records for calls that fail to authenticate as part of user authentication is also supported. Such records are necessary for users employing accounting records to manage and monitor their networks.

This section includes the following subsections:

AAA Resource Failure Stop Accounting

Before AAA resource failure stop accounting, there was no method of providing accounting records for calls that failed to reach the user authentication stage of a call setup sequence. Such records are necessary for users employing accounting records to manage and monitor their networks and their wholesale customers.

This functionality generates a “stop” accounting record for any calls that do not reach user authentication; “stop” records are generated from the moment of call setup. All calls that pass user authentication behave as they did before; that is, no additional accounting records are seen.

Figure 2 illustrates a call setup sequence with normal call flow (no disconnect) and without AAA resource failure stop accounting enabled.

Figure 2 Modem Dial-In Call Setup Sequence With Normal Flow and Without Resource Failure Stop Accounting Enabled

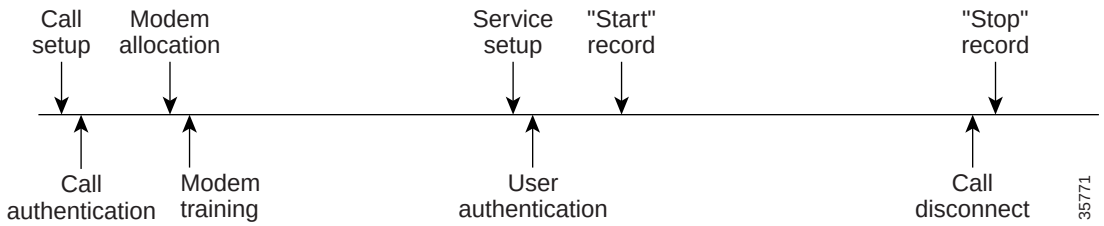


Figure 3 illustrates a call setup sequence with normal call flow (no disconnect) and with AAA resource failure stop accounting enabled.

Figure 3 Modem Dial-In Call Setup Sequence With Normal Flow and WIth Resource Failure Stop Accounting Enabled

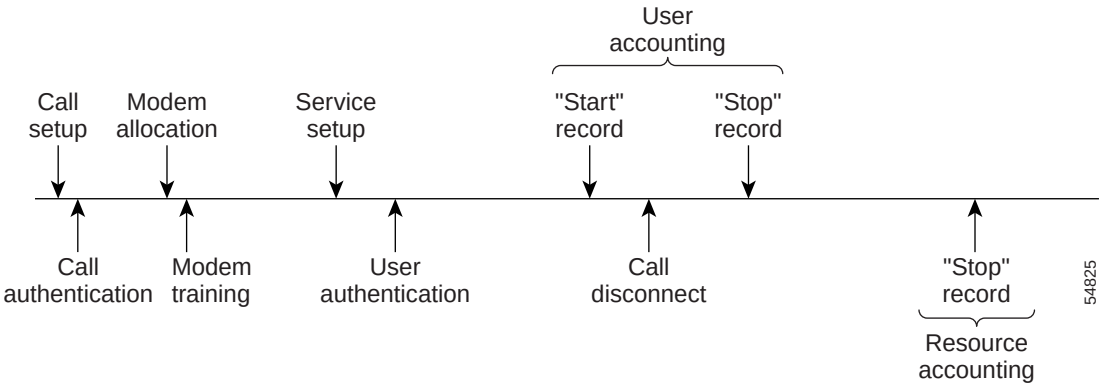
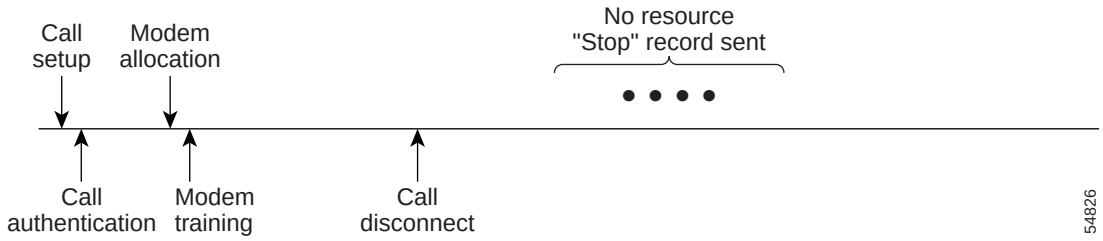


Figure 4 illustrates a call setup sequence with call disconnect occurring before user authentication and with AAA resource failure stop accounting enabled.

Figure 4 Modem Dial-In Call Setup Sequence With Call Disconnect Occurring Before User Authentication and With Resource Failure Stop Accounting Enabled

Figure 11 illustrates a call setup sequence with call disconnect occurring before user authentication and without AAA resource failure stop accounting enabled.

Figure 5 Modem Dial-In Call Setup Sequence With Call Disconnect Occurring Before User Authentication and Without Resource Failure Stop Accounting Enabled



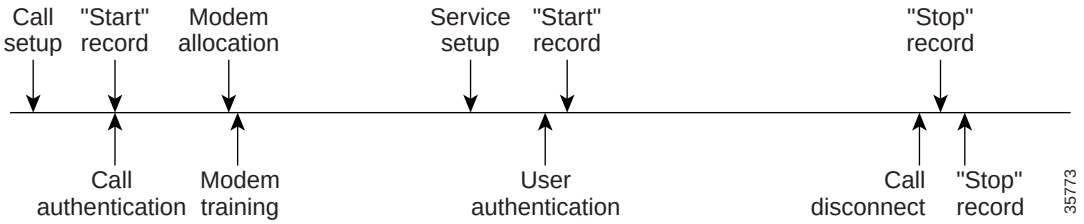
AAA Resource Accounting for Start-Stop Records

AAA resource accounting for start-stop records supports the ability to send a “start” record at each call setup, followed by a corresponding “stop” record at the call disconnect. This functionality can be used to manage and monitor wholesale customers from one source of data reporting, such as accounting records.

With this feature, a call setup and call disconnect “start-stop” accounting record tracks the progress of the resource connection to the device. A separate user authentication “start-stop” accounting record tracks the user management progress. These two sets of accounting records are interlinked by using a unique session ID for the call.

Figure 6 illustrates a call setup sequence with AAA resource start-stop accounting enabled.

Figure 6 Modem Dial-In Call Setup Sequence With Resource Start-Stop Accounting Enabled



AAA Accounting Enhancements

The section includes the following enhancements:

- AAA Broadcast Accounting, page 14
- AAA Session MIB, page 14

AAA Broadcast Accounting

AAA broadcast accounting allows accounting information to be sent to multiple AAA servers at the same time; that is, accounting information can be broadcast to one or more AAA servers simultaneously. This functionality allows service providers to send accounting information to their own private AAA servers and to the AAA servers of their end customers. It also provides redundant billing information for voice applications.

Broadcasting is allowed among groups of RADIUS or TACACS+ servers, and each server group can define its backup servers for failover independently of other groups.

Thus, service providers and their end customers can use different protocols (RADIUS or TACACS+) for the accounting server. Service providers and their end customers can also specify their backup servers independently. As for voice applications, redundant accounting information can be managed independently through a separate group with its own failover sequence.

AAA Session MIB

The AAA session MIB feature allows customers to monitor and terminate their authenticated client connections using Simple Network Management Protocol (SNMP). The data of the client is presented so that it correlates directly to the AAA accounting information reported by either the RADIUS or the TACACS+ server. AAA session MIB provides the following information:

- Statistics for each AAA function (when used in conjunction with the **show radius statistics** command)
- Status of servers providing AAA functions
- Identities of external AAA servers
- Real-time information (such as idle times), providing additional criteria for use by SNMP networks for assessing whether or not to terminate an active call

Note This command is supported only on Cisco AS5300 and Cisco AS5800 universal access server platforms.

Table 1 shows the SNMP user-end data objects that can be used to monitor and terminate authenticated client connections with the AAA session MIB feature.

Table 1 SNMP End-User Data Objects

SessionId	The session identification used by the AAA accounting protocol (same value as reported by RADIUS attribute 44 (Acct-Session-ID)).
UserId	The user login ID or zero-length string if a login is unavailable.
IpAddr	The IP address of the session or 0.0.0.0 if an IP address is not applicable or unavailable.
IdleTime	The elapsed time in seconds that the session has been idle.
Disconnect	The session termination object used to disconnect the given client.
CallId	The entry index corresponding to this accounting session that the Call Tracker record stored.

Table 2 describes the AAA summary information provided by the AAA session MIB feature using SNMP on a per-system basis.

Table 2 SNMP AAA Session Summary

ActiveTableEntries	Number of sessions currently active.
ActiveTableHighWaterMark	Maximum number of sessions present at once since last system reinstallation.
TotalSessions	Total number of sessions since last system reinstallation.
DisconnectedSessions	Total number of sessions that have been disconnected using since last system reinstallation.

How to Configure AAA Accounting

This section describes the following configuration tasks involved in configuring AAA Accounting:

Configuring AAA Accounting Using Named Method Lists

To configure AAA accounting using named method lists, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa accounting {system | network | exec | connection | commands level} {default | list-name} {start-stop | stop-only | none} [method1 [method2...]]`	Creates an accounting method list and enables accounting. The argument *list-name* is a character string used to name the created list.
Step 2	`Router(config)# line [aux | console | tty | vty] line-number [ending-line-number]` or `Router(config)# interface interface-type interface-number`	Enters the line configuration mode for the lines to which the accounting method list is applied. or Enters the interface configuration mode for the interfaces to which the accounting method list is applied.
Step 3	`Router(config-line)# accounting {arap | commands level | connection | exec} {default | list-name}` or `Router(config-if)# ppp accounting {default | list-name}`	Applies the accounting method list to a line or set of lines. or Applies the accounting method list to an interface or set of interfaces.

Note System accounting does not use named method lists. For system accounting, define only the default method list.

This section includes the following sections:

Accounting Types

Named accounting method lists are specific to the indicated type of accounting.

- **network**—To create a method list to enable authorization for all network-related service requests (including SLIP, PPP, PPP NCPs, and ARA protocols), use the **network** keyword. For example, to create a method list that provides accounting information for ARAP (network) sessions, use the **arap** keyword.
- **exec**—To create a method list that provides accounting records about user EXEC terminal sessions on the network access server, including username, date, start and stop times, use the **exec** keyword.
- **commands**—To create a method list that provides accounting information about specific, individual EXEC commands associated with a specific privilege level, use the **commands** keyword.
- **connection**—To create a method list that provides accounting information about all outbound connections made from the network access server, use the **connection** keyword.
- **resource**—Creates a method list to provide accounting records for calls that have passed user authentication or calls that failed to be authenticated.

Note System accounting does not support named method lists.

Accounting Record Types

For minimal accounting, use the **stop-only** keyword, which instructs the specified method (RADIUS or TACACS+) to send a stop record accounting notice at the end of the requested user process. For more accounting information, use the **start-stop** keyword to send a start accounting notice at the beginning of the requested event and a stop accounting notice at the end of the event. To stop all accounting activities on this line or interface, use the **none** keyword.

Accounting Methods

Table 3 lists the supported accounting methods.

Table 3 AAA Accounting Methods

Keyword	Description
group radius	Uses the list of all RADIUS servers for accounting.
group tacacs+	Uses the list of all TACACS+ servers for accounting.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for accounting as defined by the server group *group-name*.

The method argument refers to the actual method the authentication algorithm tries. Additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all other methods return an error, specify additional methods in the command. For example, to create a method list named acct_tac1 that specifies RADIUS as the backup method of authentication in the event that TACACS+ authentication returns an error, enter the following command:

```
aaa accounting network acct_tac1 stop-only group tacacs+ group radius
```

To create a default list that is used when a named list is *not* specified in the **aaa accounting** command, use the **default** keyword followed by the methods that are wanted to be used in default situations. The default method list is automatically applied to all interfaces.

For example, to specify RADIUS as the default method for user authentication during login, enter the following command:

```
aaa accounting network default stop-only group radius
```

AAA accounting supports the following methods:

- **group tacacs**—To have the network access server send accounting information to a TACACS+ security server, use the **group tacacs+** *method* keyword.
- **group radius**—To have the network access server send accounting information to a RADIUS security server, use the **group radius** *method* keyword.

Note Accounting method lists for SLIP follow whatever is configured for PPP on the relevant interface. If no lists are defined and applied to a particular interface (or no PPP settings are configured), the default setting for accounting applies.

- **group** *group-name*—To specify a subset of RADIUS or TACACS+ servers to use as the accounting method, use the **aaa accounting** command with the **group** *group-name* method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group loginrad**:

  ```
  aaa group server radius loginrad
   server 172.16.2.3
   server 172.16.2 17
   server 172.16.2.32
  ```

  This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *loginrad*.

  To specify **group loginrad** as the method of network accounting when no other method list has been defined, enter the following command:

  ```
  aaa accounting network default start-stop group loginrad
  ```

Before a group name can be used as the accounting method, communication with the RADIUS or TACACS+ security server must be enabled.

Suppressing Generation of Accounting Records for Null Username Sessions

When AAA accounting is activated, the Cisco IOS software issues accounting records for all users on the system, including users whose username string, because of protocol translation, is NULL. An example of this is users who come in on lines where the **aaa authentication login** *method-list* **none** command is applied. To prevent accounting records from being generated for sessions that do not have usernames associated with them, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa accounting suppress null-username`	Prevents accounting records from being generated for users whose username string is NULL.

Generating Interim Accounting Records

To enable periodic interim accounting records to be sent to the accounting server, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa accounting update {[newinfo] [periodic] number}`	Enables periodic interim accounting records to be sent to the accounting server.

When the **aaa accounting update** command is activated, the Cisco IOS software issues interim accounting records for all users on the system. If the keyword **newinfo** is used, interim accounting records are sent to the accounting server every time there is new accounting information to report. An example of this would be when IPCP completes IP address negotiation with the remote peer. The interim accounting record includes the negotiated IP address used by the remote peer.

When used with the keyword **periodic**, interim accounting records are sent periodically as defined by the argument number. The interim accounting record contains all of the accounting information recorded for that user up to the time the interim accounting record is sent.

Caution Using the **aaa accounting update periodic** command can cause heavy congestion when many users are logged in to the network.

Generating Accounting Records for Failed Login or Session

When AAA accounting is activated, the Cisco IOS software does not generate accounting records for system users who fail login authentication, or who succeed in login authentication but fail PPP negotiation for some reason.

To specify that accounting stop records be generated for users who fail to authenticate at login or during session negotiation, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa accounting send stop-record authentication failure`	Generates “stop” records for users who fail to authenticate at login or during session negotiation using PPP.

Specifying Accounting NETWORK-Stop Records Before EXEC-Stop Records

For PPP users who start EXEC terminal sessions, it can be specified that NETWORK records be generated before EXEC-stop records. In some cases, such as billing customers for specific services, is can be desirable to keep network start and stop records together, essentially "nesting" them within the framework of the EXEC start and stop messages. For example, a user dialing in using PPP can create the following records: EXEC-start, NETWORK-start, EXEC-stop, NETWORK-stop. By nesting the accounting records, NETWORK-stop records follow NETWORK-start messages: EXEC-start, NETWORK-start, NETWORK-stop, EXEC-stop.

To nest accounting records for user sessions, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa accounting nested`	Nests network accounting records.

Configuring AAA Resource Failure Stop Accounting

To enable resource failure stop accounting, use the following command in global configuration:

Command	Purpose
`Router(config)# aaa accounting resource` *`method-list`* `stop-failure group` *`server-group`*	Generates a "stop" record for any calls that do not reach user authentication. **Note** Before configuring this feature, the tasks described in the section "Prerequisites for Configuring Accounting" must be performed, and SNMP must be enabled on the network access server. For more information about enabling SNMP on a Cisco router or access server, see the chapter "Configuring SNMP Support" in the *Cisco IOS Network Managament Configuration Guide*.

Configuring AAA Resource Accounting for Start-Stop Records

To enable full resource accounting for start-stop records, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa accounting resource method-list start-stop group server-group`	Supports the ability to send a "start" record at each call setup. followed with a corresponding "stop" record at the call disconnect. **Note** Before configuring this feature, the tasks described in the section "Prerequisites for Configuring Accounting" must be performed, and SNMP must be enabled on the network access server. For more information about enabling SNMP on a Cisco router or access server, see the chapter "Configuring SNMP Support" in the *Cisco IOS Network Managament Configuration Guide*.

Configuring AAA Broadcast Accounting

To configure AAA broadcast accounting, use the **aaa accounting** command in global configuration mode. This command has been modified to allow the **broadcast** keyword.

Command	Purpose
`Router(config)# aaa accounting {system | network | exec | connection | commands level} {default | list-name} {start-stop | stop-only | none} [broadcast] method1 [method2...]`	Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group.

Configuring Per-DNIS AAA Broadcast Accounting

To configure AAA broadcast accounting per Dialed Number Identification Service (DNIS), use the **aaa dnis map accounting network** command in global configuration mode. This command has been modified to allow the **broadcast** keyword and multiple server groups.

Command	Purpose
`Router(config)# aaa dnis map dnis-number accounting network [start-stop | stop-only | none] [broadcast] method1 [method2...]`	Allows per-DNIS accounting configuration. This command has precedence over the global **aaa accounting** command. Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group.

Configuring AAA Session MIB

The following tasks must be performed before configuring the AAA session MIB feature:

- Configure SNMP. For information on SNMP, see the chapter "Configuring SNMP Support" in the *Cisco IOS Network Managament Configuration Guide*.
- Configure AAA.
- Define the RADIUS or TACACS+ server characteristics.

Note Overusing SNMP can affect the overall system performance; therefore, normal network management performance must be considered when this feature is used.

To configure AAA session MIB, use the following command in global configuration mode

:

	Command	Purpose
Step 1	`Router(config)# aaa session-mib disconnect`	Monitors and terminates authenticated client connections using SNMP. To terminate the call, the **disconnect** keyword must be used.

Establishing a Session with a Router if the AAA Server is Unreachable

To establish a console or telnet session with a router if the AAA server is unreachable, use the following command in Global Configuration mode:

Command	Purpose
`Router(config)# no aaa accounting system guarantee-first`	The **aaa accounting system guarantee-first** command guarantees system accounting as the first record, which is the default condition. In some situations, users may be prevented from starting a session on the console or terminal connection until after the system reloads, which can take more than three minutes. To resolve this problem, the **no aaa accounting system guarantee-first** command can be used.

Note Entering the **no aaa accounting system guarantee-first** command is not the only condition by which the console or telnet session can be started. For example, if the Privileged EXEC session is being authenticated by TACACS and the TACACS server is not reachable, then the session cannot start.

Monitoring Accounting

No specific **show** command exists for either RADIUS or TACACS+ accounting. To obtain accounting records displaying information about users currently logged in, use the following command in privileged EXEC mode:

Command	Purpose
`Router# show accounting`	Allows display of the active accountable events on the network and helps collect information in the event of a data loss on the accounting server.

Troubleshooting Accounting

To troubleshoot accounting information, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug aaa accounting`	Displays information on accountable events as they occur.

Accounting Attribute-Value Pairs

The network access server monitors the accounting functions defined in either TACACS+ attribute-value (AV) pairs or RADIUS attributes, depending on which security method is implemented.

Configuration Examples for AAA Accounting

This section contains the following examples:

- Configuring Named Method List: Example
- Configuring AAA Resource Accounting: Example
- Configuring AAA Broadcast Accounting: Example
- Configuring Per-DNIS AAA Broadcast Accounting: Example
- AAA Session MIB: Example

Configuring Named Method List: Example

The following example shows how to configure a Cisco AS5200 (enabled for AAA and communication with a RADIUS security server) in order for AAA services to be provided by the RADIUS server. If the RADIUS server fails to respond, then the local database is queried for authentication and authorization information, and accounting services are handled by a TACACS+ server.

```
aaa new-model
aaa authentication login admins local
aaa authentication ppp dialins group radius local
aaa authorization network blue1 group radius local
aaa accounting network red1 start-stop group radius group tacacs+

username root password ALongPassword

tacacs-server host 172.31.255.0
tacacs-server key goaway

radius-server host 172.16.2.7
radius-server key myRaDiUSpassWoRd

interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication chap dialins
 ppp authorization blue1
 ppp accounting red1

line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines a method list, “admins”, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins”, which specifies that first RADIUS authentication and then (if the RADIUS server does not respond) local authentication is used on serial lines using PPP.
- The **aaa authorization network blue1 group radius local** command defines the network authorization method list named “blue1”, which specifies that RADIUS authorization is used on serial lines using PPP. If the RADIUS server fails to respond, then local network authorization is performed.
- The **aaa accounting network red1 start-stop group radius group tacacs+** command defines the network accounting method list named red1, which specifies that RADIUS accounting services (in this case, start and stop records for specific events) are used on serial lines using PPP. If the RADIUS server fails to respond, accounting services are handled by a TACACS+ server.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **tacacs-server host** command defines the name of the TACACS+ server host.
- The **tacacs-server key** command defines the shared secret text string between the network access server and the TACACS+ server host.

- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication chap dialins** command selects Challenge Handshake Authentication Protocol (CHAP) as the method of PPP authentication and applies the “dialins” method list to the specified interfaces.
- The **ppp authorization blue1** command applies the blue1 network authorization method list to the specified interfaces.
- The **ppp accounting red1** command applies the red1 network accounting method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the admins method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

The **show accounting** command yields the following output for the preceding configuration:

```
Active Accounted actions on tty1, User username2 Priv 1
 Task ID 5, Network Accounting record, 00:00:52 Elapsed
 task_id=5 service=ppp protocol=ip address=10.0.0.98
```

Table 4 describes the fields contained in the preceding output.

Table 4 show accounting Field Descriptions

Field	Description
Active Accounted actions on	Terminal line or interface name user with which the user logged in.
User	User’s ID.
Priv	User’s privilege level.
Task ID	Unique identifier for each accounting session.
Accounting Record	Type of accounting session.
Elapsed	Length of time (hh:mm:ss) for this session type.
attribute=value	AV pairs associated with this accounting session.

Configuring AAA Resource Accounting: Example

The following example shows how to configure the resource failure stop accounting and resource accounting for start-stop records functions:

```
!Enable AAA on your network access server.
aaa new-model
!Enable authentication at login and list the AOL string name to use for login
authentication.
aaa authentication login AOL group radius local
!Enable authentication for ppp and list the default method to use for PPP authentication.
aaa authentication ppp default group radius local
!Enable authorization for all exec sessions and list the AOL string name to use for
authorization.
aaa authorization exec AOL group radius if-authenticated
!Enable authorization for all network-related service requests and list the default method
to use for all network-related authorizations.
aaa authorization network default group radius if-authenticated
!Enable accounting for all exec sessions and list the default method to use for all
start-stop accounting services.
aaa accounting exec default start-stop group radius
!Enable accounting for all network-related service requests and list the default method to
use for all start-stop accounting services.
aaa accounting network default start-stop group radius
!Enable failure stop accounting.
aaa accounting resource default stop-failure group radius
!Enable resource accounting for start-stop records.
aaa accounting resource default start-stop group radius
```

Configuring AAA Broadcast Accounting: Example

The following example shows how to turn on broadcast accounting using the global **aaa accounting** command:

```
aaa group server radius isp
 server 10.0.0.1
 server 10.0.0.2

aaa group server tacacs+ isp_customer
 server 172.0.0.1

aaa accounting network default start-stop broadcast group isp group isp_customer

radius-server host 10.0.0.1
radius-server host 10.0.0.2
radius-server key key1
tacacs-server host 172.0.0.1 key key2
```

The **broadcast** keyword causes “start” and “stop” accounting records for network connections to be sent simultaneously to server 10.0.0.1 in the group isp and to server 172.0.0.1 in the group isp_customer. If server 10.0.0.1 is unavailable, failover to server 10.0.0.2 occurs. If server 172.0.0.1 is unavailable, no failover occurs because backup servers are not configured for the group isp_customer.

Configuring Per-DNIS AAA Broadcast Accounting: Example

The following example shows how to turn on per DNIS broadcast accounting using the global **aaa dnis map accounting network** command:

```
aaa group server radius isp
 server 10.0.0.1
 server 10.0.0.2

aaa group server tacacs+ isp_customer
 server 172.0.0.1

aaa dnis map enable
aaa dnis map 7777 accounting network start-stop broadcast group isp group isp_customer

radius-server host 10.0.0.1
radius-server host 10.0.0.2
radius-server key key_1
tacacs-server host 172.0.0.1 key key_2
```

The **broadcast** keyword causes “start” and “stop” accounting records for network connection calls having DNIS number 7777 to be sent simultaneously to server 10.0.0.1 in the group isp and to server 172.0.0.1 in the group isp_customer. If server 10.0.0.1 is unavailable, failover to server 10.0.0.2 occurs. If server 172.0.0.1 is unavailable, no failover occurs because backup servers are not configured for the group isp_customer.

AAA Session MIB: Example

The following example shows how to set up the AAA session MIB feature to disconnect authenticated client connections for PPP users:

```
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting network default start-stop group radius
aaa session-mib disconnect
```

Feature Information for Configuring Accounting

Table 5 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or 12.0(3)S or a later release appear in the table.

Not all commands may be available in the Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 5 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 5 Feature Information for Configuring Accounting

Feature Name	Releases	Feature Information
—	Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Connection Accounting	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
AAA Session MIB	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
AAA Broadcast Accounting	Cisco IOS XE Release 2.2	This feature was introduced on Cisco ASR 1000 Series Routers.

Security Server Protocols

RADIUS

Configuring RADIUS

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes the Remote Authentication Dial-In User Service (RADIUS) security system, defines its operation, and identifies appropriate and inappropriate network environments for using RADIUS technology. The "RADIUS Configuration Task List" section describes how to configure RADIUS with the authentication, authorization, and accounting (AAA) command set.

For a complete description of the RADIUS commands used in this chapter, refer to the chapter "RADIUS Commands" in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

In This Chapter

This chapter includes the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

About RADIUS

RADIUS is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

RADIUS is a fully open protocol, distributed in source code format, that can be modified to work with any security system currently available on the market.

Cisco supports RADIUS under its AAA security paradigm. RADIUS can be used with other AAA security protocols, such as TACACS+, Kerberos, and local username lookup. RADIUS is supported on all Cisco platforms, but some RADIUS-supported features run only on specified platforms.

RADIUS has been implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

Use RADIUS in the following network environments that require access security:

- Networks with multiple-vendor access servers, each supporting RADIUS. For example, access servers from several vendors use a single RADIUS server-based security database. In an IP-based network with multiple vendors' access servers, dial-in users are authenticated through a RADIUS server that has been customized to work with the Kerberos security system.
- Turnkey network security environments in which applications support the RADIUS protocol, such as in an access environment that uses a "smart card" access control system. In one case, RADIUS has been used with Enigma's security cards to validate users and grant access to network resources.
- Networks already using RADIUS. You can add a Cisco router with RADIUS to the network. This might be the first step when you make a transition to a Terminal Access Controller Access Control System Plus (TACACS+) server.
- Networks in which a user must only access a single service. Using RADIUS, you can control user access to a single host, to a single utility such as Telnet, or to a single protocol such as Point-to-Point Protocol (PPP). For example, when a user logs in, RADIUS identifies this user as having authorization to run PPP using IP address 10.2.3.4 and the defined access list is started.
- Networks that require resource accounting. You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider (ISP) might use a freeware-based version of RADIUS access control and accounting software to meet special security and billing needs.
- Networks that wish to support preauthentication. Using the RADIUS server in your network, you can configure AAA preauthentication and set up the preauthentication profiles. Preauthentication enables service providers to better manage ports using their existing RADIUS solutions, and to efficiently manage the use of shared resources to offer differing service-level agreements.

RADIUS is not suitable in the following network security situations:

- Multiprotocol access environments. RADIUS does not support the following protocols:
  - AppleTalk Remote Access (ARA)
  - NetBIOS Frame Control Protocol (NBFCP)
  - NetWare Asynchronous Services Interface (NASI)
  - X.25 PAD connections

- Router-to-router situations. RADIUS does not provide two-way authentication. RADIUS can be used to authenticate from one router to a non-Cisco router if the non-Cisco router requires RADIUS authentication.
- Networks using a variety of services. RADIUS generally binds a user to one service model.

RADIUS Operation

When a user attempts to log in and authenticate to an access server using RADIUS, the following steps occur:

1. The user is prompted for and enters a username and password.
2. The username and encrypted password are sent over the network to the RADIUS server.

3. The user receives one of the following responses from the RADIUS server:
   a. ACCEPT—The user is authenticated.
   b. REJECT—The user is not authenticated and is prompted to reenter the username and password, or access is denied.
   c. CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
   d. CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new password.

The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and PPP, Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IP address, access list, and user timeouts.

RADIUS Configuration Task List

To configure RADIUS on your Cisco router or access server, you must perform the following tasks:

- Use the **aaa new-model** global configuration command to enable AAA. AAA must be configured if you plan to use RADIUS. For more information about using the **aaa new-model** command, refer to the "AAA Overview" chapter.
- Use the **aaa authentication** global configuration command to define method lists for RADIUS authentication. For more information about using the **aaa authentication** command, refer to the "Configuring Authentication" chapter.
- Use **line** and **interface** commands to enable the defined method lists to be used. For more information, refer to the "Configuring Authentication" chapter.

The following configuration tasks are optional:

- You may use the **aaa group server** command to group selected RADIUS hosts for specific services. For more information about using the **aaa group server** command, refer to the "Configuring AAA Server Groups" section in this chapter.
- You may use the **aaa dnis map** command to select RADIUS server groups based on DNIS number. To use this command, you must define RADIUS server groups using the **aaa group server** command. For more information about using the **aaa dnis map** command, refer to the section "Configuring AAA Server Group Selection Based on DNIS" in this chapter.
- You may use the **aaa authorization** global command to authorize specific user functions. For more information about using the **aaa authorization** command, refer to the chapter "Configuring Authorization."
- You may use the **aaa accounting** command to enable accounting for RADIUS connections. For more information about using the **aaa accounting** command, refer to the chapter "Configuring Accounting."
- You may use the **dialer aaa** interface configuration command to create remote site profiles that contain outgoing call attributes on the AAA server. For more information about using the **dialer aaa** command, refer to the section "Configuring Suffix and Password in RADIUS Access Requests" in this chapter.

This section describes how to set up RADIUS for authentication, authorization, and accounting on your network, and includes the following sections:

- Configuring Router to RADIUS Server Communication (Required)
- Configuring Router to Use Vendor-Specific RADIUS Attributes (Optional)
- Configuring Router for Vendor-Proprietary RADIUS Server Communication (Optional)
- Configuring Router to Query RADIUS Server for Static Routes and IP Addresses (Optional)
- Configuring Router to Expand Network Access Server Port Information (Optional)
- Configuring AAA Server Groups (Optional)
- Configuring AAA Server Groups with Deadtime (Optional)
- Configuring AAA DNIS Authentication
- Configuring AAA Server Group Selection Based on DNIS (Optional)
- Configuring AAA Preauthentication
- Configuring a Guard Timer
- Specifying RADIUS Authentication
- Specifying RADIUS Authorization (Optional)
- Specifying RADIUS Accounting (Optional)
- Configuring RADIUS Login-IP-Host (Optional)
- Configuring RADIUS Prompt (Optional)
- Configuring Suffix and Password in RADIUS Access Requests (Optional)

For RADIUS configuration examples using the commands in this chapter, refer to the section "RADIUS Configuration Examples" at the end of this chapter.

Configuring Router to RADIUS Server Communication

The RADIUS host is normally a multiuser system running RADIUS server software from Cisco (CiscoSecure ACS), Livingston, Merit, Microsoft, or another software provider. Configuring router to RADIUS server communication can have several components:

- Host name or IP address
- Authentication destination port
- Accounting destination port
- Timeout period
- Retransmission value
- Key string

RADIUS security servers are identified on the basis of their host name or IP address, host name and specific UDP port numbers, or IP address and specific UDP port numbers. The combination of the IP address and UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as fail-over backup to the first one. Using this

example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order they are configured.)

A RADIUS server and a Cisco router use a shared secret text string to encrypt passwords and exchange responses.To configure RADIUS to use the AAA security commands, you must specify the host running the RADIUS server daemon and a secret text (key) string that it shares with the router.

The timeout, retransmission, and encryption key values are configurable globally for all RADIUS servers, on a per-server basis, or in some combination of global and per-server settings. To apply these settings globally to all RADIUS servers communicating with the router, use the three unique global commands: **radius-server timeout**, **radius-server retransmit**, and **radius-server key**. To apply these values on a specific RADIUS server, use the **radius-server host** command.

Note You can configure both global and per-server timeout, retransmission, and key value commands simultaneously on the same Cisco network access server. If both global and per-server functions are configured on a router, the per-server timer, retransmission, and key value commands override global timer, retransmission, and key value commands.

To configure per-server RADIUS server communication, use the following command in global configuration mode:

Command	Purpose
Router(config)# **radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string] [**alias** {hostname | ip address*}]	Specifies the IP address or host name of the remote RADIUS server host and assign authentication and accounting destination port numbers. Use the **auth-port** *port-number* option to configure a specific UDP port on this RADIUS server to be used solely for authentication. Use the **acct-port** *port-number* option to configure a specific UDP port on this RADIUS server to be used solely for accounting. Use the **alias** keyword to configure up to eight multiple IP addresses for use when referring to RADIUS servers. To configure the network access server to recognize more than one host entry associated with a single IP address, simply repeat this command as many times as necessary, making sure that each UDP port number is different. Set the timeout, retransmit, and encryption key values to use with the specific RADIUS host. If no timeout is set, the global value is used; otherwise, enter a value in the range 1 to 1000. If no retransmit value is set, the global value is used; otherwise enter a value in the range 1 to 1000. If no key string is specified, the global value is used. **Note** The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command syntax because the leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.

To configure global communication settings between the router and a RADIUS server, use the following **radius-server** commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# radius-server key {0 string | 7 string | string}`	Specifies the shared secret text string used between the router and a RADIUS server. Use the **0** *line* option to configure an unencrypted shared secret. Use the **7** *line* option to configure an encrypted shared secret.
Step 2	`Router(config)# radius-server retransmit retries`	Specifies how many times the router transmits each RADIUS request to the server before giving up (the default is 3).
Step 3	`Router(config)# radius-server timeout seconds`	Specifies for how many seconds a router waits for a reply to a RADIUS request before retransmitting the request.
Step 4	`Router(config)# radius-server deadtime minutes`	Specifies for how many minutes a RADIUS server that is not responding to authentication requests is passed over by requests for RADIUS authentication.

Configuring Router to Use Vendor-Specific RADIUS Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (attribute 26). Vendor-specific attributes (VSAs) allow vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of the following format:

```
protocol : attribute sep value *
```

"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization; protocols that can be used include IP, IPX, VPDN, VOIP, SHELL, RSVP, SIP, AIRNET, OUTBOUND. "Attribute" and "value" are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS.

For example, the following AV pair causes Cisco's "multiple named ip address pools" feature to be activated during IP authorization (during PPP's IPCP address assignment):

```
cisco-avpair= ″ip:addr-pool=first“
```

If you insert an "*", the AV pair "ip:addr-pool=first" becomes optional. Note that any AV pair can be made optional.

```
cisco-avpair= ″ip:addr-pool*first“
```

The following example shows how to cause a user logging in from a network access server to have immediate access to EXEC commands:

```
cisco-avpair= ″shell:priv-lvl=15“
```

Other vendors have their own unique vendor-IDs, options, and associated VSAs. For more information about vendor-IDs and VSAs, refer to RFC 2138, Remote Authentication Dial-In User Service (RADIUS).

To configure the network access server to recognize and use VSAs, use the following command in global configuration mode:

Command	Purpose
`Router(config)# radius-server vsa send [accounting | authentication]`	Enables the network access server to recognize and use VSAs as defined by RADIUS IETF attribute 26.

For a complete list of RADIUS attributes or more information about vendor-specific attribute 26, refer to the appendix "RADIUS Attributes."

Configuring Router for Vendor-Proprietary RADIUS Server Communication

Although an Internet Engineering Task Force (IETF) draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server, some vendors have extended the RADIUS attribute set in a unique way. Cisco IOS software supports a subset of vendor-proprietary RADIUS attributes.

As mentioned earlier, to configure RADIUS (whether vendor-proprietary or IETF draft-compliant), you must specify the host running the RADIUS server daemon and the secret text string it shares with the Cisco device. You specify the RADIUS host and secret text string by using the **radius-server** commands. To identify that the RADIUS server is using a vendor-proprietary implementation of RADIUS, use the **radius-server host non-standard** command. Vendor-proprietary attributes will not be supported unless you use the **radius-server host non-standard** command.

To specify a vendor-proprietary RADIUS server host and a shared secret text string, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# radius-server host {hostname | ip-address} non-standard`	Specifies the IP address or host name of the remote RADIUS server host and identifies that it is using a vendor-proprietary implementation of RADIUS.
Step 2	`Router(config)# radius-server key {0 string | 7 string | string}`	Specifies the shared secret text string used between the router and the vendor-proprietary RADIUS server. The router and the RADIUS server use this text string to encrypt passwords and exchange responses.

Configuring Router to Query RADIUS Server for Static Routes and IP Addresses

Some vendor-proprietary implementations of RADIUS let the user define static routes and IP pool definitions on the RADIUS server instead of on each individual network access server in the network. Each network access server then queries the RADIUS server for static route and IP pool information.

To have the Cisco router or access server query the RADIUS server for static routes and IP pool definitions when the device first starts up, use the following command in global configuration mode:

Command	Purpose
`Router(config)# radius-server configure-nas`	Tells the Cisco router or access server to query the RADIUS server for the static routes and IP pool definitions used throughout its domain.

Note Because the **radius-server configure-nas** command is performed when the Cisco router starts up, it will not take effect until you issue a **copy system:running config nvram:startup-config** command.

Configuring Router to Expand Network Access Server Port Information

There are some situations when PPP or login authentication occurs on an interface different from the interface on which the call itself comes in. For example, in a V.120 ISDN call, login or PPP authentication occurs on a virtual asynchronous interface “ttt” but the call itself occurs on one of the channels of the ISDN interface.

The **radius-server attribute nas-port extended** command configures RADIUS to expand the size of the NAS-Port attribute (RADIUS IETF attribute 5) field to 32 bits. The upper 16 bits of the NAS-Port attribute display the type and number of the controlling interface; the lower 16 bits indicate the interface undergoing authentication.

To display expanded interface information in the NAS-Port attribute field, use the following command in global configuration mode:

Command	Purpose
`Router(config)# radius-server attribute nas-port format`	Expands the size of the NAS-Port attribute from 16 to 32 bits to display extended interface information.

Note This command replaces the **radius-server extended-portnames** command and the **radius-server attribute nas-port extended** command.

On platforms with multiple interfaces (ports) per slot, the Cisco RADIUS implementation will not provide a unique NAS-Port attribute that permits distinguishing between the interfaces. For example, if a dual PRI interface is in slot 1, calls on both Serial1/0:1 and Serial1/1:1 will appear as NAS-Port = 20101.

Once again, this is because of the 16-bit field size limitation associated with RADIUS IETF NAS-Port attribute. In this case, the solution is to replace the NAS-Port attribute with a vendor-specific attribute (RADIUS IETF attribute 26). Cisco's vendor-ID is 9, and the Cisco-NAS-Port attribute is subtype 2. Vendor-specific attributes (VSAs) can be turned on by entering the **radius-server vsa send** command. The port information in this attribute is provided and configured using the **aaa nas port extended** command.

To replace the NAS-Port attribute with RADIUS IETF attribute 26 and to display extended field information, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# radius-server vsa send [accounting | authentication]`	Enables the network access server to recognize and use vendor-specific attributes as defined by RADIUS IETF attribute 26.
Step 2	`Router(config)# aaa nas port extended`	Expands the size of the VSA NAS-Port field from 16 to 32 bits to display extended interface information.

The standard NAS-Port attribute (RADIUS IETF attribute 5) will continue to be sent. If you do not want this information to be sent, you can suppress it by using the **no radius-server attribute nas-port** command. When this command is configured, the standard NAS-Port attribute will no longer be sent.

For a complete list of RADIUS attributes, refer to the appendix "RADIUS Attributes."

For information about configuring RADIUS port identification for PPP, see the *Cisco IOS Wide-Area Networking Configuration Guide*.

Configuring AAA Server Groups

Configuring the router to use AAA server groups provides a way to group existing server hosts. This allows you to select a subset of the configured server hosts and use them for a particular service. A server group is used in conjunction with a global server-host list. The server group lists the IP addresses of the selected server hosts.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as failover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order in which they are configured.)

To define a server host with a server group name, enter the following commands in global configuration mode. The listed server must exist in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# radius-server host {hostname | ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string] [alias {hostname | ip address}]`	Specifies and defines the IP address of the server host before configuring the AAA server-group. Refer to the section "Configuring Router to RADIUS Server Communication" of this chapter for more information on the **radius-server host** command.

	Command	Purpose
Step 2	`Router(config-if)# aaa group server {radius | tacacs+} group-name`	Defines the AAA server group with a group name. All members of a group must be the same type; that is, RADIUS or TACACS+. This command puts the router in server group subconfiguration mode.
Step 3	`Router(config-sg)# server ip-address [auth-port port-number] [acct-port port-number]`	Associates a particular RADIUS server with the defined server group. Each security server is identified by its IP address and UDP port number. Repeat this step for each RADIUS server in the AAA server group. **Note** Each server in the group must be defined previously using the **radius-server host** command.

Configuring AAA Server Groups with Deadtime

After you have configured a server host with a server name, you can use the **deadtime** command to configure each server per server group. Configuring deadtime within a server group allows you to direct AAA traffic to separate groups of servers that have different operational characteristics.

Configuring deadtime is no longer limited to a global configuration. A separate timer has been attached to each server host in every server group. Therefore, when a server is found to be unresponsive after numerous retransmissions and timeouts, the server is assumed to be dead. The timers attached to each server host in all server groups are triggered. In essence, the timers are checked and subsequent requests to a server (once it is assumed to be dead) are directed to alternate timers, if configured. When the network access server receives a reply from the server, it checks and stops all configured timers (if running) for that server in all server groups.

If the timer has expired, only the server to which the timer is attached is assumed to be alive. This becomes the only server that can be tried for later AAA requests using the server groups to which the timer belongs.

Note Since one server has different timers and may have different deadtime values configured in the server groups, the same server may in the future have different states (dead and alive) at the same time.

Note To change the state of a server, you must start and stop all configured timers in all server groups.

The size of the server group will be slightly increased because of the addition of new timers and the deadtime attribute. The overall impact of the structure depends on the number and size of the server groups and how the servers are shared among server groups in a specific configuration.

To configure deadtime within a server group, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa group server radius group1`	Defines a RADIUS type server group.
Step 2	`Router(config-sg)# deadtime 1`	Configures and defines deadtime value in minutes. **Note** Local server group deadtime will override the global configuration. If omitted from the local server group configuration, the value will be inherited from the master list.
Step 3	`Router(config-sg)# exit`	Exits server group configuration mode.

Configuring AAA DNIS Authentication

DNIS preauthentication enables preauthentication at call setup based on the number dialed. The DNIS number is sent directly to the security server when a call is received. If authenticated by AAA, the call is accepted.

To configure DNIS authentication, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	`Router# config term`	Enters global configuration mode.
Step 2	`Router(config)# aaa preauth`	Enters AAA preauthentication mode.
Step 3	`Router(config-preauth)# group {radius | tacacs+ | server-group}`	(Optional) Selects the security server to use for AAA preauthentication requests. The default is RADIUS.
Step 4	`Router(config-preauth)# dnis [password string]`	Enables preauthentication using DNIS and optionally specifies a password to use in Access-Request packets.

Configuring AAA Server Group Selection Based on DNIS

Cisco IOS software allows you to assign a Dialed Number Identification Service (DNIS) number to a particular AAA server group so that the server group can process authentication, authorization, and accounting requests for users dialing in to the network using that particular DNIS. Any phone line (a regular home phone or a commercial T1/PRI line) can be associated with several phone numbers. The DNIS number identifies the number that was called to reach you.

For example, suppose you want to share the same phone number with several customers, but you want to know which customer is calling before you pick up the phone. You can customize how you answer the phone because DNIS allows you to know which customer is calling when you answer.

Cisco routers with either ISDN or internal modems can receive the DNIS number. This functionality allows users to assign different RADIUS server groups for different customers (that is, different RADIUS servers for different DNIS numbers). Additionally, using server groups you can specify the same server group for AAA services or a separate server group for each AAA service.

Cisco IOS software provides the flexibility to implement authentication and accounting services in several ways:

- Globally—AAA services are defined using global configuration access list commands and applied in general to all interfaces on a specific network access server.
- Per Interface—AAA services are defined using interface configuration commands and applied specifically to the interface being configured on a specific network access server.
- DNIS mapping—You can use DNIS to specify an AAA server to supply AAA services.

Because each of these AAA configuration methods can be configured simultaneously, Cisco has established an order of precedence to determine which server or groups of servers provide AAA services. The order of precedence is as follows:

- Per DNIS—If you configure the network access server to use DNIS to identify/determine which server group provides AAA services, then this method takes precedence over any additional AAA selection method.
- Per interface—If you configure the network access server per interface to use access lists to determine how a server provides AAA services, this method takes precedence over any global configuration AAA access lists.
- Globally—If you configure the network access server by using global AAA access lists to determine how the security server provides AAA services, this method has the least precedence.

Note Prior to configuring AAA Server Group Selection Based on DNIS, you must configure the list of RADIUS server hosts and configure the AAA server groups. See the sections "Configuring Router to RADIUS Server Communication" and "Configuring AAA Server Groups" of this chapter.

To configure the router to select a particular AAA server group based on the DNIS of the server group, configure DNIS mapping. To map a server group with a group name with DNIS number, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa dnis map enable`	Enables DNIS mapping.
Step 2	`Router(config)# aaa dnis map` *dnis-number* `authentication ppp group server-`*group-name*	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for authentication.
Step 3	`Router(config)# aaa dnis map dnis-number authorization network group server-group-name`	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for authorization.
Step 4	`Router(config)# aaa dnis map` *dnis-number* `accounting network [none | start-stop | stop-only] group` *server-group-name*	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for accounting.

Configuring AAA Preauthentication

Configuring AAA preauthentication with ISDN PRI or channel-associated signalling (CAS) allows service providers to better manage ports using their existing RADIUS solutions and efficiently manage the use of shared resources to offer differing service-level agreements. With ISDN PRI or CAS, information about an incoming call is available to the network access server (NAS) before the call is connected. The available call information includes the following:

- The Dialed Number Identification Service (DNIS) number, also referred to as the called number
- The Calling Line Identification (CLID) number, also referred to as the calling number
- The call type, also referred to as the bearer capability

This feature allows a Cisco NAS to decide—on the basis of the DNIS number, the CLID number, or the call type—whether to connect an incoming call. (With ISDN PRI, it enables user authentication and authorization before a call is answered. With CAS, the call must be answered; however, the call can be dropped if preauthentication fails.)

When an incoming call arrives from the public network switch, but before it is connected, AAA preauthentication enables the NAS to send the DNIS number, CLID number, and call type to a RADIUS server for authorization. If the server authorizes the call, then the NAS accepts the call. If the server does not authorize the call, then the NAS sends a disconnect message to the public network switch to reject the call.

In the event that the RADIUS server application becomes unavailable or is slow to respond, a guard timer can be set in the NAS. When the timer expires, the NAS uses a configurable parameter to accept or reject the incoming call that has no authorization.

This feature supports the use of attribute 44 by the RADIUS server application and the use of RADIUS attributes that are configured in the RADIUS preauthentication profiles to specify preauthentication behavior. They may also be used, for instance, to specify whether subsequent authentication should occur and, if so, what authentication method should be used.

The following restrictions apply to AAA preauthentication with ISDN PRI and CAS:

- Attribute 44 is available for CAS calls only when preauthentication or resource pooling is enabled.
- MMP is not available with ISDN PRI.
- AAA preauthentication is available only on the Cisco AS5300, Cisco AS5400, and Cisco AS5800 platforms.

Note Prior to configuring AAA preauthentication, you must enable the **aaa new-model** command and make sure the supporting preauthentication application is running on a RADIUS server in your network.

To configure AAA preauthentication, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa preauth`	Enters AAA preauthentication configuration mode.
Step 2	`Router(config-preauth)# group` *server-group*	Specifies the AAA RADIUS server group to use for preauthentication.
Step 3	`Router(config-preauth)# clid [if-avail | required] [accept-stop] [password` *string*`]`	Preauthenticates calls on the basis of the CLID number.

	Command	Purpose
Step 4	`Router(config-preauth)# ctype [if-avail | required] [accept-stop] [password string]`	Preauthenticates calls on the basis of the call type.
Step 5	`Router(config-preauth)# dnis [if-avail | required] [accept-stop] [password string]`	Preauthenticates calls on the basis of the DNIS number.
Step 6	`Router(config-preauth)# dnis bypass {dnis-group-name}`	Specifies a group of DNIS numbers that will be bypassed for preauthentication.

To configure DNIS preauthentication, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa preauth`	Enters AAA preauthentication mode.
Step 2	`Router(config-preauth)# group {radius | tacacs+ | server-group}`	(Optional) Selects the security server to use for AAA preauthentication requests. The default is RADIUS.
Step 3	`Router(config-preauth)# dnis [password string]`	Enables preauthentication using DNIS and optionally specifies a password to use in Access-Request packets.

In addition to configuring preauthentication on your Cisco router, you must set up the preauthentication profiles on the RADIUS server. For information on setting up the preauthentication profiles, see the following sections:

- Setting Up the RADIUS Profile for DNIS or CLID Preauthentication
- Setting Up the RADIUS Profile for Call Type Preauthentication
- Setting Up the RADIUS Profile for Preauthentication Enhancements for Callback
- Setting Up the RADIUS Profile for a Remote Host Name Used for Large-Scale Dial-Out
- Setting Up the RADIUS Profile for Modem Management
- Setting Up the RADIUS Profile for Subsequent Authentication
- Setting Up the RADIUS Profile for Subsequent Authentication Type
- Setting Up the RADIUS Profile to Include the Username
- Setting Up the RADIUS Profile for Two-Way Authentication
- Setting Up the RADIUS Profile to Support Authorization

Setting Up the RADIUS Profile for DNIS or CLID Preauthentication

To set up the RADIUS preauthentication profile, use the DNIS or CLID number as the username, and use the password defined in the **dnis** or **clid** command as the password.

Note The preauthentication profile must have “outbound” as the service type because the password is predefined on the NAS. Setting up the preauthentication profile in this manner prevents users from trying to log in to the NAS with the username of the DNIS number, CLID number, or call type and an obvious password. The “outbound” service type is also included in the access-request packet sent to the RADIUS server.

Setting Up the RADIUS Profile for Call Type Preauthentication

To set up the RADIUS preauthentication profile, use the call type string as the username, and use the password defined in the **ctype** command as the password. The following table shows the call type strings that may be used in the preauthentication profile:

Call Type String	ISDN Bearer Capabilities
digital	Unrestricted digital, restricted digital.
speech	Speech, 3.1 kHz audio, 7 kHz audio. **Note** This is the only call type available for CAS.
v.110	Anything with V.110 user information layer.
v.120	Anything with V.120 user information layer.

Note The preauthentication profile must have “outbound” as the service type because the password is predefined on the NAS. Setting up the preauthentication profile in this manner prevents users from trying to log in to the NAS with the username of the DNIS number, CLID number, or call type and an obvious password. The “outbound” service type is also included in the access-request packet sent to the RADIUS server and should be a check-in item if the RADIUS server supports check-in items.

Setting Up the RADIUS Profile for Preauthentication Enhancements for Callback

Callback allows remote network users such as telecommuters to dial in to the NAS without being charged. When callback is required, the NAS hangs up the current call and dials the caller back. When the NAS performs the callback, only information for the outgoing connection is applied. The rest of the attributes from the preauthentication access-accept message are discarded.

Note The destination IP address is not required to be returned from the RADIUS server.

The following example shows a RADIUS profile configuration with a callback number of 555-1111 and the service type set to outbound. The cisco-avpair = “preauth:send-name=<string>” uses the string “andy” and the cisco-avpair = “preauth:send-secret=<string>” uses the password “cisco.”

```
5551111 password = “cisco”, Service-Type = Outbound
    Service-Type = Callback-Framed
    Framed-Protocol = PPP,
    Dialback-No = “5551212”
    Class = “ISP12”
    cisco-avpair = “preauth:send-name=andy”
```

```
cisco-avpair = “preauth:send-secret=cisco”
```

Setting Up the RADIUS Profile for a Remote Host Name Used for Large-Scale Dial-Out

The following example adds to the previous example by protecting against accidentally calling a valid telephone number but accessing the wrong router by providing the name of the remote, for use in large-scale dial-out:

```
5551111 password = "cisco", Service-Type = Outbound
       Service-Type = Callback-Framed
       Framed-Protocol = PPP,
       Dialback-No = "5551212"
       Class = "ISP12"
       cisco-avpair = "preauth:send-name=andy"
       cisco-avpair = "preauth:send-secret=cisco"
       cisco-avpair = "preauth:remote-name=Router2"
```

Setting Up the RADIUS Profile for Modem Management

When DNIS, CLID, or call type preauthentication is used, the affirmative response from the RADIUS server may include a modem string for modem management in the NAS through vendor-specific attribute (VSA) 26. The modem management VSA has the following syntax:

```
cisco-avpair = “preauth:modem-service=modem min-speed <x> max-speed <y>
modulation <z> error-correction <a> compression <b>”
```

The modem management string within the VSA may contain the following:

Command	Argument
min-speed	<300 to 56000>, any
max-speed	<300 to 56000>, any
modulation	K56Flex, v22bis, v32bis, v34, v90, any
error-correction	lapm, mnp4
compression	mnp5, v42bis

When the modem management string is received from the RADIUS server in the form of a VSA, the information is passed to the Cisco IOS software and applied on a per-call basis. Modem ISDN channel aggregation (MICA) modems provide a control channel through which messages can be sent during the call setup time. Hence, this modem management feature is supported only with MICA modems and newer technologies. This feature is not supported with Microcom modems.

For more information on modem management, refer to the “Modem Configuration and Management” chapter of the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2.

Setting Up the RADIUS Profile for Subsequent Authentication

If preauthentication passes, you may use vendor-proprietary RADIUS attribute 201 (Require-Auth) in the preauthentication profile to determine whether subsequent authentication is to be performed. If attribute 201, returned in the access-accept message, has a value of 0, then subsequent authentication will not be performed. If attribute 201 has a value of 1, then subsequent authentication will be performed as usual.

Attribute 201 has the following syntax:

```
cisco-avpair = "preauth:auth-required=<n>"
```

where <*n*> has the same value range as attribute 201 (that is, 0 or 1).

If attribute 201 is missing in the preauthentication profile, then a value of 1 is assumed, and subsequent authentication is performed.

Note To perform subsequent authentication, you must set up a regular user profile in addition to a preauthentication profile.

Setting Up the RADIUS Profile for Subsequent Authentication Type

If you have specified subsequent authentication in the preauthentication profile, you must also specify the authentication types to be used for subsequent authentication. To specify the authentication types allowed in subsequent authentication, use the following VSA:

```
cisco-avpair = "preauth:auth-type=<string>"
```

where <*string*> can be one of the following:

String	Description
chap	Requires username and password of CHAP for PPP authentication.
ms-chap	Requires username and password of MS-CHAP for PPP authentication.
pap	Requires username and password of PAP for PPP authentication.

To specify that multiple authentication types are allowed, you can configure more than one instance of this VSA in the preauthentication profile. The sequence of the authentication type VSAs in the preauthentication profile is significant because it specifies the order of authentication types to be used in the PPP negotiation.

This VSA is a per-user attribute and replaces the authentication type list in the **ppp authentication** interface command.

Note You should use this VSA only if subsequent authentication is required because it specifies the authentication type for subsequent authentication.

Setting Up the RADIUS Profile to Include the Username

If only preauthentication is used to authenticate a call, the NAS could be missing a username when it brings up the call. RADIUS may provide a username for the NAS to use through RADIUS attribute 1 (User-Name) or through a VSA returned in the access-accept packet. The VSA for specifying the username has the following syntax:

```
cisco-avpair = "preauth:username=<string>"
```

If no username is specified, the DNIS number, CLID number, or call type is used, depending on the last preauthentication command that has been configured (for example, if **clid** was the last preauthentication command configured, the CLID number will be used as the username).

If subsequent authentication is used to authenticate a call, there might be two usernames: one provided by RADIUS and one provided by the user. In this case, the username provided by the user overrides the one contained in the RADIUS preauthentication profile; the username provided by the user is used for both authentication and accounting.

Setting Up the RADIUS Profile for Two-Way Authentication

In the case of two-way authentication, the calling networking device will need to authenticate the NAS. The Password Authentication Protocol (PAP) username and password or Challenge Handshake Authentication Protocol (CHAP) username and password need not be configured locally on the NAS. Instead, username and password can be included in the access-accept messages for preauthentication.

Note The **ppp authentication** command must be configured with the **radius** method.

To apply for PAP, do not configure the **ppp pap sent-name password** command on the interface. The vendor-specific attributes (VSAs) “preauth:send-name” and “preauth:send-secret” will be used as the PAP username and PAP password for outbound authentication.

For CHAP, “preauth:send-name” will be used not only for outbound authentication, but also for inbound authentication. For a CHAP inbound case, the NAS will use the name defined in “preauth:send-name” in the challenge packet to the caller networking device. For a CHAP outbound case, both “preauth:send-name” and “preauth:send-secret” will be used in the response packet.

The following example shows a configuration that specifies two-way authentication:

```
5551111 password = "cisco", Service-Type = Outbound
    Service-Type = Framed-User
    cisco-avpair = "preauth:auth-required=1"
    cisco-avpair = "preauth:auth-type=pap"
    cisco-avpair = "preauth:send-name=andy"
    cisco-avpair = "preauth:send-secret=cisco"
    class = "<some class>"
```

Note Two-way authentication does not work when resource pooling is enabled.

Setting Up the RADIUS Profile to Support Authorization

If only preauthentication is configured, then subsequent authentication will be bypassed. Note that because the username and password are not available, authorization will also be bypassed. However, you may include authorization attributes in the preauthentication profile to apply per-user attributes and avoid having to return subsequently to RADIUS for authorization. To initiate the authorization process, you must also configure the **aaa authorization network** command on the NAS.

You may configure authorization attributes in the preauthentication profile with one exception: the service-type attribute (attribute 6). The service-type attribute must be converted to a VSA in the preauthentication profile. This VSA has the following syntax:

```
cisco-avpair = “preauth:service-type=<n>”
```

where <*n*> is one of the standard RFC 2138 values for attribute 6. For a list of possible Service-Type values, refer to the appendix RADIUS Attributes.

Note If subsequent authentication is required, the authorization attributes in the preauthentication profile will not be applied.

Configuring a Guard Timer

Because response times for preauthentication and authentication requests can vary, the guard timer allows you to control the handling of calls. The guard timer starts when the DNIS is sent to the RADIUS server. If the NAS does not receive a response from AAA before the guard timer expires, it accepts or rejects the calls on the basis of the configuration of the timer.

To set a guard timer to accept or reject a call in the event that the RADIUS server fails to respond to an authentication or preauthentication request, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn guard-timer` *milliseconds* `[on-expiry {accept | reject}]`	Sets an ISDN guard timer to accept or reject a call in the event that the RADIUS server fails to respond to a preauthentication request.
`Router(control-config)# call guard-timer` *milliseconds* `[on-expiry {accept | reject}]`	Sets a CAS guard timer to accept or reject a call in the event that the RADIUS server fails to respond to a preauthentication request.

Specifying RADIUS Authentication

After you have identified the RADIUS server and defined the RADIUS authentication key, you must define method lists for RADIUS authentication. Because RADIUS authentication is facilitated through AAA, you must enter the **aaa authentication** command, specifying RADIUS as the authentication method. For more information, refer to the chapter "Configuring Authentication."

Specifying RADIUS Authorization

AAA authorization lets you set parameters that restrict a user's access to the network. Authorization using RADIUS provides one method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet. Because RADIUS authorization is facilitated through AAA, you must issue the **aaa authorization** command, specifying RADIUS as the authorization method. For more information, refer to the chapter "Configuring Authorization."

Specifying RADIUS Accounting

The AAA accounting feature enables you to track the services users are accessing as well as the amount of network resources they are consuming. Because RADIUS accounting is facilitated through AAA, you must issue the **aaa accounting** command, specifying RADIUS as the accounting method. For more information, refer to the chapter "Configuring Accounting."

Configuring RADIUS Login-IP-Host

To enable the network access server to attempt more than one login host when trying to connect a dial in user, you can enter as many as three Login-IP-Host entries in the user's profile on the RADIUS server. The following example shows that three Login-IP-Host instances have been configured for the user *joeuser,* and that TCP-Clear will be used for the connection:

```
joeuser       Password = xyz
        Service-Type = Login,
        Login-Service = TCP-Clear,
        Login-IP-Host = 10.0.0.0,
        Login-IP-Host = 10.2.2.2,
        Login-IP-Host = 10.255.255.255,
        Login-TCP-Port = 23
```

The order in which the hosts are entered is the order in which they are attempted. Use the **ip tcp synwait-time** command to set the number of seconds that the network access server waits before trying to connect to the next host on the list; the default is 30 seconds.

Your RADIUS server might permit more than three Login-IP-Host entries; however, the network access server supports only three hosts in access-accept packets.

Configuring RADIUS Prompt

To control whether user responses to access-challenge packets are echoed to the screen, you can configure the Prompt attribute in the user profile on the RADIUS server. This attribute is included only in access-challenge packets. The following example shows the Prompt attribute set to No-Echo, which prevents the user's responses from echoing:

```
joeuser Password = xyz
Service-Type = Login,
Login-Service = Telnet,
Prompt = No-Echo,
Login-IP-Host = 172.31.255.255
```

To allow user responses to echo, set the attribute to Echo. If the Prompt attribute is not included in the user profile, responses are echoed by default.

This attribute overrides the behavior of the **radius-server challenge-noecho** command configured on the access server. For example, if the access server is configured to suppress echoing, but the individual user profile allows echoing, then the user responses are echoed.

Note To use the Prompt attribute, your RADIUS server must be configured to support access-challenge packets.

Configuring Suffix and Password in RADIUS Access Requests

Large-scale dial-out eliminates the need to configure dialer maps on every NAS for every destination. Instead, you can create remote site profiles that contain outgoing call attributes on the AAA server. The profile is downloaded by the NAS when packet traffic requires a call to be placed to a remote site.

You can configure the username in the access-request message to RADIUS. The default suffix of the username, "-out," is appended to the username. The format for composing the username attribute is IP address plus configured suffix.

To provide username configuration capability for large-scale dial-out, the **dialer aaa** command is implemented with the new **suffix** and **password** keywords.

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables the AAA access control model.
Step 2	`Router(config)# aaa route download` *min*	Enables the download static route feature and sets the amount of time between downloads.
Step 3	`Router(config)# aaa authorization configuration default`	Downloads static route configuration information from the AAA server using TACACS+ or RADIUS.
Step 4	`Router(config)# interface dialer` *1*	Defines a dialer rotary group.
Step 5	`Router(config-if)# dialer aaa`	Allows a dialer to access the AAA server for dialing information.
Step 6	`Router(config-if)# dialer aaa suffix` *suffix* `password` *password*	Allows a dialer to access the AAA server for dialing information and specifies a suffix and nondefault password for authentication.

Monitoring and Maintaining RADIUS

To monitor and maintain RADIUS, use the following commands in privileged EXEC mode:

Command	Purpose
`Router# debug radius`	Displays information associated with RADIUS.
`Router# show radius statistics`	Displays the RADIUS statistics for accounting and authentication packets.

RADIUS Attributes

The network access server monitors the RADIUS authorization and accounting functions defined by RADIUS attributes in each user-profile. For a list of supported RADIUS attributes, refer to the appendix "RADIUS Attributes."

This section includes the following sections:

- Vendor-Proprietary RADIUS Attributes
- RADIUS Tunnel Attributes

Vendor-Proprietary RADIUS Attributes

An Internet Engineering Task Force (IETF) draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server. Some vendors, nevertheless, have extended the RADIUS attribute set in a unique way. Cisco IOS software supports a subset of vendor-proprietary RADIUS attributes. For a list of supported vendor-proprietary RADIUS attributes, refer to the appendix "RADIUS Attributes."

RADIUS Tunnel Attributes

RADIUS is a security server authentication, authorization, and accounting (AAA) protocol originally developed by Livingston, Inc. RADIUS uses attribute value (AV) pairs to communicate information between the security server and the network access server. RFC 2138 and RFC 2139 describe the basic functionality of RADIUS and the original set of Internet Engineering Task Force (IETF)-standard AV pairs used to send AAA information. Two draft IETF standards, "RADIUS Attributes for Tunnel Protocol Support" and "RADIUS Accounting Modifications for Tunnel Protocol Support," extend the IETF-defined set of AV pairs to include attributes specific to virtual private networks (VPNs); these attributes are used to carry the tunneling information between the RADIUS server and the tunnel initiator. RFC 2865 and RFC 2868 extend the IETF-defined set of AV pairs to include attributes specific to compulsory tunneling in VPNs by allowing the user to specify authentication names for the network access server and the RADIUS server.

Cisco routers and access servers now support new RADIUS IETF-standard VPDN tunnel attributes. These new RADIUS IETF-standard attributes are listed in the "RADIUS Attributes" appendix. Refer to the following three configuration examples later in this chapter:

- RADIUS User Profile with RADIUS Tunneling Attributes Example
- L2TP Access Concentrator Examples
- L2TP Network Server Examples

For more information about L2F, L2TP, VPN, or VPDN, refer to the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2.

RADIUS Configuration Examples

The following sections provide RADIUS configuration examples:

- RADIUS Authentication and Authorization Example
- RADIUS Authentication, Authorization, and Accounting Example
- Vendor-Proprietary RADIUS Configuration Example
- RADIUS Server with Server-Specific Values Example
- Multiple RADIUS Servers with Global and Server-Specific Values Example
- Multiple RADIUS Server Entries for the Same Server IP Address Example
- RADIUS Server Group Examples
- Multiple RADIUS Server Entries Using AAA Server Groups Example
- AAA Server Group Selection Based on DNIS Example
- AAA Preauthentication Examples
- RADIUS User Profile with RADIUS Tunneling Attributes Example
- Guard Timer Examples
- L2TP Access Concentrator Examples
- L2TP Network Server Examples

RADIUS Authentication and Authorization Example

The following example shows how to configure the router to authenticate and authorize using RADIUS:

```
aaa authentication login use-radius group radius local
aaa authentication ppp user-radius if-needed group radius
aaa authorization exec default group radius
aaa authorization network default group radius
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login use-radius group radius local** command configures the router to use RADIUS for authentication at the login prompt. If RADIUS returns an error, the user is authenticated using the local database. In this example, **use-radius** is the name of the method list, which specifies RADIUS and then local authentication.
- The **aaa authentication ppp user-radius if-needed group radius** command configures the Cisco IOS software to use RADIUS authentication for lines using PPP with CHAP or PAP if the user has not already been authorized. If the EXEC facility has authenticated the user, RADIUS authentication is not performed. In this example, **user-radius** is the name of the method list defining RADIUS as the if-needed authentication method.
- The **aaa authorization exec default group radius** command sets the RADIUS information that is used for EXEC authorization, autocommands, and access lists.
- The **aaa authorization network default group radius** command sets RADIUS for network authorization, address assignment, and access lists.

RADIUS Authentication, Authorization, and Accounting Example

The following example shows a general configuration using RADIUS with the AAA command set:

```
radius-server host 123.45.1.2
radius-server key myRaDiUSpassWoRd
username root password ALongPassword
aaa authentication ppp dialins group radius local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa authentication login admins local
aaa authorization exec default local
line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem ri-is-cd
interface group-async 1
 encaps ppp
 ppp authentication pap dialins
```

The lines in this example RADIUS authentication, authorization, and accounting configuration are defined as follows:

- The **radius-server host** command defines the IP address of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins,” which specifies that RADIUS authentication and then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.

- The **ppp authentication pap dialins** command applies the “dialins” method list to the lines specified.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **aaa authentication login admins local** command defines another method list, “admins,” for login authentication.
- The **login authentication admins** command applies the “admins” method list for login authentication.

Vendor-Proprietary RADIUS Configuration Example

The following example shows a general configuration using vendor-proprietary RADIUS with the AAA command set:

```
radius-server host alcatraz non-standard
radius-server key myRaDiUSpassWoRd
radius-server configure-nas
username root password ALongPassword
aaa authentication ppp dialins group radius local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa authentication login admins local
aaa authorization exec default local
line 1 16
autoselect ppp
autoselect during-login
login authentication admins
modem ri-is-cd
interface group-async 1
encaps ppp
ppp authentication pap dialins
```

The lines in this example RADIUS authentication, authorization, and accounting configuration are defined as follows:

- The **radius-server host non-standard** command defines the name of the RADIUS server host and identifies that this RADIUS host uses a vendor-proprietary version of RADIUS.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **radius-server configure-nas** command defines that the Cisco router or access server will query the RADIUS server for static routes and IP pool definitions when the device first starts up.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins,” which specifies that RADIUS authentication, and then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **ppp authentication pap dialins** command applies the “dialins” method list to the lines specified.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.

- The **aaa authentication login admins local** command defines another method list, “admins,” for login authentication.
- The **login authentication admins** command applies the “admins” method list for login authentication.

RADIUS Server with Server-Specific Values Example

The following example shows how to configure server-specific timeout, retransmit, and key values for the RADIUS server with IP address 172.31.39.46:

```
radius-server host 172.31.39.46 timeout 6 retransmit 5 key rad123
```

Multiple RADIUS Servers with Global and Server-Specific Values Example

The following example shows how to configure two RADIUS servers with specific timeout, retransmit, and key values. In this example, the **aaa new-model** command enables AAA services on the router, while specific AAA commands define the AAA services. The **radius-server retransmit** command changes the global retransmission value to 4 for all RADIUS servers. The **radius-server host** command configures specific timeout, retransmission, and key values for the RADIUS server hosts with IP addresses 172.16.1.1 and 172.29.39.46.

```
! Enable AAA services on the router and define those services.
aaa new-model
aaa authentication login default group radius
aaa authentication login console-login none
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting exec default start-stop group radius
aaa accounting network default start-stop group radius
enable password tryit1
!
! Change the global retransmission value for all RADIUS servers.
radius-server retransmit 4
!
! Configure per-server specific timeout, retransmission, and key values.
! Change the default auth-port and acct-port values.
radius-server host 172.16.1.1 auth-port 1612 acct-port 1616 timeout 3 retransmit 3 key
radkey
!
! Configure per-server specific timeout and key values. This server uses the global
! retransmission value.
radius-server host 172.29.39.46 timeout 6 key rad123
```

Multiple RADIUS Server Entries for the Same Server IP Address Example

The following example shows how to configure the network access server to recognize several RADIUS host entries with the same IP address. Two different host entries on the same RADIUS server are configured for the same services—authentication and accounting. The second host entry configured acts as fail-over backup to the first one. (The RADIUS host entries will be tried in the order they are configured.)

```
! This command enables AAA.
aaa new-model
! The next command configures default RADIUS parameters.
aaa authentication ppp default group radius
```

```
! The next set of commands configures multiple host entries for the same IP address.
radius-server host 172.20.0.1 auth-port 1000 acct-port 1001
radius-server host 172.20.0.1 auth-port 2000 acct-port 2000
```

RADIUS Server Group Examples

The following example shows how to create server group *radgroup1* with three different RADIUS server members, each using the default authentication port (1645) and accounting port (1646):

```
aaa group server radius radgroup1
 server 172.16.1.11
 server 172.17.1.21
 server 172.18.1.31
```

The following example shows how to create server group *radgroup2* with three RADIUS server members, each with the same IP address but with unique authentication and accounting ports:

```
aaa group server radius radgroup2
 server 172.16.1.1 auth-port 1000 acct-port 1001
 server 172.16.1.1 auth-port 2000 acct-port 2001
 server 172.16.1.1 auth-port 3000 acct-port 3001
```

Multiple RADIUS Server Entries Using AAA Server Groups Example

The following example shows how to configure the network access server to recognize two different RADIUS server groups. One of these groups, group1, has two different host entries on the same RADIUS server configured for the same services. The second host entry configured acts as failover backup to the first one. Each group is individually configured for deadtime; deadtime for group 1 is one minute, and deadtime for group 2 is two minutes.

Note In cases where both global commands and server commands are used, the server command will take precedence over the global command.

```
! This command enables AAA.
aaa new-model
! The next command configures default RADIUS parameters.
aaa authentication ppp default group group1
! The following commands define the group1 RADIUS server group and associate servers
! with it and configures a deadtime of one minute.
aaa group server radius group1
 server 1.1.1.1 auth-port 1645 acct-port 1646
 server 2.2.2.2 auth-port 2000 acct-port 2001
 deadtime 1
! The following commands define the group2 RADIUS server group and associate servers
! with it and configures a deadtime of two minutes.
aaa group server radius group2
 server 2.2.2.2 auth-port 2000 acct-port 2001
 server 3.3.3.3 auth-port 1645 acct-port 1646
 deadtime 2
! The following set of commands configures the RADIUS attributes for each host entry
! associated with one of the defined server groups.
radius-server host 1.1.1.1 auth-port 1645 acct-port 1646
radius-server host 2.2.2.2 auth-port 2000 acct-port 2001
radius-server host 3.3.3.3 auth-port 1645 acct-port 1646
```

AAA Server Group Selection Based on DNIS Example

The following example shows how to select RADIUS server groups based on DNIS to provide specific AAA services:

```
! This command enables AAA.
aaa new-model
!
! The following set of commands configures the RADIUS attributes for each server
! that will be associated with one of the defined server groups.
radius-server host 172.16.0.1 auth-port 1645 acct-port 1646 key cisco1
radius-server host 172.17.0.1 auth-port 1645 acct-port 1646 key cisco2
radius-server host 172.18.0.1 auth-port 1645 acct-port 1646 key cisco3
radius-server host 172.19.0.1 auth-port 1645 acct-port 1646 key cisco4
radius-server host 172.20.0.1 auth-port 1645 acct-port 1646 key cisco5

! The following commands define the sg1 RADIUS server group and associate servers
! with it.
aaa group server radius sg1
  server 172.16.0.1
  server 172.17.0.1
! The following commands define the sg2 RADIUS server group and associate a server
! with it.
aaa group server radius sg2
  server 172.18.0.1
! The following commands define the sg3 RADIUS server group and associate a server
! with it.
aaa group server radius sg3
  server 172.19.0.1
! The following commands define the default-group RADIUS server group and associate
! a server with it.
aaa group server radius default-group
  server 172.20.0.1
!
! The next set of commands configures default-group RADIUS server group parameters.
aaa authentication ppp default group default-group
aaa accounting network default start-stop group default-group
!
```

```
! The next set of commands enables DNIS mapping and maps DNIS numbers to the defined
! RADIUS server groups. In this configuration, all PPP connection requests using
! DNIS 7777 are sent to the sg1 server group. The accounting records for these
! connections (specifically, start-stop records) are handled by the sg2 server group.
! Calls with a DNIS of 8888 use server group sg3 for authentication and server group
! default-group for accounting. Calls with a DNIS of 9999 use server group
! default-group for authentication and server group sg3 for accounting records
! (stop records only). All other calls with DNIS other than the ones defined use the
! server group default-group for both authentication and stop-start accounting records.
aaa dnis map enable
aaa dnis map 7777 authentication ppp group sg1
aaa dnis map 7777 accounting network start-stop group sg2
aaa dnis map 8888 authentication ppp group sg3
aaa dnis map 9999 accounting network stop-only group sg3
```

AAA Preauthentication Examples

The following example shows a simple configuration that specifies that the DNIS number be used for preauthentication:

```
aaa preauth
 group radius
 dnis required
```

The following example shows a configuration that specifies that both the DNIS number and the CLID number be used for preauthentication. DNIS preauthentication will be performed first, followed by CLID preauthentication.

```
aaa preauth
 group radius
 dnis required
 clid required
```

The following example specifies that preauthentication be performed on all DNIS numbers except the two DNIS numbers specified in the DNIS group called “hawaii”:

```
aaa preauth
 group radius
 dnis required
 dnis bypass hawaii

dialer dnis group hawaii
 number 12345
 number 12346
```

The following example shows a sample AAA configuration with DNIS preauthentication:

```
aaa new-model
aaa authentication login CONSOLE none
aaa authentication login RADIUS_LIST group radius
aaa authentication login TAC_PLUS group tacacs+ enable
aaa authentication login V.120 none
aaa authentication enable default enable group tacacs+
aaa authentication ppp RADIUS_LIST if-needed group radius
aaa authorization exec RADIUS_LIST group radius if-authenticated
aaa authorization exec V.120 none
aaa authorization network default group radius if-authenticated
aaa authorization network RADIUS_LIST if-authenticated group radius
aaa authorization network V.120 group radius if-authenticated
aaa accounting suppress null-username
aaa accounting exec default start-stop group radius
aaa accounting commands 0 default start-stop group radius
```

```
aaa accounting network default start-stop group radius
aaa accounting connection default start-stop group radius
aaa accounting system default start-stop group radius
aaa preauth
 dnis password Cisco-DNIS
aaa nas port extended
!
radius-server configure-nas
radius-server host 10.0.0.0 auth-port 1645 acct-port 1646 non-standard
radius-server host 10.255.255.255 auth-port 1645 acct-port 1646 non-standard
radius-server retransmit 2
radius-server deadtime 1
radius-server attribute nas-port format c
radius-server unique-ident 18
radius-server key MyKey
```

Note To configure preauthentication, you must also set up preauthentication profiles on the RADIUS server.

RADIUS User Profile with RADIUS Tunneling Attributes Example

The following example shows a RADIUS user profile (Merit Daemon format) that includes RADIUS tunneling attributes. This entry supports two tunnels, one for L2F and the other for L2TP. The tag entries with :1 support L2F tunnels, and the tag entries with :2 support L2TP tunnels.

```
cisco.com Password = "cisco", Service-Type = Outbound
    Service-Type = Outbound,
    Tunnel-Type = :1:L2F,
    Tunnel-Medium-Type = :1:IP,
    Tunnel-Client-Endpoint = :1:"10.0.0.2",
    Tunnel-Server-Endpoint = :1:"10.0.0.3",
    Tunnel-Client-Auth-Id = :1:"l2f-cli-auth-id",
    Tunnel-Server-Auth-Id = :1:"l2f-svr-auth-id",
    Tunnel-Assignment-Id = :1:"l2f-assignment-id",
    Cisco-Avpair = "vpdn:nas-password=l2f-cli-pass",
    Cisco-Avpair = "vpdn:gw-password=l2f-svr-pass",
    Tunnel-Preference = :1:1,
    Tunnel-Type = :2:L2TP,
    Tunnel-Medium-Type = :2:IP,
    Tunnel-Client-Endpoint = :2:"10.0.0.2",
    Tunnel-Server-Endpoint = :2:"10.0.0.3",
    Tunnel-Client-Auth-Id = :2:"l2tp-cli-auth-id",
    Tunnel-Server-Auth-Id = :2:"l2tp-svr-auth-id",
    Tunnel-Assignment-Id = :2:"l2tp-assignment-id",
    Cisco-Avpair = "vpdn:l2tp-tunnel-password=l2tp-tnl-pass",
    Tunnel-Preference = :2:2
```

Guard Timer Examples

The following example shows an ISDN guard timer that is set at 8000 milliseconds. A call will be rejected if the RADIUS server has not responded to a preauthentication request when the timer expires.

```
interface serial1/0/0:23
 isdn guard-timer 8000 on-expiry reject

aaa preauth
 group radius
 dnis required
```

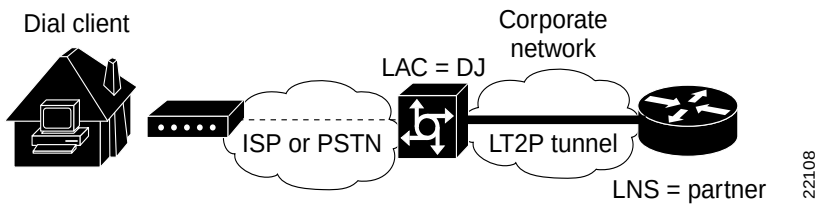
The following example shows a CAS guard timer that is set at 20,000 milliseconds. A call will be accepted if the RADIUS server has not responded to a preauthentication request when the timer expires.

```
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 ds0-group 0 timeslots 1-24 type e&m-fgb dtmf dnis
 cas-custom 0
 call guard-timer 20000 on-expiry accept

aaa preauth
group radius
dnis required
```

L2TP Access Concentrator Examples

The following example shows a basic L2TP configuration for the L2TP access concentrator (LAC) for the topology shown in Figure 12. The local name is not defined, so the host name used is the local name. Because the L2TP tunnel password is not defined, the username password is used. In this example, VPDN is configured locally on the LAC and does not take advantage of the new RADIUS tunnel attributes.

Figure 12 Topology for Configuration Examples




```
! Enable AAA globally.
aaa new-model
! Enable AAA authentication for PPP and list the default method to use for PPP
! authentication.
aaa authentication ppp default local
! Define the username as “DJ.”
username DJ password 7 030C5E070A00781B
! Enable VPDN.
vpdn enable
! Define VPDN group number 1.
vpdn-group 1
! Allow the LAC to respond to dialin requests using L2TP from IP address 172.21.9.13
! domain “cisco.com.”
 request dialin
  protocol l2tp
  domain cisco.com
 initiate-ip to 172.21.9.13
 local name nas-1
```

The following example shows how to configure the LAC if RADIUS tunnel attributes are supported. In this example, there is no local VPDN configuration on the LAC; the LAC, instead, is configured to query the remote RADIUS security server.

```
! Enable global AAA securities services.
aaa new-model
```

```
! Enable AAA authentication for PPP and list RADIUS as the default method to use
! for PPP authentication.
aaa authentication ppp default group radius local
! Enable AAA (network) authorization and list RADIUS as the default method to use for
! authorization.
aaa authorization network default group radius
! Define the username as “DJ.”
username DJ password 7 030C5E070A00781B
! Enable VPDN.
vpdn enable
! Configure the LAC to interface with the remote RADIUS security server.
radius host 171.69.1.1 auth-port 1645 acct-port 1646
radius-server key cisco
```

L2TP Network Server Examples

The following example shows a basic L2TP configuration with corresponding comments on the L2TP network server (LNS) for the topology shown in Figure 12:

```
! Enable AAA globally.
aaa new-model
! Enable AAA authentication for PPP and list the default method to use for PPP
! authentication.
aaa authentication ppp default local
! Define the username as “partner.”
username partner password 7 030C5E070A00781B
! Create virtual-template 1 and assign all values for virtual access interfaces.
interface Virtual-Template1
! Borrow the IP address from interface ethernet 1.
 ip unnumbered Ethernet0
! Disable multicast fast switching.
 no ip mroute-cache
! Use CHAP to authenticate PPP.
 ppp authentication chap
! Enable VPDN.
vpdn enable
! Create vpdn-group number 1.
vpdn-group 1
! Accept all dialin l2tp tunnels from virtual-template 1 from remote peer DJ.
 accept dialin l2tp virtual-template 1 remote DJ
  protocol any
   virtual-template 1
  terminate-from hostname nas1
local name hgw1
```

The following example shows how to configure the LNS with a basic L2F and L2TP configuration using RADIUS tunneling attributes:

```
aaa new-model
aaa authentication login default none
aaa authentication login console none
aaa authentication ppp default local group radius
aaa authorization network default group radius if-authenticated
!
username l2f-cli-auth-id password 0 l2f-cli-pass
username l2f-svr-auth-id password 0 l2f-svr-pass
username l2tp-svr-auth-id password 0 l2tp-tnl-pass
!
vpdn enable
vpdn search-order domain
!
vpdn-group 1
```

```
accept-dialin
protocol l2f
virtual-template 1
terminate-from hostname l2f-cli-auth-id
local name l2f-svr-auth-id
!
vpdn-group 2
accept-dialin
protocol l2tp
virtual-template 2
terminate-from hostname l2tp-cli-auth-id
local name l2tp-svr-auth-id
!
interface Ethernet1/0
ip address 10.0.0.3 255.255.255.0
no ip route-cache
no ip mroute-cache
!
interface Virtual-Template1
ip unnumbered Ethernet1/0
ppp authentication pap
!
interface Virtual-Template2
ip unnumbered Ethernet1/0
ppp authentication pap
!
radius-server host 1.1.1.1 auth-port 1645 acct-port 1646
radius-server key <deleted>
!
```

Per VRF AAA

First Published: June 4, 2001
Last Updated: November 17, 2008

The Per VRF AAA feature allows authentication, authorization, and accounting (AAA) on the basis of Virtual Private Network (VPN) routing and forwarding (VRF) instances.

For Cisco IOS Release 12.2(15)T or later releases, a customer template can be used, which may be stored either locally or remotely, and AAA services can be performed on the information that is stored in the customer template. This feature has also been referred to as the Dynamic Per VRF AAA feature.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Per VRF AAA" section on page 30.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Per VRF AAA

Before configuring the Per VRF AAA feature, you must enable AAA. (For information on completing this task, refer to the AAA chapters of the *Cisco IOS Security Configuration Guide*.

Restrictions for Per VRF AAA

- This feature is supported only for RADIUS servers.
- Operational parameters should be defined once per VRF rather than set per server group, because all functionality must be consistent between the network access server (NAS) and the AAA servers.
- The ability to configure a customer template either locally or remotely is available only for Cisco IOS Release 12.2(15)T and later releases.

Information About Per VRF AAA

When you use the Per VRF AAA feature, AAA services can be based on VRF instances. This feature permits the Provider Edge (PE) or Virtual Home Gateway (VHG) to communicate directly with the customer's RADIUS server, which is associated with the customer's Virtual Private Network (VPN), without having to go through a RADIUS proxy. Thus, ISPs can scale their VPN offerings more efficiently because they no longer have to use RADIUS proxies and ISPs can also provide their customers with additional flexibility.

How Per VRF AAA Works

To support AAA on a per customer basis, some AAA features must be made VRF aware. That is, ISPs must be able to define operational parameters—such as AAA server groups, method lists, system accounting, and protocol-specific parameters—and bind those parameters to a particular VRF instance. Defining and binding the operational parameters can be accomplished using one or more of the following methods:

- Virtual private dialup network (VPDN) virtual template or dialer interfaces that are configured for a specific customer
- Locally defined customer templates—Per VPN with customer definitions. The customer template is stored locally on the VHG. This method can be used to associate a remote user with a specific VPN based on the domain name or dialed number identification service (DNIS) and provide the VPN-specific configuration for virtual access interface and all operational parameters for the customer AAA server.

- Remotely defined customer templates—Per VPN with customer definitions that are stored on the service provider AAA server in a RADIUS profile. This method is used to associate a remote user with a specific VPN based on the domain name or DNIS and provide the VPN-specific configuration for the virtual access interface and all operational parameters for the AAA server of the customer.

Note The ability to configure locally or remotely defined customer templates is available only with Cisco IOS Release 12.2(15)T and later releases.

Benefits

Configuration Support

ISPs can partition AAA services on a per VRF basis. Thus, ISPs can allow their customers to control some of their own AAA services.

Server Group List Extension

The list of servers in server groups is extended to include the definitions of private servers in addition to references to the hosts in the global configuration, allowing access to both customer servers and global service provider servers simultaneously.

AAA Accounting Records

The Cisco implementation of AAA accounting provides “start” and “stop” record support for calls that have passed user authentication. Start and stop records are necessary for users employing accounting records to manage and monitor their networks.

New Vendor-Specific Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (VSA) attribute 26. Attribute 26 encapsulates VSAs, thereby, allowing vendors to support their own extended attributes otherwise not suitable for general use.

The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco’s vendor-ID is 9, and the supported option has vendor-type 1, which is named “cisco-avpair.” The value is a string of the following format:

```
protocol : attribute sep value *
```

“Protocol” is a value of the Cisco “protocol” attribute for a particular type of authorization. “Attribute” and “value” are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and “sep” is “=” for mandatory attributes and “*” for optional attributes. This format allows the full set of features available for TACACS+ authorization to be used also for RADIUS.

Table 1 summarizes the VSAs that are now supported with Per VRF AAA.

Table 1 VSAs supported with Per VRF AAA

VSA Name	Value Type	Description
Note Each VSA must have the prefix "template:" before the VSA name, unless a different prefix is explicitly stated.		
account-delay	string	This VSA must be "on." The functionality of this VSA is equal to the **aaa accounting delay-start** command for the customer template.
account-send-stop	string	This VSA must be "on." The functionality of this VSA is equal to the **aaa accounting send stop-record authentication** command with the **failure** keyword.
account-send-success-remote	string	This VSA must be "on." The functionality of this VSA is equal to the **aaa accounting send stop-record authentication** command with the **success** keyword.
attr-44	string	This VSA must be "access-req." The functionality of this VSA is equal to the **radius-server attribute 44 include-in-access-req** command.
ip-addr	string	This VSA specifies the IP address, followed by the mask that the router uses to indicate its own IP address and mask in negotiation with the client; for example, ip-addr=1.2.3.4 255.255.255.255
ip-unnumbered	string	This VSA specifies the name of an interface on the router. The functionality of this VSA is equal to the **ip unnumbered** command, which specifies an interface name such as "Loopback 0."
ip-vrf	string	This VSA specifies which VRF will be used for the packets of the end user. This VRF name should match the name that is used on the router via the **ip vrf forwarding** command.
peer-ip-pool	string	This VSA specifies the name of an IP address pool from which an address will be allocated for the peer. This pool should be configured using the **ip local pool** command or should be automatically downloadable via RADIUS.
ppp-acct-list	string	This VSA defines the accounting method list that is to be used for PPP sessions. The VSA syntax is as follows: "ppp-acct-list=[start-stop | stop-only | none] group X [group Y] [broadcast]." It is equal to the **aaa accounting network mylist** command functionality. The user must specify at least one of the following options: start-stop, stop-only, or none. If either start-stop or stop-only is specified, the user must specify at least one, but not more than four, group arguments. Each group name must consist of integers. The servers in the group should have already been identified in the access-accept via the VSA "rad-serv." After each group has been specified, the user can specify the broadcast option

VSA Name	Value Type	Description
ppp-authen-list	string	This VSA defines which authentication method list is to be used for PPP sessions and, if more than one method is specified, in what order the methods should be used. The VSA syntax is as follows: “ppp-authen-list=[groupX | local | local-case | none | if-needed],” which is equal to the **aaa authentication ppp mylist** command functionality. The user must specify at least one, but no more than four, authentication methods. If a server group is specified, the group name must be an integer. The servers in the group should have already been identified in the access-accept via the VSA “rad-serv.”
ppp-authen-type	string	This VSA allows the end user to specify at least one of the following authentication types: pap, chap, eap, ms-chap, ms-chap-v2, any, or a combination of the available types that is separated by spaces. The end user will be permitted to log in using only the methods that are specified in this VSA. PPP will attempt these authentication methods in the order presented in the attribute.
ppp-author-list	string	This VSA defines the authorization method list that is to be used for PPP sessions. It indicates which methods will be used and in what order. The VSA syntax is as follows: “ppp-author-list=[groupX] [local] [if-authenticated] [none],” which is equal to the **aaa authorization network mylist** command functionality. The user must specify at least one, but no more than four, authorization methods. If a server group is specified, the group name must be an integer. The servers in the group should have already been identified in the access-accept via the VSA “rad-serv.”
Note The RADIUS VSAs—rad-serv, rad-server-filter, rad-serv-source-if, and rad-serv-vrf—must have the prefix “aaa:” before the VSA name.		
rad-serv	string	This VSA indicates the IP address, key, timeout, and retransmit number of a server, as well as the group of the server. The VSA syntax is as follows: “rad-serv=a.b.c.d [key SomeKey] [auth-port X] [acct-port Y] [retransmit V] [timeout W].” Other than the IP address, all parameters are optional and can be issued in any order. If the optional parameters are not specified, their default values will be used. The key cannot contain any spaces; for “retransmit V,” “V” can range from 1-100; for “timeout W,” the “W” can range from 1-1000.

VSA Name	Value Type	Description
rad-serv-filter	string	The VSA syntax is as follows: “rad-serv-filter=authorization | accounting-request | reply-accept | reject-filtername.” The filtername must be defined via the **radius-server attribute list filtername** command.
rad-serv-source-if	string	This VSA specifies the name of the interface that is used for transmitting RADIUS packets. The specified interface must match the interface configured on the router.
rad-serv-vrf	string	This VSA specifies the name of the VRF that is used for transmitting RADIUS packets. The VRF name should match the name that was specified via the **ip vrf forwarding** command.

How to Configure Per VRF AAA

The following sections contain procedures for possible deployment scenarios for using the Per VRF AAA feature.

- Configuring Per VRF AAA, page 7 (required)
- Configuring Per VRF AAA Using Local Customer Templates, page 12 (optional)
- Configuring Per VRF AAA Using Remote Customer Templates, page 15 (optional)
- Verifying VRF Routing Configurations, page 17 (optional)
- Troubleshooting Per VRF AAA Configurations, page 18 (optional)

Configuring Per VRF AAA

This section contains the following procedures.

Configuring AAA

To enable AAA you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`aaa new-model` **Example:** `Router(config)# aaa new-model`	Enables AAA globally.

Configuring Server Groups

To configure server groups you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa group server radius** *groupname*

5. **server-private** *ip-address* [**auth-port port-number** | **acct-port port-number**] [**non-standard**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa group server radius** *groupname* **Example:** `Router(config)# aaa group server radius v2.44.com`	Groups different RADIUS server hosts into distinct lists and distinct methods. Enters server-group configuration mode.
Step 5	**server-private** *ip-address* [**auth-port** *port-number* | **acct-port** *port-number*] [**non-standard**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] **Example:** `Router(config-sg-radius)# server-private 10.10.130.2 auth-port 1600 key ww`	Configures the IP address of the private RADIUS server for the group server. **Note** If private server parameters are not specified, global configurations will be used. If global configurations are not specified, default values will be used.
Step 6	**exit** **Example:** `Router(config-sg-radius)# exit`	Exits from server-group configuration mode; returns to global configuration mode.

Configuring Authentication, Authorization, and Accounting for Per VRF AAA

To configure authentication, authorization, and accounting for Per VRF AAA, you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**

4. **aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*]
5. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} *method1* [*method2...*]
6. **aaa accounting system default** [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *groupname*
7. **aaa accounting delay-start** [**vrf** *vrf-name*]
8. **aaa accounting send stop-record authentication** {**failure** | **success** {**remote-server**}} [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authentication ppp method_list_v2.44.com group v2.44.com`	Specifies one or more AAA authentication methods for use on serial interfaces that are running PPP.
Step 5	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authorization network method_list_v2.44.com group v2.44.com`	Sets parameters that restrict user access to a network.
Step 6	**aaa accounting system default** [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *groupname* **Example:** `Router(config)# aaa accounting system default vrf v2.44.com start-stop group v2.44.com`	Enables AAA accounting of requested services for billing or security purposes when you use RADIUS.

	Command or Action	Purpose
Step 7	**aaa accounting delay-start** [**vrf** *vrf-name*] **Example:** `Router(config)# aaa acounting delay-start vrf v2.44.com`	Displays generation of the start accounting records until the user IP address is established.
Step 8	**aaa accounting send stop-record authentication** {**failure** | **success** {**remote-server**}} [**vrf** *vrf-name*] **Example:** `Router(config)# aaa accounting send stop-record authentication failure vrf v2.44.com`	Generates accounting stop records. When using the **failure** keyword a "stop" record will be sent for calls that are rejected during authentication. When using the **success** keyword a "stop" record will be sent for calls that meet one of the following criteria: • Calls that are authenticated by a remote AAA server when the call is terminated. • Calls that are not authenticated by a remote AAA server and the start record has been sent. • Calls that are successfully established and then terminated with the "stop-only" **aaa accounting** configuration. **Note** The **success** and **remote-server** keywords are available in Cisco IOS Release 12.4(2)T and later releases.

Configuring RADIUS-Specific Commands for Per VRF AAA

To configure RADIUS-specific commands for Per VRF AAA you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip radius source-interface** *subinterface-name* [**vrf** *vrf-name*]
4. **radius-server attribute 44 include-in-access-req** [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip radius source-interface** *subinterface-name* [**vrf** *vrf-name*] **Example:** `Router(config)# ip radius source-interface loopback55`	Forces RADIUS to use the IP address of a specified interface for all outgoing RADIUS packets and enables the specification on a per-VRF basis.
Step 4	**radius-server attribute 44 include-in-access-req** [**vrf** *vrf-name*] **Example:** `Router(config)# radius-server attribute 44 include-in-access-req vrf v2.44.com`	Sends RADIUS attribute 44 in access request packets before user authentication and enables the specification on a per-VRF basis.

Configuring Interface-Specific Commands for Per VRF AAA

To configure interface-specific commands for Per VRF AAA, you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip vrf forwarding** *vrf-name*
5. **ppp authentication** {*protocol1* [*protocol2*...]} *listname*
6. **ppp authorization** *list-name*
7. **ppp accounting default**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface loopback11`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding v2.44.com`	Associates a VRF with an interface.
Step 5	**ppp authentication** {*protocol1* [*protocol2...*]} *listname* **Example:** `Router(config-if)# ppp authentication chap callin V2_44_com`	Enables Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) or both and specifies the order in which CHAP and PAP authentication are selected on the interface.
Step 6	**ppp authorization** *list-name* **Example:** `Router(config-if)# ppp authorization V2_44_com`	Enables AAA authorization on the selected interface.
Step 7	**ppp accounting default** **Example:** `Router(config-if)# ppp accounting default`	Enables AAA accounting services on the selected interface.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits interface configuration mode.

Configuring Per VRF AAA Using Local Customer Templates

This section contains the following procedures:

Configuring AAA

Perform the tasks as outlined in the "Configuring Per VRF AAA" section on page 7.

Configuring Server Groups

Perform the tasks as outlined in the "Configuring Server Groups" section on page 7.

Configuring Authentication, Authorization, and Accounting for Per VRF AAA

Perform the tasks as outlined in the "Configuring Authentication, Authorization, and Accounting for Per VRF AAA" section on page 8.

Configuring Authorization for Per VRF AAA with Local Customer Templates

To configure authorization for Per VRF AAA with local templates, you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization template**
4. **aaa authorization network default local**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authorization template** **Example:** `Router(config)# aaa authorization template`	Enables the use of local or remote templates.
Step 4	**aaa authorization network default local** **Example:** `Router(config)# aaa authorization network default local`	Specifies local as the default method for authorization.

Configuring Local Customer Templates

To configure local customer templates, you need to complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn search-order domain**
4. **template** *name* [**default** | **exit** | **multilink** | **no** | **peer** | **ppp**]
5. **peer default ip address pool** *pool-name*
6. **ppp authentication** {*protocol1* [*protocol2*...]} [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**]

7. **ppp authorization** [**default** | *list-name*]
8. **aaa accounting** {**auth-proxy** | **system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *groupname*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vpdn search-order domain** **Example:** `Router (config)# vpdn search-order domain`	Looks up the profiles based on domain.
Step 4	**template** *name* [**default** | **exit** | **multilink** | **no** | **peer** | **ppp**] **Example:** `Router (config)# template v2.44.com`	Creates a customer profile template and assigns a unique name that relates to the customer that will be receiving it. Enters template configuration mode. **Note** Steps 5, 6, and 7 are optional. Enter **multilink**, **peer**, and **ppp** keywords appropriate to customer application requirements.
Step 5	**peer default ip address pool** *pool-name* **Example:** `Router(config-template)# peer default ip address pool v2_44_com_pool`	(Optional) Specifies that the customer profile to which this template is attached will use a local IP address pool with the specified name.
Step 6	**ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] **Example:** `Router(config-template)# ppp authentication chap`	(Optional) Sets the PPP link authentication method.
Step 7	**ppp authorization** [**default** | *list-name*] **Example:** `Router(config-template)# ppp authorization v2_44_com`	(Optional) Sets the PPP link authorization method.

	Command or Action	Purpose
Step 8	**aaa accounting** {**auth-proxy** | **system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **none**} [**broadcast**] **group** *groupname* **Example:** `Router(config-template)# aaa accounting v2_44_com`	(Optional) Enables AAA operational parameters for the specified customer profile.
Step 9	**exit** **Example:** `Router(config-template)# exit`	Exits from template configuration mode; returns to global configuration mode.

Configuring Per VRF AAA Using Remote Customer Templates

This section contains the following procedures:

Configuring AAA

Perform the tasks as outlined in the "Configuring Per VRF AAA" section on page 7.

Configuring Server Groups

Perform the tasks as outlined in the "Configuring Server Groups" section on page 12.

Configuring Authentication for Per VRF AAA with Remote Customer Profiles

To configure authentication for Per VRF AAA with remote customer profiles, you need to perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*]
4. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [[*method1* [*method2...*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authentication ppp** {**default** | *list-name*} *method1* [*method2*...] **Example:** `Router(config)# ppp authentication ppp default group radius`	Specifies one or more authentication, authorization, and accounting (AAA) authentication methods for use on serial interfaces that are running PPP.
Step 4	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [[*method1* [*method2*...] **Example:** `Router(config)# aaa authorization network default group sp`	Sets parameters that restrict user access to a network.

Configuring Authorization for Per VRF AAA with Remote Customer Profiles

To configuring authorization for Per VRF AAA with remote customer profiles, you need to perform the following step.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization template**
4. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [[*method1* [*method2*...]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authorization template** **Example:** `Router(config)# aaa authorization template`	Enables use of local or remote templates.
Step 4	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [[*method1* [*method2*...] **Example:** `Router(config)# aaa authorization network default sp`	Specifies the server group that is named as the default method for authorization.

Configuring the RADIUS Profile on the SP RADIUS Server

Configure the RADIUS profile on the Service Provider (SP) RADIUS server. See the "Per VRF AAA Using a Remote RADIUS Customer Template: Example" section on page 20 for an example of how to update the RADIUS profile.

Verifying VRF Routing Configurations

To verify VRF routing configurations, you need to complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **show ip route vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**show ip route vrf** *vrf-name* **Example:** `Router(config)# show ip route vrf northvrf`	Displays the IP routing table associated with a VRF.

Troubleshooting Per VRF AAA Configurations

To troubleshoot the Per VRF AAA feature, use at least one of the following commands in EXEC mode:

Command	Purpose
`Router#` **debug aaa accounting**	Displays information on accountable events as they occur.
`Router#` **debug aaa authentication**	Displays information on AAA authentication.
`Router#` **debug aaa authorization**	Displays information on AAA authorization.
`Router#` **debug ppp negotiation**	Displays information on traffic and exchanges in an internetwork implementing PPP.
`Router#` **debug radius**	Displays information associated with RADIUS.
`Router#` **debug vpdn event**	Displays Layer 2 Transport Protocol (L2TP) errors and events that are a part of normal tunnel establishment or shutdown for VPNs.
`Router#` **debug vpdn error**	Displays debug traces for VPN.

Configuration Examples for Per VRF AAA

This section provides the following configuration examples:

- Per VRF Configuration: Examples, page 19
- Customer Template: Examples, page 20
- AAA Accounting Stop Records: Examples, page 22

Per VRF Configuration: Examples

This section provides the following configuration examples:

Per VRF AAA: Example

The following example shows how to configure the Per VRF AAA feature using a AAA server group with associated private servers:

```
aaa new-model

aaa authentication ppp method_list_v1.55.com group v1.55.com
aaa authorization network method_list_v1.55.com group v1.55.com
aaa accounting network method_list_v1.55.com start-stop group v1.55.com
aaa accounting system default vrf v1.55.com start-stop group v1.55.com
aaa accounting delay-start vrf v1.55.com
aaa accounting send stop-record authentication failure vrf v1.55.com

aaa group server radius v1.55.com
    server-private 10.10.132.4 auth-port 1645 acct-port 1646 key ww
    ip vrf forwarding v1.55.com

ip radius source-interface loopback55
radius-server attribute 44 include-in-access-req vrf v1.55.com
```

Per VRF AAA Using a Locally Defined Customer Template: Example

The following example shows how to configure the Per VRF AAA feature using a locally defined customer template with a AAA server group that has associated private servers:

```
aaa new-model
aaa authentication ppp method_list_v1.55.com group v1.55.com
aaa authorization network method_list_v1.55.com group v1.55.com
aaa authorization network default local
aaa authorization template
aaa accounting network method_list_v1.55.com start-stop group v1.55.com
aaa accounting system default vrf v1.55.com start-stop group v1.55.com

aaa group server radius V1_55_com
  server-private 10.10.132.4 auth-port 1645 acct-port 1646 key ww
  ip vrf forwarding V1.55.com

template V1.55.com
  peer default ip address pool V1_55_com_pool
  ppp authentication chap callin V1_55_com
  ppp authorization V1_55_com
  ppp accounting V1_55_com
  aaa accounting delay-start
  aaa accounting send stop-record authentication failure
  radius-server attribute 44 include-in-access-req
  ip vrf forwarding v1.55.com
  ip radius source-interface Loopback55
```

Per VRF AAA Using a Remote RADIUS Customer Template: Example

The following examples shows how to configure the Per VRF AAA feature using a remotely defined customer template on the SP RADIUS server with a AAA server group that has associated private servers:

```
aaa new-model
aaa authentication ppp default group radius
aaa authorization template
aaa authorization network default group sp

aaa group server radius sp
   server 10.3.3.3

radius-server host 10.3.3.3 auth-port 1645 acct-port 1646 key sp_key
```

The following RADIUS server profile is configured on the SP RADIUS server:

```
cisco-avpair = "aaa:rad-serv#1=10.10.132.4 key ww"
cisco-avpair = "aaa:rad-serv-vrf#1=V1.55.com"
cisco-avpair = "aaa:rad-serv-source-if#1=Loopback 55"
cisco-avpair = "template:ppp-authen-list=group 1"
cisco-avpair = "template:ppp-author-list=group 1"
cisco-avpair = "template:ppp-acct-list= start-stop group 1"
cisco-avpair = "template:account-delay=on"
cisco-avpair = "template:account-send-stop=on"
cisco-avpair = "template:rad-attr44=access-req"
cisco-avpair = "template:peer-ip-pool=V1.55-pool"
cisco-avpair = "template:ip-vrf=V1.55.com"
cisco-avpair = "template:ip-unnumbered=Loopback 55"
framed-protocol = ppp
service-type = framed
```

Customer Template: Examples

This section provides the following configuration examples:

- Locally Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting: Example, page 20
- Remotely Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting: Example, page 21

Locally Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting: Example

The following example shows how to create a locally configured template for a single customer, configuring additional features including RADIUS attribute screening and broadcast accounting:

```
aaa authentication ppp default local group radius
aaa authentication ppp V1_55_com group V1_55_com
aaa authorization template
aaa authorization network default local group radius
aaa authorization network V1_55_com group V1_55_com
aaa accounting network V1_55_com start-stop broadcast group V1_55_com group SP_AAA_server

aaa group server radius SP_AAA_server
 server 10.10.100.7 auth-port 1645 acct-port 1646
```

```
aaa group server radius V1_55_com
 server-private 10.10.132.4 auth-port 1645 acct-port 1646
 authorization accept min-author
 accounting accept usage-only
 ip vrf forwarding V1.55.com

ip vrf V1.55.com
 rd 1:55
 route-target export 1:55
 route-target import 1:55

template V1.55.com
 peer default ip address pool V1.55-pool
 ppp authentication chap callin V1_55_com
 ppp authorization V1_55_com
 ppp accounting V1_55_com
 aaa accounting delay-start
 aaa accounting send stop-record authentication failure
 radius-server attribute 44 include-in-access-req

vpdn-group V1.55
 accept-dialin
  protocol l2tp
  virtual-template 13
 terminate-from hostname lac-lb-V1.55
 source-ip 10.10.104.12
 lcp renegotiation always
 l2tp tunnel password 7 060506324F41

interface Virtual-Template13
 ip vrf forwarding V1.55.com
 ip unnumbered Loopback55
 ppp authentication chap callin
 ppp multilink

ip local pool V1.55-pool 10.1.55.10 10.1.55.19 group V1.55-group

ip radius source-interface Loopback0
ip radius source-interface Loopback55 vrf V1.55.com

radius-server attribute list min-author
 attribute 6-7,22,27-28,242
radius-server attribute list usage-only
 attribute 1,40,42-43,46

radius-server host 10.10.100.7 auth-port 1645 acct-port 1646 key ww
radius-server host 10.10.132.4 auth-port 1645 acct-port 1646 key ww
```

Remotely Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting: Example

The following example shows how to create a remotely configured template for a single customer, configuring additional features including RADIUS attribute screening and broadcast accounting:

```
aaa authentication ppp default local group radius
aaa authorization template
aaa authorization network default local group radius

ip vrf V1.55.com
 rd 1:55
```

```
 route-target export 1:55
 route-target import 1:55

vpdn-group V1.55
 accept-dialin
  protocol l2tp
  virtual-template 13
 terminate-from hostname lac-lb-V1.55
 source-ip 10.10.104.12
 lcp renegotiation always
 l2tp tunnel password 7 060506324F41

interface Virtual-Template13
 no ip address
 ppp authentication chap callin
 ppp multilink

ip local pool V1.55-pool 10.1.55.10 10.1.55.19 group V1.55-group

radius-server attribute list min-author
 attribute 6-7,22,27-28,242
radius-server attribute list usage-only
 attribute 1,40,42-43,46
```

The customer template is stored as a RADIUS server profile for v1.55.com.

```
cisco-avpair = "aaa:rad-serv#1=10.10.132.4 key ww"
cisco-avpair = "aaa:rad-serv-vrf#1=V1.55.com"
cisco-avpair = "aaa:rad-serv-source-if#1=Loopback 55"
cisco-avpair = "aaa:rad-serv#2=10.10.100.7 key ww"
cisco-avpair = "aaa:rad-serv-source-if#2=Loopback 0"
cisco-avpair = "template:ppp-authen-list=group 1"
cisco-avpair = "template:ppp-author-list=group 1"
cisco-avpair = "template:ppp-acct-list= start-stop group 1 group 2 broadcast"
cisco-avpair = "template:account-delay=on"
cisco-avpair = "template:account-send-stop=on"
cisco-avpair = "template:rad-attr44=access-req"
cisco-avpair = "aaa:rad-serv-filter#1=authorization accept min-author"
cisco-avpair = "aaa:rad-serv-filter#1=accounting accept usage-only"
cisco-avpair = "template:peer-ip-pool=V1.55-pool"
cisco-avpair = "template:ip-vrf=V1.55.com"
cisco-avpair = "template:ip-unnumbered=Loopback 55"
framed-protocol = ppp
service-type = framed
```

AAA Accounting Stop Records: Examples

The following AAA accounting stop record examples show how to configure the **aaa accounting send stop-record authentication** command to control the generation of “stop” records when the **aaa accounting** command is issued with the **start-stop** or **stop-only** keyword.

Note The **success** and **remote-server** keywords are available in Cisco IOS Release 12.4(2)T and later releases.

This section provides the following configuration examples:

AAA Accounting Stop Record and Successful Call: Example

The following example shows “start” and “stop” records being sent for a successful call when the **aaa accounting send stop-record authentication** command is issued with the **failure** keyword.

```
Router# show running config | include aaa
.
.
.
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default local
aaa accounting send stop-record authentication failure
aaa accounting network default start-stop group radius
.
.
.
*Jul  7 03:28:31.543: AAA/BIND(00000018): Bind i/f Virtual-Template2
*Jul  7 03:28:31.547: ppp14 AAA/AUTHOR/LCP: Authorization succeeds trivially
*Jul  7 03:28:33.555: AAA/AUTHOR (0x18): Pick method list 'default'
*Jul  7 03:28:33.555: AAA/BIND(00000019): Bind i/f
*Jul  7 03:28:33.555:  Tnl 5192 L2TP: O SCCRQ
*Jul  7 03:28:33.555:  Tnl 5192 L2TP: O SCCRQ, flg TLS, ver 2, len 141, tnl 0,
ns 0, nr 0
         C8 02 00 8D 00 00 00 00 00 00 00 80 08 00 00
         00 00 00 01 80 08 00 00 00 02 01 00 00 08 00 00
         00 06 11 30 80 10 00 00 00 07 4C 41 43 2D 74 75
         6E 6E 65 6C 00 19 00 00 00 08 43 69 73 63 6F 20
         53 79 73 74 65 6D 73 ...
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse  AVP 0, len 8, flag 0x8000 (M)
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse SCCRP
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse  AVP 2, len 8, flag 0x8000 (M)
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Protocol Ver 256
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse  AVP 3, len 10, flag 0x8000 (M)
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Framing Cap 0x0
*Jul  7 03:28:33.563:  Tnl 5192 L2TP: Parse  AVP 4, len 10, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Bearer Cap 0x0
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 6, len 8, flag 0x0
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Firmware Ver 0x1120
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 7, len 16, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Hostname LNS-tunnel
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 8, len 25, flag 0x0
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Vendor Name Cisco Systems, Inc.
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 9, len 8, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Assigned Tunnel ID 6897
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 10, len 8, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Rx Window Size 20050
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 11, len 22, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Chlng
         81 13 03 F6 A8 E4 1D DD 25 18 25 6E 67 8C 7C 39
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Parse  AVP 13, len 22, flag 0x8000 (M)
*Jul  7 03:28:33.567:  Tnl 5192 L2TP: Chlng Resp
         4D 52 91 DC 1A 43 B3 31 B4 F5 B8 E1 88 22 4F 41
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: No missing AVPs in SCCRP
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: I SCCRP, flg TLS, ver 2, len 157, tnl
5192, ns 0, nr 1
contiguous pak, size 157
```

```
         C8 02 00 9D 14 48 00 00 00 00 01 80 08 00 00
         00 00 00 02 80 08 00 00 00 02 01 00 80 0A 00 00
         00 03 00 00 00 00 80 0A 00 00 00 04 00 00 00 00
         00 08 00 00 00 06 11 20 80 10 00 00 00 07 4C 4E
         53 2D 74 75 6E 6E 65 6C ...
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: I SCCRP from LNS-tunnel
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: O SCCCN  to LNS-tunnel tnlid 6897
*Jul  7 03:28:33.571:  Tnl 5192 L2TP: O SCCCN, flg TLS, ver 2, len 42, tnl
6897, ns 1, nr 1
         C8 02 00 2A 1A F1 00 00 00 01 00 01 80 08 00 00
         00 00 00 03 80 16 00 00 00 0D 32 24 17 BC 6A 19
         B1 79 F3 F9 A9 D4 67 7D 9A DB
*Jul  7 03:28:33.571: uid:14 Tnl/Sn 5192/11 L2TP: O ICRQ to LNS-tunnel 6897/0
*Jul  7 03:28:33.571: uid:14 Tnl/Sn 5192/11 L2TP: O ICRQ, flg TLS, ver 2, len
63, tnl 6897, lsid 11, rsid 0, ns 2, nr 1
         C8 02 00 3F 1A F1 00 00 00 02 00 01 80 08 00 00
         00 00 00 0A 80 0A 00 00 00 0F C8 14 B4 03 80 08
         00 00 00 0E 00 0B 80 0A 00 00 00 12 00 00 00 00
         00 0F 00 09 00 64 0F 10 09 02 02 00 1B 00 00
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: Parse  AVP 0, len 8, flag
0x8000 (M)
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: Parse ICRP
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: Parse  AVP 14, len 8, flag
0x8000 (M)
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: Assigned Call ID 5
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: No missing AVPs in ICRP
*Jul  7 03:28:33.575: uid:14 Tnl/Sn 5192/11 L2TP: I ICRP, flg TLS, ver 2, len
28, tnl 5192, lsid 11, rsid 0, ns 1, nr 3
contiguous pak, size 28
         C8 02 00 1C 14 48 00 0B 00 01 00 03 80 08 00 00
         00 00 00 0B 80 08 00 00 00 0E 00 05
*Jul  7 03:28:33.579: uid:14 Tnl/Sn 5192/11 L2TP: O ICCN to LNS-tunnel 6897/5
*Jul  7 03:28:33.579: uid:14 Tnl/Sn 5192/11 L2TP: O ICCN, flg TLS, ver 2, len
167, tnl 6897, lsid 11, rsid 5, ns 3, nr 2
         C8 02 00 A7 1A F1 00 05 00 03 00 02 80 08 00 00
         00 00 00 0C 80 0A 00 00 00 18 06 1A 80 00 00 0A
         00 00 00 26 06 1A 80 00 80 0A 00 00 00 13 00 00
         00 01 00 15 00 00 00 1B 01 04 05 D4 03 05 C2 23
         05 05 06 0A 0B E2 7A ...
*Jul  7 03:28:33.579: RADIUS/ENCODE(00000018):Orig. component type = PPoE
*Jul  7 03:28:33.579: RADIUS(00000018): Config NAS IP: 10.0.0.0
*Jul  7 03:28:33.579: RADIUS(00000018): sending
*Jul  7 03:28:33.579: RADIUS/ENCODE: Best Local IP-Address 10.0.1.123 for
Radius-Server 172.19.192.238
*Jul  7 03:28:33.579: RADIUS(00000018): Send Accounting-Request to
172.19.192.238:2196 id 1646/23, len 176
*Jul  7 03:28:33.579: RADIUS:  authenticator 3C 81 D6 C5 2B 6D 21 8E - 19 FF
43 B5 41 86 A8 A5
*Jul  7 03:28:33.579: RADIUS:  Acct-Session-Id     [44]  10  "00000023"
*Jul  7 03:28:33.579: RADIUS:  Framed-Protocol     [7]   6
PPP                      [1]
*Jul  7 03:28:33.579: RADIUS:  Tunnel-Medium-Type  [65]  6
00:IPv4                  [1]
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Client-Endpoi[66]  10  "10.0.0.1"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Server-Endpoi[67]  10  "10.0.0.2"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Assignment-Id[82]  5   "lac"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Type         [64]  6
00:L2TP                  [3]
*Jul  7 03:28:33.583: RADIUS:  Acct-Tunnel-Connecti[68]  12  "3356800003"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Client-Auth-I[90]  12  "LAC-tunnel"
*Jul  7 03:28:33.583: RADIUS:  Tunnel-Server-Auth-I[91]  12  "LNS-tunnel"
*Jul  7 03:28:33.583: RADIUS:  User-Name           [1]   16  "user@example.com"
*Jul  7 03:28:33.583: RADIUS:  Acct-Authentic      [45]  6
Local                    [2]
```

```
*Jul  7 03:28:33.583: RADIUS:  Acct-Status-Type    [40]  6
Start                     [1]
*Jul  7 03:28:33.583: RADIUS:  NAS-Port-Type       [61]  6
Virtual                   [5]
*Jul  7 03:28:33.583: RADIUS:  NAS-Port            [5]   6
0
*Jul  7 03:28:33.583: RADIUS:  NAS-Port-Id         [87]  9   "0/0/0/0"
*Jul  7 03:28:33.583: RADIUS:  Service-Type        [6]   6
Framed                    [2]
*Jul  7 03:28:33.583: RADIUS:  NAS-IP-Address      [4]   6
10.0.1.123
*Jul  7 03:28:33.583: RADIUS:  Acct-Delay-Time     [41]  6
0
*Jul  7 03:28:33.683: RADIUS: Received from id 1646/23 172.19.192.238:2196,
Accounting-response, len 20
*Jul  7 03:28:33.683: RADIUS:  authenticator 1C E9 53 42 A2 8A 58 9A - C3 CC
1D 79 9F A4 6F 3A
```

AAA Accounting Stop Record and Rejected Call: Example

The following example shows the “stop” record being sent for a rejected call during authentication when the **aaa accounting send stop-record authentication** command is issued with the **success** keyword.

```
Router# show running config | include aaa
.
.
.
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default local
aaa accounting send stop-record authentication success remote-server
aaa accounting network default start-stop group radius

Router#

*Jul  7 03:39:40.199: AAA/BIND(00000026): Bind i/f Virtual-Template2
*Jul  7 03:39:40.199: ppp21 AAA/AUTHOR/LCP: Authorization succeeds trivially
*Jul  7 03:39:42.199: RADIUS/ENCODE(00000026):Orig. component type = PPoE
*Jul  7 03:39:42.199: RADIUS:  AAA Unsupported     [156] 7
*Jul  7 03:39:42.199: RADIUS:   30 2F 30 2F
30                                  [0/0/0]
*Jul  7 03:39:42.199: RADIUS(00000026): Config NAS IP: 10.0.0.0
*Jul  7 03:39:42.199: RADIUS/ENCODE(00000026): acct_session_id: 55
*Jul  7 03:39:42.199: RADIUS(00000026): sending
*Jul  7 03:39:42.199: RADIUS/ENCODE: Best Local IP-Address 10.0.1.123 for
Radius-Server 172.19.192.238
*Jul  7 03:39:42.199: RADIUS(00000026): Send Access-Request to
172.19.192.238:2195 id 1645/14, len 94
*Jul  7 03:39:42.199: RADIUS:  authenticator A6 D1 6B A4 76 9D 52 CF - 33 5D
16 BE AC 7E 5F A6
*Jul  7 03:39:42.199: RADIUS:  Framed-Protocol     [7]   6
PPP                       [1]
*Jul  7 03:39:42.199: RADIUS:  User-Name           [1]   16  "user@example.com"
*Jul  7 03:39:42.199: RADIUS:  CHAP-Password       [3]   19  *
*Jul  7 03:39:42.199: RADIUS:  NAS-Port-Type       [61]  6
Virtual                   [5]
*Jul  7 03:39:42.199: RADIUS:  NAS-Port            [5]   6
0
*Jul  7 03:39:42.199: RADIUS:  NAS-Port-Id         [87]  9   "0/0/0/0"
*Jul  7 03:39:42.199: RADIUS:  Service-Type        [6]   6
Framed                    [2]
```

```
*Jul  7 03:39:42.199: RADIUS:  NAS-IP-Address      [4]   6
10.0.1.123
*Jul  7 03:39:42.271: RADIUS: Received from id 1645/14 172.19.192.238:2195,
Access-Accept, len 194
*Jul  7 03:39:42.271: RADIUS:  authenticator 30 AD FF 8E 59 0C E4 6C - BA 11
23 63 81 DE 6F D7
*Jul  7 03:39:42.271: RADIUS:  Framed-Protocol     [7]   6
PPP                       [1]
*Jul  7 03:39:42.275: RADIUS:  Service-Type        [6]   6
Framed                    [2]
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  26
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   20  "vpdn:tunnel-
id=lac"
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  29
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   23  "vpdn:tunnel-
type=l2tp"
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  30
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   24  "vpdn:gw-
password=cisco"
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  31
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   25  "vpdn:nas-
password=cisco"
*Jul  7 03:39:42.275: RADIUS:  Vendor, Cisco       [26]  34
*Jul  7 03:39:42.275: RADIUS:   Cisco AVpair       [1]   28  "vpdn:ip-
addresses=10.0.0.2"
*Jul  7 03:39:42.275: RADIUS:  Service-Type        [6]   6
Framed                    [2]
*Jul  7 03:39:42.275: RADIUS:  Framed-Protocol     [7]   6
PPP                       [1]
*Jul  7 03:39:42.275: RADIUS(00000026): Received from id 1645/14
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: Framed-Protocol
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: service-type
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: tunnel-id
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: tunnel-type
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: gw-password
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: nas-password
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: ip-addresses
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: service-type
*Jul  7 03:39:42.275: ppp21 PPP/AAA: Check Attr: Framed-Protocol
*Jul  7 03:39:42.279: AAA/BIND(00000027): Bind i/f
*Jul  7 03:39:42.279:  Tnl 21407 L2TP: O SCCRQ
*Jul  7 03:39:42.279:  Tnl 21407 L2TP: O SCCRQ, flg TLS, ver 2, len 134, tnl
0, ns 0, nr 0
         C8 02 00 86 00 00 00 00 00 00 00 80 08 00 00
         00 00 00 01 80 08 00 00 00 02 01 00 00 08 00 00
         00 06 11 30 80 09 00 00 00 07 6C 61 63 00 19 00
         00 00 08 43 69 73 63 6F 20 53 79 73 74 65 6D 73
         2C 20 49 6E 63 2E 80 ...
*Jul  7 03:39:49.279:  Tnl 21407 L2TP: O StopCCN
*Jul  7 03:39:49.279:  Tnl 21407 L2TP: O StopCCN, flg TLS, ver 2, len 66, tnl
0, ns 1, nr 0
         C8 02 00 42 00 00 00 00 00 01 00 00 80 08 00 00
         00 00 00 04 80 1E 00 00 00 01 00 02 00 06 54 6F
         6F 20 6D 61 6E 79 20 72 65 74 72 61 6E 73 6D 69
         74 73 00 08 00 09 00 69 00 01 80 08 00 00 00 09
         53 9F
*Jul  7 03:39:49.279: RADIUS/ENCODE(00000026):Orig. component type = PPoE
*Jul  7 03:39:49.279: RADIUS(00000026): Config NAS IP: 10.0.0.0
*Jul  7 03:39:49.279: RADIUS(00000026): sending
*Jul  7 03:39:49.279: RADIUS/ENCODE: Best Local IP-Address 10.0.1.123 for
Radius-Server 172.19.192.238
*Jul  7 03:39:49.279: RADIUS(00000026): Send Accounting-Request to
172.19.192.238:2196 id 1646/32, len 179
*Jul  7 03:39:49.279: RADIUS:  authenticator 0A 85 2F F0 65 6F 25 E1 - 97 54
```

```
CC BF EA F7 62 89
*Jul  7 03:39:49.279: RADIUS:  Acct-Session-Id     [44]  10  "00000037"
*Jul  7 03:39:49.279: RADIUS:  Framed-Protocol     [7]   6
PPP                       [1]
*Jul  7 03:39:49.279: RADIUS:  Tunnel-Medium-Type  [65]  6
00:IPv4                   [1]
*Jul  7 03:39:49.279: RADIUS:  Tunnel-Client-Endpoi[66]  10  "10.0.0.1"
*Jul  7 03:39:49.279: RADIUS:  Tunnel-Server-Endpoi[67]  10  "10.0.0.2"
*Jul  7 03:39:49.283: RADIUS:  Tunnel-Type         [64]  6
00:L2TP                   [3]
*Jul  7 03:39:49.283: RADIUS:  Acct-Tunnel-Connecti[68]  3   "0"
*Jul  7 03:39:49.283: RADIUS:  Tunnel-Client-Auth-I[90]  5   "lac"
*Jul  7 03:39:49.283: RADIUS:  User-Name           [1]   16  "user@example.com"
*Jul  7 03:39:49.283: RADIUS:  Acct-Authentic      [45]  6
RADIUS                    [1]
*Jul  7 03:39:49.283: RADIUS:  Acct-Session-Time   [46]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Input-Octets   [42]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Output-Octets  [43]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Input-Packets  [47]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Output-Packets [48]  6
0
*Jul  7 03:39:49.283: RADIUS:  Acct-Terminate-Cause[49]  6   nas-
error                  [9]
*Jul  7 03:39:49.283: RADIUS:  Acct-Status-Type    [40]  6
Stop                      [2]
*Jul  7 03:39:49.283: RADIUS:  NAS-Port-Type       [61]  6
Virtual                   [5]
*Jul  7 03:39:49.283: RADIUS:  NAS-Port            [5]   6
0
*Jul  7 03:39:49.283: RADIUS:  NAS-Port-Id         [87]  9   "0/0/0/0"
*Jul  7 03:39:49.283: RADIUS:  Service-Type        [6]   6
Framed                    [2]
*Jul  7 03:39:49.283: RADIUS:  NAS-IP-Address      [4]   6
10.0.1.123
*Jul  7 03:39:49.283: RADIUS:  Acct-Delay-Time     [41]  6
0
*Jul  7 03:39:49.335: RADIUS: Received from id 1646/32 172.19.192.238:2196,
Accounting-response, len 20
*Jul  7 03:39:49.335: RADIUS:  authenticator C8 C4 61 AF 4D 9F 78 07 - 94 2B
44 44 17 56 EC 03
```

Additional References

The following sections provide references related to Per VRF AAA.

Related Documents

Related Topic	Document Title
AAA: Configuring Server Groups	*Cisco IOS Security Configuration Guide*, Release 12.4
RADIUS Attribute Screening	
RADIUS Debug Enhancements	
Broadcast Accounting	*AAA Broadcast Accounting*, Release 12.1(1)T
Cisco IOS Security Commands	*Cisco IOS Security Command Reference*, Release 12.4
Cisco IOS Switching Services Commands	*Cisco Switching Services Command Reference*, Release 12.2
Configuring Multiprotocol Label Switching	"Configuring Multiprotocol Label Switching" chapter in the *Cisco IOS Switching Services Configuration Guide*, Release 12.2
Configuring Virtual Templates section	"Virtual Templates, Profiles, and Networks" chapter in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.
For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **aaa accounting**
- **aaa accounting delay-start**
- **aaa accounting send stop-record authentication**
- **aaa authorization template**
- **ip radius source-interface**
- **ip vrf forwarding (server-group)**
- **radius-server attribute 44 include-in-access-req**
- **radius-server domain-stripping**
- **server-private (RADIUS)**

Feature Information for Per VRF AAA

Table 2 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS, and Cisco IOS XE, software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for Per VRF AAA

Feature Name	Releases	Feature Information
Per VRF AAA	12.2(1)DX 12.2(2)DD 12.2(4)B 12.2(13)T 12.2(15)T 12.4(2)T 12.2(28)SB 12.2(33)SR 12.2(33)SXI 12.2(33)SXH4 XE 2.1	The Per VRF AAA feature allows authentication, authorization, and accounting (AAA) on the basis of Virtual Private Network (VPN) routing and forwarding (VRF) instances. For Cisco IOS Release 12.2(15)T or later releases, you can use a customer template, which may be stored either locally or remotely, and AAA services can be performed on the information that is stored in the customer template. In 12.2(1)DX, this feature was introduced on the Cisco 7200 series and the Cisco 7401ASR. In 12.2(2)DD, the **ip vrf forwarding (server-group)** and **radius-server domain-stripping** commands were added. In 12.2(15)T, the **aaa authorization template** command was added In 12.4(2)T, the **aaa accounting send stop-record authentication** command was updated with additional support for AAA accounting stop records. In 12.2(33)SRC, dynamic configuration of AAA was introduced. In Cisco IOS Release 12.2(33)SXI, this feature was introduced. In Cisco IOS Release 12.2(33)SXH4, this feature was introduced. The following commands were introduced or modified: **aaa accounting**, **aaa accounting delay-start**, **ip radius source-interface**, **radius-server attribute 44 include-in-access-req**, **server-private (RADIUS)**. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers.

Glossary

AAA—authentication, authorization, and accounting. A framework of security services that provide the method for identifying users (authentication), for remote access control (authorization), and for collecting and sending security server information used for billing, auditing, and reporting (accounting).

L2TP—Layer 2 Tunnel Protocol. A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

PE—Provider Edge. Networking devices that are located on the edge of a service provider network.

RADIUS—Remote Authentication Dial-In User Service. RADIUS is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

VPN—Virtual Private Network. A system that permits dial-in networks to exist remotely to home networks, while giving the appearance of being directly connected. VPNs use L2TP and L2F to terminate the Layer 2 and higher parts of the network connection at the LNS instead of the LAC.

VRF—Virtual Route Forwarding. Initially, a router has only one global default routing/forwarding table. VRFs can be viewed as multiple disjoined routing/forwarding tables, where the routes of a user have no correlation with the routes of another user.

TACACS+

Configuring TACACS+

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter discusses how to enable and configure TACACS+, which provides detailed accounting information and flexible administrative control over authentication and authorization processes. TACACS+ is facilitated through AAA and can be enabled only through AAA commands.

For a complete description of the TACACS+ commands used in this chapter, refer to the chapter "TACACS+ Commands" in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature, or refer to the software release notes for a specific release. For more information, see the section "Identifying Supported Platforms" in the chapter "Using Cisco IOS Software."

In This Chapter

This chapter includes the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

About TACACS+

TACACS+ is a security application that provides centralized validation of users attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your network access server are available.

TACACS+ provides for separate and modular authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service—authentication, authorization, and accounting—independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The goal of TACACS+ is to provide a methodology for managing multiple network access points from a single management service. The Cisco family of access servers and routers and the Cisco IOS user interface (for both routers and access servers) can be network access servers.

Network access points enable traditional “dumb” terminals, terminal emulators, workstations, personal computers (PCs), and routers in conjunction with suitable adapters (for example, modems or ISDN adapters) to communicate using protocols such as Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), Compressed SLIP (CSLIP), or AppleTalk Remote Access (ARA) protocol. In other words, a network access server provides connections to a single user, to a network or subnetwork, and to interconnected networks. The entities connected to the network through a network access server are called *network access clients*; for example, a PC running PPP over a voice-grade circuit is a network access client. TACACS+, administered through the AAA security services, can provide the following services:

- Authentication—Provides complete control of authentication through login and password dialog, challenge and response, messaging support.

  The authentication facility provides the ability to conduct an arbitrary dialog with the user (for example, after a login and password are provided, to challenge a user with a number of questions, like home address, mother’s maiden name, service type, and social security number). In addition, the TACACS+ authentication service supports sending messages to user screens. For example, a message could notify users that their passwords must be changed because of the company’s password aging policy.

- Authorization—Provides fine-grained control over user capabilities for the duration of the user’s session, including but not limited to setting autocommands, access control, session duration, or protocol support. You can also enforce restrictions on what commands a user may execute with the TACACS+ authorization feature.

- Accounting—Collects and sends information used for billing, auditing, and reporting to the TACACS+ daemon. Network managers can use the accounting facility to track user activity for a security audit or to provide information for user billing. Accounting records include user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

The TACACS+ protocol provides authentication between the network access server and the TACACS+ daemon, and it ensures confidentiality because all protocol exchanges between a network access server and a TACACS+ daemon are encrypted.

You need a system running TACACS+ daemon software to use the TACACS+ functionality on your network access server.

Cisco makes the TACACS+ protocol specification available as a draft RFC for those customers interested in developing their own TACACS+ software.

TACACS+ Operation

When a user attempts a simple ASCII login by authenticating to a network access server using TACACS+, the following process typically occurs:

1. When the connection is established, the network access server will contact the TACACS+ daemon to obtain a username prompt, which is then displayed to the user. The user enters a username and the network access server then contacts the TACACS+ daemon to obtain a password prompt. The network access server displays the password prompt to the user, the user enters a password, and the password is then sent to the TACACS+ daemon.

Note TACACS+ allows an arbitrary conversation to be held between the daemon and the user until the daemon receives enough information to authenticate the user. This is usually done by prompting for a username and password combination, but may include other items, such as mother's maiden name, all under the control of the TACACS+ daemon.

2. The network access server will eventually receive one of the following responses from the TACACS+ daemon:
   a. ACCEPT—The user is authenticated and service may begin. If the network access server is configured to requite authorization, authorization will begin at this time.
   b. REJECT—The user has failed to authenticate. The user may be denied further access, or will be prompted to retry the login sequence depending on the TACACS+ daemon.
   c. ERROR—An error occurred at some time during authentication. This can be either at the daemon or in the network connection between the daemon and the network access server. If an ERROR response is received, the network access server will typically try to use an alternative method for authenticating the user.
   d. CONTINUE—The user is prompted for additional authentication information.
3. A PAP login is similar to an ASCII login, except that the username and password arrive at the network access server in a PAP protocol packet instead of being typed in by the user, so the user is not prompted. PPP CHAP logins are also similar in principle.

Following authentication, the user will also be required to undergo an additional authorization phase, if authorization has been enabled on the network access server. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

4. If TACACS+ authorization is required, the TACACS+ daemon is again contacted and it returns an ACCEPT or REJECT authorization response. If an ACCEPT response is returned, the response will contain data in the form of attributes that are used to direct the EXEC or NETWORK session for that user, determining services that the user can access.

   Services include the following:
   a. Telnet, rlogin, Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services
   b. Connection parameters, including the host or client IP address, access list, and user timeouts

TACACS+ Configuration Task List

To configure your router to support TACACS+, you must perform the following tasks:

- Use the **aaa new-model** global configuration command to enable AAA. AAA must be configured if you plan to use TACACS+. For more information about using the **aaa new-model** command, refer to the chapter “AAA Overview”.
- Use the **tacacs-server host** command to specify the IP address of one or more TACACS+ daemons. Use the **tacacs-server key** command to specify an encryption key that will be used to encrypt all exchanges between the network access server and the TACACS+ daemon. This same key must also be configured on the TACACS+ daemon.
- Use the **aaa authentication** global configuration command to define method lists that use TACACS+ for authentication. For more information about using the **aaa authentication** command, refer to the chapter “Configuring Authentication”.
- Use **line** and **interface** commands to apply the defined method lists to various interfaces. For more information, refer to the chapter “Configuring Authentication”.
- If needed, use the **aaa authorization** global command to configure authorization for the network access server. Unlike authentication, which can be configured per line or per interface, authorization is configured globally for the entire network access server. For more information about using the **aaa authorization** command, refer to the “Configuring Authorization” chapter.
- If needed, use the **aaa accounting** command to enable accounting for TACACS+ connections. For more information about using the **aaa accounting** command, refer to the “Configuring Accounting” chapter.

To configure TACACS+, perform the tasks in the following sections:

- Identifying the TACACS+ Server Host (Required)
- Setting the TACACS+ Authentication Key (Optional)
- Configuring AAA Server Groups (Optional)
- Configuring AAA Server Group Selection Based on DNIS (Optional)
- Specifying TACACS+ Authentication (Required)
- Specifying TACACS+ Authorization (Optional)
- Specifying TACACS+ Accounting (Optional)

For TACACS+ configuration examples using the commands in this chapter, refer to the “TACACS+ Configuration Examples” section at the end of the this chapter.

Identifying the TACACS+ Server Host

The **tacacs-server host** command enables you to specify the names of the IP host or hosts maintaining a TACACS+ server. Because the TACACS+ software searches for the hosts in the order specified, this feature can be useful for setting up a list of preferred daemons.

To specify a TACACS+ host, use the following command in global configuration mode:

Command	Purpose
`Router(config)# tacacs-server host` *hostname* `[single-connection] [port` *integer*`] [timeout` *integer*`] [key` *string*`]`	Specifies a TACACS+ host.

Using the **tacacs-server host** command, you can also configure the following options:

- Use the **single-connection** keyword to specify single-connection (only valid with CiscoSecure Release 1.0.1 or later). Rather than have the router open and close a TCP connection to the daemon each time it must communicate, the single-connection option maintains a single open connection between the router and the daemon. This is more efficient because it allows the daemon to handle a higher number of TACACS operations.

  Note The daemon must support single-connection mode for this to be effective, otherwise the connection between the network access server and the daemon will lock up or you will receive spurious errors.

- Use the **port** *integer* argument to specify the TCP port number to be used when making connections to the TACACS+ daemon. The default port number is 49.
- Use the **timeout** *integer* argument to specify the period of time (in seconds) the router will wait for a response from the daemon before it times out and declares an error.

  Note Specifying the timeout value with the **tacacs-server host** command overrides the default timeout value set with the **tacacs-server timeout** command for this server only.

- Use the **key** *string* argument to specify an encryption key for encrypting and decrypting all traffic between the network access server and the TACACS+ daemon.

  Note Specifying the encryption key with the **tacacs-server host** command overrides the default key set by the global configuration **tacacs-server key** command for this server only.

Because some of the parameters of the **tacacs-server host** command override global settings made by the **tacacs-server timeout** and **tacacs-server key** commands, you can use this command to enhance security on your network by uniquely configuring individual TACACS+ connections.

Setting the TACACS+ Authentication Key

To set the global TACACS+ authentication key and encryption key, use the following command in global configuration mode:

Command	Purpose
`Router(config)# tacacs-server key` *key*	Sets the encryption key to match that used on the TACACS+ daemon.

Note You must configure the same key on the TACACS+ daemon for encryption to be successful.

Configuring AAA Server Groups

Configuring the router to use AAA server groups provides a way to group existing server hosts. This allows you to select a subset of the configured server hosts and use them for a particular service. A server group is used in conjunction with a global server-host list. The server group lists the IP addresses of the selected server hosts.

Server groups can include multiple host entries as long as each entry has a unique IP address. If two different host entries in the server group are configured for the same service—for example, accounting—the second host entry configured acts as fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry for accounting services. (The TACACS+ host entries will be tried in the order in which they are configured.)

To define a server host with a server group name, enter the following commands starting in global configuration mode. The listed server must exist in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# tacacs-server host name [single-connection] [port integer] [timeout integer] [key string]`	Specifies and defines the IP address of the server host before configuring the AAA server-group. Refer to the "Identifying the TACACS+ Server Host" section of this chapter for more information on the **tacacs-server host** command.
Step 2	`Router(config-if)# aaa group server {radius | tacacs+} group-name`	Defines the AAA server-group with a group name. All members of a group must be the same type; that is, RADIUS or TACACS+. This command puts the router in server group subconfiguration mode.
Step 3	`Router(config-sg)# server ip-address [auth-port port-number] [acct-port port-number]`	Associates a particular TACACS+ server with the defined server group. Use the **auth-port** *port-number* option to configure a specific UDP port solely for authentication. Use the **acct-port** *port-number* option to configure a specific UDP port solely for accounting. Repeat this step for each TACACS+ server in the AAA server group. **Note** Each server in the group must be defined previously using the **tacacs-server host** command.

Configuring AAA Server Group Selection Based on DNIS

Cisco IOS software allows you to authenticate users to a particular AAA server group based on the Dialed Number Identification Service (DNIS) number of the session. Any phone line (a regular home phone or a commercial T1/PRI line) can be associated with several phone numbers. The DNIS number identifies the number that was called to reach you.

For example, suppose you want to share the same phone number with several customers, but you want to know which customer is calling before you pick up the phone. You can customize how you answer the phone because DNIS allows you to know which customer is calling when you answer.

Cisco routers with either ISDN or internal modems can receive the DNIS number. This functionality allows users to assign different TACACS+ server groups for different customers (that is, different TACACS+ servers for different DNIS numbers). Additionally, using server groups you can specify the same server group for AAA services or a separate server group for each AAA service.

Cisco IOS software provides the flexibility to implement authentication and accounting services in several ways:

- Globally—AAA services are defined using global configuration access list commands and applied in general to all interfaces on a specific network access server.
- Per interface—AAA services are defined using interface configuration commands and applied specifically to the interface being configured on a specific network access server.
- DNIS mapping—You can use DNIS to specify an AAA server to supply AAA services.

Because AAA configuration methods can be configured simultaneously, Cisco has established an order of precedence to determine which server or groups of servers provide AAA services. The order of precedence is as follows:

- Per DNIS—If you configure the network access server to use DNIS to identify which server group provides AAA services, then this method takes precedence over any additional AAA selection method.
- Per interface—If you configure the network access server per interface to use access lists to determine how a server provides AAA services, this method takes precedence over any global configuration AAA access lists.
- Globally—If you configure the network access server by using global AAA access lists to determine how the security server provides AAA services, this method has the lowest precedence.

Note Prior to configuring AAA Server Group Selection Based on DNIS, you must configure the remote security servers associated with each AAA server group. See the sections "Identifying the TACACS+ Server Host" and "Configuring AAA Server Groups" in this chapter.

To configure the router to select a particular AAA server group based on the DNIS of the server group, configure DNIS mapping. To map a server group with a group name with DNIS number, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa dnis map enable`	Enables DNIS mapping.
Step 2	`Router(config)# aaa dnis map` *dnis-number* `authentication ppp group` *server-group-name*	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for authentication.
Step 3	`Router(config)# aaa dnis map` *dnis-number* `accounting network [none | start-stop | stop-only] group` *server-group-name*	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for accounting.

Specifying TACACS+ Authentication

After you have identified the TACACS+ daemon and defined an associated TACACS+ encryption key, you must define method lists for TACACS+ authentication. Because TACACS+ authentication is operated via AAA, you need to issue the **aaa authentication** command, specifying TACACS+ as the authentication method. For more information, refer to the chapter "Configuring Authentication."

Specifying TACACS+ Authorization

AAA authorization enables you to set parameters that restrict a user's access to the network. Authorization via TACACS+ may be applied to commands, network connections, and EXEC sessions. Because TACACS+ authorization is facilitated through AAA, you must issue the **aaa authorization** command, specifying TACACS+ as the authorization method. For more information, refer to the chapter "Configuring Authorization."

Specifying TACACS+ Accounting

AAA accounting enables you to track the services users are accessing as well as the amount of network resources they are consuming. Because TACACS+ accounting is facilitated through AAA, you must issue the **aaa accounting** command, specifying TACACS+ as the accounting method. For more information, refer to the chapter "Configuring Accounting."

TACACS+ AV Pairs

The network access server implements TACACS+ authorization and accounting functions by transmitting and receiving TACACS+ attribute-value (AV) pairs for each user session. For a list of supported TACACS+ AV pairs, refer to the appendix "TACACS+ Attribute-Value Pairs."

TACACS+ Configuration Examples

The following sections provide TACACS+ configuration examples:

- TACACS+ Authentication Examples
- TACACS+ Authorization Example
- TACACS+ Accounting Example
- TACACS+ Server Group Example
- AAA Server Group Selection Based on DNIS Example
- TACACS+ Daemon Configuration Example

TACACS+ Authentication Examples

The following example shows how to configure TACACS+ as the security protocol for PPP authentication:

```
aaa new-model
```

```
aaa authentication ppp test group tacacs+ local
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication chap pap test
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, “test,” to be used on serial interfaces running PPP. The keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be “goaway.”
- The **interface** command selects the line, and the **ppp authentication** command applies the test method list to this line.

The following example shows how to configure TACACS+ as the security protocol for PPP authentication, but instead of the “test” method list, the “default” method list is used.

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication chap default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, “default,” to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be “goaway.”
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

The following example shows how to create the same authentication algorithm for PAP, but it calls the method list “MIS-access” instead of “default”:

```
aaa new-model
aaa authentication pap MIS-access if-needed group tacacs+ local
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication pap MIS-access
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.

- The **aaa authentication** command defines a method list, “MIS-access,” to be used on serial interfaces running PPP. The method list, “MIS-access,” means that PPP authentication is applied to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be “goaway.”
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

The following example shows the configuration for a TACACS+ daemon with an IP address of 10.2.3.4 and an encryption key of “apple”:

```
aaa new-model
aaa authentication login default group tacacs+ local
tacacs-server host 10.2.3.4
tacacs-server key apple
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines the default method list. Incoming ASCII logins on all interfaces (by default) will use TACACS+ for authentication. If no TACACS+ server responds, then the network access server will use the information contained in the local username database for authentication.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.2.3.4. The **tacacs-server key** command defines the shared encryption key to be “apple.”

TACACS+ Authorization Example

The following example shows how to configure TACACS+ as the security protocol for PPP authentication using the default method list; it also shows how to configure network authorization via TACACS+:

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
aaa authorization network default group tacacs+
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication chap default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, “default,” to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done

through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.

- The **aaa authorization** command configures network authorization via TACACS+. Unlike authentication lists, this authorization list always applies to all incoming network connections made to the network access server.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be “goaway.”
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

TACACS+ Accounting Example

The following example shows how to configure TACACS+ as the security protocol for PPP authentication using the default method list; it also shows how to configure accounting via TACACS+:

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
aaa accounting network default stop-only group tacacs+
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication chap default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, “default,” to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **aaa accounting** command configures network accounting via TACACS+. In this example, accounting records describing the session that just terminated will be sent to the TACACS+ daemon whenever a network connection terminates.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be “goaway.”
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

TACACS+ Server Group Example

The following example shows how to create a server group with three different TACACS+ servers members:

```
aaa group server tacacs tacgroup1
    server 172.16.1.1
    server 172.16.1.21
```

```
server 172.16.1.31
```

AAA Server Group Selection Based on DNIS Example

The following example shows how to select TACAC+ server groups based on DNIS to provide specific AAA services:

```
! This command enables AAA.
aaa new-model
!
! The following set of commands configures the TACACS+ servers that will be associated
! with one of the defined server groups.
tacacs-server host 172.16.0.1
tacacs-server host 172.17.0.1
tacacs-server host 172.18.0.1
tacacs-server host 172.19.0.1
tacacs-server host 172.20.0.1
tacacs-server key abcdefg

! The following commands define the sg1 TACACS+ server group and associate servers
! with it.
aaa group server tacacs sg1
  server 172.16.0.1
  server 172.17.0.1
! The following commands define the sg2 TACACS+ server group and associate a server
! with it.
aaa group server tacacs sg2
  server 172.18.0.1
! The following commands define the sg3 TACACS+ server group and associate a server
! with it.
aaa group server tacacs sg3
  server 172.19.0.1
! The following commands define the default-group TACACS+ server group and associate
! a server with it.
aaa group server tacacs default-group
  server 172.20.0.1
!
! The next set of commands configures default-group tacacs server group parameters.
aaa authentication ppp default group default-group
aaa accounting network default start-stop group default-group
!
! The next set of commands enables DNIS mapping and maps DNIS numbers to the defined
! RADIUS server groups. In this configuration, all PPP connection requests using DNIS
! 7777 are sent to the sg1 server group. The accounting records for these connections
! (specifically, start-stop records) are handled by the sg2 server group. Calls with a
! DNIS of 8888 use server group sg3 for authentication and server group default-group
! for accounting. Calls with a DNIS of 9999 use server group default-group for
! authentication and server group sg3 for accounting records (stop records only). All
! other calls with DNIS other than the ones defined use the server group default-group
! for both authentication and stop-start accounting records.
aaa dnis map enable
aaa dnis map 7777 authentication ppp group sg1
aaa dnis map 7777 accounting network start-stop group sg2
aaa dnis map 8888 authentication ppp group sg3
aaa dnis map 9999 accounting network stop-only group sg3
```

TACACS+ Daemon Configuration Example

The following example shows a sample configuration of the TACACS+ daemon. The precise syntax used by your TACACS+ daemon may be different from what is included in this example.

```
user = mci_customer1 {
 chap = cleartext “some chap password”
 service = ppp protocol = ip {
inacl#1=”permit ip any any precedence immediate”
inacl#2=”deny igrp 0.0.1.2 255.255.0.0 any”
 }
}
```

Per VRF for TACACS+ Servers

First Published: March 1, 2004
Last Updated: November 17, 2008

The Per VRF for TACACS+ Servers feature allows per virtual route forwarding (per VRF) to be configured for authentication, authorization, and accounting (AAA) on TACACS+ servers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Per VRF for TACACS+ Servers" section on page 8.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Per VRF for TACACS+ Servers

- TACACS+ server access is required.
- Experience configuring TACACS+, AAA and per VRF AAA, and group servers is necessary.

Restrictions for Per VRF for TACACS+ Servers

- The VRF instance must be specified before per VRF for a TACACS+ server is configured.

Information About Per VRF for TACACS+ Servers

To configure the Per VRF for TACACS+ Servers feature, the following concept should be understood:

- Per VRF for TACACS+ Servers Overview, page 2

Per VRF for TACACS+ Servers Overview

The Per VRF for TACACS+ Servers feature allows per VRF AAA to be configured on TACACS+ servers. Prior to Cisco IOS Release 12.3(7)T, this functionality was available only on RADIUS servers.

How to Configure Per VRF for TACACS+ Servers

This section contains the following procedures:

- Configuring Per VRF on a TACACS+ Server, page 2 (required)
- Verifying Per VRF for TACACS+ Servers, page 4 (optional)

Configuring Per VRF on a TACACS+ Server

The initial steps in this procedure are used to configure AAA and a server group, create a VRF routing table, and configure an interface. Steps 10 through 13 are used to configure the per VRF on a TACACS+ server feature:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **exit**
6. **interface** *interface-name*
7. **ip vrf forwarding** *vrf-name*

8. **ip address** *ip-address mask* [**secondary**]
9. **exit**
10. **aaa group server tacacs+** *group-name*
11. **server-private** {*ip-address* | *name*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*]
12. **ip vrf forwarding** *vrf-name*
13. **ip tacacs source-interface** *subinterface-name*
14. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router (config)# ip vrf cisco`	Configures a VRF table and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Router (config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF instance.
Step 5	**exit** **Example:** `Router (config-vrf)# exit`	Exits VRF configuration mode.
Step 6	**interface** *interface-name* **Example:** `Router (config)# interface Loopback0`	Configures an interface and enters interface configuration mode.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Router (config-if)# ip vrf forwarding cisco`	Configures a VRF for the interface.
Step 8	**ip address** *ip-address mask* [**secondary**] **Example:** `Router (config-if)# ip address 10.0.0.2 255.0.0.0`	Sets a primary or secondary IP address for an interface.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 10	**aaa group server tacacs+** *group-name* **Example:** `Router (config)# aaa group server tacacs+ tacacs1`	Groups different TACACS+ server hosts into distinct lists and distinct methods and enters server-group configuration mode.
Step 11	**server-private** {*ip-address* | *name*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*] **Example:** `Router (config-sg-tacacs+)# server-private 10.1.1.1 port 19 key cisco`	Configures the IP address of the private TACACS+ server for the group server.
Step 12	**ip vrf forwarding** *vrf-name* **Example:** `Router (config-sg-tacacs+)# ip vrf forwarding cisco`	Configures the VRF reference of a AAA TACACS+ server group.
Step 13	**ip tacacs source-interface** *subinterface-name* **Example:** `Router (config-sg-tacacs+)# ip tacacs source-interface Loopback0`	Uses the IP address of a specified interface for all outgoing TACACS+ packets.
Step 14	**exit** **Example:** `Router (config-sg-tacacs)# exit`	Exits server-group configuration mode.

Verifying Per VRF for TACACS+ Servers

To verify the per VRF TACACS+ configuration, perform the following steps:

Note The **debug** commands may be used in any order.

SUMMARY STEPS

1. **enable**
2. **debug tacacs authentication**
3. **debug tacacs authorization**
4. **debug tacacs accounting**
5. **debug tacacs packets**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug tacacs authentication** **Example:** `Router# debug tacacs authentication`	Displays information about AAA/TACACS+ authentication.
Step 3	**debug tacacs authorization** **Example:** `Router# debug tacacs authorization`	Displays information about AAA/TACACS+ authorization.
Step 4	**debug tacacs accounting** **Example:** `Router# debug tacacs accounting`	Displays information about accountable events as they occur.
Step 5	**debug tacacs packets** **Example:** `Router# debug tacacs packets`	Displays information about TACACS+ packets.

Configuration Examples for Per VRF for TACACS+ Servers

This section includes the following configuration example:

• Configuring Per VRF for TACACS+ Servers: Example, page 5

Configuring Per VRF for TACACS+ Servers: Example

The following output example shows that the group server **tacacs1** is configured for per VRF AAA services:

```
aaa group server tacacs+ tacacs1
    server-private 10.1.1.1 port 19 key cisco
    ip vrf forwarding cisco
    ip tacacs source-interface Loopback0

  ip vrf cisco
   rd 100:1

  interface Loopback0
   ip address 10.0.0.2 255.0.0.0
   ip vrf forwarding cisco
```

Additional References

The following sections provide references related to Per VRF for TACACS+ Servers.

Related Documents

Related Topic	Document Title
Configuring TACACS+	"Configuring TACACS+" chapter of the "Security Server Protocols" section of the *Cisco IOS Security Configuration Guide*
Per VRF AAA	*Per VRF AAA*
Cisco IOS commands	*Cisco Master Commands* list, Release 12.4T
Security commands	*Cisco IOS Security Command Reference*, Release 12.4T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **ip tacacs source-interface**
- **ip vrf forwarding (server-group)**
- **server-private (TACACS+)**

Feature Information for Per VRF for TACACS+ Servers

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Per VRF for TACACS+ Servers

Feature Name	Releases	Feature Information
Per VRF for TACACS+ Servers	12.3(7)T 12.3(11)T 12.2(33)SXI 12.2(33)SXH4	The Per VRF for TACACS+ Servers feature allows per virtual route forwarding (per VRF) to be configured for authentication, authorization, and accounting (AAA) on TACACS+ servers. This feature was introduced in Cisco IOS Release 12.3(7)T. This feature was integrated into Cisco IOS Release 12.2(33)SRA1. This feature was integrated into Cisco IOS Release 12.2(33)SXI. This feature was integrated into Cisco IOS Release 12.2(33)SXH4.

Configuring Kerberos

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes the Kerberos security system. For a complete description of the Kerberos commands used in this chapter, refer to the "Kerberos Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature, or refer to the software release notes for a specific release. For more information, see the section "Identifying Supported Platforms" in the chapter "Using Cisco IOS Software."

In This Chapter

This chapter includes the following topics and tasks:

About Kerberos

Kerberos is a secret-key network authentication protocol, developed at the Massachusetts Institute of Technology (MIT), that uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication. Kerberos was designed to authenticate requests for network resources.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Kerberos, like other secret-key systems, is based on the concept of a trusted third party that performs secure verification of users and services. In the Kerberos protocol, this trusted third party is called the key distribution center (KDC).

The primary use of Kerberos is to verify that users and the network services they use are really who and what they claim to be. To accomplish this, a trusted Kerberos server issues tickets to users. These tickets, which have a limited lifespan, are stored in a user's credential cache and can be used in place of the standard username-and-password authentication mechanism.

The Kerberos credential scheme embodies a concept called "single logon." This process requires authenticating a user once, and then allows secure authentication (without encrypting another password) wherever that user's credential is accepted.

Starting with Cisco IOS Release 11.2, Cisco IOS software includes Kerberos 5 support, which allows organizations already deploying Kerberos 5 to use the same Kerberos authentication database on their routers that they are already using on their other network hosts (such as UNIX servers and PCs).

The following network services are supported by the Kerberos authentication capabilities in Cisco IOS software:

- Telnet
- rlogin
- rsh
- rcp

Note Cisco Systems' implementation of Kerberos client support is based on code developed by CyberSafe, which was derived from the MIT code. As a result, the Cisco Kerberos implementation has successfully undergone full compatibility testing with the CyberSafe Challenger commercial Kerberos server and MIT's server code, which is freely distributed.

Table 20 lists common Kerberos-related terms and their definitions.

Table 20 Kerberos Terminology

Term	Definition
authentication	A process by which a user or service identifies itself to another service. For example, a client can authenticate to a router or a router can authenticate to another router.
authorization	A means by which the router determines what privileges you have in a network or on the router and what actions you can perform.
credential	A general term that refers to authentication tickets, such as ticket granting tickets (TGTs) and service credentials. Kerberos credentials verify the identity of a user or service. If a network service decides to trust the Kerberos server that issued a ticket, it can be used in place of retyping in a username and password. Credentials have a default lifespan of eight hours.

Table 20 Kerberos Terminology (continued)

Term	Definition
instance	An authorization level label for Kerberos principals. Most Kerberos principals are of the form user@REALM (for example, smith@EXAMPLE.COM). A Kerberos principal with a Kerberos instance has the form user/instance@REALM (for example, smith/admin@EXAMPLE.COM). The Kerberos instance can be used to specify the authorization level for the user if authentication is successful. It is up to the server of each network service to implement and enforce the authorization mappings of Kerberos instances. Note that the Kerberos realm name must be in uppercase characters.
Kerberized	Applications and services that have been modified to support the Kerberos credential infrastructure.
Kerberos realm	A domain consisting of users, hosts, and network services that are registered to a Kerberos server. The Kerberos server is trusted to verify the identity of a user or network service to another user or network service. Kerberos realms must always be in uppercase characters.
Kerberos server	A daemon running on a network host. Users and network services register their identity with the Kerberos server. Network services query the Kerberos server to authenticate to other network services.
key distribution center (KDC)	A Kerberos server and database program running on a network host.
principal	Also known as a Kerberos identity, this is who you are or what a service is according to the Kerberos server.
service credential	A credential for a network service. When issued from the KDC, this credential is encrypted with the password shared by the network service and the KDC, and with the user's TGT.
SRVTAB	A password that a network service shares with the KDC. The network service authenticates an encrypted service credential by using the SRVTAB (also known as a KEYTAB) to decrypt it.
ticket granting ticket (TGT)	A credential that the key distribution center (KDC) issues to authenticated users. When users receive a TGT, they can authenticate to network services within the Kerberos realm represented by the KDC.

Kerberos Client Support Operation

This section describes how the Kerberos security system works with a Cisco router functioning as the security server. Although (for convenience or technical reasons) you can customize Kerberos in a number of ways, remote users attempting to access network services must pass through three layers of security before they can access network services.

This section includes the following sections:

- Authenticating to the Boundary Router
- Obtaining a TGT from a KDC
- Authenticating to Network Services

Authenticating to the Boundary Router

This section describes the first layer of security that remote users must pass through when they attempt to access a network. The first step in the Kerberos authentication process is for users to authenticate themselves to the boundary router. The following process describes how users authenticate to a boundary router:

1. The remote user opens a PPP connection to the corporate site router.
2. The router prompts the user for a username and password.
3. The router requests a TGT from the KDC for this particular user.
4. The KDC sends an encrypted TGT to the router that includes (among other things) the user's identity.
5. The router attempts to decrypt the TGT using the password the user entered. If the decryption is successful, the remote user is authenticated to the router.

A remote user who successfully initiates a PPP session and authenticates to the boundary router is inside the firewall but still must authenticate to the KDC directly before being allowed to access network services. This is because the TGT issued by the KDC is stored on the router and is not useful for additional authentication unless the user physically logs on to the router.

Obtaining a TGT from a KDC

This section describes how remote users who are authenticated to the boundary router authenticate themselves to a KDC.

When a remote user authenticates to a boundary router, that user technically becomes part of the network; that is, the network is extended to include the remote user and the user's machine or network. To gain access to network services, however, the remote user must obtain a TGT from the KDC. The following process describes how remote users authenticate to the KDC:

1. The remote user, at a workstation on a remote site, launches the KINIT program (part of the client software provided with the Kerberos protocol).
2. The KINIT program finds the user's identity and requests a TGT from the KDC.
3. The KDC creates a TGT, which contains the identity of the user, the identity of the KDC, and the expiration time of the TGT.
4. Using the user's password as a key, the KDC encrypts the TGT and sends the TGT to the workstation.
5. When the KINIT program receives the encrypted TGT, it prompts the user for a password (this is the password that is defined for the user in the KDC).
6. If the KINIT program can decrypt the TGT with the password the user enters, the user is authenticated to the KDC, and the KINIT program stores the TGT in the user's credential cache.

At this point, the user has a TGT and can communicate securely with the KDC. In turn, the TGT allows the user to authenticate to other network services.

Authenticating to Network Services

The following process describes how a remote user with a TGT authenticates to network services within a given Kerberos realm. Assume the user is on a remote workstation (Host A) and wants to log in to Host B.

1. The user on Host A initiates a Kerberized application (such as Telnet) to Host B.
2. The Kerberized application builds a service credential request and sends it to the KDC. The service credential request includes (among other things) the user's identity and the identity of the desired network service. The TGT is used to encrypt the service credential request.
3. The KDC tries to decrypt the service credential request with the TGT it issued to the user on Host A. If the KDC can decrypt the packet, it is assured that the authenticated user on Host A sent the request.
4. The KDC notes the network service identity in the service credential request.
5. The KDC builds a service credential for the appropriate network service on Host B on behalf of the user on Host A. The service credential contains the client's identity and the desired network service's identity.
6. The KDC then encrypts the service credential twice. It first encrypts the credential with the SRVTAB that it shares with the network service identified in the credential. It then encrypts the resulting packet with the TGT of the user (who, in this case, is on Host A).
7. The KDC sends the twice-encrypted credential to Host A.
8. Host A attempts to decrypt the service credential with the user's TGT. If Host A can decrypt the service credential, it is assured the credential came from the real KDC.
9. Host A sends the service credential to the desired network service. Note that the credential is still encrypted with the SRVTAB shared by the KDC and the network service.
10. The network service attempts to decrypt the service credential using its SRVTAB.
11. If the network service can decrypt the credential, it is assured the credential was in fact issued from the KDC. Note that the network service trusts anything it can decrypt from the KDC, even if it receives it indirectly from a user. This is because the user first authenticated with the KDC.

At this point, the user is authenticated to the network service on Host B. This process is repeated each time a user wants to access a network service in the Kerberos realm.

Kerberos Configuration Task List

For hosts and the KDC in your Kerberos realm to communicate and mutually authenticate, you must identify them to each other. To do this, you add entries for the hosts to the Kerberos database on the KDC and add SRVTAB files generated by the KDC to all hosts in the Kerberos realm. You also make entries for users in the KDC database.

This section describes how to set up a Kerberos-authenticated server-client system and contains the following topics:

- Configuring the KDC Using Kerberos Commands
- Configuring the Router to Use the Kerberos Protocol

This section assumes that you have installed the Kerberos administrative programs on a UNIX host, known as the KDC, initialized the database, and selected a Kerberos realm name and password. For instructions about completing these tasks, refer to documentation that came with your Kerberos software.

Note Write down the host name or IP address of the KDC, the port number you want the KDC to monitor for queries, and the name of the Kerberos realm it will serve. You need this information to configure the router.

Configuring the KDC Using Kerberos Commands

After you set up a host to function as the KDC in your Kerberos realm, you must make entries to the KDC database for all principals in the realm. Principals can be network services on Cisco routers and hosts or they can be users.

To use Kerberos commands to add services to the KDC database (and to modify existing database information), complete the tasks in the following sections:

- Adding Users to the KDC Database
- Creating SRVTABs on the KDC
- Extracting SRVTABs

Note All Kerberos command examples are based on Kerberos 5 Beta 5 of the original MIT implementation. Later versions use a slightly different interface.

Adding Users to the KDC Database

To add users to the KDC and create privileged instances of those users, use the **su** command to become root on the host running the KDC and use the kdb5_edit program to use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# ank username@REALM`	Use the **ank** (add new key) command to add a user to the KDC. This command prompts for a password, which the user must enter to authenticate to the router.
Step 2	`Router# ank username/instance@REALM`	Use the **ank** command to add a privileged instance of a user.

For example, to add user *loki* of Kerberos realm CISCO.COM, enter the following Kerberos command:

```
ank loki@CISCO.COM
```

Note The Kerberos realm name must be in uppercase characters.

You might want to create privileged instances to allow network administrators to connect to the router at the enable level, for example, so that they need not enter a clear text password (and compromise security) to enter enable mode.

To add an instance of *loki* with additional privileges (in this case, *enable*, although it could be anything) enter the following Kerberos command:

```
ank loki/enable@CISCO.COM
```

In each of these examples, you are prompted to enter a password, which you must give to user *loki* to use at login.

The "Enabling Kerberos Instance Mapping" section describes how to map Kerberos instances to various Cisco IOS privilege levels.

Creating SRVTABs on the KDC

All routers that you want to authenticate to use the Kerberos protocol must have an SRVTAB. This section and the "Extracting SRVTABs" section describe how to create and extract SRVTABs for a router called *router1*. The section "Copying SRVTAB Files" describes how to copy SRVTAB files to the router.

To make SRVTAB entries on the KDC, use the following command in privileged EXEC mode:

Command	Purpose
`Router# ark SERVICE/HOSTNAME@REALM`	Use the **ark** (add random key) command to add a network service supported by a host or router to the KDC.

For example, to add a Kerberized authentication service for a Cisco router called *router1* to the Kerberos realm CISCO.COM, enter the following Kerberos command:

```
ark host/router1.cisco.com@CISCO.COM
```

Make entries for all network services on all Kerberized hosts that use this KDC for authentication.

Extracting SRVTABs

SRVTABs contain (among other things) the passwords or randomly generated keys for the service principals you entered into the KDC database. Service principal keys must be shared with the host running that service. To do this, you must save the SRVTAB entries to a file, then copy the file to the router and all hosts in the Kerberos realm. Saving SRVTAB entries to a file is called *extracting* SRVTABs. To extract SRVTABs, use the following command in privileged EXEC mode:

Command	Purpose
`Router# xst router-name host`	Use the kdb5_edit command **xst** to write an SRVTAB entry to a file.

For example, to write the host/router1.cisco.com@CISCO.COM SRVTAB to a file, enter the following Kerberos command:

```
xst router1.cisco.com@CISCO.COM host
```

Use the **quit** command to exit the kdb5_edit program.

Configuring the Router to Use the Kerberos Protocol

To configure a Cisco router to function as a network security server and authenticate users using the Kerberos protocol, complete the tasks in the following sections:

Defining a Kerberos Realm

For a router to authenticate a user defined in the Kerberos database, it must know the host name or IP address of the host running the KDC, the name of the Kerberos realm and, optionally, be able to map the host name or Domain Name System (DNS) domain to the Kerberos realm.

To configure the router to authenticate to a specified KDC in a specified Kerberos realm, use the following commands in global configuration mode. Note that DNS domain names must begin with a leading dot (.):

	Command	Purpose
Step 1	`Router(config)# kerberos local-realm kerberos-realm`	Defines the default realm for the router.
Step 2	`Router(config)# kerberos server kerberos-realm {hostname | ip-address} [port-number]`	Specifies to the router which KDC to use in a given Kerberos realm and, optionally, the port number that the KDC is monitoring. (The default is 88.)
Step 3	`Router(config)# kerberos realm {dns-domain | host} kerberos-realm`	(Optional) Maps a host name or DNS domain to a Kerberos realm.

Note Because the machine running the KDC and all Kerberized hosts must interact within a 5-minute window or authentication fails, all Kerberized machines, and especially the KDC, should be running the Network Time Protocol (NTP).

The **kerberos local-realm**, **kerberos realm**, and **kerberos server** commands are equivalent to the UNIX *krb.conf* file. Table 21 identifies mappings from the Cisco IOS configuration commands to a Kerberos 5 configuration file (krb5.conf).

Table 21 Kerberos 5 Configuration File and Commands

krb5.conf File	Cisco IOS Configuration Command
[libdefaults] `default_realm = DOMAIN.COM`	(in configuration mode) **`kerberos local-realm`** *`DOMAIN.COM`*
[domain_realm] `.domain.com = DOMAIN.COM` `domain.com = DOMAIN.COM`	(in configuration mode) **`kerberos realm`***`.domain.com DOMAIN.COM`* **`kerberos realm`** *`domain.com DOMAIN.COM`*
[realms] `kdc = DOMAIN.PIL.COM:750` `admin_server = DOMAIN.PIL.COM` `default_domain = DOMAIN.COM`	(in configuration mode) **`kerberos server`** *`DOMAIN.COM 172.65.44.2`* (*`172.65.44.2`* is the example IP address for *`DOMAIN.PIL.COM`*)

For an example of defining a Kerberos realm, see the section "Defining a Kerberos Realm" later in this chapter.

Copying SRVTAB Files

To make it possible for remote users to authenticate to the router using Kerberos credentials, the router must share a secret key with the KDC. To do this, you must give the router a copy of the SRVTAB you extracted on the KDC.

The most secure method to copy SRVTAB files to the hosts in your Kerberos realm is to copy them onto physical media and go to each host in turn and manually copy the files onto the system. To copy SRVTAB files to the router, which does not have a physical media drive, you must transfer them via the network using TFTP.

To remotely copy SRVTAB files to the router from the KDC, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`kerberos srvtab remote`** *`{hostname | ip-address} {filename}`*	Retrieves an SRVTAB file from the KDC.

When you copy the SRVTAB file from the router to the KDC, the **kerberos srvtab remote** command parses the information in this file and stores it in the router's running configuration in the **kerberos srvtab entry** format. To ensure that the SRVTAB is available (does not need to be acquired from the KDC) when you reboot the router, use the **write memory** configuration command to write your running configuration (which contains the parsed SRVTAB file) to NVRAM.

For an example of copying SRVTAB files, see the section "SRVTAB File Copying Example" later in this chapter.

Specifying Kerberos Authentication

You have now configured Kerberos on your router. This makes it possible for the router to authenticate using Kerberos. The next step is to tell it to do so. Because Kerberos authentication is facilitated through AAA, you need to enter the **aaa authentication** command, specifying Kerberos as the authentication method. For more information, refer to the chapter "Configuring Authentication".

Enabling Credentials Forwarding

With Kerberos configured thus far, a user authenticated to a Kerberized router has a TGT and can use it to authenticate to a host on the network. However, if the user tries to list credentials after authenticating to a host, the output will show no Kerberos credentials present.

You can optionally configure the router to forward users' TGTs with them as they authenticate from the router to Kerberized remote hosts on the network when using Kerberized Telnet, rcp, rsh, and rlogin (with the appropriate flags).

To force all clients to forward users' credentials as they connect to other hosts in the Kerberos realm, use the following command in global configuration mode:

Command	Purpose
`Router(config)# kerberos credentials forward`	Forces all clients to forward user credentials upon successful Kerberos authentication.

With credentials forwarding enabled, users' TGTs are automatically forwarded to the next host they authenticate to. In this way, users can connect to multiple hosts in the Kerberos realm without running the KINIT program each time to get a new TGT.

Opening a Telnet Session to the Router

To use Kerberos to authenticate users opening a Telnet session to the router from within the network, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authentication login {default | list-name} krb5_telnet`	Sets login authentication to use the Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router.

Although Telnet sessions to the router are authenticated, users must still enter a clear text password if they want to enter enable mode. The **kerberos instance map** command, discussed in a later section, allows them to authenticate to the router at a predefined privilege level.

Establishing an Encrypted Kerberized Telnet Session

Another way for users to open a secure Telnet session is to use Encrypted Kerberized Telnet. With Encrypted Kerberized Telnet, users are authenticated by their Kerberos credentials before a Telnet session is established. The Telnet session is encrypted using 56-bit Data Encryption Standard (DES) encryption with 64-bit Cipher Feedback (CFB). Because data sent or received is encrypted, not clear text, the integrity of the dialed router or access server can be more easily controlled.

Note This feature is available only if you have the 56-bit encryption image. 56-bit DES encryption is subject to U.S. Government export control regulations.

To establish an encrypted Kerberized Telnet session from a router to a remote host, use either of the following commands in EXEC command mode:

Command	Purpose
`Router(config)# connect host [port] /encrypt kerberos` or `Router(config)# telnet host [port] /encrypt kerberos`	Establishes an encrypted Telnet session.

When a user opens a Telnet session from a Cisco router to a remote host, the router and remote host negotiate to authenticate the user using Kerberos credentials. If this authentication is successful, the router and remote host then negotiate whether or not to use encryption. If this negotiation is successful, both inbound and outbound traffic is encrypted using 56-bit DES encryption with 64-bit CFB.

When a user dials in from a remote host to a Cisco router configured for Kerberos authentication, the host and router will attempt to negotiate whether or not to use encryption for the Telnet session. If this negotiation is successful, the router will encrypt all outbound data during the Telnet session.

If encryption is not successfully negotiated, the session will be terminated and the user will receive a message stating that the encrypted Telnet session was not successfully established.

For information about enabling bidirectional encryption from a remote host, refer to the documentation specific to the remote host device.

For an example of using encrypted Kerberized Telnet to open a secure Telnet session, see the section "Encrypted Telnet Session Example" later in this chapter.

Enabling Mandatory Kerberos Authentication

As an added layer of security, you can optionally configure the router so that, after remote users authenticate to it, these users can authenticate to other services on the network only with Kerberized Telnet, rlogin, rsh, and rcp. If you do not make Kerberos authentication mandatory and Kerberos authentication fails, the application attempts to authenticate users using the default method of authentication for that network service; for example, Telnet and rlogin prompt for a password, and rsh attempts to authenticate using the local rhost file.

To make Kerberos authentication mandatory, use the following command in global configuration mode:

Command	Purpose
`Router(config)# kerberos clients mandatory`	Sets Telnet, rlogin, rsh, and rcp to fail if they cannot negotiate the Kerberos protocol with the remote server.

Enabling Kerberos Instance Mapping

As mentioned in the section "Creating SRVTABs on the KDC," you can create administrative instances of users in the KDC database. The **kerberos instance map** command allows you to map those instances to Cisco IOS privilege levels so that users can open secure Telnet sessions to the router at a predefined privilege level, obviating the need to enter a clear text password to enter enable mode.

To map a Kerberos instance to a Cisco IOS privilege level, use the following command in global configuration mode:

Command	Purpose
`Router(config)# kerberos instance map instance privilege-level`	Maps a Kerberos instance to a Cisco IOS privilege level.

If there is a Kerberos instance for user *loki* in the KDC database (for example, *loki/admin*), user *loki* can now open a Telnet session to the router as loki/admin and authenticate automatically at privilege level 15, assuming instance "admin" is mapped to privilege level 15. (See the section "Adding Users to the KDC Database" earlier in this chapter.)

Cisco IOS commands can be set to various privilege levels using the **privilege level** command.

After you map a Kerberos instance to a Cisco IOS privilege level, you must configure the router to check for Kerberos instances each time a user logs in. To run authorization to determine if a user is allowed to run an EXEC shell based on a mapped Kerberos instance, use the **aaa authorization** command with the **krb5-instance** keyword. For more information, refer to the chapter "Configuring Authorization."

Monitoring and Maintaining Kerberos

To display or remove a current user's credentials, use the following commands in EXEC mode:

	Command	Purpose
Step 1	`Router# show kerberos creds`	Lists the credentials in a current user's credentials cache.
Step 2	`Router# clear kerberos creds`	Destroys all credentials in a current user's credentials cache, including those forwarded.

For an example of Kerberos configuration, see the section "Kerberos Configuration Examples".

Kerberos Configuration Examples

The following sections provide Kerberos configuration examples:

- Kerberos Realm Definition Examples
- SRVTAB File Copying Example
- Kerberos Configuration Examples
- Encrypted Telnet Session Example

Kerberos Realm Definition Examples

To define CISCO.COM as the default Kerberos realm, use the following command:

```
kerberos local-realm CISCO.COM
```

To tell the router that the CISCO.COM KDC is running on host 10.2.3.4 at port number 170, use the following Kerberos command:

```
kerberos server CISCO.COM 10.2.3.4 170
```

To map the DNS domain cisco.com to the Kerberos realm CISCO.COM, use the following command:

```
kerberos realm.cisco.com CISCO.COM
```

SRVTAB File Copying Example

To copy over the SRVTAB file on a host named host123.cisco.com for a router named router1.cisco.com, the command would look like this:

```
kerberos srvtab remote host123.cisco.com router1.cisco.com-new-srvtab
```

Kerberos Configuration Examples

This section provides a typical non-Kerberos router configuration and shows output for this configuration from the **write term** command, then builds on this configuration by adding optional Kerberos functionality. Output for each configuration is presented for comparison against the previous configuration.

This example shows how to use the kdb5_edit program to perform the following configuration tasks:

- Adding user chet to the Kerberos database
- Adding a privileged Kerberos instance of user chet (chet/admin) to the Kerberos database
- Adding a restricted instance of chet (chet/restricted) to the Kerberos database
- Adding workstation chet-ss20.cisco.com
- Adding router chet-2500.cisco.com to the Kerberos database
- Adding workstation chet-ss20.cisco.com to the Kerberos database
- Extracting SRVTABs for the router and workstations
- Listing the contents of the KDC database (with the **ldb** command)

Note that, in this sample configuration, host chet-ss20 is also the KDC:

```
chet-ss20# sbin/kdb5_edit
kdb5_edit:  ank chet
Enter password:
Re-enter password for verification:
kdb5_edit:  ank chet/admin
Enter password:
Re-enter password for verification:
kdb5_edit:  ank chet/restricted
Enter password:
Re-enter password for verification:
kdb5_edit:  ark host/chet-ss20.cisco.com
kdb5_edit:  ark host/chet-2500.cisco.com
kdb5_edit:  xst chet-ss20.cisco.com host
'host/chet-ss20.cisco.com@CISCO.COM' added to keytab
'WRFILE:chet-ss20.cisco.com-new-srvtab'
kdb5_edit:  xst chet-2500.cisco.com host
'host/chet-2500.cisco.com@CISCO.COM' added to keytab
'WRFILE:chet-2500.cisco.com-new-srvtab'
kdb5_edit:  ldb
entry: host/chet-2500.cisco.com@CISCO.COM
entry: chet/restricted@CISCO.COM
entry: chet@CISCO.COM
entry: K/M@CISCO.COM
entry: host/chet-ss20.cisco.com@CISCO.COM
entry: krbtgt/CISCO.COM@CISCO.COM
entry: chet/admin@CISCO.COM
```

```
kdb5_edit:  q
chet-ss20#
```

The following example shows output from a **write term** command, which displays the configuration of router chet-2500. This is a typical configuration with no Kerberos authentication.

```
chet-2500# write term
Building configuration...

Current configuration:
!
! Last configuration
change at 14:03:55 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
clock timezone PST -8
clock summer-time PDT recurring
aaa new-model
aaa authentication login console none
aaa authentication ppp local local
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
```

```
!
router eigrp 109
 network 172.17.0.0
 no auto-summary
!
ip default-gateway 172.30.55.64
ip domain-name cisco.com
ip name-server 192.168.0.0
ip classless
!
!

line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end
```

The following example shows how to enable user authentication on the router via the Kerberos database. To enable user authentication via the Kerberos database, you would perform the following tasks:

- Entering configuration mode
- Defining the Kerberos local realm
- Identifying the machine hosting the KDC
- Enabling credentials forwarding
- Specifying Kerberos as the method of authentication for login
- Exiting configuration mode (CTL-Z)
- Writing the new configuration to the terminal

```
chet-2500# configure term
Enter configuration commands, one per line. End with CNTL/Z.
chet-2500(config)# kerberos local-realm CISCO.COM
chet-2500(config)# kerberos server CISCO.COM chet-ss20
Translating "chet-ss20"...domain server (192.168.0.0) [OK]

chet-2500(config)# kerberos credentials forward
chet-2500(config)# aaa authentication login default krb5
chet-2500(config)#
chet-2500#
%SYS-5-CONFIG_I: Configured from console by console
chet-2500# write term
```

Compare the following configuration with the previous one. In particular, look at the lines beginning with the words “aaa,” “username,” and “kerberos” (lines 10 through 20) in this new configuration.

```
Building configuration...

Current configuration:
!
! Last configuration change at 14:05:54 PDT Mon May 13 1996
!
```

```
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
clock timezone PST -8
clock summer-time PDT recurring
aaa new-model
aaa authentication login default krb5
aaa authentication login console none
aaa authentication ppp local local
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
kerberos local-realm CISCO.COM
kerberos server CISCO.COM 172.71.54.14
kerberos credentials forward
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
router eigrp 109
 network 172.17.0.0
 no auto-summary
!
ip default-gateway 172.30.55.64
ip domain-name cisco.com
ip name-server 192.168.0.0
ip classless
!
```

```
!
line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end
```

With the router configured thus far, user chet can log in to the router with a username and password and automatically obtain a TGT, as illustrated in the next example. With possession of a credential, user chet successfully authenticates to host chet-ss20 without entering a username/password.

```
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.


User Access Verification

Username: chet
Password:

chet-2500> show kerberos creds
Default Principal:  chet@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 14:05:39    13-May-1996 22:06:40    krbtgt/CISCO.COM@CISCO.COM

chet-2500> telnet chet-ss20
Trying chet-ss20.cisco.com (172.71.54.14)... Open
Kerberos:       Successfully forwarded credentials


SunOS UNIX (chet-ss20) (pts/7)

Last login: Mon May 13 13:47:35 from chet-ss20.cisco.c
Sun Microsystems Inc.   SunOS 5.4       Generic July 1994
unknown mode: new
chet-ss20%
```

The following example shows how to authenticate to the router using Kerberos credentials. To authenticate using Kerberos credentials, you would perform the following tasks:

- Entering configuration mode
- Remotely copying over the SRVTAB file from the KDC
- Setting authentication at login to use the Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router
- Writing the configuration to the terminal

Note that the new configuration contains a **kerberos srvtab entry** line. This line is created by the **kerberos srvtab remote** command.

```
chet-2500# configure term
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
chet-2500(config)# kerberos srvtab remote earth chet/chet-2500.cisco.com-new-srvtab
Translating "earth"...domain server (192.168.0.0) [OK]

Loading chet/chet-2500.cisco.com-new-srvtab from 172.68.1.123 (via Ethernet0): !
[OK - 66/1000 bytes]

chet-2500(config)# aaa authentication login default krb5-telnet krb5
chet-2500(config)#
chet-2500#
%SYS-5-CONFIG_I: Configured from console by console
chet-2500# write term
Building configuration...

Current configuration:
!
! Last configuration change at 14:08:32 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
clock timezone PST -8
clock summer-time PDT recurring
aaa new-model
aaa authentication login default krb5-telnet krb5
aaa authentication login console none
aaa authentication ppp local local
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
kerberos local-realm CISCO.COM
kerberos srvtab entry host/chet-2500.cisco.com@CISCO.COM 0 832015393 1 1 8 7 sMudgkin
kerberos server CISCO.COM 172.71.54.14
kerberos credentials forward
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!

interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
```

```
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
router eigrp 109
 network 172.17.0.0
 no auto-summary
!
ip default-gateway 172.30.55.64
ip domain-name cisco.com
ip name-server 192.168.0.0
ip classless
!
!
line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end

chet-2500#
```

With this configuration, the user can Telnet in to the router using Kerberos credentials, as illustrated in the next example:

```
chet-ss20% bin/telnet -a -F chet-2500
Trying 172.16.0.0...
Connected to chet-2500.cisco.com.
Escape character is '^]'.
[ Kerberos V5 accepts you as "chet@CISCO.COM" ]

User Access Verification

chet-2500>[ Kerberos V5 accepted forwarded credentials ]

chet-2500> show kerberos creds
Default Principal:  chet@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 15:06:25    14-May-1996 00:08:29    krbtgt/CISCO.COM@CISCO.COM

chet-2500>q
Connection closed by foreign host.
chet-ss20%
```

The following example shows how to map Kerberos instances to Cisco's privilege levels. To map Kerberos instances to privilege levels, you would perform the following tasks:

- Entering configuration mode

- Mapping the Kerberos instance admin to privilege level 15
- Mapping the Kerberos instance restricted to privilege level 3
- Specifying that the instance defined by the **kerberos instance map** command be used for AAA Authorization
- Writing the configuration to the terminal

```
chet-2500# configure term
Enter configuration commands, one per line.  End with CNTL/Z.
chet-2500(config)# kerberos instance map admin 15
chet-2500(config)# kerberos instance map restricted 3
chet-2500(config)# aaa authorization exec default krb5-instance
chet-2500(config)#
chet-2500#
%SYS-5-CONFIG_I: Configured from console by console
chet-2500# write term
Building configuration...

Current configuration:
!
! Last configuration change at 14:59:05 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
aaa new-model
aaa authentication login default krb5-telnet krb5
aaa authentication login console none
aaa authentication ppp default krb5 local
aaa authorization exec default krb5-instance
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
ip domain-name cisco.com
ip name-server 192.168.0.0
kerberos local-realm CISCO.COM
kerberos srvtab entry host/chet-2500.cisco.com@CISCO.COM 0 832015393 1 1 8 7 sMudgkin
kerberos server CISCO.COM 172.71.54.14
kerberos instance map admin 15
kerberos instance map restricted 3
kerberos credentials forward
clock timezone PST -8
clock summer-time PDT recurring
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
```

```
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
router eigrp 109
 network 172.17.0.0
no auto-summary
!
ip default-gateway 172.30.55.64
ip classless
!
!
line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end

chet-2500#
```

The following example shows output from the three types of sessions now possible for user chet with Kerberos instances turned on:

```
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.

User Access Verification

Username: chet
Password:

chet-2500> show kerberos creds
Default Principal:  chet@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 14:58:28    13-May-1996 22:59:29    krbtgt/CISCO.COM@CISCO.COM

chet-2500> show privilege
```

```
Current privilege level is 1
chet-2500> q
Connection closed by foreign host.
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.


User Access Verification

Username: chet/admin
Password:

chet-2500# show kerberos creds
Default Principal:  chet/admin@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 14:59:44    13-May-1996 23:00:45    krbtgt/CISCO.COM@CISCO.COM
chet-2500# show privilege
Current privilege level is 15
chet-2500# q
Connection closed by foreign host.
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.

User Access Verification

Username: chet/restricted
Password:

chet-2500# show kerberos creds
Default Principal:  chet/restricted@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 15:00:32    13-May-1996 23:01:33    krbtgt/CISCO.COM@CISCO.COM

chet-2500# show privilege
Current privilege level is 3
chet-2500# q
Connection closed by foreign host.
chet-ss20%
```

Encrypted Telnet Session Example

The following example shows how to establish an encrypted Telnet session from a router to a remote host named “host1”:

```
Router> telnet host1 /encrypt kerberos
```

Traffic Filtering, Firewalls, and Virus Detection

Cisco IOS Firewall Overview

This chapter describes how you can configure your Cisco networking device to function as a firewall, using Cisco IOS Firewall security features.

This chapter has the following sections:

About Firewalls

Firewalls are networking devices that control access to your organization's network assets. Firewalls are positioned at the entrance points into your network. If your network has multiple entrance points, you must position a firewall at each point to provide effective network access control.

Firewalls are often placed in between the internal network and an external network such as the Internet. With a firewall between your network and the Internet, all traffic coming from the Internet must pass through the firewall before entering your network.

Firewalls can also be used to control access to a specific part of your network. For example, you can position firewalls at all the entry points into a research and development network to prevent unauthorized access to proprietary information.

The most basic function of a firewall is to monitor and filter traffic. Firewalls can be simple or elaborate, depending on your network requirements. Simple firewalls are usually easier to configure and manage. However, you might require the flexibility of a more elaborate firewall.

The Cisco IOS Firewall Solution

Cisco IOS software provides an extensive set of security features, allowing you to configure a simple or elaborate firewall, according to your particular requirements. You can configure a Cisco device as a firewall if the device is positioned appropriately at a network entry point. Security features that provide firewall functionality are listed in the "Creating a Customized Firewall" section.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

In addition to the security features available in standard Cisco IOS feature sets, Cisco IOS Firewall gives your router additional firewall capabilities.

The Cisco IOS Firewall Feature Set

The Cisco IOS Firewall feature set combines existing Cisco IOS firewall technology and Context-based Access Control (CBAC). When you configure the Cisco IOS Firewall on your Cisco router, you turn your router into an effective, robust firewall.

The Cisco IOS Firewall features are designed to prevent unauthorized external individuals from gaining access to your internal network and to block attacks on your network, while at the same time allowing authorized users to access network resources.

You can use the Cisco IOS Firewall features to configure your Cisco IOS router as one of the following:

- An Internet firewall or part of an Internet firewall
- A firewall between groups in your internal network
- A firewall providing secure connections to or from branch offices
- A firewall between your company's network and your company's partners' networks

The Cisco IOS Firewall features provide the following benefits:

- Protection of internal networks from intrusion
- Monitoring of traffic through network perimeters
- Enabling of network commerce via the World Wide Web

Creating a Customized Firewall

To create a firewall customized to fit your organization's security policy, you should determine which Cisco IOS Firewall features are appropriate, and configure those features. At a minimum, you must configure basic traffic filtering to provide a basic firewall. You can configure your Cisco networking device to function as a firewall by using the following Cisco IOS Firewall features:

- Standard Access Lists and Static Extended Access Lists
- Reflexive Access Lists
- Lock-and-Key (Dynamic Access Lists)
- TCP Intercept
- Context-based Access Control
- Intrusion Prevention System (IPS) (formerly known as Cisco IOS Firewall Intrusion Detection System)
- Authentication Proxy
- Port to Application Mapping
- Security Server Support
- Network Address Translation
- IPSec Network Security
- Neighbor Router Authentication

- Event Logging
- User Authentication and Authorization

In addition to configuring these features, you should follow the guidelines listed in the "Other Guidelines for Configuring Your Firewall" section. This section outlines important security practices to protect your firewall and network. Table 23 describes Cisco IOS security features.

Table 23 Cisco IOS Features for a Robust Firewall

Feature	Chapter	Comments
Standard Access Lists and Static Extended Access Lists	"Access Control Lists: Overview and Guidelines"	Standard and static extended access lists provide basic traffic filtering capabilities. You configure criteria that describe which packets should be forwarded, and which packets should be dropped at an interface, based on each packet's network layer information. For example, you can block all UDP packets from a specific source IP address or address range. Some extended access lists can also examine transport layer information to determine whether to block or forward packets. To configure a basic firewall, you should at a minimum configure basic traffic filtering. You should configure basic access lists for all network protocols that will be routed through your firewall, such as IP, IPX, AppleTalk, and so forth.
Lock-and-Key (Dynamic Access Lists)	"Configuring Lock-and-Key Security (Dynamic Access Lists)"	Lock-and-Key provides traffic filtering with the ability to allow temporary access through the firewall for certain individuals. These individuals must first be authenticated (by a username/password mechanism) before the firewall allows their traffic through the firewall. Afterwards, the firewall closes the temporary opening. This provides tighter control over traffic at the firewall than with standard or static extended access lists.
Reflexive Access Lists	"Configuring IP Session Filtering (Reflexive Access Lists)"	Reflexive access lists filter IP traffic so that TCP or UDP "session" traffic is only permitted through the firewall if the session originated from within the internal network. You would only configure Reflexive Access Lists when not using Context-based Access Control.
TCP Intercept	"Configuring TCP Intercept (Preventing Denial-of-Service Attacks)"	TCP Intercept protects TCP servers within your network from TCP SYN-flooding attacks, a type of denial-of-service attack. You would only configure TCP Intercept when not using Context-based Access Control.

Table 23 Cisco IOS Features for a Robust Firewall (continued)

Feature	Chapter	Comments
Context-based Access Control	"Configuring Context-Based Access Control"	CBAC examines not only network layer and transport layer information, but also examines the application-layer protocol information (such as FTP information) to learn about the state of TCP and UDP connections. CBAC maintains connection state information for individual connections. This state information is used to make intelligent decisions about whether packets should be permitted or denied, and dynamically creates and deletes temporary openings in the firewall. CBAC also generates real-time alerts and audit trails. Enhanced audit trail features use SYSLOG to track all network transactions. Real-time alerts send SYSLOG error messages to central management consoles upon detecting suspicious activity. Using CBAC inspection rules, you can configure alerts and audit trail information on a per-application protocol basis. CBAC is only available in the Cisco IOS Firewall feature set.
Cisco IOS Intrusion Prevention System (IPS)	"Configuring Cisco IOS Intrusion Prevention System (IPS) "	The Cisco IOS IPS acts as an in-line intrusion detection sensor, "watching" packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When Cisco IOS IPS detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or Security Device Event Exchange (SDEE). The network administrator can configure Cisco IOS IPS to choose the appropriate response to various threats. When packets in a session match a signature, Cisco IOS IPS can take any of the following actions, as appropriate: • Send an alarm to a syslog server or a centralized management interface • Drop the packet • Reset the connection • Deny traffic from the source IP address of the attacker for a specified amount of time • Deny traffic on the connection for which the signature was seen for a specified amount of time
Authentication Proxy	"Configuring Authentication Proxy"	The Cisco IOS Firewall authentication proxy feature allows network administrators to apply specific security policies on a per-user basis. Previously, user identity and related authorized access was associated with a user's IP address, or a single security policy had to be applied to an entire user group or sub network. Now, users can be identified and authorized on the basis of their per-user policy, and access privileges tailored on an individual basis are possible, as opposed to general policy applied across multiple users.

Table 23 Cisco IOS Features for a Robust Firewall (continued)

Feature	Chapter	Comments
Port to Application Mapping	"Configuring Port to Application Mapping"	Port to Application Mapping (PAM) is a feature of Cisco IOS Firewall. PAM allows you to customize TCP or UDP port numbers for network services or applications. PAM uses this information to support network environments that run services using ports that are different from the registered or well-known ports associated with an application. The information in the PAM table enables CBAC supported services to run on nonstandard ports.
Security Server Support	"Configuring TACACS+," "Configuring RADIUS," and "Configuring Kerberos"	The Cisco IOS Firewall feature set can be configured as a client of the following supported security servers: • TACACS+ (including CiscoSecure) • RADIUS • Kerberos You can use any of these security servers to store a database of user profiles. To gain access into your firewall or to gain access through the firewall into another network, users must enter authentication information (such as a username and password), which is matched against the information on the security server. When users pass authentication, they are granted access according to their specified privileges.
Network Address Translation	"Configuring NAT for IP Address Conservation"	You can use Network Address Translation (NAT) to hide internal IP network addresses from the world outside the firewall. NAT was designed to provide IP address conservation and for internal IP networks that have unregistered (not globally unique) IP addresses: NAT translates these unregistered IP addresses into legal addresses at the firewall. NAT can also be configured to advertise only one address for the entire internal network to the outside world. This provides security by effectively hiding the entire internal network from the world. NAT gives you limited spoof protection because internal addresses are hidden. Additionally, NAT removes all your internal services from the external name space. NAT does not work with the application-layer protocols RPC, VDOLive, or SQL*Net "Redirected." (NAT does work with SQL*Net "Bequeathed.") Do not configure NAT with networks that will carry traffic for these incompatible protocols.

Table 23 Cisco IOS Features for a Robust Firewall (continued)

Feature	Chapter	Comments
IPSec Network Security	"Configuring Security for VPNs with IPSec "	IPSec is a framework of open standards developed by the Internet Engineering Task Force (IETF) that provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices ("peers") such as Cisco routers.
Neighbor Router Authentication	"Neighbor Router Authentication: Overview and Guidelines"	Neighbor router authentication requires the firewall to authenticate all neighbor routers before accepting any route updates from that neighbor. This ensures that the firewall receives legitimate route updates from a trusted source.
Event Logging	"System Monitoring and Logging" chapter in the *Cisco IOS Network Management Configuration Guide*	Event logging automatically logs output from system error messages and other events to the console terminal. You can also redirect these messages to other destinations such as virtual terminals, internal buffers, or syslog servers. You can also specify the severity of the event to be logged, and you can configure the logged output to be timestamped. The logged output can be used to assist real-time debugging and management, and to track potential security breaches or other nonstandard activities throughout a network.
User Authentication and Authorization	"Configuring Authentication" and "Configuring Authorization"	Authentication and authorization help protect your network from access by unauthorized users.

Other Guidelines for Configuring Your Firewall

As with all networking devices, you should always protect access into the firewall by configuring passwords as described in the module "Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices." You should also consider configuring user authentication, authorization, and accounting as described in the chapters in the "Authentication, Authorization, and Accounting (AAA)" part of this guide.

You should also consider the following recommendations:

- When setting passwords for privileged access to the firewall, use the **enable secret** command rather than the **enable password** command, which does not have as strong an encryption algorithm.
- Put a password on the console port. In authentication, authorization, and accounting (AAA) environments, use the same authentication for the console as for elsewhere. In a non-AAA environment, at a minimum configure the **login** and **password** *password* commands.
- Think about access control *before* you connect a console port to the network in any way, including attaching a modem to the port. Be aware that a *break* on the console port might give total control of the firewall, even with access control configured.
- Apply access lists and password protection to all virtual terminal ports. Use access lists to limit who can Telnet into your router.

- Do not enable any local service (such as SNMP or NTP) that you do not use. Cisco Discovery Protocol (CDP) and Network Time Protocol (NTP) are on by default, and you should turn these off if you do not need them.

  To turn off CDP, enter the **no cdp run** global configuration command. To turn off NTP, enter the **ntp disable** interface configuration command on each interface not using NTP.

  If you must run NTP, configure NTP only on required interfaces, and configure NTP to listen only to certain peers.

  Any enabled service could present a potential security risk. A determined, hostile party might be able to find creative ways to misuse the enabled services to access the firewall or the network.

  For local services that are enabled, protect against misuse. Protect by configuring the services to communicate only with specific peers, and protect by configuring access lists to deny packets for the services at specific interfaces.

- Protect against spoofing: protect the networks on both sides of the firewall from being spoofed from the other side. You could protect against spoofing by configuring input access lists at all interfaces to pass only traffic from expected source addresses, and to deny all other traffic.

  You should also disable source routing. For IP, enter the **no ip source-route** global configuration command. Disabling source routing at *all* routers can also help prevent spoofing.

  You should also disable minor services. For IP, enter the **no service tcp-small-servers** and **no service udp-small-servers** global configuration commands.

- Prevent the firewall from being used as a relay by configuring access lists on any asynchronous Telnet ports.

- Normally, you should disable directed broadcasts for all applicable protocols on your firewall and on all your other routers. For IP, use the **no ip directed-broadcast** command. Rarely, some IP networks do require directed broadcasts; if this is the case, do not disable directed broadcasts.

  Directed broadcasts can be misused to multiply the power of denial-of-service attacks, because every denial-of-service packet sent is broadcast to every host on a subnet. Furthermore, some hosts have other intrinsic security risks present when handling broadcasts.

- Configure the **no ip proxy-arp** command to prevent internal addresses from being revealed. (This is important to do if you do not already have NAT configured to prevent internal addresses from being revealed.)

- Keep the firewall in a secured (locked) room.

---

coincidental.

Access Control Lists (ACLs)

IP Access List Features Roadmap

First Published: August 18, 2006
Last Updated: August 18, 2006

This roadmap lists the access list features documented in the *Cisco IOS Security Configuration Guide* and maps them to the modules in which they appear.

Feature and Release Support

Table 1 lists access list feature support for the Cisco IOS software releases 12.2S, 12.3T, and 12.4T.

Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table. *Not all features may be supported in your Cisco IOS software release*

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note

Table 1 Supported Access List Features

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Releases 12.2S, 12.3T, and 12.4T			
12.2(25)S	ACL Support for Filtering IP Options	This feature allows you to filter packets having IP Options, in order to prevent routers from becoming saturated with spurious packets.	Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Supported Access List Features (continued)

12.3(4)T 12.2(25)S	ACL TCP Flags Filtering	This feature provides a flexible mechanism for filtering on TCP flags. Before Cisco IOS Release 12.3(4)T, an incoming packet was matched as long as any TCP flag in the packet matched a flag specified in the access control entry (ACE). This behavior allows for a security loophole, because packets with all flags set could get past the access control list (ACL). The ACL TCP Flags Filtering feature allows you to select any combination of flags on which to filter. The ability to match on a flag set and on a flag not set gives you a greater degree of control for filtering on TCP flags, thus enhancing security.	Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values
12.3(7)T 12.2(25)S	ACL—Named ACL Support for Noncontiguous Ports on an Access Control Entry	This feature allows you to specify noncontiguous ports in a single access control entry, which greatly reduces the number of entries required in an access control list when several entries have the same source address, destination address, and protocol, but differ only in the ports.	Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values
12.4(2)T	ACL Support for Filtering on TTL Value	You may use extended IP access lists (named or numbered) to filter packets based on their time-to-live (TTL) value, from 0 to 255. This filtering enhances your control over which packets reach a router.	Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values
12.4(6)T	ACL Manageability	The ACL Manageability feature enables users to display and clear Access Control Entry (ACE) statistics per interface and per incoming or outgoing traffic direction for access control lists (ACLs).	Displaying and Clearing IP Access List Data Using ACL Manageability http://lbj/push_targets1/ucdit/cc/td/doc/product/software/ios124/124tcg/tsec_c/tsaclsho.htm

IP Access List Overview

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

First Published: August 18, 2006
Last Updated: August 18, 2006

Access control lists (ACLs) perform packet filtering to control which packets move through the network and where. Such control provides security by helping to limit network traffic, restrict the access of users and devices to the network, and prevent traffic from leaving a network. IP access lists can reduce the chance of spoofing and denial-of-service attacks and allow dynamic, temporary user access through a firewall.

IP access lists can also be used for purposes other than security, such as bandwidth control, restricting the content of routing updates, redistributing routes, triggering dial-on-demand (DDR) calls, limiting debug output, and identifying or classifying traffic for quality of service (QoS) features. This module provides an overview of IP access lists.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About IP Access Lists

This module contains the following concepts, which you should understand before configuring an IP access control list (ACL), also known as an access list:

Benefits of IP Access Lists

Access control lists (ACLs) perform packet filtering to control which packets move through the network and where. Such control can restrict the access of users and devices to the network, providing a measure of security. Access lists can save network resources by reducing traffic. Access lists provide diverse benefits, depending on how they are used. Many of the benefits fall into the following categories:

Block Unwanted Traffic or Users

Access lists can filter incoming or outgoing packets on an interface, thereby controlling access based on source addresses, destination addresses, or user authentication. You can also use access lists to determine which types of traffic are forwarded or blocked at the router interfaces. For example, you can permit e-mail traffic to be routed, but at the same time block all Telnet traffic.

Reduce the Chance of DOS Attacks

There are a number of ways to reduce the chance of denial-of-service attacks. For example, by specifying IP source addresses, you can control whether traffic from hosts, networks, or users access your network. You can filter on specific time-to-live (TTL) values in packets to control how many hops a packet can take before reaching a router in your network. By configuring the TCP Intercept feature, you can prevent servers from being flooded with requests for a connection.

Control Access to Virtual Terminal Lines

You can place an access list on inbound vty (Telnet) line access from certain nodes or networks. You can also place an access list on outbound vty access, blocking or permitting Telnet access to other devices.

Restrict the Content of Routing Updates

Access lists can control routing updates being sent, received, or redistributed.

Provide Bandwidth Control

An access list on a slow link can prevent excess traffic.

Identify or Classify Traffic for QoS Features

Access lists can provide congestion avoidance by setting IP precedence for WRED or CAR. It can provide congestion management for class-based weighted fair queuing (WFQ), priority queuing, and custom queuing.

Trigger Dial-on-Demand (DDR) Calls

An access list can enforce dialing and disconnect criteria.

Limit Debug Command Output

An access list can limit debug output based on an address or protocol.

Provide NAT Control

Access lists can control which addresses are translated by Network Address Translation (NAT).

Authenticate Incoming RSH and RCP Requests

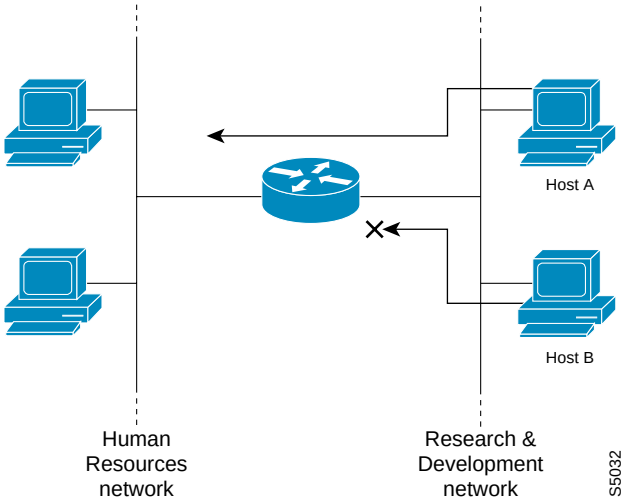
To enable the Cisco IOS software to receive incoming remote shell (rsh) protocol and remote copy (rcp) protocol requests, customers must configure an authentication database to control access to the router. Access lists can simplify the identification of local users, remote hosts, and remote users in the database authentication configuration.

Border Routers and Firewall Routers Should Use Access Lists

There are many reasons to configure access lists; for example, you can use access lists to restrict contents of routing updates or to provide traffic flow control. One of the most important reasons to configure access lists is to provide a basic level of security for your network by controlling access to it. If you do not configure access lists on your router, all packets passing through the router could be allowed onto all parts of your network.

An access list can allow one host to access a part of your network and prevent another host from accessing the same area. In Figure 1, by applying an appropriate access list to the interfaces of the router, Host A is allowed to access the Human Resources network and Host B is prevented from accessing the Human Resources network.

Figure 1 Traffic Filters to Prevent Traffic from Being Routed to a Network



Access lists should be used in firewall routers, which are often positioned between your internal network and an external network such as the Internet. You can also use access lists on a router positioned between two parts of your network, to control traffic entering or exiting a specific part of your internal network.

To provide some security benefits of access lists, you should at least configure access lists on border routers—routers located at the edges of your networks. Such an access list provides a basic buffer from the outside network or from a less controlled area of your own network into a more sensitive area of your network. On these border routers, you should configure access lists for each network protocol configured on the router interfaces. You can configure access lists so that inbound traffic or outbound traffic or both are filtered on an interface.

Access lists are defined on a per-protocol basis. In other words, you should define access lists for every protocol enabled on an interface if you want to control traffic flow for that protocol.

Definition of an Access List

An access list is a sequential list consisting of at least one **permit** statement and possibly one or more **deny** statements. In the case of IP access lists, the statements can apply to IP addresses, upper-layer IP protocols, or other fields in IP packets. The access list is identified and referenced by a name or a number. The access list acts as a packet filter, filtering packets based on the criteria defined in the access list.

An access list may be configured, but it does not take effect until the access list is either applied to an interface (with the **ip access-group** command), a virtual terminal line (vty) (with the **access-class** command), or referenced by some other command that accepts an access list. Access lists have many uses, and therefore many Cisco IOS software commands accept a reference to an access list in their command syntax. Multiple commands can reference the same access list.

In the following configuration excerpt, the first three lines are an example of an IP access list named branchoffices, which is applied to serial interface 0 on incoming packets. No sources other than those on the networks specified by each source address and mask pair can access this interface. The destinations for packets coming from sources on network 172.20.7.0 are unrestricted. The destination for packets coming from sources on network 172.29.2.0 must be 172.25.5.4.

```
ip access-list extended branchoffices
 10 permit 172.20.7.0 0.0.0.3 any
 20 permit 172.29.2.0 0.0.0.255 host 172.25.5.4
```

```
!
interface serial 0
 ip access-group branchoffices in
```

Software Processing of an Access List

The following general steps describe how the Cisco IOS software processes an access list when it is applied to an interface, a vty, or referenced by some other Cisco IOS command. These steps apply to an access list that has 13 or fewer access list entries.

- The software receives an IP packet and tests parts of each packet being filtered against the conditions in the access list, one condition (**permit** or **deny** statement) at a time. For example, the software tests the source and destination addresses of the packet against the source and destination addresses in a **permit** or **deny** statement.
- If a packet does not match an access list statement, the packet is then tested against the next statement in the list.
- If a packet and an access list statement match, the rest of the statements in the list are skipped and the packet is permitted or denied as specified in the matched statement. The first entry that the packet matches determines whether the software permits or denies the packet. That is, after the first match, no subsequent entries are considered.
- If the access list denies a packet, the software discards the packet and returns an ICMP Host Unreachable message.
- If no conditions match, the software drops the packet. This is because each access list ends with an unwritten, implicit **deny** statement. That is, if the packet has not been permitted by the time it was tested against each statement, it is denied.

In later Cisco IOS releases such as Release 12.4, 12.2S, and 12.0S, by default, an access list that has more than 13 access list entries is processed differently from one that has 13 or fewer entries. In order to be more efficient, an access list with more than 13 entries is processed using a trie-based lookup algorithm. This process will happen automatically; it does not need to be configured.

Access List Rules

Keep the following rules and characteristics of access lists in mind when creating one:

- Only one access list per interface, per protocol, per direction is allowed.
- The access list must contain at least one **permit** statement or else all packets are denied.
- Because the software stops testing conditions after the first match, the order of the conditions is critical. The same **permit** or **deny** statements specified in a different order could result in a packet being passed under one circumstance and denied in another circumstance.
- If an access list is referenced by name in a command, but the access list does not exist, all packets pass. That is, an interface or command with an empty access list applied to it permits all traffic.
- Standard access lists and extended access lists cannot have the same name.
- Inbound access lists process packets arriving at the router. Incoming packets are processed before being routed to an outbound interface. An inbound access list is efficient because it saves the overhead of routing lookups if the packet is to be discarded because it is denied by the filtering tests. If the packet is permitted by the tests, it is then processed for routing. For inbound lists, **permit** means continue to process the packet after receiving it on an inbound interface; **deny** means discard the packet.

- Outbound access lists process packets before they leave the router. Incoming packets are routed to the outbound interface and then processed through the outbound access list. For outbound lists, **permit** means send it to the output buffer; **deny** means discard the packet.
- An access list can control traffic arriving at the router or leaving the router, but not traffic originating at the router.

Helpful Hints for Creating IP Access Lists

The following tips will help you avoid unintended consequences and help you create more efficient, useful access lists.

- Create the access list before applying it to an interface (or elsewhere), because if you apply a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause you immediate access problems.
- Another reason to configure an access list before applying it is because an interface with an empty access list applied to it permits all traffic.
- All access lists need at least one **permit** statement; otherwise, all packets are denied and no traffic passes.
- Because the software stops testing conditions after it encounters the first match (to either a **permit** or **deny** statement), you will reduce processing time and resources if you put the statements that packets are most likely to match at the beginning of the access list. Place more frequently occurring conditions before less frequent conditions.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- Use the statement **permit any any** if you want to allow all other packets not already denied. Using the statement **permit any any** in effect avoids denying all other packets with the implicit deny statement at the end of an access list. Do not make your first access list entry **permit any any** because all traffic will get through; no packets will reach the subsequent testing. In fact, once you specify **permit any any**, all traffic not already denied will get through.
- Although all access lists end with an implicit **deny** statement, we recommend use of an explicit **deny** statement (for example, **deny ip any any**). On most platforms, you can display the count of packets denied by issuing the **show access-list** command, thus finding out more information about who your access list is disallowing. Only packets denied by explicit **deny** statements are counted, which is why the explicit **deny** statement will yield more complete data for you.
- While you are creating an access list or after it is created, you might want to delete an entry.
  - You cannot delete an entry from a *numbered* access list; trying to do so will delete the entire access list. If you need to delete an entry, you need to delete the entire access list and start over.
  - You can delete an entry from a *named* access list. Use the **no permit** or **no deny** command to delete the appropriate entry.
- In order to make the purpose of individual statements more scannable and easily understood at a glance, you can write a helpful remark before or after any statement by using the **remark** command.
- If you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you.

- This hint applies to the placement of your access list. When trying to save resources, remember that an inbound access list applies the filter conditions *before* the routing table lookup. An outbound access list applies the filter conditions *after* the routing table lookup.

Named or Numbered Access Lists

All access lists must be identified by a name or a number. Named and numbered access lists have different command syntax. Named access lists are compatible with Cisco IOS Release 11.2 and later. Named access lists are more convenient than numbered access lists because you can specify a meaningful name that is easier to remember and associate with a purpose. You may reorder statements in or add statements to a named access list.

Named access list are newer than numbered access lists and support the following features that are not supported in numbered access lists:

- TCP flag filtering
- IP option filtering
- noncontiguous ports
- reflexive access lists
- ability to delete entries with the **no permit** or **no deny** command

Not all commands that accept a numbered access list will accept a named access list. For example, virtual terminal lines use only numbered access lists.

Standard or Extended Access Lists

All access lists are either standard or extended access lists. If you only intend to filter on a source address, the simpler standard access list is sufficient. For filtering on anything other than a source address, an extended access list is necessary.

- Named access lists are specified as standard or extended based on the keyword **standard** or **extended** in the **ip access-list** command syntax.
- Numbered access lists are specified as standard or extended based on their number in the **access-list** command syntax. Standard IP access lists are numbered 1 to 99 or 1300 to 1999; extended IP access lists are numbered 100 to 199 or 2000 to 2699. The range of standard IP access lists was initially only 1 to 99, and was subsequently expanded with the range 1300 to 1999 (the intervening numbers were assigned to other protocols). The extended access list range was similarly expanded.

Standard Access Lists

Standard IP access lists test only source addresses of packets (except for two exceptions). Because standard access lists test source addresses, they are very efficient at blocking traffic close to a destination. There are two exceptions when the address in a standard access list is not a source address:

- On outbound VTY access lists, when someone is trying to telnet, the address in the access list entry is used as a destination address rather than a source address.
- When filtering routes, you are filtering the network being advertised to you rather than a source address.

Extended Access Lists

Extended access lists are good for blocking traffic anywhere. Extended access lists test source and destination addresses and other IP packet data, such as protocols, TCP or UDP port numbers, type of service (ToS), precedence, TCP flags, IP options, and TTL value. Extended access lists can also provide capabilities that standard access lists cannot, such as the following:

- Filtering IP Options
- Filtering TCP flags
- Filtering noninitial fragments of packets (see the module “Refining an IP Access List”)
- Time-based entries (see the “Time-Based and Distributed Time-Based Access Lists” section on page 11 and the module “Refining an IP Access List”)
- Dynamic access lists (see the section “Types of IP Access Lists” section on page 11)
- Reflexive access lists (see the section “Types of IP Access Lists” section on page 11 and the module “Configuring IP Session Filtering [Reflexive Access Lists])

Note Packets that are subject to an extended access list will not be autonomous switched.

IP Packet Fields You Can Filter to Control Access

You can use an extended access list to filter on any of the following fields in an IP packet. Source address and destination address are the two most frequently specified fields on which to base an access list:

- Source address—Specifies a source address to control packets coming from certain networking devices or hosts.
- Destination address—Specifies a destination address to control packets being sent to certain networking devices or hosts.
- Protocol—Specifies an IP protocol indicated by the keyword **eigrp**, **gre**, **icmp**, **igmp**, **ip**, **ipinip**, **nos**, **ospf**, **tcp**, or **udp**, or indicated by an integer in the range from 0 to 255 (representing an Internet protocol). If you specify a transport layer protocol (**icmp**, **igmp**, **tcp**, or **udp**), the command has a specific syntax.
  - Ports and non-contiguous ports—Specifies TCP or UDP ports by a port name or port number. The port numbers can be noncontiguous port numbers. Port numbers can be useful to filter Telnet traffic or HTTP traffic, for example.
  - TCP flags—Specifies that packets match any flag or all flags set in TCP packets. Filtering on specific TCP flags can help prevent false synchronization packets.
- IP options—Specifies IP options; one reason to filter on IP options is to prevent routers from being saturated with spurious packets containing them.
- TTL value—Specifies TTL values indicated by an operator and possibly a range of values. Filtering on TTL value can control who can reach an interface based on how many hops away the source is. Such filtering can also prevent packets from reaching the process level.

Wildcard Mask for Addresses in an Access List

Address filtering uses wildcard masking to indicate to the software whether to check or ignore corresponding IP address bits when comparing the address bits in an access list entry to a packet being submitted to the access list. By carefully setting wildcard masks, you can specify one or more IP addresses for permit or deny tests.

Wildcard masking for IP address bits uses the number 1 and the number 0 to specify how the software treats the corresponding IP address bits. A wildcard mask is sometimes referred to as an inverted mask because a 1 and 0 mean the opposite of what they mean in a subnet (network) mask.

- A wildcard mask bit 0 means *check* the corresponding bit value; they must match.
- A wildcard mask bit 1 means *ignore* that corresponding bit value; they need not match.

If you do not supply a wildcard mask with a source or destination address in an access list statement, the software assumes an implicit wildcard mask of 0.0.0.0, meaning all values must match.

Unlike subnet masks, which require contiguous bits indicating network and subnet to be ones, wildcard masks allow noncontiguous bits in the mask.

Table 1 shows examples of IP addresses and masks from an access list, along with the corresponding addresses that are considered a match.

Table 1 Sample IP Addresses, Wildcard Masks, and Match Results

Address	Wildcard Mask	Match Results
0.0.0.0	255.255.255.255	All addresses will match the access list conditions.
172.18.0.0/16	0.0.255.255	Network 172.18.0.0
172.18.5.2/16	0.0.0.0	Only host 172.18.5.2 matches
172.18.8.0	0.0.0.7	Only subnet 172.18.8.0/29 matches
172.18.8.8	0.0.0.7	Only subnet 172.18.8.8/29 matches
172.18.8.15	0.0.0.3	Only subnet 172.18.8.15/30 matches
10.1.2.0	0.0.252.255 (noncontiguous bits in mask)	Matches any even-numbered network in the range of 10.1.2.0 to 10.1.254.0

Access List Sequence Numbers

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If you wanted to insert an entry in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

This feature allows users to add sequence numbers to access list entries and resequence them. When you add a new entry, you specify the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry.

Access List Logging

The Cisco IOS software can provide logging messages about packets permitted or denied by a single standard or extended IP access list entry. That is, any packet that matches the entry will cause an informational logging message about the packet to be sent to the console. The level of messages logged to the console is controlled by the **logging console** global configuration command.

The first packet that triggers the access list entry causes an immediate logging message, and subsequent packets are collected over 5-minute intervals before they are displayed or logged. The logging message includes the access list number, whether the packet was permitted or denied, the source IP address of the packet, and the number of packets from that source permitted or denied in the prior 5-minute interval.

However, you can use the **ip access-list log-update** command to set the number of packets that, when match an access list (and are permitted or denied), cause the system to generate a log message. You might want to do this to receive log messages more frequently than at 5-minute intervals.

Caution If you set the *number-of-matches* argument to 1, a log message is sent right away, rather than caching it; every packet that matches an access list causes a log message. A setting of 1 is not recommended because the volume of log messages could overwhelm the system.

Even if you use the **ip access-list log-update** command, the 5-minute timer remains in effect, so each cache is emptied at the end of 5 minutes, regardless of the count of messages in each cache. Regardless of when the log message is sent, the cache is flushed and the count reset to 0 for that message the same way it is when a threshold is not specified.

Note The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from crashing due to too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list.

Alternative to Access List Logging

Packets matching an entry in an ACL with a log option are process switched. It is not recommended to use the log option on ACLs, but rather use NetFlow export and match on a destination interface of Null0. This is done in the CEF path. The destination interface of Null0 is set for any packet that is dropped by the ACL.

Additional IP Access List Features

Beyond the basic steps to create a standard or extended access list, you can enhance your access lists as mentioned below. Each of these methods is described completely in the module entitled "Refining an Access List."

- You can impose dates and times when **permit** or **deny** statements in an extended access list are in effect, making your access list more granular and specific to an absolute or periodic time period.
- After you create a named access list, you might want to add entries or change the order of the entries, known as resequencing an access list.
- You can achieve finer granularity when filtering packets by filtering on noninitial fragments of packets.

Time-Based and Distributed Time-Based Access Lists

Time-based access lists implement access list entries based on particular times of the day or week. This is an advantage when you don't want access list entries always in effect or in effect as soon as they are applied. Use time-based access lists to make the enforcement of permit or deny conditions granular, based on time and date.

Distributed time-based access lists are those that are supported on line cards for the Cisco 7500 series routers. Packets destined for an interface configured with time-based access lists are distributed switched through the line card.

Types of IP Access Lists

There are several types of access lists that are distinct because of how they are triggered, their temporary nature, or how their behavior differs from an ordinary access list.

Authentication Proxy

Authentication proxy provides dynamic, per-user authentication and authorization, authenticating users against industry standard TACACS+ and RADIUS authentication protocols. Authenticating and authorizing connections by users provides more robust protection against network attacks.

Context-Based Access Control

Context-based access control (CBAC) examines not only network layer and transport layer information, but also the application-layer protocol information (such as FTP information) to learn about the state of TCP and UDP connections. CBAC maintains connection state information for individual connections. This state information is used to make intelligent decisions about whether packets should be permitted or denied, and dynamically creates and deletes temporary openings in the firewall.

Dynamic Access Lists with the Lock-and-Key Feature

Dynamic access lists provide temporary access to designated users who are using Telnet to reach designated hosts through a firewall. Dynamic access lists involve user authentication and authorization.

Reflexive Access Lists

Reflexive access lists provide filtering on upper-layer IP protocol sessions. They contain temporary entries that are automatically created when a new IP session begins. They are nested within extended, named IP access lists that are applied to an interface. Reflexive access lists are typically configured on border routers, which pass traffic between an internal and external network. These are often firewall routers. Reflexive access lists do not end with an implicit deny statement because they are nested within an access list and the subsequent statements need to be examined.

Where to Apply an Access List

If you are applying an access list to an interface, carefully consider whether to specify it as **in** (inbound) or **out** (outbound). Applying an access list to an incoming or outgoing interface controls the traffic that will enter or leave the router's interface or process level (in the case of filtering on TTL values).

- When an inbound access list is applied to an interface, after the software receives a packet, the software checks the packet against the access list statements. If the access list permits the packet, the software continues to process the packet. Therefore, filtering on incoming packets can save router resources because filtered packets will not go through the router.

- Access lists that apply to outbound packets are filtering packets that have already gone through the router. Packets that pass the access list are transmitted (sent) out the interface.
- The TCP ACL splitting feature of Rate-Based Satellite Control Protocol (RBSCP) is an example of a feature that can be used on an outgoing interface. The access list controls which packets are subject to TCP ACK splitting.

Access lists can be used in ways other than applying them to interfaces. The following are additional places to apply an access list.

- To restrict incoming and outgoing connections between a particular vty (into a Cisco device) and the network devices at addresses in an access list, apply an access list to a line. See the "Controlling Access to a Virtual Terminal Line" module.
- Referencing an access list from a **debug** command limits the amount of information displayed to only the information permitted by the access list, such as sources, destinations, or protocols, for example.
- Access lists can be used to control routing updates, to control dial-on-demand routing (DDR), and to control quality of service (QoS) features, for example. See the appropriate configuration chapters for using access lists with these features.

Where to Go Next

You must first decide what you want to restrict, and then select the type of access list that achieves your goal. Next, you will create an access list that permits or denies packets based on values in the fields you specify, and finally, you will apply the access list (which determines its placement).

Assuming you have decided what you want to restrict and what type of access list you need, your next step is to create an access list. Creating an access list based on source address, destination address, or protocol is described in the "Creating an IP Access List and Applying It to an Interface" module. You could create an access list that filters on other fields, as described in "Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values." If you want to control access to a virtual line, see "Controlling Access to a Virtual Terminal Line." If the purpose of your access list is to control routing updates or QoS features, for example, see the appropriate technology chapter.

Additional References

The following sections provide references related to IP access lists.

Related Documents

Related Topic	Document Title
IP access list commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Application Services Command Reference,* Release 12.4
Filtering on source address, destination address, or protocol	"Creating an IP Access List and Applying It to an Interface"
Filtering on IP Options, TCP flags, noncontiguous ports, or TTL	"Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, and TTL Values"
Restricting access to a vty line.	"Controlling Access to a Virtual Terminal Line"

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	—

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Creating an IP Access List and Applying It to an Interface

First Published: August 18, 2006
Last Updated: August 18, 2006

IP access lists provide many benefits for securing a network and achieving nonsecurity goals, such as determining quality of service (QOS) factors or limiting **debug** command output. This module describes how to create standard, extended, named, and numbered IP access lists. An access list can be referenced by a name or a number. Standard access lists filter on only the source address in IP packets. Extended access lists can filter on source address, destination address, and other fields in an IP packet.

After you create an access list, you must apply it to something in order for it to have any effect. This module describes how to apply an access list to an interface. However, there are many other uses for an access list, which are referenced in this module and described in other modules and in other configuration guides for various technologies.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Creating IP Access Lists" section on page 24.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Creating an IP Access List and Applying It to an Interface

Before you create or apply an IP access list, you should understand the concepts in the "IP Access List Overview" module. You should also have IP running in your network.

Information About Creating an IP Access List and Applying It to an Interface

You should understand the following concepts before creating an IP access list.

Helpful Hints for Creating IP Access Lists

The following tips will help you avoid unintended consequences and help you create more efficient access lists.

- Create the access list before applying it to an interface (or elsewhere), because if you apply a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause you immediate access problems.
- Another reason to configure an access list before applying it is because an interface with an empty access list applied to it permits all traffic.
- All access lists need at least one **permit** statement; otherwise, all packets are denied and no traffic passes.
- Because the software stops testing conditions after it encounters the first match (to either a **permit** or **deny** statement), you will reduce processing time and resources if you put the statements that packets are most likely to match at the beginning of the access list. Place more frequently occurring conditions before less frequent conditions.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- Use the statement **permit any any** if you want to allow all other packets not already denied. Using the statement **permit any any** in effect avoids denying all other packets with the implicit deny statement at the end of an access list. Do not make your first access list entry **permit any any** because all traffic will get through; no packets will reach the subsequent testing. In fact, once you specify **permit any any**, all traffic not already denied will get through.

- Although all access lists end with an implicit **deny** statement, we recommend use of an explicit **deny** statement (for example, **deny ip any any**). On most platforms, you can display the count of packets denied by issuing the **show access-list** command, thus finding out more information about who your access list is disallowing. Only packets denied by explicit **deny** statements are counted, which is why the explicit **deny** statement will yield more complete data for you.
- While you are creating an access list or after it is created, you might want to delete an entry.
  - You cannot delete an entry from a *numbered* access list; trying to do so will delete the entire access list. If you need to delete an entry, you need to delete the entire access list and start over.
  - You can delete an entry from a *named* access list. Use the **no permit** or **no deny** command to delete the appropriate entry.
- In order to make the purpose of individual statements more scannable and easily understood at a glance, you can write a helpful remark before or after any statement by using the **remark** command.
- If you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you.
- This hint applies to the placement of your access list. When trying to save resources, remember that an inbound access list applies the filter conditions *before* the routing table lookup. An outbound access list applies the filter conditions *after* the routing table lookup.

Access List Remarks

You can include comments (remarks) about entries in a named IP access list. An access list remark is an optional comment before or after an access list entry that describes the entry for you at a glance, so you do not have to interpret the purpose of the entry by its command syntax. Each remark is limited to 100 characters.

The remark can go before or after a **permit** or **deny** statement. You should be consistent about where you put your remarks so that it is clear which remark describes which statement. It could be confusing to have some remarks before the associated **permit** or **deny** statements and some remarks after the associated statements.

The following example of a remark is a user-friendly description of what the subsequent **deny** statement does.

```
ip access-list extended telnetting
 remark Do not allow host1 subnet to telnet out
 deny tcp host 172.69.2.88 any eq telnet
```

Additional IP Access List Features

Beyond the basic steps to create a standard or extended access list, you can enhance your access lists as mentioned below. Each of these methods is described completely in the module entitled "Refining an Access List."

- You can impose dates and times when **permit** or **deny** statements in an extended access list are in effect, making your access list more granular and specific to an absolute or periodic time period.
- After you create a named or numbered access list, you might want to add entries or change the order of the entries, known as resequencing an access list.
- You can achieve finer granularity when filtering packets by filtering on noninitial fragments of packets.

How to Create an IP Access List and Apply It to an Interface

This section describes the general ways to create a standard or extended access list using either a name or a number. Access lists are very flexible; the tasks simply illustrate one **permit** command and one **deny** command to provide you the command syntax of each. Only you can determine how many **permit** and **deny** commands you need and their order.

Note The first two tasks in this module create an access list; you must apply the access list in order for it to function. If you want to apply the access list to an interface, perform the task "Applying the Access List to an Interface" section on page 16.

If you don't intend to apply the access list to an interface, see the "Where to Go Next" section on page 21 for pointers to modules that describe other ways to apply access lists.

Creating a Standard Access List to Filter on Source Address

If you want to filter on source address only, a standard access list is simple and sufficient. There are two alternative types of standard access list: named and numbered. Named access lists allow you to identify your access lists with a more intuitive name rather than a number, and they also support more features than numbered access lists.

Creating a Named Access List to Filter on Source Address

Use a standard, named access list if you need to filter on source address only. This task illustrates one **permit** statement and one **deny** statement, but the actual statements you use and their order depend on what you want to filter or allow. Define your **permit** and **deny** statements in the order that achieves your filtering goals.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list standard** *name*
4. **remark** *remark*
5. **deny** {*source* [*source-wildcard*] | **any**} [**log**]
6. **remark** *remark*
7. **permit** {*source* [*source-wildcard*] | **any**} [**log**]
8. Repeat some combination of Steps 4 through 7 until you have specified the source networks and hosts on which you want to base your access list.

9. **end**
10. **show ip access-list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list standard** *name* **Example:** `Router(config)# ip access-list standard R&D`	Defines a standard IP access list using a name and enters standard named access list configuration mode.
Step 4	**remark** *remark* **Example:** `Router(config-std-nacl)# remark deny Sales network`	(Optional) Adds a user-friendly comment about an access list entry. • A remark can precede or follow an access list entry. • In this example, the remark reminds the network administrator that the subsequent entry denies the Sales network access to the interface (assuming this access list is later applied to an interface).
Step 5	**deny** {*source* [*source-wildcard*] | **any**} [**log**] **Example:** `Router(config-std-nacl)# deny 172.16.0.0 0.0.255.255 log`	(Optional) Denies the specified source based on a source address and wildcard mask. • If the *source-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, all hosts on network 172.16.0.0 are denied passing the access list. • Because this example explicitly denies a source address and the **log** keyword is specified, any packets from that source are logged when they are denied. This is a way to be notified that someone on a network or host is trying to gain access.
Step 6	**remark** *remark* **Example:** `Router(config-std-nacl)# remark Give access to Tester's host`	(Optional) Adds a user-friendly comment about an access list entry. • A remark can precede or follow an access list entry. • This remark reminds the network administrator that the subsequent entry allows the Tester's host access to the interface.

	Command or Action	Purpose
Step 7	**permit** {*source* [*source-wildcard*] | **any**} [**log**] **Example:** `Router(config-std-nacl)# permit 172.18.5.22 0.0.0.0`	Permits the specified source based on a source address and wildcard mask. • Every access list needs at least one **permit** statement; it need not be the first entry. • If the *source-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, host 172.18.5.22 is allowed to pass the access list.
Step 8	Repeat some combination of Steps 4 through 7 until you have specified the sources on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 9	**end** **Example:** `Router(config-std-nacl)# end`	Ends configuration mode and brings the system to privileged EXEC mode.
Step 10	**show ip access-list** **Example:** `Router# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

What to Do Next

The access list you created is not in effect until you apply it to an interface, a vty line, or reference it from a command that uses an access list. See the "Applying the Access List to an Interface" section on page 16 or the "Where to Go Next" section on page 21 for pointers to modules that describe other ways to use access lists.

Creating a Numbered Access List to Filter on Source Address

Configure a standard, numbered access list if you need to filter on source address only and you prefer not to use a named access list.

IP standard access lists are numbered 1 to 99 or 1300 to 1999. This task illustrates one **permit** statement and one **deny** statement, but the actual statements you use and their order depend on what you want to filter or allow. Define your **permit** and **deny** statements in the order that achieves your filtering goals.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **remark** *remark*
4. **access-list** *access-list-number* **permit** {*source* [*source-wildcard*] | **any**} [**log**]

5. **access-list** *access-list-number* **remark** *remark*
6. **access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} [**log**]
7. Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.
8. **end**
9. **show ip access-list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **remark** *remark* **Example:** `Router(config)# access-list 1 remark Give access to Jones`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.
Step 4	**access-list** *access-list-number* **permit** {*source* [*source-wildcard*] | **any**} [**log**] **Example:** `Router(config)# access-list 1 permit 172.16.5.22 0.0.0.0`	Permits the specified source based on a source address and wildcard mask. • Every access list needs at least one **permit** statement; it need not be the first entry. • Standard IP access lists are numbered 1 to 99 or 1300 to 1999. • If the *source-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, host 172.16.5.22 is allowed to pass the access list.
Step 5	**access-list** *access-list-number* **remark** *remark* **Example:** `Router(config)# access-list 1 remark Don't give access to Johnson and log any attempts`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.
Step 6	**access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} [**log**] **Example:** `Router(config)# access-list 1 deny 172.16.7.34 0.0.0.0`	Denies the specified source based on a source address and wildcard mask. • If the *source-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the abbreviation **any** as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, host 172.16.7.34 is denied passing the access list.

	Command or Action	Purpose
Step 7	Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 8	**end** **Example:** `Router(config-std-nacl)# end`	Ends configuration mode and brings the system to privileged EXEC mode.
Step 9	**show ip access-list** **Example:** `Router# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

What to Do Next

The access list you created is not in effect until you apply it to an interface, a vty line, or reference it from a command that uses an access list. See the "Applying the Access List to an Interface" section on page 16 or the "Where to Go Next" section on page 21 for pointers to modules that describe other ways to use access lists.

Creating an Extended Access List

If you want to filter on anything other than source address, you need to create an extended access list. There are two alternative types of extended access list: named and numbered. Named access lists allow you to identify your access lists with a more intuitive name rather than a number, and they also support more features.

For details on how to filter something other than source or destination address, see the syntax descriptions in the command reference documentation.

Creating a Named Extended Access List

Create a named extended access list if you want to filter on source and destination address, or a combination of addresses and other IP fields.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *name*
4. **remark** *remark*
5. **deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]

6. **remark** *remark*
7. **permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*]] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
8. Repeat some combination of Steps 4 through 7 until you have specified the fields and values on which you want to base your access list.
9. **end**
10. **show ip access-list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *name* **Example:** `Router(config)# ip access-list extended nomarketing`	Defines an extended IP access list using a name and enters extended named access list configuration mode.
Step 4	**remark** *remark* **Example:** `Router(config-ext-nacl)# remark protect server by denying access from the Marketing network`	(Optional) Adds a user-friendly comment about an access list entry. • A remark can precede or follow an access list entry. • In this example, the remark reminds the network administrator that the subsequent entry denies the Sales network access to the interface.
Step 5	**deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# deny ip 172.18.0.0 0.0.255.255 host 172.16.40.10 log`	(Optional) Denies any packet that matches all of the conditions specified in the statement. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255. • Optionally use the keyword **host** *source* to indicate a source and source wildcard of *source* 0.0.0.0 or the abbreviation **host** *destination* to indicate a destination and destination wildcard of *destination* 0.0.0.0. • In this example, packets from all sources are denied access to the destination network 172.18.0.0. Logging messages about packets permitted or denied by the access list are sent to the facility configured by the **logging facility** command (for example, console, terminal, or syslog). That is, any packet that matches the access list will cause an informational logging message about the packet to be sent to the configured facility. The level of messages logged to the console is controlled by the **logging console** command.

	Command or Action	Purpose
Step 6	**remark** *remark* **Example:** `Router(config-ext-nacl)# remark allow TCP from any source to any destination`	(Optional) Adds a user-friendly comment about an access list entry. • A remark can precede or follow an access list entry.
Step 7	**permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# permit tcp any any`	Permits any packet that matches all of the conditions specified in the statement. • Every access list needs at least one permit statement. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255. • In this example, TCP packets are allowed from any source to any destination. • Use the **log-input** keyword to include input interface, source MAC address, or virtual circuit in the logging output.
Step 8	Repeat some combination of Steps 4 through 7 until you have specified the fields and values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 9	**end** **Example:** `Router(config-ext-nacl)# end`	Ends configuration mode and brings the system to privileged EXEC mode.
Step 10	**show ip access-list** **Example:** `Router# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

What to Do Next

The access list you created is not in effect until you apply it to an interface, a vty line, or reference it from a command that uses an access list. See the "Applying the Access List to an Interface" section on page 16 or the "Where to Go Next" section on page 21 for pointers to modules that describe other ways to use access lists.

Creating a Numbered Extended Access List

Create a numbered extended access list if you want to filter on source and destination address, or a combination of addresses and other IP fields, and you prefer not to use a name. Extended IP access lists are numbered 100 to 199 or 2000 to 2699.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **remark** *remark*
4. **access-list** *access-list-number* **permit** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
5. **access-list** *access-list-number* **remark** *remark*
6. **access-list** *access-list-number* **deny** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
7. Repeat some combination of Steps 3 through 6 until you have specified the fields and values on which you want to base your access list.
8. **end**
9. **show ip access-list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **remark** *remark* **Example:** `Router(config)# access-list 107 remark allow Telnet packets from any source to network 173.69.0.0 (headquarters)`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.
Step 4	**access-list** *access-list-number* **permit** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config)# access-list 107 permit tcp any 173.69.0.0 0.0.255.255 eq telnet`	Permits any packet that matches all of the conditions specified in the statement. • Every access list needs at least one **permit** statement; it need not be the first entry. • Extended IP access lists are numbered 100 to 199 or 2000 to 2699. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255. • TCP and other protocols have additional syntax available. See the **access-list** command in the command reference for complete syntax.
Step 5	**access-list** *access-list-number* **remark** *remark* **Example:** `Router(config)# access-list 107 remark deny all other TCP packets`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.

	Command or Action	Purpose
Step 6	**access-list** *access-list-number* **deny** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config)# access-list 107 deny tcp any any`	Denies any packet that matches all of the conditions specified in the statement. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255.
Step 7	Repeat some combination of Steps 3 through 6 until you have specified the fields and values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 8	**end** **Example:** `Router(config)# end`	Ends configuration mode and brings the system to privileged EXEC mode.
Step 9	**show ip access-list** **Example:** `Router# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

Applying the Access List to an Interface

Perform this task to apply an access list to an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface and enters interface configuration mode.
Step 4	**ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**} **Example:** `Router(config-if)# ip access-group noncorp in`	Applies the specified access list to the incoming or outgoing interface. • When you are filtering on source addresses, you typically apply the access list to an incoming interface. • Filtering on source addresses is most efficient when applied near the destination.

What to Do Next

The access list you created is not in effect until you apply it to an interface, a vty line, or reference it from a command that uses an access list. See the "Applying the Access List to an Interface" section on page 16 or the "Where to Go Next" section on page 21 for pointers to modules that describe other ways to use access lists.

Configuration Examples for IP Access Lists

This section contains the following examples of named and numbered, standard and extended IP access lists that are applied to an interface or referenced by a command:

Filtering on Source Address (Hosts): Example

In the following example, the workstation belonging to Jones is allowed access to Ethernet interface 0 and the workstation belonging to Smith is not allowed access:

```
interface ethernet 0
 ip access-group workstations in
!
ip access-list standard workstations
 remark Permit only Jones workstation through
 permit 172.16.2.88
 remark Do not allow Smith workstation through
 deny 172.16.3.13
```

Filtering on Source Address (Subnet): Example

In the following example, the Jones subnet is not allowed access to Ethernet interface 0, but the Main subnet is allowed access:

```
interface ethernet 0
 ip access-group prevention in
!
ip access-list standard prevention
 remark Do not allow Jones subnet through
 deny 172.22.0.0 0.0.255.255
 remark Allow Main subnet
 permit 172.25.0.0 0.0.255.255
```

Filtering on Source Address, Destination Address, and IP Protocols: Example

The following configuration example shows an interface with two access lists, one applied to outgoing packets and one applied to incoming packets. The standard access list named Internet_filter filters outgoing packets on source address. The only packets allowed out the interface must be from source 172.16.3.4.

The extended access list named marketing_group filters incoming packets. The access list permits Telnet packets from any source to network 172.26.0.0 and denies all other TCP packets. It permits any ICMP packets. It denies UDP packets from any source to network 172.26.0 0 on port numbers less than 1024. Finally, the access list denies all other IP packets and performs logging of packets passed or denied by that entry.

```
interface Ethernet0/5
 ip address 172.20.5.1 255.255.255.0
 ip access-group Internet_filter out
 ip access-group marketing_group in
!
ip access-list standard Internet_filter
 permit 172.16.3.4
ip access-list extended marketing_group
 permit tcp any 172.26.0.0 0.0.255.255 eq telnet
 deny tcp any any
 permit icmp any any
 deny udp any 172.26.0.0 0.0.255.255 lt 1024
 deny ip any any
```

Filtering on Source Address (Host and Subnets) Using a Numbered Access List: Example

In the following example, network 10.0.0.0 is a Class A network whose second octet specifies a subnet; that is, its subnet mask is 255.255.0.0. The third and fourth octets of a network 10.0.0.0 address specify a particular host. Using access list 2, the Cisco IOS software would accept one address on subnet 48 and reject all others on that subnet. The last line of the list shows that the software would accept addresses on all other network 10.0.0.0 subnets.

```
interface ethernet 0
 ip access-group 2 in
!
access-list 2 permit 10.48.0.3
access-list 2 deny 10.48.0.0  0.0.255.255
access-list 2 permit 10.0.0.0  0.255.255.255
```

Preventing Telnet Access to a Subnet: Example

In the following example, the Jones subnet is not allowed to Telnet out Ethernet interface 0:

```
interface ethernet 0
 ip access-group telnetting out
!
ip access-list extended telnetting
 remark Do not allow Jones subnet to telnet out
 deny tcp 172.20.0.0 0.0.255.255 any eq telnet
 remark Allow Top subnet to telnet out
 permit tcp 172.33.0.0 0.0.255.255 any eq telnet
```

Filtering on TCP and ICMP Using Port Numbers: Example

In the following example, the first line of the extended access list named goodports permits any incoming TCP connections with destination ports greater than 1023. The second line permits incoming TCP connections to the Simple Mail Transfer Protocol (SMTP) port of host 172.28.1.2. The last line permits incoming ICMP messages for error feedback.

```
interface ethernet 0
 ip access-group goodports in
!
ip access-list extended goodports
 permit tcp any 172.28.0.0 0.0.255.255 gt 1023
 permit tcp any host 172.28.1.2 eq 25
 permit icmp any 172.28.0.0 255.255.255.255
```

Allowing SMTP (E-mail) and Established TCP Connections: Example

Suppose you have a network connected to the Internet, and you want any host on an Ethernet to be able to form TCP connections to any host on the Internet. However, you do not want IP hosts to be able to form TCP connections to hosts on the Ethernet except to the mail (SMTP) port of a dedicated mail host.

SMTP uses TCP port 25 on one end of the connection and a random port number on the other end. The same two port numbers are used throughout the life of the connection. Mail packets coming in from the Internet will have a destination port of 25. Outbound packets will have the port numbers reversed. The

fact that the secure system behind the router always will accept mail connections on port 25 is what makes possible separate control of incoming and outgoing services. The access list can be configured on either the outbound or inbound interface.

In the following example, the Ethernet network is a Class B network with the address 172.18.0.0, and the address of the mail host is 172.18.1.2. The **established** keyword is used only for the TCP protocol to indicate an established connection. A match occurs if the TCP datagram has the ACK or RST bits set, which indicate that the packet belongs to an existing connection.

```
interface ethernet 0
 ip access-group 102 in
!
access-list 102 permit tcp any 172.18.0.0 0.0.255.255 established
access-list 102 permit tcp any host 172.18.1.2 eq 25
```

Preventing Access to the Web By Filtering on Port Name: Example

In the following example, the Winter and Smith workstations are not allowed web access; other hosts on network 172.20.0.0 are allowed web access:

```
interface ethernet 0
 ip access-group no_web out
!
ip access-list extended no_web
 remark Do not allow Winter to browse the web
 deny host 172.20.3.85 any eq http
 remark Do not allow Smith to browse the web
 deny host 172.20.3.13 any eq http
 remark Allow others on our network to browse the web
 permit 172.20.0.0 0.0.255.255 any eq http
```

Filtering on Source Address and Logging the Packets Permitted and Denied: Example

The following example defines access lists 1 and 2, both of which have logging enabled:

```
interface ethernet 0
 ip address 172.16.1.1 255.0.0.0
 ip access-group 1 in
 ip access-group 2 out
!
access-list 1 permit 172.25.0.0 0.0.255.255 log
access-list 1 deny 172.30.0.0 0.0.255.255 log
!
access-list 2 permit 172.27.3.4 log
access-list 2 deny 172.17.0.0 0.0.255.255 log
```

If the interface receives 10 packets from 172.25.7.7 and 14 packets from 172.17.23.21, the first log will look like the following:

```
list 1 permit 172.25.7.7 1 packet
list 2 deny 172.17.23.21 1 packet
```

Five minutes later, the console will receive the following log:

```
list 1 permit 172.25.7.7 9 packets
list 2 deny 172.17.23.21 13 packets
```

Limiting Debug Output: Example

The following example configuration example uses an access list to limit the **debug** command output displayed. Limiting debug output narrows the volume of data to what you are interested in, saving you time and resources.

```
ip access-list idaho
 remark Displays only advertisements for LDP peer in idaho
 permit host 10.0.0.44

Router# debug mpls ldp advertisements peer-acl idaho

tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.17.0.33

tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.16.0.31

tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.22.0.33

tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.1

tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.3

tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.1.33
```

Where to Go Next

This module describes how to create an access list that permits or denies packets based on source or destination address or protocol. However, there are other fields you could filter on, and other ways to use access lists. If you want to create an access list that filters on other fields or if you want to apply an access list to something other than an interface, you should decide what you want to restrict in your network and determine the type of access list that achieves your goal.

See the following table for references to other fields to filter and other ways to use an IP access list.

If you want to...	See
Filter based on IP Options, TCP flags, noncontiguous ports, or TTL value	"Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values" module
Reorder your access list entries	"Refining an IP Access List" module
Limit access list entries to a time of day or week	"Refining an IP Access List" module
Restrict packets with noninitial fragments	"Refining an IP Access List" module
Restrict access to virtual terminal lines	"Controlling Access to a Virtual Terminal Line"
Control routing updates	"Configuring Routing Protocol-Independent Features" module in the *Cisco IOS IP Routing Protocols Configuration Guide*, Release 12.4
Identify or classify traffic for features such as congestion avoidance, congestion management, and priority queuing	"Regulating Packet Flow on a Per-Interface Basis—Using Generic Traffic Shaping" module in the *Quality of Service Solutions Configuration Guide*, Release 12.4

If you want to...	See
Trigger dial-on-demand (DOD) calls	"Preparing to Configure DDR" module in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.4
Configure authentication proxy	"Configuring Authentication Proxy" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
Configure reflexive access lists	"Configuring IP Session Filtering (Reflexive Access Lists)" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
Configure Context-Based Access Control (CBAC)	"Configuring Lock-and-Key Security (Dynamic Access Lists)" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
Configure dynamic access lists	"Configuring Context-Based Access Control" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
Configure TCP Intercept	"Configuring TCP Intercept (Preventing Denial-of-Service Attacks)" module in the *Cisco IOS Security Configuration Guide*, Release 12.4

Additional References

The following sections provide references related to IP access lists.

Related Documents

Related Topic	Document Title
Order of access list entries	"Refining an IP Access List" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
Access list entries based on time of day or week	"Refining an IP Access List" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
Packets with noninitial fragments	"Refining an IP Access List" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
Filtering on IP Options, TCP flags, noncontiguous ports, or TTL values	"Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
Access to virtual terminal lines	"Controlling Access to a Virtual Terminal Line" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
Routing updates and policy routing	"Configuring Routing Protocol-Independent Features" modules in the *Cisco IOS IP Routing Protocols Configuration Guide*, Release 12.4
Traffic identification or classification for features such as congestion avoidance, congestion management, and priority queuing	"Regulating Packet Flow on a Per-Interface Basis—Using Generic Traffic Shaping" module in the *Quality of Service Solutions Configuration Guide*, Release 12.4
Dial-on-demand (DOD) calls	"Preparing to Configure DDR" module in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.4
Authentication proxy	"Configuring Authentication Proxy" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
Reflexive access lists	"Configuring IP Session Filtering (Reflexive Access Lists)" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
Context-Based Access Control (CBAC)	"Configuring Lock-and-Key Security (Dynamic Access Lists)" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
Dynamic access lists	"Configuring Context-Based Access Control" module in the *Cisco IOS Security Configuration Guide*, Release 12.4
TCP Intercept	"Configuring TCP Intercept (Preventing Denial-of-Service Attacks)" module in the *Cisco IOS Security Configuration Guide*, Release 12.4

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	—

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Creating IP Access Lists

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the IP Access List Roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Creating IP Access Lists

Feature Name	Releases	Feature Configuration Information
—	Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Commented IP Access List Entries	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Standard IP Access List Logging	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

First Published: August 18, 2006
Last Updated: August 18, 2006

This module describes how to use an IP access list to filter IP packets that contain certain IP Options, TCP flags, noncontiguous ports, or time-to-live (TTL) values.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Using an IP Access List to Filter Packets" section on page 21.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

Before you perform any of the tasks in this module, you should be familiar with the information in the following modules:

- "IP Access List Overview"
- "Creating an IP Access List and Applying It to an Interface"

How to Create an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

This section includes the following optional tasks:

Filtering Packets That Contain IP Options

The task in this section configures an access list to filter packets that contain IP Options and verifies that the access list has been configured correctly.

IP Options

IP uses four key mechanisms in providing its service: Type of Service, Time to Live, Options, and Header Checksum.

The Options, commonly referred to as IP Options, provide for control functions that are required in some situations but unnecessary for the most common communications. IP Options include provisions for time stamps, security, and special routing.

IP Options may or may not appear in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation. In some environments the security option may be required in all datagrams.

The option field is variable in length. There may be zero or more options. IP Options can have one of two formats:

- Format 1: A single octet of option-type.
- Format 2: An option-type octet, an option-length octet, and the actual option-data octets.

The option-length octet counts the option-type octet, the option-length octet, and the option-data octets.

The option-type octet is viewed as having three fields: a 1-bit copied flag, a 2-bit option class, and a 5-bit option number. These fields form an 8-bit value for the option type field. IP Options are commonly referred to by their 8-bit value.

For a complete list and description of IP Options, refer to RFC 791, *Internet Protocol* at the following URL:

http://www.faqs.org/rfcs/rfc791.html

Benefits of Filtering IP Options

- Filtering of packets that contain IP Options from the network relieves downstream routers and hosts of the load from options packets.
- This feature also minimizes load to the Route Processor (RP) for packets with IP Options that require RP processing on distributed systems. Previously, the packets were always routed to or processed by the RP CPU. Filtering the packets prevents them from impacting the RP.

Restrictions

- The ACL Support for Filtering IP Options feature can be used only with named, extended ACLs.
- Resource Reservation Protocol (RSVP) Multiprotocol Label Switching Traffic Engineering (MPLS TE), Internet Group Management Protocol Version 2 (IGMPV2), and other protocols that use IP Options packets may not function in drop or ignore mode if this feature is configured.
- On most Cisco routers, a packet with IP Options is not switched in hardware, but requires control plane software processing (primarily because there is a need to process the options and rewrite the IP header), so all IP packets with IP Options will be filtered and switched in software.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*sequence-number*] **deny** *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
5. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
6. Repeat Step 4 or Step 5 as necessary.
7. **end**
8. **show ip access-lists** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Router(config)# ip access-list extended mylist1`	Specifies the IP access list by name and enters named access list configuration mode. **Note** The ACL Support for Filtering IP Options feature works only with named, extended ACLs.
Step 4	[*sequence-number*] **deny** *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# deny ip any any option traceroute`	(Optional) Specifies a **deny** statement in named IP access list mode. • This access list happens to use a **deny** statement first, but a **permit** statement could appear first, depending on the order of statements you need. • Use the **option** keyword and *option-value* argument to filter packets that contain a particular IP Option. • In this example, any packet that contains the traceroute IP Option will be filtered out. • Use the **no** *sequence-number* form of this command to delete an entry.
Step 5	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# permit ip any any option security`	Specifies a **permit** statement in named IP access list mode. • In this example, any packet (not already filtered) that contains the security IP Option will be permitted. • Use the **no** *sequence-number* form of this command to delete an entry.
Step 6	Repeat Step 4 or Step 5 as necessary.	Allows you to revise the access list.
Step 7	**end** **Example:** `Router(config-ext-nacl)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
Step 8	**show ip access-lists** *access-list-name* **Example:** `Router# show ip access-lists mylist1`	(Optional) Displays the contents of the IP access list. • Review the output to verify that the access list includes the new entry.

What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

Note To effectively eliminate all packets that contain IP Options, we recommend that you configure the global **ip options drop** command.

Filtering Packets That Contain TCP Flags

The task in this section configures an access list to filter packets that contain TCP flags and verifies that the access list has been configured correctly.

Benefits of Filtering on TCP Flags

The ACL TCP Flags Filtering feature provides a flexible mechanism for filtering on TCP flags. Before Cisco IOS Release 12.3(4)T, an incoming packet was matched as long as any TCP flag in the packet matched a flag specified in the access control entry (ACE). This behavior allows for a security loophole, because packets with all flags set could get past the access control list (ACL). The ACL TCP Flags Filtering feature allows you to select any combination of flags on which to filter. The ability to match on a flag set and on a flag not set gives you a greater degree of control for filtering on TCP flags, thus enhancing security.

Because TCP packets can be sent as false synchronization packets that can be accepted by a listening port, it is recommended that administrators of firewall devices set up some filtering rules to drop false TCP packets.

The ACEs that make up an access list can be configured to detect and drop unauthorized TCP packets by allowing only the packets that have a very specific group of TCP flags set or not set. The ACL TCP Flags Filtering feature gives users a greater degree of packet-filtering control in the following ways:

- Users can select any desired combination of TCP flags on which to filter TCP packets.
- Users can configure ACEs in order to allow matching on a flag that is set, as well as on a flag that is not set.

TCP Flags

Table 1 lists the TCP flags, which are further described in RFC 793, *Transmission Control Protocol*.

Table 1 TCP Flags

TCP Flag	Purpose
ACK	Acknowledge flag—Indicates that the acknowledgment field of a segment specifies the next sequence number the sender of this segment is expecting to receive.
FIN	Finish flag—Used to clear connections.
PSH	Push flag— Indicates the data in the call should be immediately pushed through to the receiving user.
RST	Reset flag—Indicates that the receiver should delete the connection without further interaction.

Table 1 TCP Flags (continued)

TCP Flag	Purpose
SYN	Synchronize flag—Used to establish connections.
URG	Urgent flag—Indicates that the urgent field is meaningful and must be added to the segment sequence number.

Restrictions

- TCP flag filtering can be used only with named, extended ACLs.
- The ACL TCP Flags Filtering feature is supported only for Cisco IOS ACLs.
- Before Cisco IOS Release 12.3(4)T, the following command-line interface (CLI) format could be used to configure a TCP flag-checking mechanism:

  permit tcp any any rst

  The following format that represents the same ACE can be used with Cisco IOS Release 12.3(4)T and later releases:

  permit tcp any any match-any +rst

  Both the CLI formats are accepted; however, if the new keywords **match-all** or **match-any** are chosen, they must be followed by the new flags that are prefixed with "**+**" or "**-**". It is advisable to use only the old format or the new format in a single ACL. You cannot mix and match the old and new CLI formats.

Caution If a router having ACEs with the new syntax format is reloaded with an older version of Cisco IOS software that does not support the ACL TCP Flags Filtering feature, the ACEs will not be applied, leading to possible security loopholes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*sequence-number*] **permit tcp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** | {**match-any** | **match-all**} {**+** | **-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
5. [*sequence-number*] **deny tcp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** | {**match-any** | **match-all**} {**+** | **-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
6. Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.
7. **end**
8. **show ip access-lists** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Router(config)# ip access-list extended kmd1`	Specifies the IP access list by name and enters named access list configuration mode. **Note** The ACL TCP Flags Filtering feature works only with named, extended ACLs.
Step 4	[*sequence-number*] **permit tcp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** | {**match-any** | **match-all**} {**+** | **-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# permit tcp any any match-any +rst`	Specifies a **permit** statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • Use the TCP command syntax of the **permit** command. • Any packet with the RST TCP header flag set will be matched and allowed to pass the named access list kmd1 in Step 3.
Step 5	[*sequence-number*] **deny tcp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** | {**match-any** | **match-all**} {**+** | **-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# deny tcp any any match-all -ack -fin`	(Optional) Specifies a **deny** statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • Use the TCP command syntax of the **deny** command. • Any packet that does not have the ACK flag set, and also does not have the FIN flag set, will not be allowed to pass the named access list kmd1 in Step 3. • See the **deny** (IP) command for additional command syntax to permit upper-layer protocols (ICMP, IGMP, TCP, and UDP).
Step 6	Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise the access list.

	Command or Action	Purpose
Step 7	**end** **Example:** `Router(config-ext-nacl)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
Step 8	**show ip access-lists** *access-list-name* **Example:** `Router# show ip access-lists kmd1`	(Optional) Displays the contents of the IP access list. • Review the output to confirm that the access list includes the new entry.

What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

Configuring an Access Control Entry with Noncontiguous Ports

Perform this task to create access list entries that use noncontiguous TCP or UDP port numbers. Although this task uses TCP ports, you could use the UDP syntax of the **permit** and **deny** commands to filter noncontiguous UDP ports.

Although this task uses a **permit** command first, use the **permit** and **deny** commands in the order that achieves your filtering goals.

Benefits of Using the ACL—Named ACL Support for Noncontiguous Ports on an Access Control Entry Feature

This feature greatly reduces the number of ACEs required in an access control list to handle multiple entries for the same source address, destination address, and protocol. If you maintain large numbers of ACEs, we recommend that you use this feature to consolidate existing groups of access list entries wherever it is possible and also when you create new access list entries. When you configure access list entries with noncontiguous ports, you will have fewer access list entries to maintain.

Restrictions

The ACL—Named ACL Support for Noncontiguous Ports on an Access Control Entry feature can be used only with named, extended ACLs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*sequence-number*] **permit tcp** *source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** | {**match-any** | **match-all**} {**+** | **-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

5. [*sequence-number*] **deny tcp** *source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** | {**match-any** | **match-all**} {**+** | **-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
6. Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.
7. **end**
8. **show ip access-lists** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Router(config)# ip access-list extended kmd1`	Specifies the IP access list by name and enters named access list configuration mode.
Step 4	[*sequence-number*] **permit tcp** *source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established**	{**match-any**	**match-all**} {**+**	**-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# permit tcp any eq telnet ftp any eq 450 679`	Specifies a **permit** statement in named IP access list configuration mode. • Operators include **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). • If the operator is positioned after the source and source-wildcard arguments, it must match the source port. If the operator is positioned after the destination and destination-wildcard arguments, it must match the destination port. • The **range** operator requires two port numbers. You can configure up to 10 ports after the **eq** and **neq** operators. All other operators require one port number. • To filter UDP ports, use the UDP syntax of this command.

	Command or Action	Purpose
Step 5	[*sequence-number*] **deny tcp** *source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** | {**match-any** | **match-all**} {**+** | **-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# deny tcp any neq 45 565 632`	(Optional) Specifies a **deny** statement in named access list configuration mode. • Operators include **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). • If the *operator* is positioned after the *source* and *source-wildcard* arguments, it must match the source port. If the *operator* is positioned after the *destination* and *destination-wildcard* arguments, it must match the destination port. • The **range** operator requires two port numbers. You can configure up to 10 ports after the **eq** and **neq** operators. All other operators require one port number. • To filter UDP ports, use the UDP syntax of this command.
Step 6	Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise the access list.
Step 7	**end** **Example:** `Router(config-ext-nacl)# end`	(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.
Step 8	**show ip access-lists** *access-list-name* **Example:** `Router# show ip access-lists kmd1`	(Optional) Displays the contents of the access list. • Review the output to verify that the access list displays the new entries that you created.

Consolidating Access List Entries with Noncontiguous Ports into One Access List Entry

Perform this task to consolidate a group of access list entries with noncontiguous ports into one access list entry.

Although this task uses TCP ports, you could use the UDP syntax of the **permit** and **deny** commands to filter noncontiguous UDP ports.

Although this task uses a **permit** command first, use the **permit** and **deny** commands in the order that achieves your filtering goals.

SUMMARY STEPS

1. **enable**
2. **show ip access-lists** *access-list-name*
3. **configure terminal**
4. **ip access-list extended** *access-list-name*

5. **no** [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
6. [*sequence-number*] **permit** *protocol source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator port* [*port*]] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
7. Repeat Steps 5 and 6 as necessary, adding **permit** or **deny** statements to consolidate access list entries where possible. Use the **no** *sequence-number* command to delete an entry.
8. **end**
9. **show ip access-lists** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip access-lists** *access-list-name* **Example:** `Router# show ip access-lists mylist1`	(Optional) Displays the contents of the IP access list. • Review the output to see if you can consolidate any access list entries.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**ip access-list extended** *access-list-name* **Example:** `Router(config)# ip access-list extended mylist1`	Specifies the IP access list by name and enters named access list configuration mode.
Step 5	**no** [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# no 10`	Removes the redundant access list entry that can be consolidated. • Repeat this step to remove entries to be consolidated because only the port numbers differ. • After this step is repeated to remove the access list entries 20, 30, and 40, for example, those entries are removed because they will be consolidated into one **permit** statement. • If a *sequence-number* is specified, the rest of the command syntax is optional.

	Command or Action	Purpose
Step 6	[*sequence-number*] **permit** *protocol source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator port* [*port*]] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# permit tcp any neq 45 565 632 any eq 23 45 34 43`	Specifies a **permit** statement in named access list configuration mode. • In this instance, a group of access list entries with noncontiguous ports was consolidated into one **permit** statement. • You can configure up to 10 ports after the **eq** and **neq** operators.
Step 7	Repeat Steps 5 and 6 as necessary, adding **permit** or **deny** statements to consolidate access list entries where possible. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise the access list.
Step 8	**end** **Example:** `Router(config-std-nacl)# end`	(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.
Step 9	**show ip access-lists** *access-list-name* **Example:** `Router# show ip access-lists mylist1`	(Optional) Displays the contents of the access list. • Review the output to verify that the redundant access list entries have been replaced with your new consolidated entries.

What To Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

Filtering Packets Based on TTL Value

Because access lists are very flexible, it is not possible to define only one combination of **permit** and **deny** commands to filter packets based on the TTL value. This task illustrates just one example that achieves TTL filtering. Configure the appropriate **permit** and **deny** statements that will accomplish your filtering plan.

How Filtering on TTL Works

IP extended named and numbered access lists may filter on the TTL value of packets arriving at or leaving an interface. Packets with any possible TTL values 0 through 255 may be permitted or denied (filtered). Like filtering on other fields, such as source or destination address, the **ip access-group** command specifies **in** or **out**, which makes the access list ingress or egress and applies it to incoming or outgoing packets, respectively. The TTL value is checked in conjunction with the specified protocol, application, and any other settings in the access list entry, and all conditions must be met.

Special Handling for Packets with TTL or 0 or 1 Arriving on Ingress Interface

The software switching paths—distributed Cisco Express Forwarding (dCEF), CEF, fast switching, and process switching—will usually permit or discard the packets based on the access list statements. However, when the TTL value of packets arriving on an *ingress* interface have a TTL of 0 or 1, special

handling is required. The packets with a TTL of 0 or 1 get sent to the process level before the ingress access list is checked in CEF, dCEF, or fast switching paths. The ingress access list is applied to packets with TTL values 2 through 255 and a permit or deny decision is made.

Packets with a TTL value of 0 or 1 are sent to the process level because they will never be forwarded out of the device; the process level must check whether each packet is destined for the router or not and whether an Internet Control Message Protocol (ICMP) TTL Expire message needs to be sent back or not. This means that even if an ACL with TTL value 0 or 1 filtering is configured on the ingress interface with the intention to drop packets with a TTL of 0 or 1, the dropping of the packets will not happen in the faster paths. It will instead happen in the process level when the process applies the ACL. This is also true for hardware switching platforms. Packets with TTL 0 or 1 are sent to the process level of the route processor (RP) or Multilayer Switch Feature Card (MSFC).

On egress interfaces, access list filtering on TTL work just like other access list features. The check will happen in the fastest switching path enabled in the device. This is because the faster switching paths handle all the TTL values (0-255) equally on the egress interface.

Control Plane Policing for Filtering TTL Values 0 and 1

The special behavior for packets with a TTL of 0 or 1 results in higher CPU usage for the device. If you are filtering on TTL value 0 or 1, you should use control plane policing (CPP) to protect the CPU from being overwhelmed. In order to leverage CPP, you must configure an access list especially for filtering TTL values 0 and 1 and apply the access list through CPP. This access list will be a separate access list from any interface access lists. Because CPP works for the entire system, not just on individual interfaces, you would need to configure only one such special access list for the entire device. This task is described in the section "Enabling Control Plane Policing to Filter on TTL Values 0 and 1" section on page 15.

Benefits of Filtering on TTL

- Filtering on TTL provides a way to control which packets are allowed to reach the router or prevented from reaching the router. By looking at your network layout, you can choose whether to accept or deny packets from a certain router based on how many hops away it is. For example, in a small network, you can deny packets from a location more than three hops away. Filtering on TTL allows you to validate if the traffic originated from a neighboring device, as follows. You can accept only packets that reach you in one hop, for example, by accepting only packets with a TTL of one less than the initial TTL value of a particular protocol.
- Many control plane protocols communicate only with their neighbors, but receive packets from everyone. By applying to receiving routers an access list that filters on TTL, you can block unwanted packets.
- The Cisco IOS software sends all packets with a TTL of 0 or 1 to the process level to be processed. The device must then send an ICMP TTL expire message to the source. By filtering packets that have a TTL of 0 through 2, you can reduce the load on the process level.

Restrictions

- When the access list specifies the operation EQ or NEQ, routers running Cisco IOS Release 12.2S can have that access list specify up to ten TTL values. However, for Release 12.0S, only one TTL value can be specified.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**log**] [**time-range** *time-range-name*] [**fragments**]
5. Continue to add **permit** or **deny** statements to achieve the filtering you want.
6. **exit**
7. **interface** *type number*
8. **ip access-group** *access-list-name* {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip access-list extended` *access-list-name* **Example:** `Router(config)# ip access-list extended ttlfilter`	Defines an IP access list by name. • An access list that filters on TTL value must be an extended access list.
Step 4	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# permit ip host 172.16.1.1 any ttl lt 2`	Sets conditions to allow a packet to pass a named IP access list. • Every access list must have at least one **permit** statement. • This example permits packets from source 172.16.1.1 to any destination with a TTL value less than 2.
Step 5	Continue to add **permit** or **deny** statements to achieve the filtering you want.	—
Step 6	`exit` **Example:** `Router(config-ext-nacl)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.

	Command or Action	Purpose
Step 7	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Configures an interface type and enters interface configuration mode.
Step 8	**ip access-group** *access-list-name* {**in** | **out**} **Example:** `Router(config-if)# ip access-group ttlfilter in`	Applies the access list to an interface.

Enabling Control Plane Policing to Filter on TTL Values 0 and 1

Perform this task if you want to filter IP packets based on a TTL value of 0 or 1 and you want to protect the CPU from being overwhelmed. This task configures an access list for classification on TTL 0 and 1, configures Modular QoS CLI (MQC), and applies the policy map to the control plane. Any packets that pass the access list are dropped. This special access list is separate from any interface access lists.

Because access lists are very flexible, it is not possible to define only one combination of **permit** and **deny** commands to filter packets based on the TTL value. This task illustrates just one example that achieves TTL filtering. Configure the appropriate **permit** and **deny** statements that will accomplish your filtering plan.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* **ttl** *operator value*
5. Continue to add **permit** or **deny** statements to achieve the filtering you want.
6. **exit**
7. class-map *class-map-name* [match-all | match-any]
8. match access-group {*access-group* | name *access-group-name*}
9. **exit**
10. policy-map *policy-map-name*
11. class {*class-name* | class-default}
12. **drop**
13. **exit**
14. **exit**
15. **control-plane**
16. service-policy {input | output} *policy-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Router(config)# ip access-list extended ttlfilter`	Defines an IP access list by name. • An access list that filters on a TTL value must be an extended access list.
Step 4	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* **ttl** *operator* value **Example:** `Router(config-ext-nacl)# permit ip host 172.16.1.1 any ttl lt 2`	Sets conditions to allow a packet to pass a named IP access list. • Every access list must have at least one **permit** statement. • This example permits packets from source 172.16.1.1 to any destination with a TTL value less than 2.
Step 5	Continue to add **permit** or **deny** statements to achieve the filtering you want.	The packets that pass the access list will be dropped.
Step 6	**exit** **Example:** `Router(config-ext-nacl)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 7	**class-map** *class-map-name* [**match-all** | **match-any**] **Example:** `Router(config)# class-map acl-filtering`	Creates a class map to be used for matching packets to a specified class.
Step 8	**match access-group** {*access-group* | **name** *access-group-name*} **Example:** `Router(config-cmap)# match access-group name ttlfilter`	Configures the match criteria for a class map on the basis of the specified access control list.
Step 9	**exit** **Example:** `Router(config-cmap)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.

	Command or Action	Purpose
Step 10	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map acl-filter`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
Step 11	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class acl-filter-class`	Specifies the name of the class whose policy you want to create or change or to specify the default class (commonly known as the class-default class) before you configure its policy.
Step 12	**drop** **Example:** `Router(config-pmap-c)# drop`	Configures a traffic class to discard packets belonging to a specific class.
Step 13	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 14	**exit** **Example:** `Router(config-pmap)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 15	**control-plane** **Example:** `Router(config)# control-plane`	Associates or modifies attributes or parameters that are associated with the control plane of the device.
Step 16	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-cp)# service-policy input acl-filter`	Attaches a policy map to a control plane for aggregate control plane services.

Configuration Examples for Filtering IP Options, TCP Flags, Noncontiguous Ports, and TTL Values

This section provides the following configuration examples:

Filtering Packets That Contain IP Options: Example

The following example shows an extended access list named mylist2 that contains access list entries (ACEs) that are configured to permit TCP packets only if they contain the IP Options that are specified in the ACEs:

```
ip access-list extended mylist2
 10 permit ip any any option eool
 20 permit ip any any option record-route
 30 permit ip any any option zsu
 40 permit ip any any option mtup
```

The **show access-list** command has been entered to show how many packets were matched and therefore permitted:

```
Router# show ip access-list mylist2

Extended IP access list test
10 permit ip any any option eool (1 match)
20 permit ip any any option record-route (1 match)
30 permit ip any any option zsu (1 match)
40 permit ip any any option mtup (1 match)
```

Filtering Packets That Contain TCP Flags: Example

The following access list allows TCP packets only if the TCP flags ACK and SYN are set and the FIN flag is not set:

```
ip access-list extended aaa
 permit tcp any any match-all +ack +syn -fin
 end
```

The **show access-list** command has been entered to display the ACL:

```
Router# show access-list aaa

Extended IP access list aaa
 10 permit tcp any any match-all +ack +syn -fin
```

Creating an Access List Entry with Noncontiguous Ports: Example

The following access list entry can be created because up to ten ports can be entered after the **eq** and **neq** operators:

```
ip access-list extended aaa
 permit tcp any eq telnet ftp any eq 23 45 34
 end
```

Enter the **show access-lists** command to display the newly created access list entry.

```
Router# show access-lists aaa

Extended IP access list aaa
 10 permit tcp any eq telnet ftp any eq 23 45 34
```

Consolidating Some Existing Access List Entries into One Access List Entry with Noncontiguous Ports: Example

The **show access-lists** command is used to display a group of access list entries for the access list named abc:

```
Router# show access-lists abc

Extended IP access list abc
 10 permit tcp any eq telnet any eq 450
 20 permit tcp any eq telnet any eq 679
 30 permit tcp any eq ftp any eq 450
 40 permit tcp any eq ftp any eq 679
```

Because the entries are all for the same **permit** statement and simply show different ports, they can be consolidated into one new access list entry. The following example shows the removal of the redundant access list entries and the creation of a new access list entry that consolidates the previously displayed group of access list entries:

```
ip access-list extended abc
 no 10
 no 20
 no 30
 no 40
 permit tcp any eq telnet ftp any eq 450 679
 end
```

When the **show access-lists** command is reentered, the consolidated access list entry is displayed:

```
Router# show access-lists abc

Extended IP access list abc
 10 permit tcp any eq telnet ftp any eq 450 679
```

Filtering on TTL Value: Example

The following access list filters IP packets containing type of service (ToS) level 3 with TTL values 10 and 20. It also filters IP packets with a TTL greater than 154 and applies that rule to noninitial fragments. It permits IP packets with a precedence level of flash and a TTL not equal to 1, and it sends log messages about such packets to the console. All other packets are denied.

```
ip access-list extended incomingfilter
 deny ip any any tos 3 ttl eq 10 20
 deny ip any any ttl gt 154 fragments
 permit ip any any precedence flash ttl neq 1 log
!
interface ethernet 0
ip access-group incomingfilter in
```

Control Plane Policing to Filter on TTL Values 0 and 1: Example

The following example configures a traffic class called acl-filter-class for use in a policy map called acl-filter. An access list permits IP packets from any source having a TTL of 0 or 1. Any packets matching the access list are dropped. The policy map is attached to the control plane.

```
ip access-list extended ttlfilter
 permit ip any any ttl eq 0 1
class-map acl-filter-class
 match access-group name ttlfilter
policy-map acl-filter
 class acl-filter-class
 drop
control-plane
 service-policy input acl-filter
```

Additional References

The following sections provide references related to IP access list filtering described in this module.

Related Documents

Related Topic	Document Title
Configuring the router to drop or ignore packets containing IP Options by using the **no ip options** command.	"IP Options Selective Drop" module in *Cisco IOS IP Application Services*, Release 12.3(4)T.
QoS commands	*Cisco IOS Quality of Service Solutions Command Reference, Release 12.4*

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 791	Internet Protocol http://www.faqs.org/rfcs/rfc791.html
RFC 793	*Transmission Control Protocol*
RFC 1393	*Traceroute Using an IP Option*

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Using an IP Access List to Filter Packets

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

Feature Name	Releases	Feature Configuration Information
ACL Support for Filtering IP Options	12.3(4)T 12.2(25)S	This feature allows you to filter packets having IP Options, in order to prevent routers from becoming saturated with spurious packets. See the following sections: • Filtering Packets That Contain IP Options, page 2 • Filtering Packets That Contain IP Options: Example, page 18
ACL TCP Flags Filtering	12.3(4)T 12.2(25)S	This feature provides a flexible mechanism for filtering on TCP flags. Before Cisco IOS Release 12.3(4)T, an incoming packet was matched as long as any TCP flag in the packet matched a flag specified in the access control entry (ACE). This behavior allows for a security loophole, because packets with all flags set could get past the access control list (ACL). The ACL TCP Flags Filtering feature allows you to select any combination of flags on which to filter. The ability to match on a flag set and on a flag not set gives you a greater degree of control for filtering on TCP flags, thus enhancing security. See the following sections: • Filtering Packets That Contain TCP Flags, page 5 • Filtering Packets That Contain TCP Flags: Example, page 18
ACL—Named ACL Support for Noncontiguous Ports on an Access Control Entry	12.3(7)T 12.2(25)S	This feature allows you to specify noncontiguous ports in a single access control entry, which greatly reduces the number of entries required in an access control list when several entries have the same source address, destination address, and protocol, but differ only in the ports. See the following sections: • Configuring an Access Control Entry with Noncontiguous Ports, page 8 • Consolidating Access List Entries with Noncontiguous Ports into One Access List Entry, page 10 • Creating an Access List Entry with Noncontiguous Ports: Example, page 18 • Consolidating Some Existing Access List Entries into One Access List Entry with Noncontiguous Ports: Example, page 19

Table 2 Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values (continued)

Feature Name	Releases	Feature Configuration Information
ACL Support for Filtering on TTL Value	12.4(2)T	Customers may use extended IP access lists (named or numbered) to filter packets based on their time-to-live (TTL) value, from 0 to 255. This filtering enhances a customer's control over which packets reach a router. See the following sections: • Filtering Packets Based on TTL Value, page 12 • Enabling Control Plane Policing to Filter on TTL Values 0 and 1, page 15 • Filtering on TTL Value: Example, page 19 • Control Plane Policing to Filter on TTL Values 0 and 1: Example, page 20
ACL - Named ACL Support for Noncontiguous Ports on an Access Control Entry	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
ACL - TCP Flags Filtering	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
ACL Support for Filtering IP Options	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Selective Drop/ Ignore of IP Options	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Refining an IP Access List

First Published: August 18, 2006
Last Updated: August 18, 2006

There are several ways to refine an access list while or after you create it. You can change the order of the entries in an access list or add entries to an access list. You can restrict access list entries to a certain time of day or week, or achieve finer granularity when filtering packets by filtering noninitial fragments of packets.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Refining an IP Access List" section on page 19.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Prerequisites for Refining an IP Access List

Before you perform any of the tasks in this module, you should be familiar with the concepts in the "IP Access List Overview" module.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About Refining an IP Access List

You should understand the following concept before configuring an IP access list with sequence numbers:

Access List Sequence Numbers

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If you wanted to insert an entry in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

Sequence numbers allow users to add access list entries and resequence them. When you add a new entry, you specify the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry.

How to Refine an IP Access List

The tasks in this module provide you with various ways to refine an access list if you did not already do so while you were creating it. You can change the order of the entries in an access list, add entries to an access list, restrict access list entries to a certain time of day or week, or achieve finer granularity when filtering packets by filtering on noninitial fragments of packets.

This section includes the following tasks:

Revising an Access List Using Sequence Numbers

Perform this task if you want to add entries to an existing access list, change the order of entries, or simply number the entries in an access list to accommodate future changes.

Note Remember that if you want to delete an entry from an access list, you can simply use the **no deny** or **no permit** form of the command, or the **no** *sequence-number* command if the statement already has a sequence number.

Benefits of Access List Sequence Numbers

An access list sequence number is a number at the beginning of a **permit** or **deny** command in an access list. The sequence number determines the order that the entry appears in the access list. The ability to apply sequence numbers to IP access list entries simplifies access list changes.

Prior to having sequence numbers, users could only add access list entries to the end of an access list; therefore, needing to add statements anywhere except the end of the list required reconfiguring the entire access list. There was no way to specify the position of an entry within an access list. If a user wanted to insert an entry (statement) in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

This feature allows users to add sequence numbers to access list entries and resequence them. When a user adds a new entry, the user chooses the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry. Sequence numbers make revising an access list much easier.

Sequence Numbering Behavior

- For backward compatibility with previous releases, if entries with no sequence numbers are applied, the first entry is assigned a sequence number of 10, and successive entries are incremented by 10. The maximum sequence number is 2147483647. If the generated sequence number exceeds this maximum number, the following message is displayed:

  ```
  Exceeded maximum sequence number.
  ```

- If the user enters an entry without a sequence number, it is assigned a sequence number that is 10 greater than the last sequence number in that access list and is placed at the end of the list.
- If the user enters an entry that matches an already existing entry (except for the sequence number), then no changes are made.
- If the user enters a sequence number that is already present, the following error message is generated:

  ```
  Duplicate sequence number.
  ```

- If a new access list is entered from global configuration mode, then sequence numbers for that access list are generated automatically.
- Distributed support is provided so that the sequence numbers of entries in the Route Processor (RP) and line card are in synchronization at all times.
- Sequence numbers are not nvgened. That is, the sequence numbers themselves are not saved. In the event that the system is reloaded, the configured sequence numbers revert to the default sequence starting number and increment. The function is provided for backward compatibility with software releases that do not support sequence numbering.
- This feature works with named and numbered, standard and extended IP access lists.

Restrictions

- Access list sequence numbers do not support dynamic, reflexive, or firewall access lists.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list resequence** *access-list-name starting-sequence-number increment*
4. **ip access-list** {**standard** | **extended**} *access-list-name*
5. *sequence-number* **permit** *source source-wildcard*

or

sequence-number **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

6. *sequence-number* **deny** *source source-wildcard*

   or

   sequence-number **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

7. Repeat Step 5 and Step 6 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.
8. **end**
9. **show ip access-lists** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list resequence** *access-list-name starting-sequence-number increment* **Example:** `Router(config)# ip access-list resequence kmd1 100 15`	Resequences the specified IP access list using the starting sequence number and the increment of sequence numbers. • This example resequences an access list named kmd1. The starting sequence number is 100 and the increment is 15.
Step 4	**ip access-list** {**standard** | **extended**} *access-list-name* **Example:** `Router(config)# ip access-list standard xyz123`	Specifies the IP access list by name and enters named access list configuration mode. • If you specify **standard**, make sure you specify subsequent **permit** and **deny** statements using the standard access list syntax. • If you specify **extended**, make sure you specify subsequent **permit** and **deny** statements using the extended access list syntax.
Step 5	*sequence-number* **permit** *source source-wildcard* or *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-std-nacl)# 105 permit 10.5.5.5 0.0.0.255`	Specifies a permit statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the **permit** (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be `Router(config-ext-nacl)#` and you would use the extended **permit** command syntax.

	Command or Action	Purpose
Step 6	*sequence-number* **deny** *source source-wildcard* or *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-std-nacl)# 110 deny 10.6.6.7 0.0.0.255`	(Optional) Specifies a deny statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the **deny** (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be `Router(config-ext-nacl)#` and you would use the extended **deny** command syntax.
Step 7	Repeat Step 5 and Step 6 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise the access list.
Step 8	**end** **Example:** `Router(config-std-nacl)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
Step 9	**show ip access-lists** *access-list-name* **Example:** `Router# show ip access-lists xyz123`	(Optional) Displays the contents of the IP access list. • Review the output to see that the access list includes the new entry.

Examples

The following is sample output from the **show ip access-lists** command when the **xyz123** access list is specified.

```
Router# show ip access-lists xyz123

Standard IP access list xyz123
100 permit 10.4.4.0, wildcard bits 0.0.0.255
105 permit 10.5.5.5, wildcard bits 0.0.0.255
115 permit 10.0.0.0, wildcard bits 0.0.0.255
130 permit 10.5.5.0, wildcard bits 0.0.0.255
145 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Restricting an Access List Entry to a Time of Day or Week

By default, access list statements are always in effect once they are applied. However, you can define the times of the day or week that **permit** or **deny** statements are in effect by defining a time range, and then referencing the time range by name in an individual access list statement. IP and Internetwork Packet Exchange (IPX) named or numbered extended access lists can use time ranges.

Benefits of Time Ranges

Benefits and possible uses of time ranges include the following:

- The network administrator has more control over permitting or denying a user access to resources. These resources could be an application (identified by an IP address/mask pair and a port number), policy routing, or an on-demand link (identified as interesting traffic to the dialer).
- Network administrators can set time-based security policy, including the following:
  - Perimeter security using the Cisco IOS Firewall feature set or access lists
  - Data confidentiality with Cisco Encryption Technology or IP Security Protocol (IPSec)
- Policy-based routing (PBR) and queueing functions are enhanced.
- When provider access rates vary by time of day, it is possible to automatically reroute traffic cost effectively.
- Service providers can dynamically change a committed access rate (CAR) configuration to support the quality of service (QoS) service level agreements (SLAs) that are negotiated for certain times of day.
- Network administrators can control logging messages. Access list entries can log traffic at certain times of the day, but not constantly. Therefore, administrators can simply deny access without needing to analyze many logs generated during peak hours.

Distributed Time-Based Access Lists

Before the introduction of the Distributed Time-Based Access Lists feature, time-based access lists were not supported on line cards for the Cisco 7500 series routers. If time-based access lists were configured, they behaved as normal access lists. If an interface on a line card were configured with a time-based access list, the packets switched into the interface were not distributed switched through the line card, but were forwarded to the Route Processor for processing.

The Distributed Time-Based Access Lists feature allows packets destined for an interface configured with a time-based access list to be distributed switched through the line card.

For this functionality to work, the software clock must remain synchronized between the Route Processor and the line card. This synchronization occurs through an exchange of interprocess communications (IPC) messages from the Route Processor to the line card. When a time range or a time-range entry is changed, added, or deleted, an IPC message is sent by the Route Processor to the line card.

There is no difference between how the user configures a time-based access list and a distributed time-based access list.

Prerequisites

The time range relies on the software clock of the routing device. For the time range feature to work the way you intend, you need a reliable clock source. We recommend that you use Network Time Protocol (NTP) to synchronize the software clock of the routing device.

Restrictions

The Distributed Time-Based Access Lists feature is supported on Cisco 7500 series routers with a Versatile Interface Processor (VIP) enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **time-range** *time-range-name*
4. **periodic** *days-of-the-week hh*:*mm* **to** [*days-of-the-week*] *hh*:*mm*
5. Repeat Step 4 if you want more than one period of time applied to an access list statement.
6. **absolute** [**start** *time date*] [**end** *time date*]
7. **exit**
8. Repeat Steps 3 through 7 if you want different time ranges to apply to **permit** or **deny** statements.
9. **ip access-list extended** *name*
10. **deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] **time-range** *time-range-name*
11. **permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] **time-range** *time-range-name*
12. Optionally repeat some combination of Steps 10 and 11 until you have specified the values on which you want to base your access list.
13. **end**
14. **show ip access-list**
15. **show time-range**
16. **show time-range ipc**
17. **clear time-range ipc**
18. debug time-range ipc

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**time-range** *time-range-name* **Example:** `Router(config)# time-range limit_http`	Defines a time range and enters time-range configuration mode. • The name cannot contain a space or quotation mark, and must begin with a letter. • Multiple time ranges can occur in a single access list.
Step 4	**periodic** *days-of-the-week hh:mm* **to** [*days-of-the-week*] *hh:mm* **Example:** `Router(config-time-range)# periodic Monday 6:00 to Wednesday 19:00`	(Optional) Specifies a recurring (weekly) time range. • The first occurrence of *days-of-the-week* is the starting day or day of the week that the associated time range is in effect. The second occurrence is the ending day or day of the week the associated statement is in effect. • The *days-of-the-week* argument can be any single day or combinations of days: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. Other possible values are: – **daily**—Monday through Sunday – **weekdays**—Monday through Friday – **weekend**—Saturday and Sunday • If the ending days of the week are the same as the starting days of the week, they can be omitted. • The first occurrence of *hh:mm* is the starting hours:minutes that the associated time range is in effect. The second occurrence is the ending hours:minutes the associated statement is in effect. • The hours:minutes are expressed in a 24-hour clock. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m.
Step 5	Repeat Step 4 if you want more than one period of time applied to an access list statement.	(Optional) Multiple **periodic** commands are allowed in a time range.

	Command or Action	Purpose
Step 6	**absolute** [**start** *time date*] [**end** *time date*] **Example:** `Router(config-time-range)# absolute start 6:00 1 August 2005 end 18:00 31 October 2005`	(Optional) Specifies an absolute time when a time range is in effect. • Only one **absolute** command is allowed in a time range. • The time is expressed in 24-hour notation, in the form of hours:minutes. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m. The date is expressed in the format *day month year*. The minimum start is 00:00 1 January 1993. If no start time and date are specified, the **permit** or **deny** statement is in effect immediately. • Absolute time and date that the **permit** or **deny** statement of the associated access list is no longer in effect. Same time and date format as described for the **start** keyword. The end time and date must be after the start time and date. The maximum end time is 23:59 31 December 2035. If no end time and date are specified, the associated **permit** or **deny** statement is in effect indefinitely.
Step 7	**exit** **Example:** `Router(config-time-range)# exit`	Exits to the next highest mode.
Step 8	Repeat Steps 3 through 7 if you want different time ranges to apply to **permit** or **deny** statements.	—
Step 9	**ip access-list extended** *name* **Example:** `Router(config)# ip access-list extended autumn`	Defines an extended IP access list using a name and enters extended named access list configuration mode.
Step 10	**deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] **time-range** *time-range-name* **Example:** `Router(config-ext-nacl)# deny tcp 172.16.22.23 any eq http time-range limit_http`	(Optional) Denies any packet that matches all of the conditions specified in the statement. • Specify the time range you created in Step 3. • In this example, one host is denied HTTP access during the time defined by the time range called “limit_http.”
Step 11	**permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] **time-range** *time-range-name* **Example:** `Router(config-ext-nacl)# permit tcp any any eq http time-range limit_http`	Permits any packet that matches all of the conditions specified in the statement. • You can specify the time range you created in Step 3 or in a different instance of Step 3, depending on whether you want the time ranges for your statements to be the same or different. • In this example, all other sources are given access to HTTP during the time defined by the time range called “limit_http.”

	Command or Action	Purpose
Step 12	Optionally repeat some combination of Steps 10 and 11 until you have specified the values on which you want to base your access list.	—
Step 13	**end** **Example:** `Router(config-ext-nacl)# end`	Ends configuration mode and returns the system to privileged EXEC mode.
Step 14	**show ip access-list** **Example:** `Router# show ip access-list`	(Optional) Displays the contents of all current IP access lists.
Step 15	**show time-range** **Example:** `Router# show time-range`	(Optional) Displays the time ranges that are set.
Step 16	**show time-range ipc** **Example:** `Router# show time-range ipc`	(Optional) Displays the statistics about the time-range IPC messages between the Route Processor and line card on the Cisco 7500 series router.
Step 17	**clear time-range ipc** **Example:** `Router# clear time-range ipc`	(Optional) Clears the time-range IPC message statistics and counters between the Route Processor and line card on the Cisco 7500 series router.
Step 18	**debug time-range ipc** **Example:** `Router# debug time-range ipc`	(Optional) Enables debugging output for monitoring the time-range IPC messages between the Route Processor and line card on the Cisco 7500 series router.

What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

Filtering Noninitial Fragments of Packets

Filter noninitial fragments of packets with an extended access list if you want to block more of the traffic you intended to block, not just the initial fragment of such packets. You should first understand the following concepts.

Benefits of Filtering Noninitial Fragments of Packets

If the **fragments** keyword is used in additional IP access list entries that deny fragments, the fragment control feature provides the following benefits:

Additional Security

You are able to block more of the traffic you intended to block, not just the initial fragment of such packets. The unwanted fragments no longer linger at the receiver until the reassembly timeout is reached because they are blocked before being sent to the receiver. Blocking a greater portion of unwanted traffic improves security and reduces the risk from potential hackers.

Reduced Cost

By blocking unwanted noninitial fragments of packets, you are not paying for traffic you intended to block.

Reduced Storage

By blocking unwanted noninitial fragments of packets from ever reaching the receiver, that destination does not have to store the fragments until the reassembly timeout period is reached.

Expected Behavior Is Achieved

The noninitial fragments will be handled in the same way as the initial fragment, which is what you would expect. There are fewer unexpected policy routing results and fewer fragments of packets being routed when they should not be.

Access List Processing of Fragments

The behavior of access list entries regarding the use or lack of use of the **fragments** keyword can be summarized as follows:

If the Access-List Entry Has...	Then...
...no **fragments** keyword (the default), and assuming all of the access-list entry information matches,	For an access list entry that contains only Layer 3 information: • The entry is applied to nonfragmented packets, initial fragments, and noninitial fragments. For an access list entry that contains Layer 3 and Layer 4 information: • The entry is applied to nonfragmented packets and initial fragments. – If the entry is a **permit** statement, then the packet or fragment is permitted. – If the entry is a **deny** statement, then the packet or fragment is denied. • The entry is also applied to noninitial fragments in the following manner. Because noninitial fragments contain only Layer 3 information, only the Layer 3 portion of an access list entry can be applied. If the Layer 3 portion of the access list entry matches, and – If the entry is a **permit** statement, then the noninitial fragment is permitted. – If the entry is a **deny** statement, then the next access list entry is processed. **Note** The **deny** statements are handled differently for noninitial fragments versus nonfragmented or initial fragments.
...the **fragments** keyword, and assuming all of the access-list entry information matches,	The access list entry is applied only to noninitial fragments. The **fragments** keyword cannot be configured for an access list entry that contains any Layer 4 information.

Be aware that you should not add the **fragments** keyword to every access list entry because the first fragment of the IP packet is considered a nonfragment and is treated independently of the subsequent fragments. An initial fragment will not match an access list **permit** or **deny** entry that contains the **fragments** keyword. The packet is compared to the next access list entry, and so on, until it is either permitted or denied by an access list entry that does not contain the **fragments** keyword. Therefore, you may need two access list entries for every **deny** entry. The first **deny** entry of the pair will not include the **fragments** keyword and applies to the initial fragment. The second **deny** entry of the pair will include the **fragments** keyword and applies to the subsequent fragments. In the cases in which there are multiple **deny** entries for the same host but with different Layer 4 ports, a single **deny** access list entry with the **fragments** keyword for that host is all that needs to be added. Thus all the fragments of a packet are handled in the same manner by the access list.

Packet fragments of IP datagrams are considered individual packets, and each counts individually as a packet in access list accounting and access list violation counts.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *name*
4. [*sequence-number*] **deny** *protocol source* [*source-wildcard*] [*operator port* [*port*]] *destination* [*destination-wildcard*][*operator port* [*port*]]
5. [*sequence-number*] **deny** *protocol source* [*source-wildcard*] [*operator port* [*port*]] *destination* [*destination-wildcard*] [*operator port* [*port*]] [**fragments**]
6. [*sequence-number*] **permit** *protocol source* [*source-wildcard*] [*operator port* [*port*]] *destination* [*destination-wildcard*] [*operator port* [*port*]]
7. Repeat some combination of Steps 4 through 6 until you have specified the values on which you want to base your access list.
8. **end**
9. **show ip access-list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *name* **Example:** `Router(config)# ip access-list extended rstrct4`	Defines an extended IP access list using a name and enters extended named access list configuration mode.
Step 4	[*sequence-number*] **deny** *protocol source* [*source-wildcard*] [*operator port* [*port*]] *destination* [*destination-wildcard*] [*operator port* [*port*]] **Example:** `Router(config-ext-nacl)# deny ip any 172.20.1.1`	(Optional) Denies any packet that matches all of the conditions specified in the statement. • This statement will apply to nonfragmented packets and initial fragments.
Step 5	[*sequence-number*] **deny** *protocol source* [*source-wildcard*][*operator port* [*port*]] *destination* [*destination-wildcard*] [*operator port* [*port*]] **fragments** **Example:** `Router(config-ext-nacl)# deny ip any 172.20.1.1 fragments`	(Optional) Denies any packet that matches all of the conditions specified in the statement • This statement will apply to noninitial fragments.
Step 6	[*sequence-number*] **permit** *protocol source* [*source-wildcard*] [*operator port* [*port*]] *destination* [*destination-wildcard*] [*operator port* [*port*]] **Example:** `Router(config-ext-nacl)# permit tcp any any`	Permits any packet that matches all of the conditions specified in the statement. • Every access list needs at least one **permit** statement. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255.
Step 7	Repeat some combination of Steps 4 through 6 until you have specified the values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.

	Command or Action	Purpose
Step 8	**end** **Example:** `Router(config-ext-nacl)# end`	Ends configuration mode and returns the system to privileged EXEC mode.
Step 9	**show ip access-list** **Example:** `Router# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

Configuration Examples for Refining an IP Access List

This section provides the following configuration examples:

Resequencing Entries in an Access List: Example

The following example shows an access list before and after resequencing. The starting value is 1, and increment value is 2. The subsequent entries are ordered based on the increment values that users provide, and the range is from 1 to 2147483647.

When an entry with no sequence number is entered, by default it has a sequence number of 10 more than the last entry in the access list.

```
Router# show access-list carls

Extended IP access list carls
    10 permit ip host 10.3.3.3 host 172.16.5.34
    20 permit icmp any any
    30 permit tcp any host 10.3.3.3
    40 permit ip host 10.4.4.4 any
    50 Dynamic test permit ip any any
    60 permit ip host 172.16.2.2 host 10.3.3.12
    70 permit ip host 10.3.3.3 any log
    80 permit tcp host 10.3.3.3 host 10.1.2.2
    90 permit ip host 10.3.3.3 any
    100 permit ip any any

Router(config)# ip access-list extended carls
Router(config)# ip access-list resequence carls 1 2
Router(config)# end
```

```
Router# show access-list carls

Extended IP access list carls
    1 permit ip host 10.3.3.3 host 172.16.5.34
    3 permit icmp any any
    5 permit tcp any host 10.3.3.3
    7 permit ip host 10.4.4.4 any
    9 Dynamic test permit ip any any
    11 permit ip host 172.16.2.2 host 10.3.3.12
    13 permit ip host 10.3.3.3 any log
    15 permit tcp host 10.3.3.3 host 10.1.2.2
    17 permit ip host 10.3.3.3 any
    19 permit ip any any
```

Adding an Entry with a Sequence Number: Example

In the following example, an new entry (sequence number 15) is added to an access list:

```
Router# show ip access-list

Standard IP access list tryon
2 permit 10.4.4.2, wildcard bits 0.0.255.255
5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
20 permit 10.0.0.2, wildcard bits 0.0.0.255

Router(config)# ip access-list standard tryon

Router(config-std-nacl)# 15 permit 10.5.5.5 0.0.0.255

Router# show ip access-list

Standard IP access list tryon
2 permit 10.4.0.0, wildcard bits 0.0.255.255
5 permit 10.0.0.0, wildcard bits 0.0.0.255
10 permit 10.0.0.0, wildcard bits 0.0.0.255
15 permit 10.5.5.0, wildcard bits 0.0.0.255
20 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Adding an Entry with No Sequence Number: Example

The following example shows how an entry with no specified sequence number is added to the end of an access list. When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list.

```
Router(config)# ip access-list standard resources

Router(config-std-nacl)# permit 10.1.1.1 0.0.0.255
Router(config-std-nacl)# permit 10.2.2.2 0.0.0.255
Router(config-std-nacl)# permit 10.3.3.3 0.0.0.255

Router# show access-list
Standard IP access list resources
10 permit 10.1.1.1, wildcard bits 0.0.0.255
20 permit 10.2.2.2, wildcard bits 0.0.0.255
30 permit 10.3.3.3, wildcard bits 0.0.0.255

Router(config)# ip access-list standard resources
```

```
Router(config-std-nacl)# permit 10.4.4.4 0.0.0.255
Router(config-std-nacl)# end

Router# show access-list

Standard IP access list resources
10 permit 10.1.1.1, wildcard bits 0.0.0.255
20 permit 10.2.2.2, wildcard bits 0.0.0.255
30 permit 10.3.3.3, wildcard bits 0.0.0.255
40 permit 10.4.4.4, wildcard bits 0.0.0.255
```

Time Ranges Applied to IP Access List Entries: Example

The following example creates a time range called no-http, which extends from Monday to Friday from 8:00 a.m. to 6:00 p.m. That time range is applied to the **deny** statement, thereby denying HTTP traffic on Monday through Friday from 8:00 a.m. to 6:00 p.m.

The time range called udp-yes defines weekends from noon to 8:00 p.m. That time range is applied to the **permit** statement, thereby allowing UDP traffic on Saturday and Sunday from noon to 8:00 p.m. only. The access list containing both statements is applied to inbound packets on Ethernet interface 0.

```
time-range no-http
 periodic weekdays 8:00 to 18:00
!
time-range udp-yes
 periodic weekend 12:00 to 20:00
!
ip access-list extended strict
 deny tcp any any eq http time-range no-http
 permit udp any any time-range udp-yes
!
interface ethernet 0
 ip access-group strict in
```

Filtering IP Packet Fragments: Example

In the following access list, the first statement will deny only noninitial fragments destined for host 172.16.1.1. The second statement will permit only the remaining nonfragmented and initial fragments that are destined for host 172.16.1.1 TCP port 80. The third statement will deny all other traffic. In order to block noninitial fragments for any TCP port, we must block noninitial fragments for all TCP ports, including port 80 for host 172.16.1.1. That is, non-initial fragments will not contain Layer 4 port information, so, in order to block such traffic for a given port, we have to block fragments for all ports.

```
access-list 101 deny ip any host 172.16.1.1 fragments
access-list 101 permit tcp any host 172.16.1.1 eq 80
access-list 101 deny ip any any
```

Additional References

The following sections provide references related to access list entry resequencing, time-based access lists, or IP fragment filtering.

Related Documents

Related Topic	Document Title
Using the **time-range** command to establish time ranges	"Performing Basic System Management" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	—

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Refining an IP Access List

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Refining an IP Access List

Feature Name	Releases	Feature Configuration Information
Distributed Time-Based Access Lists	12.2(2)T	Before the introduction of this feature, time-based access lists were not supported on line cards for the Cisco 7500 series routers. If time-based access lists were configured, they behaved as normal access lists. If an interface on a line card were configured with a time-based access list, the packets switched into the interface were not distributed switched through the line card, but were forwarded to the Route Processor for processing. The Distributed Time-Based Access Lists feature allows packets destined for an interface configured with a time-based access list to be distributed switched through the line card. See the following section: • Distributed Time-Based Access Lists, page 7
Time-Based Access Lists	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Configuring Lock-and-Key Security (Dynamic Access Lists)

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes how to configure lock-and-key security at your router. Lock-and-key is a traffic filtering security feature available for the IP protocol.

For a complete description of lock-and-key commands, refer to the "Lock-and-Key Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the chapter "Identifying Supported Platforms" section in the "Using Cisco IOS Software."

In This Chapter

This chapter has the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

About Lock-and-Key

Lock-and-key is a traffic filtering security feature that dynamically filters IP protocol traffic. Lock-and-key is configured using IP dynamic extended access lists. Lock-and-key can be used in conjunction with other standard access lists and static extended access lists.

When lock-and-key is configured, designated users whose IP traffic is normally blocked at a router can gain temporary access through the router. When triggered, lock-and-key reconfigures the interface's existing IP access list to permit designated users to reach their designated host(s). Afterwards, lock-and-key reconfigures the interface back to its original state.

For a user to gain access to a host through a router with lock-and-key configured, the user must first open a Telnet session to the router. When a user initiates a standard Telnet session to the router, lock-and-key automatically attempts to authenticate the user. If the user is authenticated, they will then gain temporary access through the router and be able to reach their destination host.

This section has the following sections:

Benefits of Lock-and-Key

Lock-and-key provides the same benefits as standard and static extended access lists (these benefits are discussed in the chapter "Access Control Lists: Overview and Guidelines"). However, lock-and-key also has the following security benefits over standard and static extended access lists:

- Lock-and-key uses a challenge mechanism to authenticate individual users.
- Lock-and-key provides simpler management in large internetworks.
- In many cases, lock-and-key reduces the amount of router processing required for access lists.
- Lock-and-key reduces the opportunity for network break-ins by network hackers.

With lock-and-key, you can specify which users are permitted access to which source and destination hosts. These users must pass a user authentication process before they are permitted access to their designated hosts. Lock-and-key creates dynamic user access through a firewall, without compromising other configured security restrictions.

When to Use Lock-and-Key

Two examples of when you might use lock-and-key follow:

- When you want a specific remote user (or group of remote users) to be able to access a host within your network, connecting from their remote hosts via the Internet. Lock-and-key authenticates the user, then permits limited access through your firewall router for the individual's host or subnet, for a finite period of time.

- When you want a subset of hosts on a local network to access a host on a remote network protected by a firewall. With lock-and-key, you can enable access to the remote host only for the desired set of local user's hosts. Lock-and-key require the users to authenticate through a TACACS+ server, or other security server, before allowing their hosts to access the remote hosts.

How Lock-and-Key Works

The following process describes the lock-and-key access operation:

1. A user opens a Telnet session to a border (firewall) router configured for lock-and-key. The user connects via the virtual terminal port on the router.
2. The Cisco IOS software receives the Telnet packet, opens a Telnet session, prompts for a password, and performs a user authentication process. The user must pass authentication before access through the router is allowed. The authentication process can be done by the router or by a central access security server such as a TACACS+ or RADIUS server.
3. When the user passes authentication, they are logged out of the Telnet session, and the software creates a temporary entry in the dynamic access list. (Per your configuration, this temporary entry can limit the range of networks to which the user is given temporary access.)
4. The user exchanges data through the firewall.
5. The software deletes the temporary access list entry when a configured timeout is reached, or when the system administrator manually clears it. The configured timeout can either be an idle timeout or an absolute timeout.

Note The temporary access list entry is not automatically deleted when the user terminates a session. The temporary access list entry remains until a configured timeout is reached or until it is cleared by the system administrator.

Compatibility with Releases Before Cisco IOS Release 11.1

Enhancements to the **access-list** command are used for lock-and-key. These enhancements are backward compatible—if you migrate from a release before Cisco IOS Release 11.1 to a newer release, your access lists will be automatically converted to reflect the enhancements. However, if you try to use lock-and-key with a release before Cisco IOS Release 11.1, you might encounter problems as described in the following caution paragraph:

Caution Cisco IOS releases before Release 11.1 are not upwardly compatible with the lock-and-key access list enhancements. Therefore, if you save an access list with software older than Release 11.1, and then use this software, the resulting access list will not be interpreted correctly. *This could cause you severe security problems.* You must save your old configuration files with Cisco IOS Release 11.1 or later software before booting an image with these files.

Risk of Spoofing with Lock-and-Key

Caution Lock-and-key access allows an external event (a Telnet session) to place an opening in the firewall. While this opening exists, the router is susceptible to source address spoofing.

When lock-and-key is triggered, it creates a dynamic opening in the firewall by temporarily reconfiguring an interface to allow user access. While this opening exists, another host might spoof the authenticated user's address to gain access behind the firewall. Lock-and-key does not cause the address spoofing problem; the problem is only identified here as a concern to the user. Spoofing is a problem inherent to all access lists, and lock-and-key does not specifically address this problem.

To prevent spoofing, configure encryption so that traffic from the remote host is encrypted at a secured remote router, and decrypted locally at the router interface providing lock-and-key. You want to ensure that all traffic using lock-and-key will be encrypted when entering the router; this way no hackers can spoof the source address, because they will be unable to duplicate the encryption or to be authenticated as is a required part of the encryption setup process.

Router Performance Impacts with Lock-and-Key

When lock-and-key is configured, router performance can be affected in the following ways:

- When lock-and-key is triggered, the dynamic access list forces an access list rebuild on the silicon switching engine (SSE). This causes the SSE switching path to slow down momentarily.
- Dynamic access lists require the idle timeout facility (even if the timeout is left to default) and therefore cannot be SSE switched. These entries must be handled in the protocol fast-switching path.
- When remote users trigger lock-and-key at a border router, additional access list entries are created on the border router interface. The interface's access list will grow and shrink dynamically. Entries are dynamically removed from the list after either the idle-timeout or max-timeout period expires. Large access lists can degrade packet switching performance, so if you notice performance problems, you should look at the border router configuration to see if you should remove temporary access list entries generated by lock-and-key.

Prerequisites to Configuring Lock-and-Key

Lock-and-key uses IP extended access lists. You must have a solid understanding of how access lists are used to filter traffic, before you attempt to configure lock-and-key. Access lists are described in the chapter "Access Control Lists: Overview and Guidelines."

Lock-and-key employs user authentication and authorization as implemented in Cisco's authentication, authorization, and accounting (AAA) paradigm. You must understand how to configure AAA user authentication and authorization before you configure lock-and-key. User authentication and authorization is explained in the "Authentication, Authorization, and Accounting (AAA)" part of this document.

Lock-and-key uses the **autocommand** command, which you should understand. This command is described in the "Modem Support and Asynchronous Device Commands" chapter of the *Cisco IOS Dial Technologies Command Reference.*

Configuring Lock-and-Key

To configure lock-and-key, use the following commands beginning in global configuration mode. While completing these steps, be sure to follow the guidelines listed in the "Lock-and-Key Configuration Guidelines" section of this chapter.

	Command	Purpose
Step 1	`Router(config)# access-list access-list-number [dynamic dynamic-name [timeout minutes]] {deny | permit} telnet source source-wildcard destination destination-wildcard [precedence precedence] [tos tos] [established] [log]`	Configures a dynamic access list, which serves as a template and placeholder for temporary access list entries.
Step 2	`Router(config)# access-list dynamic-extend`	(Optional) Extends the absolute timer of the dynamic ACL by six minutes when you open another Telnet session into the router to re-authenticate yourself using lock-and-key. Use this command if your job will run past the ACL's absolute timer.
Step 3	`Router(config)# interface type number`	Configures an interface and enters interface configuration mode.
Step 4	`Router(config-if)# ip access-group access-list-number`	Applies the access list to the interface.
Step 5	`Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 6	`Router(config)# line vty line-number [ending-line-number]`	Defines one or more virtual terminal (VTY) ports and enters line configuration mode. If you specify multiple VTY ports, they must all be configured identically because the software hunts for available VTY ports on a round-robin basis. If you do not want to configure all your VTY ports for lock-and-key access, you can specify a group of VTY ports for lock-and-key support only.
Step 7	`Router(config-line)# login tacacs` or `Router(config-line)# password password` or `Router(config-line)# login local` or `Router(config-line)# exit` then `Router(config)# username name password secret`	Configures user authentication in line or global configuration mode.
Step 8	`Router(config-line)# autocommand access-enable [host] [timeout minutes]` or `Router(config)# autocommand access-enable [host] [timeout minutes]`	Enables the creation of temporary access list entries in line or global configuration mode. If the optional **host** keyword is *not* specified, all hosts on the entire network are allowed to set up a temporary access list entry. The dynamic access list contains the network mask to enable the new network connection.

For an example of a lock-and-key configuration, see the section "Lock-and-Key Configuration Examples" later in this chapter.

Lock-and-Key Configuration Guidelines

Before you configure lock-and-key, you should understand the guidelines discussed in the following sections:

- Dynamic Access Lists
- Lock-and-Key Authentication
- The autocommand Command

Dynamic Access Lists

Use the following guidelines for configuring dynamic access lists:

- Do *not* create more than one dynamic access list for any one access list. The software only refers to the first dynamic access list defined.
- Do *not* assign the same *dynamic-name* to another access list. Doing so instructs the software to reuse the existing list. All named entries must be globally unique within the configuration.
- Assign attributes to the dynamic access list in the same way you assign attributes for a static access list. The temporary access list entries inherit the attributes assigned to this list.
- Configure Telnet as the protocol so that users must open a Telnet session into the router to be authenticated before they can gain access through the router.
- Either define an idle timeout now with the **timeout** keyword in the **access-enable** command in the **autocommand** command, or define an absolute timeout value later with the **access-list** command. You must define either an idle timeout or an absolute timeout—otherwise, the temporary access list entry will remain configured indefinitely on the interface (even after the user has terminated their session) until the entry is removed manually by an administrator. (You could configure both idle and absolute timeouts if you wish.)
- If you configure an idle timeout, the idle timeout value should be equal to the WAN idle timeout value.
- If you configure both idle and absolute timeouts, the idle timeout value must be less than the absolute timeout value.
- If you realize that a job will run past the ACL's absolute timer, use the **access-list dynamic-extend** command to extend the absolute timer of the dynamic ACL by six minutes. This command allows you to open a new Telnet session into the router to re-authentication yourself using lock-and-key.
- The only values replaced in the temporary entry are the source or destination address, depending whether the access list was in the input access list or output access list. All other attributes, such as port, are inherited from the main dynamic access list.
- Each addition to the dynamic list is always put at the beginning of the dynamic list. You cannot specify the order of temporary access list entries.
- Temporary access list entries are never written to NVRAM.
- To manually clear or to display dynamic access lists, refer to the section "Maintaining Lock-and-Key" later in this chapter.

Lock-and-Key Authentication

There are three possible methods to configure an authentication query process. These three methods are described in this section.

Note Cisco recommends that you use the TACACS+ server for your authentication query process. TACACS+ provides authentication, authorization, and accounting services. It also provides protocol support, protocol specification, and a centralized security database. Using a TACACS+ server is described in the next section, "Method 1—Configuring a Security Server."

Method 1—Configuring a Security Server

Use a network access security server such as TACACS+ server. This method requires additional configuration steps on the TACACS+ server but allows for stricter authentication queries and more sophisticated tracking capabilities.

```
Router(config-line)# login tacacs
```

Method 2—Configuring the username Command

Use the **username** command. This method is more effective because authentication is determined on a user basis.

```
Router(config)# username name {nopassword | password {mutual-password | encryption-type
encryption-password}}
```

Method 3—Configuring the password and login Commands

Use the **password** and **login** commands. This method is less effective because the password is configured for the port, not for the user. Therefore, any user who knows the password can authenticate successfully.

```
Router(config-line)# password password
Router(config-line)# login local
```

The autocommand Command

Use the following guidelines for configuring the **autocommand** command:

- If you use a TACACS+ server to authenticate the user, you should configure the **autocommand** command on the TACACS+ server as a per-user autocommand. If you use local authentication, use the **autocommand** command on the line.
- Configure all virtual terminal (VTY) ports with the same **autocommand** command. Omitting an **autocommand** command on a VTY port allows a random host to gain EXEC mode access to the router and does not create a temporary access list entry in the dynamic access list.
- If you did not previously define an idle timeout with the **autocommand access-enable** command, you must define an absolute timeout now with the **access-list** command. You must define either an idle timeout or an absolute timeout—otherwise, the temporary access list entry will remain configured indefinitely on the interface (even after the user has terminated their session) until the entry is removed manually by an administrator. (You could configure both idle and absolute timeouts if you wish.)
- If you configure both idle and absolute timeouts, the absolute timeout value must be greater than the idle timeout value.

Verifying Lock-and-Key Configuration

You can verify that lock-and-key is successfully configured on the router by asking a user to test the connection. The user should be at a host that is permitted in the dynamic access list, and the user should have AAA authentication and authorization configured.

To test the connection, the user should Telnet to the router, allow the Telnet session to close, and then attempt to access a host on the other side of the router. This host must be one that is permitted by the dynamic access list. The user should access the host with an application that uses the IP protocol.

The following sample display illustrates what end-users might see if they are successfully authenticated. Notice that the Telnet connection is closed immediately after the password is entered and authenticated. The temporary access list entry is then created, and the host that initiated the Telnet session now has access inside the firewall.

```
Router% telnet corporate
Trying 172.21.52.1 ...
Connected to corporate.example.com.
Escape character is '^]'.
User Access Verification
Password:Connection closed by foreign host.
```

You can then use the **show access-lists** command at the router to view the dynamic access lists, which should include an additional entry permitting the user access through the router.

Maintaining Lock-and-Key

When lock-and-key is in use, dynamic access lists will dynamically grow and shrink as entries are added and deleted. You need to make sure that entries are being deleted in a timely way, because while entries exist, the risk of a spoofing attack is present. Also, the more entries there are, the bigger the router performance impact will be.

If you do not have an idle or absolute timeout configured, entries will remain in the dynamic access list until you manually remove them. If this is the case, make sure that you are extremely vigilant about removing entries.

Displaying Dynamic Access List Entries

You can display temporary access list entries when they are in use. After a temporary access list entry is cleared by you or by the absolute or idle timeout parameter, it can no longer be displayed. The number of matches displayed indicates the number of times the access list entry was hit.

To view dynamic access lists and any temporary access list entries that are currently established, use the following command in privileged EXEC mode:

Command	Purpose
Router# **show access-lists** [*access-list-number*]	Displays dynamic access lists and temporary access list entries.

Manually Deleting Dynamic Access List Entries

To manually delete a temporary access list entry, use the following command in privileged EXEC mode:

Command	Purpose
Router# **clear access-template** [*access-list-number* | *name*] [*dynamic-name*] [*source*] [*destination*]	Deletes a dynamic access list.

Lock-and-Key Configuration Examples

The following sections provide lock-and-key configuration examples:

- Lock-and-Key with Local Authentication Example
- Lock-and-Key with TACACS+ Authentication Example

Cisco recommends that you use a TACACS+ server for authentication, as shown in the second example.

Lock-and-Key with Local Authentication Example

This example shows how to configure lock-and-key access, with authentication occurring locally at the router. Lock-and-key is configured on the Ethernet 0 interface.

```
interface ethernet0
 ip address 172.18.23.9 255.255.255.0
 ip access-group 101 in

access-list 101 permit tcp any host 172.18.21.2 eq telnet
access-list 101 dynamic mytestlist timeout 120 permit ip any any

line vty 0
login local
autocommand access-enable timeout 5
```

The first access-list entry allows only Telnet into the router. The second access-list entry is always ignored until lock-and-key is triggered.

In the **access-list** command, the timeout is the absolute timeout. In this example, the lifetime of the mytestlist ACL is 120 minutes; that is, when a user logs in and enable the **access-enable** command, a dynamic ACL is created for 120 minutes (the maximum absolute time). The session is closed after 120 minutes, whether or not anyone is using it.

In the **autocommand** command, the timeout is the idle timeout. In this example, each time the user logs in or authenticates there is a 5-minute session. If there is no activity, the session closes in 5 minutes and the user has to reauthenticate. If the user uses the connection, the absolute time takes affect and the session closes in 120 minutes.

After a user opens a Telnet session into the router, the router will attempt to authenticate the user. If authentication is successful, the **autocommand** executes and the Telnet session terminates. The **autocommand** creates a temporary inbound access list entry at the Ethernet 0 interface, based on the second access-list entry (mytestlist). This temporary entry will expire after 5 minutes, as specified by the timeout.

Lock-and-Key with TACACS+ Authentication Example

The following example shows how to configure lock-and-key access, with authentication on a TACACS+ server. Lock-and-key access is configured on the BRI0 interface. Four VTY ports are defined with the password “cisco”.

```
aaa authentication login default group tacacs+ enable
aaa accounting exec stop-only group tacacs+
aaa accounting network stop-only group tacacs+
enable password ciscotac
!
isdn switch-type basic-dms100
!
interface ethernet0
ip address 172.18.23.9 255.255.255.0
!
interface BRI0
 ip address 172.18.21.1 255.255.255.0
 encapsulation ppp
 dialer idle-timeout 3600
 dialer wait-for-carrier-time 100
 dialer map ip 172.18.21.2 name diana
 dialer-group 1
 isdn spid1 2036333715291
 isdn spid2 2036339371566
 ppp authentication chap
 ip access-group 102 in
!
access-list 102 permit tcp any host 172.18.21.2 eq telnet
access-list 102 dynamic testlist timeout 5 permit ip any any
!
!
ip route 172.18.250.0 255.255.255.0 172.18.21.2
priority-list 1 interface BRI0 high
tacacs-server host 172.18.23.21
tacacs-server host 172.18.23.14
tacacs-server key test1
tftp-server rom alias all
!
dialer-list 1 protocol ip permit
!
line con 0
 password cisco
line aux 0
 line VTY 0 4
 autocommand access-enable timeout 5
 password cisco
!
```

Configuring IP Session Filtering (Reflexive Access Lists)

This chapter describes how to configure reflexive access lists on your router. Reflexive access lists provide the ability to filter network traffic at a router, based on IP upper-layer protocol "session" information.

For a complete description of reflexive access list commands, refer to the "Reflexive Access List Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the chapter "Identifying Supported Platforms" section in the "Using Cisco IOS Software."

In This Chapter

This chapter has the following sections:

About Reflexive Access Lists

Reflexive access lists allow IP packets to be filtered based on upper-layer session information. You can use reflexive access lists to permit IP traffic for sessions originating from within your network but to deny IP traffic for sessions originating from outside your network. This is accomplished by reflexive filtering, a kind of session filtering.

Reflexive access lists can be defined with extended named IP access lists only. You cannot define reflexive access lists with numbered or standard named IP access lists or with other protocol access lists.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

You can use reflexive access lists in conjunction with other standard access lists and static extended access lists.

This section has the following sections:

Benefits of Reflexive Access Lists

Reflexive access lists are an important part of securing your network against network hackers, and can be included in a firewall defense. Reflexive access lists provide a level of security against spoofing and certain denial-of-service attacks. Reflexive access lists are simple to use, and, compared to basic access lists, provide greater control over which packets enter your network.

What Is a Reflexive Access List?

Reflexive access lists are similar in many ways to other access lists. Reflexive access lists contain condition statements (entries) that define criteria for permitting IP packets. These entries are evaluated in order, and when a match occurs, no more entries are evaluated.

However, reflexive access lists have significant differences from other types of access lists. Reflexive access lists contain only temporary entries; these entries are automatically created when a new IP session begins (for example, with an outbound packet), and the entries are removed when the session ends. Reflexive access lists are not themselves applied directly to an interface, but are “nested” within an extended named IP access list that is applied to the interface. (For more information about this, see the section “Reflexive Access Lists Configuration Task List” later in this chapter.) Also, reflexive access lists do not have the usual implicit “deny all traffic” statement at the end of the list, because of the nesting.

How Reflexive Access Lists Implement Session Filtering

This section compares session filtering with basic access lists to session filtering with reflexive access lists. This section contains the following sections:

With Basic Access Lists

With basic standard and static extended access lists, you can approximate session filtering by using the **established** keyword with the **permit** command. The **established** keyword filters TCP packets based on whether the ACK or RST bits are set. (Set ACK or RST bits indicate that the packet is not the first in the session, and therefore, that the packet belongs to an established session.) This filter criterion would be part of an access list applied permanently to an interface.

With Reflexive Access Lists

Reflexive access lists, however, provide a truer form of session filtering, which is much harder to spoof because more filter criteria must be matched before a packet is permitted through. (For example, source and destination addresses and port numbers are checked, not just ACK and RST bits.) Also, session filtering uses temporary filters which are removed when a session is over. This limits the hacker's attack opportunity to a smaller time window.

Moreover, the previous method of using the **established** keyword was available only for the TCP upper-layer protocol. So, for the other upper-layer protocols (such as UDP, ICMP, and so forth), you would have to either permit all incoming traffic or define all possible permissible source/destination host/port address pairs for each protocol. (Besides being an unmanageable task, this could exhaust NVRAM space.)

Where to Configure Reflexive Access Lists

Configure reflexive access lists on border routers—routers that pass traffic between an internal and external network. Often, these are firewall routers.

Note In this chapter, the words "within your network" and "internal network" refer to a network that is controlled (secured), such as your organization's intranet, or to a part of your organization's internal network that has higher security requirements than another part. "Outside your network" and "external network" refer to a network that is uncontrolled (unsecured) such as the Internet or to a part of your organization's network that is not as highly secured.

How Reflexive Access Lists Work

A reflexive access list is triggered when a new IP upper-layer session (such as TCP or UDP) is initiated from inside your network, with a packet traveling to the external network. When triggered, the reflexive access list generates a new, temporary entry. This entry will permit traffic to enter your network if the traffic is part of the session, but will not permit traffic to enter your network if the traffic is not part of the session.

For example, if an outbound TCP packet is forwarded to outside of your network, and this packet is the first packet of a TCP session, then a new, temporary reflexive access list entry will be created. This entry is added to the reflexive access list, which applies to inbound traffic. The temporary entry has characteristics as described next.

This section contains the following sections:

Temporary Access List Entry Characteristics

- The entry is always a **permit** entry.
- The entry specifies the same protocol (TCP) as the original outbound TCP packet.
- The entry specifies the same source and destination addresses as the original outbound TCP packet, except the addresses are swapped.
- The entry specifies the same source and destination port numbers as the original outbound TCP packet, except the port numbers are swapped.

  (This entry characteristic applies only for TCP and UDP packets. Other protocols, such as ICMP and IGMP, do not have port numbers, and other criteria are specified. For example, for ICMP, type numbers are used instead.)
- Inbound TCP traffic will be evaluated against the entry, until the entry expires. If an inbound TCP packet matches the entry, the inbound packet will be forwarded into your network.
- The entry will expire (be removed) after the last packet of the session passes through the interface.
- If no packets belonging to the session are detected for a configurable length of time (the timeout period), the entry will expire.

When the Session Ends

Temporary reflexive access list entries are removed at the end of the session. For TCP sessions, the entry is removed 5 seconds after two set FIN bits are detected, or immediately after matching a TCP packet with the RST bit set. (Two set FIN bits in a session indicate that the session is about to end; the 5-second window allows the session to close gracefully. A set RST bit indicates an abrupt session close.) Or, the temporary entry is removed after no packets of the session have been detected for a configurable length of time (the timeout period).

For UDP and other protocols, the end of the session is determined differently than for TCP. Because other protocols are considered to be connectionless (sessionless) services, there is no session tracking information embedded in packets. Therefore, the end of a session is considered to be when no packets of the session have been detected for a configurable length of time (the timeout period).

Restrictions on Using Reflexive Access Lists

Reflexive access lists do not work with some applications that use port numbers that change during a session. For example, if the port numbers for a return packet are different from the originating packet, the return packet will be denied, even if the packet is actually part of the same session.

The TCP application of FTP is an example of an application with changing port numbers. With reflexive access lists, if you start an FTP request from within your network, the request will not complete. Instead, you must use Passive FTP when originating requests from within your network.

Prework: Before You Configure Reflexive Access Lists

Before you configure reflexive access lists, you must decide whether to configure reflexive access lists on an internal or external interface, as described in the next section, “Choosing an Interface: Internal or External.”

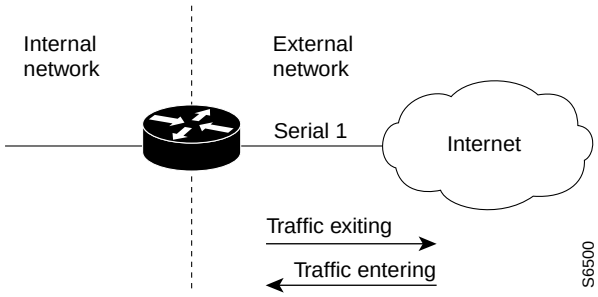
You should also be sure that you have a basic understanding of the IP protocol and of access lists; specifically, you should know how to configure extended named IP access lists. To learn about configuring IP extended access lists, refer to the "Configuring IP Services" chapter of the *Cisco IOS IP Configuration Guide*.

Choosing an Interface: Internal or External

Reflexive access lists are most commonly used with one of two basic network topologies. Determining which of these topologies is most like your own can help you decide whether to use reflexive access lists with an internal interface or with an external interface (the interface connecting to an internal network, or the interface connecting to an external network).

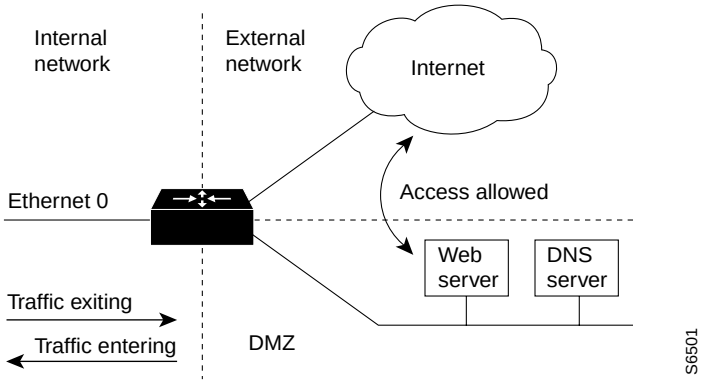
The first topology is shown in Figure 18. In this simple topology, reflexive access lists are configured for the *external* interface Serial 1. This prevents IP traffic from entering the router and the internal network, unless the traffic is part of a session already established from within the internal network.

Figure 18 Simple Topology—Reflexive Access Lists Configured at the External Interface



The second topology is shown in Figure 19. In this topology, reflexive access lists are configured for the *internal* interface Ethernet 0. This allows external traffic to access the services in the Demilitarized Zone (DMZ), such as DNS services, but prevents IP traffic from entering your internal network—unless the traffic is part of a session already established from within the internal network.

Figure 19 DMZ Topology—Reflexive Access Lists Configured at the Internal Interface



Use these two example topologies to help you decide whether to configure reflexive access lists for an internal or external interface.

Reflexive Access Lists Configuration Task List

In the previous section, "Prework: Before You Configure Reflexive Access Lists," you decided whether to configure reflexive access lists for an internal or external interface.

Now, complete the tasks in one of the following configuration task lists:

- External Interface Configuration Task List
- Internal Interface Configuration Task List

For configuration examples, refer to the "Reflexive Access List Configuration Examples" section at the end of this chapter.

External Interface Configuration Task List

To configure reflexive access lists for an external interface, perform the following tasks:

1. Defining the reflexive access list(s) in an *outbound* IP extended named access list
2. Nesting the reflexive access list(s) in an *inbound* IP extended named access list
3. Setting a global timeout value

These tasks are described in the sections following the "Internal Interface Configuration Task List" section.

Note The defined (outbound) reflexive access list evaluates traffic traveling out of your network: if the defined reflexive access list is matched, temporary entries are created in the nested (inbound) reflexive access list. These temporary entries will then be applied to traffic traveling into your network.

Internal Interface Configuration Task List

To configure reflexive access lists for an internal interface, perform the following tasks:

1. Defining the reflexive access list(s) in an *inbound* IP extended named access list
2. Nesting the reflexive access list(s) in an *outbound* IP extended named access list
3. Setting a global timeout value

These tasks are described in the next sections.

Note The defined (inbound) reflexive access list is used to evaluate traffic traveling out of your network: if the defined reflexive access list is matched, temporary entries are created in the nested (outbound) reflexive access list. These temporary entries will then be applied to traffic traveling into your network.

Defining the Reflexive Access List(s)

To define a reflexive access list, you use an entry in an extended named IP access list. This entry must use the **reflect** keyword.

- If you are configuring reflexive access lists for an external interface, the extended named IP access list should be one that is applied to outbound traffic.

- If you are configuring reflexive access lists for an internal interface, the extended named IP access list should be one that is applied to inbound traffic.

To define reflexive access lists, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip access-list extended` *name*	External interface: Specifies the outbound access list. or Internal interface: Specifies the inbound access list. (This command enters access-list configuration mode.)
Step 2	`Router(config-ext-nacl)# permit` *protocol* `any any reflect` *name* `[timeout` *seconds*`]`	Defines the reflexive access list using the reflexive **permit** entry. Repeat this step for each IP upper-layer protocol; for example, you can define reflexive filtering for TCP sessions and also for UDP sessions. You can use the same *name* for multiple protocols. For additional guidelines for this task, see the following section, "Mixing Reflexive Access List Statements with Other Permit and Deny Entries."

If the extended named IP access list you just specified has never been applied to the interface, you must also apply the extended named IP access list to the interface.

To apply the extended named IP access list to the interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip access-group` *name* `out` or `Router(config-if)# ip access-group` *name* `in`	External interface: Applies the extended access list to the interface's outbound traffic. Internal interface: Applies the extended access list to the interface's inbound traffic.

Mixing Reflexive Access List Statements with Other Permit and Deny Entries

The extended IP access list that contains the reflexive access list **permit** statement can also contain other normal **permit** and **deny** statements (entries). However, as with all access lists, the order of entries is important, as explained in the next few paragraphs.

If you configure reflexive access lists for an external interface, when an outbound IP packet reaches the interface, the packet will be evaluated sequentially by each entry in the outbound access list until a match occurs.

If the packet matches an entry prior to the reflexive **permit** entry, the packet will not be evaluated by the reflexive **permit** entry, and no temporary entry will be created for the reflexive access list (reflexive filtering will not be triggered).

The outbound packet will be evaluated by the reflexive **permit** entry only if no other match occurs first. Then, if the packet matches the protocol specified in the reflexive **permit** entry, the packet is forwarded out of the interface and a corresponding temporary entry is created in the inbound reflexive access list (unless the corresponding entry already exists, indicating the outbound packet belongs to a session in progress). The temporary entry specifies criteria that permits inbound traffic only for the same session.

Nesting the Reflexive Access List(s)

After you define a reflexive access list in one IP extended access list, you must "nest" the reflexive access list within a different extended named IP access list.

- If you are configuring reflexive access lists for an external interface, nest the reflexive access list within an extended named IP access list applied to inbound traffic.
- If you are configuring reflexive access lists for an internal interface, nest the reflexive access list within an extended named IP access list applied to outbound traffic.

After you nest a reflexive access list, packets heading into your internal network can be evaluated against any reflexive access list temporary entries, along with the other entries in the extended named IP access list.

To nest reflexive access lists, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip access-list extended` *name*	External interface: Specifies the inbound access list. or Internal interface: Specifies the outbound access list. (This command enters access-list configuration mode.)
Step 2	`Router(config-ext-nacl)# evaluate` *name*	Adds an entry that "points" to the reflexive access list. Adds an entry for each reflexive access list *name* previously defined.

Again, the order of entries is important. Normally, when a packet is evaluated against entries in an access list, the entries are evaluated in sequential order, and when a match occurs, no more entries are evaluated. With a reflexive access list nested in an extended access list, the extended access list entries are evaluated sequentially up to the nested entry, then the reflexive access list entries are evaluated sequentially, and then the remaining entries in the extended access list are evaluated sequentially. As usual, after a packet matches *any* of these entries, no more entries will be evaluated.

If the extended named IP access list you just specified has never been applied to the interface, you must also apply the extended named IP access list to the interface.

To apply the extended named IP access list to the interface, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ip access-group name in` or `Router(config-if)# ip access-group name out`	External interface: Applies the extended access list to the interface's inbound traffic. Internal interface: Applies the extended access list to the interface's outbound traffic.

Setting a Global Timeout Value

Reflexive access list entries expire after no packets in the session have been detected for a certain length of time (the "timeout" period). You can specify the timeout for a particular reflexive access list when you define the reflexive access list. But if you do not specify the timeout for a given reflexive access list, the list will use the global timeout value instead.

The global timeout value is 300 seconds by default. But, you can change the global timeout to a different value at any time.

To change the global timeout value, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip reflexive-list timeout seconds`	Changes the global timeout value for temporary reflexive access list entries. Use a positive integer from 0 to 2,147,483.

Reflexive Access List Configuration Examples

The following sections provide reflexive access list configuration examples:

- External Interface Configuration Example
- Internal Interface Configuration Example

External Interface Configuration Example

This example shows reflexive access lists configured for an external interface, for a topology similar to the one in Figure 18 (shown earlier in this chapter).

This configuration example permits both inbound and outbound TCP traffic at interface Serial 1, but only if the first packet (in a given session) originated from inside your network. The interface Serial 1 connects to the Internet.

Define the interface where the session-filtering configuration is to be applied:

```
interface serial 1
 description Access to the Internet via this interface
```

Apply access lists to the interface, for inbound traffic and for outbound traffic:

```
ip access-group inboundfilters in
ip access-group outboundfilters out
```

Define the outbound access list. This is the access list that evaluates all outbound traffic on interface Serial 1.

```
ip access-list extended outboundfilters
```

Define the reflexive access list *tcptraffic*. This entry permits *all* outbound TCP traffic and creates a new access list named *tcptraffic*. Also, when an outbound TCP packet is the first in a new session, a corresponding temporary entry will be automatically created in the reflexive access list *tcptraffic*.

```
permit tcp any any reflect tcptraffic
```

Define the inbound access list. This is the access list that evaluates all inbound traffic on interface Serial 1.

```
ip access-list extended inboundfilters
```

Define the inbound access list entries. This example shows Enhanced IGRP permitted on the interface. Also, no ICMP traffic is permitted. The last entry points to the reflexive access list. If a packet does not match the first two entries, the packet will be evaluated against all the entries in the reflexive access list *tcptraffic*.

```
permit eigrp any any
deny icmp any any
evaluate tcptraffic
```

Define the global idle timeout value for all reflexive access lists. In this example, when the reflexive access list *tcptraffic* was defined, no timeout was specified, so *tcptraffic* uses the global timeout. Therefore, if for 120 seconds there is no TCP traffic that is part of an established session, the corresponding reflexive access list entry will be removed.

```
ip reflexive-list timeout 120
```

The example configuration looks as follows:

```
interface Serial 1
 description Access to the Internet via this interface
 ip access-group inboundfilters in
 ip access-group outboundfilters out
!
ip reflexive-list timeout 120
!
ip access-list extended outboundfilters
 permit tcp any any reflect tcptraffic
!
ip access-list extended inboundfilters
 permit eigrp any any
 deny icmp any any
 evaluate tcptraffic
```

With this configuration, before any TCP sessions have been initiated the **show access-list** EXEC command displays the following:

```
Extended IP access list inboundfilters
 permit eigrp any any
 deny icmp any any
 evaluate tcptraffic
Extended IP access list outboundfilters
 permit tcp any any reflect tcptraffic
```

Notice that the reflexive access list does not appear in this output. This is because before any TCP sessions have been initiated, no traffic has triggered the reflexive access list, and the list is empty (has no entries). When empty, reflexive access lists do not show up in **show access-list** output.

After a Telnet connection is initiated from within your network to a destination outside of your network, the **show access-list** EXEC command displays the following:

```
Extended IP access list inboundfilters
 permit eigrp any any
 deny icmp any any
 evaluate tcptraffic
Extended IP access list outboundfilters
 permit tcp any any reflect tcptraffic
Reflexive IP access list tcptraffic
 permit tcp host 172.19.99.67 eq telnet host 192.168.60.185 eq 11005 (5 matches) (time
left 115 seconds)
```

Notice that the reflexive access list *tcptraffic* now appears and displays the temporary entry generated when the Telnet session initiated with an outbound packet.

Internal Interface Configuration Example

This is an example configuration for reflexive access lists configured for an internal interface. This example has a topology similar to the one in Figure 19 (shown earlier in this chapter).

This example is similar to the previous example; the only difference between this example and the previous example is that the entries for the outbound and inbound access lists are swapped. Please refer to the previous example for more details and descriptions.

```
interface Ethernet 0
 description Access from the I-net to our Internal Network via this interface
 ip access-group inboundfilters in
 ip access-group outboundfilters out
!
ip reflexive-list timeout 120
!
ip access-list extended outboundfilters
 permit eigrp any any
 deny icmp any any
 evaluate tcptraffic
!
ip access-list extended inboundfilters
 permit tcp any any reflect tcptraffic
```

Configuring TCP Intercept (Preventing Denial-of-Service Attacks)

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes how to configure your router to protect TCP servers from TCP SYN-flooding attacks, a type of denial-of-service attack. This is accomplished by configuring the Cisco IOS feature known as TCP Intercept.

For a complete description of TCP Intercept commands, refer to the "TCP Intercept Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the chapter "Identifying Supported Platforms" section in the "Using Cisco IOS Software."

In This Chapter

This chapter has the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

About TCP Intercept

The TCP intercept feature implements software to protect TCP servers from TCP SYN-flooding attacks, which are a type of denial-of-service attack.

A SYN-flooding attack occurs when a hacker floods a server with a barrage of requests for connection. Because these messages have unreachable return addresses, the connections cannot be established. The resulting volume of unresolved open connections eventually overwhelms the server and can cause it to deny service to valid requests, thereby preventing legitimate users from connecting to a web site, accessing e-mail, using FTP service, and so on.

The TCP intercept feature helps prevent SYN-flooding attacks by intercepting and validating TCP connection requests. In intercept mode, the TCP intercept software intercepts TCP synchronization (SYN) packets from clients to servers that match an extended access list. The software establishes a connection with the client on behalf of the destination server, and if successful, establishes the connection with the server on behalf of the client and knits the two half-connections together transparently. Thus, connection attempts from unreachable hosts will never reach the server. The software continues to intercept and forward packets throughout the duration of the connection. The number of SYNs per second and the number of concurrent connections proxied depends on the platform, memory, processor, and other factors.

In the case of illegitimate requests, the software's aggressive timeouts on half-open connections and its thresholds on TCP connection requests protect destination servers while still allowing valid requests.

When establishing your security policy using TCP intercept, you can choose to intercept all requests or only those coming from specific networks or destined for specific servers. You can also configure the connection rate and threshold of outstanding connections.

You can choose to operate TCP intercept in watch mode, as opposed to intercept mode. In watch mode, the software passively watches the connection requests flowing through the router. If a connection fails to get established in a configurable interval, the software intervenes and terminates the connection attempt.

TCP options that are negotiated on handshake (such as RFC 1323 on window scaling) will not be negotiated because the TCP intercept software does not know what the server can do or will negotiate.

TCP Intercept Configuration Task List

To configure TCP intercept, perform the tasks in the following sections. The first task is required; the rest are optional.

- Enabling TCP Intercept (Required)
- Setting the TCP Intercept Mode (Optional)
- Setting the TCP Intercept Drop Mode (Optional)
- Changing the TCP Intercept Timers (Optional)
- Changing the TCP Intercept Aggressive Thresholds (Optional)
- Monitoring and Maintaining TCP Intercept (Optional)

For TCP intercept configuration examples using the commands in this chapter, refer to the "TCP Intercept Configuration Example" section at the end of this chapter.

Enabling TCP Intercept

To enable TCP intercept, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# access-list access-list-number {deny | permit} tcp any destination destination-wildcard`	Defines an IP extended access list.
Step 2	`Router(config)# ip tcp intercept list access-list-number`	Enables TCP intercept.

You can define an access list to intercept all requests or only those coming from specific networks or destined for specific servers. Typically the access list will define the source as **any** and define specific destination networks or servers. That is, you do not attempt to filter on the source addresses because you do not necessarily know who to intercept packets from. You identify the destination in order to protect destination servers.

If no access list match is found, the router allows the request to pass with no further action.

Setting the TCP Intercept Mode

The TCP intercept can operate in either active intercept mode or passive watch mode. The default is intercept mode.

In intercept mode, the software actively intercepts each incoming connection request (SYN) and responds on behalf of the server with an SYN-ACK, then waits for an ACK from the client. When that ACK is received, the original SYN is sent to the server and the software performs a three-way handshake with the server. When this is complete, the two half-connections are joined.

In watch mode, connection requests are allowed to pass through the router to the server but are watched until they become established. If they fail to become established within 30 seconds (configurable with the **ip tcp intercept watch-timeout** command), the software sends a Reset to the server to clear up its state.

To set the TCP intercept mode, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip tcp intercept mode {intercept | watch}`	Sets the TCP intercept mode.

Setting the TCP Intercept Drop Mode

When under attack, the TCP intercept feature becomes more aggressive in its protective behavior. If the number of incomplete connections exceeds 1100 or the number of connections arriving in the last one minute exceeds 1100, each new arriving connection causes the oldest partial connection to be deleted. Also, the initial retransmission timeout is reduced by half to 0.5 seconds (so the total time trying to establish a connection is cut in half).

By default, the software drops the oldest partial connection. Alternatively, you can configure the software to drop a random connection. To set the drop mode, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip tcp intercept drop-mode {oldest | random}`	Sets the drop mode.

Changing the TCP Intercept Timers

By default, the software waits for 30 seconds for a watched connection to reach established state before sending a Reset to the server. To change this value, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip tcp intercept watch-timeout seconds`	Changes the time allowed to reach established state.

By default, the software waits for 5 seconds from receipt of a reset or FIN-exchange before it ceases to manage the connection. To change this value, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip tcp intercept finrst-timeout seconds`	Changes the time between receipt of a reset or FIN-exchange and dropping the connection.

By default, the software still manages a connection for 24 hours after no activity. To change this value, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip tcp intercept connection-timeout seconds`	Changes the time the software will manage a connection after no activity.

Changing the TCP Intercept Aggressive Thresholds

Two factors determine when aggressive behavior begins and ends: total incomplete connections and connection requests during the last one-minute sample period. Both thresholds have default values that can be redefined.

When a threshold is exceeded, the TCP intercept assumes the server is under attack and goes into aggressive mode. When in aggressive mode, the following occurs:

- Each new arriving connection causes the oldest partial connection to be deleted. (You can change to a random drop mode.)
- The initial retransmission timeout is reduced by half to 0.5 seconds, and so the total time trying to establish the connection is cut in half. (When not in aggressive mode, the code does exponential back-off on its retransmissions of SYN segments. The initial retransmission timeout is 1 second. The subsequent timeouts are 2 seconds, 4 seconds, 8 seconds, and 16 seconds. The code retransmits 4 times before giving up, so it gives up after 31 seconds of no acknowledgment.)
- If in watch mode, the watch timeout is reduced by half. (If the default is in place, the watch timeout becomes 15 seconds.)

The drop strategy can be changed from the oldest connection to a random connection with the **ip tcp intercept drop-mode** command.

Note The two factors that determine aggressive behavior are related and work together. When *either* of the **high** values is exceeded, aggressive behavior begins. When *both* quantities fall below the **low** value, aggressive behavior ends.

You can change the threshold for triggering aggressive mode based on the total number of incomplete connections. The default values for **low** and **high** are 900 and 1100 incomplete connections, respectively. To change these values, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip tcp intercept max-incomplete low` *number*	Sets the threshold for stopping aggressive mode.
Step 2	`Router(config)# ip tcp intercept max-incomplete high` *number*	Sets the threshold for triggering aggressive mode.

You can also change the threshold for triggering aggressive mode based on the number of connection requests received in the last 1-minute sample period. The default values for **low** and **high** are 900 and 1100 connection requests, respectively. To change these values, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip tcp intercept one-minute low` *number*	Sets the threshold for stopping aggressive mode.
Step 2	`Router(config)# ip tcp intercept one-minute high` *number*	Sets the threshold for triggering aggressive mode.

Monitoring and Maintaining TCP Intercept

To display TCP intercept information, use either of the following commands in EXEC mode:

Command	Purpose
`Router# show tcp intercept connections`	Displays incomplete connections and established connections.
`Router# show tcp intercept statistics`	Displays TCP intercept statistics.

TCP Intercept Configuration Example

The following configuration defines extended IP access list 101, causing the software to intercept packets for all TCP servers on the 192.168.1.0/24 subnet:

```
ip tcp intercept list 101
!
access-list 101 permit tcp any 192.168.1.0 0.0.0.255
```

IP Access List Entry Sequence Numbering

Users can apply sequence numbers to **permit** or **deny** statements and also reorder, add, or remove such statements from a named IP access list. This feature makes revising IP access lists much easier. Prior to this feature, users could add access list entries to the end of an access list only; therefore needing to add statements anywhere except the end required reconfiguring the access list entirely.

Feature History for the IP Access List Entry Additions Feature

Release	Modification
12.2(14)S	This feature was introduced.
12.2(15)T	This feature was integrated into Cisco IOS Release 12.2(15)T.
12.3(2T	This feature was integrated into Cisco IOS Release 12.3(2)T.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for IP Access List Entry Sequence Numbering

- This feature does not support dynamic, reflexive, or firewall access lists.
- This feature does not support old-style numbered access lists, which existed before named access lists. Keep in mind that you can name an access list with a number, so numbers are allowed when they are entered in the standard or extended named access list (NACL) configuration mode.

Information About IP Access Lists

Before you resequence or add entries to an IP access list, you should understand the following concepts:

- Purpose of IP Access Lists, page 2
- How an IP Access List Works, page 2
- IP Access List Entry Sequence Numbering, page 4

Purpose of IP Access Lists

Access lists perform packet filtering to control which packets move through the network and where. Such control can help limit network traffic and restrict the access of users and devices to the network. Access lists have many uses, and therefore many commands accept a reference to an access list in their command syntax. Access lists can be used to do the following:

- Filter incoming packets on an interface.
- Filter outgoing packets on an interface.
- Restrict the contents of routing updates.
- Limit debug output based on an address or protocol.
- Control virtual terminal line access.
- Identify or classify traffic for advanced features, such as congestion avoidance, congestion management, and priority and custom queuing.
- Trigger dial-on-demand routing (DDR) calls.

How an IP Access List Works

An access list is a sequential list consisting of at least one **permit** statement and possibly one or more **deny** statements that apply to IP addresses and possibly upper-layer IP protocols. The access list has a name by which it is referenced. Many software commands accept an access list as part of their syntax.

An access list can be configured and named, but it is not in effect until the access list is referenced by a command that accepts an access list. Multiple commands can reference the same access list. An access list can control traffic arriving at the router or leaving the router, but not traffic originating at the router.

IP Access List Process and Rules

- The software tests the source or destination address or the protocol of each packet being filtered against the conditions in the access list, one condition (**permit** or **deny** statement) at a time.

- If a packet does not match an access list statement, the packet is then tested against the next statement in the list.
- If a packet and an access list statement match, the rest of the statements in the list are skipped and the packet is permitted or denied as specified in the matched statement. The first entry that the packet matches determines whether the software permits or denies the packet. That is, after the first match, no subsequent entries are considered.
- If the access list denies the address or protocol, the software discards the packet and returns an ICMP Host Unreachable message.
- If no conditions match, the software drops the packet. This is because each access list ends with an unwritten or implicit **deny** statement. That is, if the packet has not been permitted by the time it was tested against each statement, it is denied.
- The access list must contain at least one **permit** statement or else all packets are denied.
- Because the software stops testing conditions after the first match, the order of the conditions is critical. The same **permit** or **deny** statements specified in a different order could result in a packet being passed under one circumstance and denied in another circumstance.
- If an access list is referenced by name in a command, but the access list does not exist, all packets pass.
- Only one access list per interface, per protocol, per direction is allowed.
- Inbound access lists process packets arriving at the router. Incoming packets are processed before being routed to an outbound interface. An inbound access list is efficient because it saves the overhead of routing lookups if the packet is to be discarded because it is denied by the filtering tests. If the packet is permitted by the tests, it is then processed for routing. For inbound lists, **permit** means continue to process the packet after receiving it on an inbound interface; **deny** means discard the packet.
- Outbound access lists process packets before they leave the router. Incoming packets are routed to the outbound interface and then processed through the outbound access list. For outbound lists, **permit** means send it to the output buffer; **deny** means discard the packet.

Helpful Hints for Creating IP Access Lists

- Create the access list before applying it to an interface. An interface with an empty access list applied to it permits all traffic.
- Another reason to configure an access list before applying it is because if you applied a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause you immediate access problems.
- Because the software stops testing conditions after it encounters the first match (to either a **permit** or **deny** statement), you will reduce processing time and resources if you put the statements that packets are most likely to match at the beginning of the access list. Place more frequently occurring conditions before less frequent conditions.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- In order to make the purpose of individual statements more easily understood at a glance, you can write a helpful remark before or after any statement.

Source and Destination Addresses

Source address and destination addresses are two of the most typical fields in an IP packet on which to base an access list. Specify source addresses to control packets from certain networking devices or hosts. Specify destination addresses to control packets being sent to certain networking devices or hosts.

Wildcard Mask and Implicit Wildcard Mask

Address filtering uses wildcard masking to indicate to the software whether to check or ignore corresponding IP address bits when comparing the address bits in an access list entry to a packet being submitted to the access list. By carefully setting wildcard masks, an administrator can select single or several IP addresses for permit or deny tests.

Wildcard masking for IP address bits uses the number 1 and the number 0 to specify how the software treats the corresponding IP address bits. A wildcard mask is sometimes referred to as an inverted mask because a 1 and 0 mean the opposite of what they mean in a subnet (network) mask.

- A wildcard mask bit 0 means *check* the corresponding bit value.
- A wildcard mask bit 1 means *ignore* that corresponding bit value.

If you do not supply a wildcard mask with a source or destination address in an access list statement, the software assumes a default wildcard mask of 0.0.0.0.

Unlike subnet masks, which require contiguous bits indicating network and subnet to be ones, wildcard masks allow noncontiguous bits in the mask.

Transport Layer Information

You can filter packets based on transport layer information, such as whether the packet is a TCP, UDP, ICMP or IGMP packet.

IP Access List Entry Sequence Numbering

Benefits

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If a user wanted to insert an entry (statement) in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

This feature allows users to add sequence numbers to access list entries and resequence them. When a user adds a new entry, the user chooses the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry.

Sequence Numbering Behavior

- For backward compatibility with previous releases, if entries with no sequence numbers are applied, the first entry is assigned a sequence number of 10, and successive entries are incremented by 10. The maximum sequence number is 2147483647. If the generated sequence number exceeds this maximum number, the following message is displayed:

```
Exceeded maximum sequence number.
```

- If the user enters an entry without a sequence number, it is assigned a sequence number that is 10 greater than the last sequence number in that access list and is placed at the end of the list.
- If the user enters an entry that matches an already existing entry (except for the sequence number), then no changes are made.
- If the user enters a sequence number that is already present, the following error message is generated:

  ```
  Duplicate sequence number.
  ```

- If a new access list is entered from global configuration mode, then sequence numbers for that access list are generated automatically.
- Distributed support is provided so that the sequence numbers of entries in the Route Processor (RP) and line card (LC) are in synchronization at all times.
- Sequence numbers are not nvgened. That is, the sequence numbers themselves are not saved. In the event that the system is reloaded, the configured sequence numbers revert to the default sequence starting number and increment. The function is provided for backward compatibility with software releases that do not support sequence numbering.
- This feature works with named standard and extended IP access lists. Because the name of an access list can be designated as a number, numbers are acceptable.

How to Use Sequence Numbers in an IP Access List

This section describes how to use sequence numbers in an IP access list.

- Sequencing Access-List Entries and Revising the Access List, page 5

Sequencing Access-List Entries and Revising the Access List

This task shows how to assign sequence numbers to entries in a named IP access list and how to add or delete an entry to or from an access list. It is assumed a user wants to revise an access list. The context of this task is the following:

- A user need not resequence access lists for no reason; resequencing in general is optional. The resequencing step in this task is shown as required because that is one purpose of this feature and this task demonstrates the feature.
- Step 5 happens to be a **permit** statement and Step 6 happens to be a **deny** statement, but they need not be in that order.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list resequence** *access-list-name starting-sequence-number increment*
4. **ip access-list {standard** | **extended}** *access-list-name*
5. *sequence-number* **permit** *source source-wildcard*

or

sequence-number **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

6. *sequence-number* **deny** *source source-wildcard*

   or

   sequence-number **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

7. Repeat Step 5 and/or Step 6 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.
8. **end**
9. **show ip access-lists** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list resequence** *access-list-name starting-sequence-number increment* **Example:** `Router(config)# ip access-list resequence kmd1 100 15`	Resequences the specified IP access list using the starting sequence number and the increment of sequence numbers. • This example resequences an access list named kmd1. The starting sequence number is 100 and the increment is 15.
Step 4	**ip access-list** {**standard** | **extended**} *access-list-name* **Example:** `Router(config)# ip access-list standard kmd1`	Specifies the IP access list by name and enters named access list configuration mode. • If you specify **standard**, make sure you subsequently specify **permit** and/or **deny** statements using the standard access list syntax. • If you specify **extended**, make sure you subsequently specify **permit** and/or **deny** statements using the extended access list syntax.

	Command or Action	Purpose
Step 5	*sequence-number* **permit** *source source-wildcard* or *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-std-nacl)# 105 permit 10.5.5.5 0.0.0 255`	Specifies a permit statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the permit (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be `Router(config-ext-nacl)` and you would use the extended **permit** command syntax.
Step 6	*sequence-number* **deny** *source source-wildcard* or *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-std-nacl)# 105 deny 10.6.6.7 0.0.0 255`	(Optional) Specifies a deny statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the deny (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be `Router(config-ext-nacl)` and you would use the extended **deny** command syntax.
Step 7	Repeat Step 5 and/or Step 6 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise the access list.

	Command or Action	Purpose
Step 8	**end** **Example:** `Router(config-std-nacl)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
Step 9	**show ip access-lists** *access-list-name* **Example:** `Router# show ip access-lists kmd1`	(Optional) Displays the contents of the IP access list. • Review the output to see that the access list includes the new entry. `Router# show ip access-lists kmd1` `Standard IP access list kmd1` `100 permit 10.4.4.0, wildcard bits 0.0.0.255` `105 permit 10.5.5.0, wildcard bits 0.0.0.255` `115 permit 10.0.0.0, wildcard bits 0.0.0.255` `130 permit 10.5.5.0, wildcard bits 0.0.0.255` `145 permit 10.0.0.0, wildcard bits 0.0.0.255`

What to Do Next

If your access list is not already applied to an interface or line or otherwise referenced, apply the access list. Refer to the "Configuring IP Services" chapter of the *Cisco IOS IP Configuration Guide* for information about how to apply an IP access list.

Configuration Examples for IP Access List Entry Sequence Numbering

This section provides the following examples related to sequence numbering of entries in an IP access list:

Resequencing Entries in an Access List: Example

The following example shows access list resequencing. The starting value is 1, and increment value is 2. The subsequent entries are ordered based on the increment values that users provide, and the range is from 1 to 2147483647.

When an entry with no sequence number is entered, by default it has a sequence number of 10 more than the last entry in the access list.

```
Router# show access-list 150

Extended IP access list 150
    10 permit ip host 10.3.3.3 host 172.16.5.34
    20 permit icmp any any
    30 permit tcp any host 10.3.3.3
    40 permit ip host 10.4.4.4 any
    50 Dynamic test permit ip any any
```

```
    60 permit ip host 172.16.2.2 host 10.3.3.12
    70 permit ip host 10.3.3.3 any log
    80 permit tcp host 10.3.3.3 host 10.1.2.2
    90 permit ip host 10.3.3.3 any
    100 permit ip any any

Router(config)# ip access-list extended 150
Router(config)# ip access-list resequence 150 1 2
Router(config)# end

Router# show access-list 150

Extended IP access list 150
    1 permit ip host 10.3.3.3 host 172.16.5.34
    3 permit icmp any any
    5 permit tcp any host 10.3.3.3
    7 permit ip host 10.4.4.4 any
    9 Dynamic test permit ip any any
    11 permit ip host 172.16.2.2 host 10.3.3.12
    13 permit ip host 10.3.3.3 any log
    15 permit tcp host 10.3.3.3 host 10.1.2.2
    17 permit ip host 10.3.3.3 any
    19 permit ip any any
```

Adding Entries with Sequence Numbers: Example

In the following example, an new entry is added to a specified access list:

```
Router# show ip access-list

Standard IP access list tryon
2 permit 10.4.4.2, wildcard bits 0.0.255.255
5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
20 permit 10.0.0.2, wildcard bits 0.0.0.255

Router(config)# ip access-list standard tryon

Router(config-std-nacl)# 15 permit 10.5.5.5 0.0.0.255

Router# show ip access-list

Standard IP access list tryon
2 permit 10.4.0.0, wildcard bits 0.0.255.255
5 permit 10.0.0.0, wildcard bits 0.0.0.255
10 permit 10.0.0.0, wildcard bits 0.0.0.255
15 permit 10.5.5.0, wildcard bits 0.0.0.255
20 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Entry without Sequence Number: Example

The following example shows how an entry with no specified sequence number is added to the end of an access list. When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list.

```
Router(config)# ip access-list standard 1

Router(config-std-nacl)# permit 1.1.1.1 0.0.0.255
```

```
Router(config-std-nacl)# permit 2.2.2.2 0.0.0.255
Router(config-std-nacl)# permit 3.3.3.3 0.0.0.255

Router# show access-list
Standard IP access list 1
10 permit 0.0.0.0, wildcard bits 0.0.0.255
20 permit 0.0.0.0, wildcard bits 0.0.0.255
30 permit 0.0.0.0, wildcard bits 0.0.0.255

Router(config)# ip access-list standard 1
Router(config-std-nacl)# permit 4.4.4.4 0.0.0.255
Router(config-std-nacl)# end

Router# show access-list

Standard IP access list 1
10 permit 0.0.0.0, wildcard bits 0.0.0.255
20 permit 0.0.0.0, wildcard bits 0.0.0.255
30 permit 0.0.0.0, wildcard bits 0.0.0.255
40 permit 0.4.0.0, wildcard bits 0.0.0.255
```

Additional References

The following sections provide references related to IP access lists.

Related Documents

Related Topic	Document Title
Configuring IP access lists	*"Configuring IP Services" chapter in the Cisco IOS IP Configuration Guide, Release 12.2*
IP access list commands	*"IP Services Commands" chapter in the Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services, Release 12.2*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and lots more. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature.

New Command

- **ip access-list resequence**

Revised Commands

- **deny (IP)**
- **permit (IP)**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Turbo Access Control List Scalability Enhancements

First Published: December 5, 2006
Last Updated: December 5, 2006

The Turbo Access Control List (ACL) Scalability Enhancements feature introduced in Cisco IOS Release 12.2(31)SB2 improves overall performance on the Cisco 7304 router using a Network Services Engine (NSE) by allowing Turbo ACLs to be processed in PXF using less memory, thereby allowing more traffic traversing the Cisco 7304 router using an NSE to be PXF-accelerated. This feature also introduces user-configuration options that allow users to define the amount of memory used for Turbo ACL purposes in the Route Processor (RP) processing path.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Turbo ACL Scalability Enhancements on the NSEs" section on page 20.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Turbo Access Control List Scalability Enhancements on the NSEs

Because the portion of this feature that more expediently removes older entries works in the PXF processing path, PXF must be enabled for this particular functionality to have any benefit. PXF processing is enabled by default.

Restrictions for Turbo Access Control List Scalability Enhancements on the NSEs

This feature is not available for Cisco 7304 routers using an NPE-G100.

Information About Turbo Access Control List Scalability Enhancements on the NSEs

To benefit from the Turbo Access Control List Scalability Enhancements for the NSEs, you should understand the following concepts:

How Turbo ACL on the Cisco 7304 Router Using an NSE Works

With the exception that most Turbo ACL classification is PXF-accelerated on a Cisco 7304 router using an NSE-100 or an NSE-150, Turbo ACL classification on the Cisco 7304 router using an NSE-100 or NSE-150 is similar in behavior to Turbo ACL on other platforms. For information on Turbo ACL, see *Turbo Access Control Lists*.

For information on PXF on Cisco 7304 routers using an NSE-100 or an NSE-150, including the Turbo ACL features that are PXF-accelerated, see *PXF Information for the Cisco 7304 Router*.

How Turbo ACL Scalability Enhancements on the NSEs Improves Overall PXF Performance

The memory allocated in PXF for Turbo Access Control Lists (ACLs) on the NSE-100 especially is limited to the point where even modestly-sized ACL configurations cause a large amount of PXF memory to be used for Turbo ACL processing. As a result, a large amount of network traffic that should be processed through the PXF processing path is instead processed through the RP path.

This enhancement is part of a series of enhancements to improve Turbo ACL functionality on the Cisco 7304 router using the NSE-100. Specifically, this feature keeps the entries for PXF-based Turbo ACL classification current by more actively removing older entries. The older entries, which are no longer used for current traffic flows, still consume memory and, therefore, cause traffic that would normally be PXF-accelerated to instead be punted to the RP. This portion of the feature, which does not require user configuration, improves overall traffic flow on the Cisco 7304 router using an NSE by allowing more network traffic to be PXF-accelerated.

How Turbo ACL Scalability Enhancements on the NSEs Improves Overall Route Processing Performance

These Turbo ACL scalability enhancements also introduce an enhancement that allows users, via configuration commands, to configure the amount of memory reserved for ACL processing on the RP. The ability to configure the amount of memory reserved for ACL processing in the RP path gives users the option either to improve ACL processing performance in the RP path by reserving more memory for ACL processing, or to improve all other RP path functionality by reserving less memory for ACL processing.

In Cisco IOS releases not containing this feature, the amount of memory reserved for RP ACL handling is fixed.

Understanding Memory Limits for Turbo ACL Processes on the Route Processor

An NSE-150 has 2 GB of DRAM. NSE-100 RAM is user-configurable using an SDRAM SODIMM. While most NSE-100s have 512 MB of RAM, 256-MB and 128-MB SDRAM SODIMMs for the NSE-100 exist.

On a Cisco 7304 router using an NSE-150, the default memory limit for Turbo ACL processes (such as classification, compilation, and table storage) of Layer 3 and Layer 4 data in the RP path is always 256 MB. The default memory limit for Turbo ACL processes for Layer 2 data in the RP path for a Cisco 7304 router using an NSE-150 is always 128 MB.

On a Cisco 7304 router using an NSE-100, the default amount of memory reserved for Turbo ACL processes in the RP path is dependant upon the amount of SDRAM configured on the NSE-100. If the NSE has 512 MB of SDRAM or more, the default memory limit for Turbo ACL processes for Layer 3 and Layer 4 traffic processing is 256 MB. If the processor has less than 512 MB of SDRAM, the default memory limit for Turbo ACL processes for Layer 3 and Layer 4 traffic is 128 MB.

The default amount of memory reserved for Layer 2 Turbo ACL processes for a Cisco 7304 router using an NSE-100 is always 128 MB, regardless of the amount of memory configured on the processor.

To see the default amount of memory reserved for Layer 2 or for Layer 3 and Layer 4 Turbo ACL processing on your Cisco 7304 router, enter the **show access-list compiled** command. The “Mb default limit” output, which appears in both the “Compiled ACL statistics for IPv4” and “Compiled ACL

statistics for Data-Link" sections of the output, shows you the default memory reservations for either Layer 2 or Layer 3 and Layer 4 Turbo ACL processing. See the "Monitoring Turbo ACL Memory Usage in the Route Processing Path" section on page 5 for a more detailed explanation of this procedure.

To change the default amount of memory reserved for Layer 2 or Layer 3 and Layer 4 Turbo ACL processing on your Cisco 7304 router, enter the **access-list compiled** [**ipv4** | **data-link**] **limit memory** *number* command.

To restore the default amount of memory reserved for Layer 2 or Layer 3 and Layer 4 Turbo ACL processing on your Cisco 7304 router, enter the **default access-list compiled** [**ipv4** | **data-link**] **limit memory** command.

To learn more about the SDRAM SODIMMs that determine the amount of SDRAM available for Cisco 7304 routers using an NSE-100, see *NSE-100 Memory Information*.

Benefits

Improved Traffic Flow

This feature improves the Turbo ACL processing process in PXF by more expediently removing older entries. As a result, more Turbo ACL processing can be done in the PXF processing path, thereby allowing more router traffic to be accelerated using the PXF processing path.

Configuration of Route Processor Memory Limits for ACL Processing

This feature allows users to set the amount of memory reserved for ACL processes (such as compilation, storage, and classification) in the RP path. Users who need more memory for ACL processes now have the ability to set aside additional memory resources in the RP path for ACL processes. Users who need more more memory for other processes in the RP path now can set aside less memory for ACL processes.

How to Configure Turbo Access Control List Scalability Enhancements on the NSEs

It is important to note that the portion of this feature that more expediently removes older ACL entries for ACLs being processed in the PXF processing path occurs automatically without user configuration.

The following sections contain procedures for configuring memory reservations for Turbo ACL processing on the RP:

Monitoring Turbo ACL Memory Usage in the Route Processing Path

Before setting the actual memory limits for RP-based Turbo ACL usage, it may be helpful to gather information regarding the amount of memory being used for Turbo ACL usage.

To monitor your Turbo ACL memory usage in the RP path, you must complete the following steps.

SUMMARY STEPS

1. **enable**
2. **show access-list compiled**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`show access-list compiled` **Example:** `Router# show access-list compiled`	Displays the status and condition of the Turbo ACL tables associated with each access list. When using this command to verify memory limitation settings for Turbo ACL processing, look for the following: • The output for **show access-list compiled** is separated for Layer 2 and for Layer 3 and Layer 4 data. Layer 3 and Layer 4 ACL compilation tables and information can be seen in the "Compiled ACL statistics for IPv4" section of the output, while Layer 2 ACL compilation tables and information can be seen in the "Compiled ACL statistics for Data-Link" section. • The "mem limits" output that shows the number of times a compile has occurred and the ACL has reached its configured limit. • The "Mb limit" output that shows the current memory limit setting. • The "Mb max memory" output that shows the maximum amount of memory the current ACL configuration could actually consume under maximum usage conditions. For additional information and an example, see the "Monitoring Memory Limitations for Layer 2 or Layer 3 and Layer 4 ACL Processing: Example" section on page 14.

Configuring a User-Defined Memory Limitations for Turbo ACL Processing of Layer 3 and Layer 4 Data in the Route Processing Path

To enable memory limitations for Turbo ACL processing of Layer 3 and Layer 4 data in the RP path, you must complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list compiled ipv4 limit memory** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`access-list compiled ipv4 limit memory` *number* **Example:** `Router(config)# access-list compiled ipv4 limit memory 300`	Specifies the limit, in megabytes, reserved for Turbo ACL instance 0, which is used for processing Layer 3 and Layer 4 data.

Removing Memory Limits for Turbo ACL Processing of Layer 3 and Layer 4 Data in the Route Processing Path

Removing all memory limits for Turbo ACL processes in the Route Processor allows all route processing memory to be used for Turbo ACL processing of Layer 3 and Layer 4 data, if necessary. It is important to note that this functionality is not used to remove a previously configured limit, even though it is a **no** form of a command.

To remove all memory limits for Turbo ACL processing for Layer 3 and Layer 4 data and to allow as much memory as needed for Layer 3 and Layer 4 Turbo ACL processing in the RP path, you must complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no access-list compiled ipv4 limit memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no access-list compiled ipv4 limit memory** **Example:** `Router(config)# no access-list compiled ipv4 limit memory`	Removes any memory limits for Layer 3 and Layer 4 Turbo ACL processing, thereby allowing all available memory to be used for Layer 3 and Layer 4 Turbo ACL processing, if necessary.

Restoring the Default Memory Limits for Turbo ACL Processing of Layer 3 and 4 Data in the Route Processing Path

The default memory limit for Turbo ACL processing of Layer 3 and Layer 4 data in the RP path is always 256 MB on the NSE-150.

On the NSE-100, the default memory limit for Turbo ACL processing of Layer 3 and Layer 4 data in the RP path is dependant on the amount of memory on your NSE-100. If you have more than 512 MB of memory configured on your processor, your default memory limit for RP-based Turbo ACL processing is 256 MB. If you have less than 512 MB of memory, your default memory limit for RP-based Turbo ACL processing is 128 MB.

To restore the default RP memory limit settings for Turbo ACL processing of Layer 3 and Layer 4 traffic, you must complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **default access-list compiled ipv4 limit memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**default access-list compiled ipv4 limit memory** **Example:** `Router(config)# default access-list compiled ipv4 limit memory`	Restores the default memory limit setting for Layer 3 and Layer 4 Turbo ACL traffic processing. The default memory limit for Turbo ACL processing of Layer 3 and Layer 4 data in the RP path is always 256 MB on the NSE-150. On the NSE-100, the default memory limit for Turbo ACL processing of Layer 3 and Layer 4 data in the RP path is dependant on the amount of memory on your NSE-100. If you have more than 512 MB of memory configured on your processor, your default memory limit for RP-based Turbo ACL processing is 256 MB. If you have less than 512 MB of memory, your default memory limit for RP-based Turbo ACL processing is 128 MB.

Configuring a User-Defined Memory Limitation for Turbo ACL Processing of Layer 2 Data in the Route Processing Path

To enable a memory limitation setting for Turbo ACL processing of Layer 2 data in the RP path, you must complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list compiled data-link limit memory** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list compiled data-link limit memory** *number* **Example:** `Router(config)# access-list compiled data-link limit memory 150`	Specifies the limit, in megabytes, reserved for Turbo ACL instance 1, which is used by the Turbo ACL algorithm to classify Layer 2 frames.

Removing Memory Limits for Turbo ACL Processing of Layer 2 Data in the Route Processing Path

Removing all memory limits for Turbo ACL processing of Layer 2 data in the Route Processor allows all route processing memory to be used for Turbo ACL processing of Layer 2 data, if necessary. It is important to note that this functionality is not used to remove a previously configured limit, even though it is a **no** form of a command.

To remove all RP-based memory limits for Turbo ACL processing for Layer 2 data and to allow as much memory as needed for Layer 2 Turbo ACL processing, you must complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no access-list compiled data-link limit memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`no access-list compiled data-link limit memory` **Example:** `Router(config)# no access-list compiled data-link limit memory`	Removes any memory limits for Layer 2 Turbo ACL processing, thereby allowing all available memory to be used for Layer 2 Turbo ACL processing, if necessary.

Restoring the Default Memory Limits for Turbo ACL Processing of Layer 2 Data in the Route Processing Path

The default memory limit for Turbo ACL processing of Layer 2 data in the RP processing path is 128 MB for the NSE-100 and NSE-150.

To restore the default RP-based memory limit setting for Turbo ACL processing of Layer 2 data, you must complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **default access-list compiled data-link limit memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**default access-list compiled data-link limit memory** **Example:** `Router(config)# default access-list compiled data-link limit memory`	Restores the default memory limit setting for Layer 2 Turbo ACL processing. The default memory limit setting for Layer 2 Turbo ACL processing is always 128 MB.

Verifying Memory Limitation Settings for Turbo ACL Processing

To verify RP-based memory limitation settings for Turbo ACL processing, you must complete the following steps.

SUMMARY STEPS

1. **enable**
2. **show access-list compiled**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`show access-list compiled` **Example:** `Router# show access-list compiled`	Displays the status and condition of the Turbo ACL tables associated with each access list. When using this command to verify memory limitation settings for Turbo ACL processing, look at the "Mb limit" output for both IPv4 and Data-Link. The new MB limit setting should be listed in the "Mb limit" output for IPv4 or Data-Link, depending on which memory limit was changed. For an example of the **show access-list compiled** command with these outputs highlighted, see the "Verifying ACL Memory Limit Configurations: Example" section on page 16.

Configuration Examples for Turbo Access Control List Scalability Enhancements on the NSEs

This section provides the following configuration examples:

Monitoring Memory Limitations for Layer 2 or Layer 3 and Layer 4 ACL Processing: Example

In the following example, the **show access-list compiled** command is entered.

Note the following, which are italicized in the example output:

- The output for **show access-list compiled** is separated for Layer 2 and for Layer 3 and Layer 4 data. Layer 3 and Layer 4 ACL compilation tables and information can be seen in the "Compiled ACL statistics for IPv4" section of the output, while Layer 2 ACL compilation tables and information can be seen in the "Compiled ACL statistics for Data-Link" section.
- The "mem limits" output shows the number of times a compile has occurred and the ACL has reached its configured limit. If you have reached the configured limit numerous times, you may want to consider modifying the memory limit to allow more memory.
  In this example, ACL memory for Layer 3 and Layer 4 data has never reached its configured limit. The same is true for Layer 2 data in this example.
- The "Mb limit" output shows the current memory limit setting.
  In this example, the Layer 3 and Layer 4 memory limit was previously set to 65 MB (via the **access-list compiled ipv4 limit memory 65** command), while the Layer 2 memory limit has not been changed from its default limit of 128 MB.
- The "Mb default limit" output shows the current default memory limit setting. If the **default** form of the **access-list compiled ipv4 limit memory** command or **access-list compiled data-link limit memory** command is entered, the "Mb default limit" will become the "Mb limit."
  In this example, the default limits are 256 MB for Layer 3 and Layer 4 data and 128 MB for Layer 2 data.
- The "Mb max memory" output shows the maximum amount of memory the current ACL configuration could actually consume under maximum usage conditions. This number is helpful for configuring memory limits for ACL processing. If you want to free up RP memory, for instance, and you have a small number of ACLs with a low "max memory," you could configure a reservation of a small amount of memory for ACL processing using the **access-list compiled** [**ipv4** | **data-link**] **limit memory** *number* command, thereby freeing up memory for other RP processes. Conversely, if you have a high memory limit, you may want to use the **access-list compiled** [**ipv4** | **data-link**] **limit memory** *number* command to commit more memory to ACL processing, or even the **no access-list compiled** [**ipv4** | **data-link**] **limit memory** command to allow as much memory as is available for ACL processing.
  In this example, the max memory for the current Layer 3 and Layer 4 Turbo ACL configuration data on the router is 1 MB, and the max memory for Layer 2 Turbo ACL configuration data is 0 Mb.

```
Router# show access-lists compiled
Compiled ACL statistics for IPv4:
ACL State      Entries Config Fragment Redundant
102 Operational    1       1      0        0
103 Operational    1       1      0        0
104 Operational    1       1      0        0
105 Operational    1       1      0        0
106 Operational    1       1      0        0
112 Operational    1       1      0        0
ws_def_acl Operational 1 1 0 0
7 ACLs, 7 active, 1 builds, 7 entries, 1408 ms last compile
1 history updates, 2000 history entries
0 mem limits, 65 Mb limit, 256 Mb default limit, 1 Mb max memory
0 compile failures, 0 priming failures
Overflows: L1 0, L2 0, L3 0
Table expands:[9]=0 [10]=0 [11]=0 [12]=0 [13]=0 [14]=0 [15]=0
L0: 1803Kb 2/3 8/9 3/4 2/3 2/3 2/3 2/3 2/3
```

```
L1: 5Kb 3/27 3/12 2/9 2/9
L2: 4Kb 3/150 2/81
L3: 7Kb 3/250
Ex: 8Kb
Tl: 1828Kb 41 equivs (18 dynamic)
Compiled ACL statistics for Data-Link:
ACL       State       Entries Config Fragment Redundant
int-l2-0  Operational    1       1      0        0
int-l2-1  Operational    2       2      0        0
int-l2-2  Operational    3       3      0        0
int-l2-3  Operational    4       4      0        0
int-l2-4  Operational    1       1      0        0
int-l2-5  Operational  199     199      0        0
int-l2-6  Operational  200     200      0        0
int-l2-8  Operational    3       3      0        0
int-l2-10 Operational    2       2      0        0
int-l2-15 Operational    1       1      0        0
int-l2-16 Operational    2       2      0        0
int-l2-17 Operational    3       3      0        0
int-l2-18 Operational    1       1      0        0
19 ACLs, 13 active, 22 builds, 422 entries, 832 ms last compile
0 history updates, 524288 history entries
0 mem limits, 128 Mb limit, 128 Mb default limit, 0 Mb max memory
0 compile failures, 0 priming failures
Overflows: L1 3
Table expands:[3]=3
L0: 593Kb 1013/1014 2/3
L1: 86Kb 1013/1518
Ex: 191Kb
Tl: 871Kb 2028 equivs (1013 dynamic)
```

Reserving a Set Amount of Memory for Layer 2 ACL Processing: Example

The following example reserves 100 MB of memory for Layer 2 ACL processing in the RP path:

```
access-list compiled data-link limit memory 100
```

Allowing All Available Memory to Be Used for Layer 2 ACL Processing: Example

The following example allows Layer 2 ACL processing to use as much memory as is needed for Layer 2 ACL processing:

```
no access-list compiled data-link limit memory
```

Restoring the Default Amount of Memory Reserved for Layer 2 ACL Processing: Example

The following example restores the default amount of memory reserved for Layer 2 ACL processing in the RP path:

default access-list compiled data-link limit memory

Reserving a Set Amount of Memory for Layer 3 and Layer 4 ACL Processing: Example

The following example reserves 100 MB of memory for Layer 3 and Layer 4 ACL processing in the RP path:

```
access-list compiled ipv4 limit memory 100
```

Allowing All Available Memory to Be Used for Layer 3 and Layer 4 ACL Processing: Example

The following example allows Layer 3 and Layer 4 ACL processing to use as much memory as is needed for Layer 3 and Layer 4 ACL data:

```
no access-list compiled ipv4 limit memory
```

Restoring the Default Amount of Memory Reserved for Layer 3 and Layer 4 ACL Processing: Example

The following example restores the default amount of memory reserved for Layer 3 and Layer 4 ACL processing in the RP path:

default access-list compiled ipv4 limit memory

Verifying ACL Memory Limit Configurations: Example

In the following example, a 65-MB limit has been configured for Layer 3 and Layer 4 ACL processing, while the Layer 2 ACL memory reservations have not been changed.

See the italicized output in the following example to view the changes:

```
Router# show access-lists compiled
Compiled ACL statistics for IPv4:
ACL State      Entries Config Fragment Redundant
102 Operational    1       1      0        0
103 Operational    1       1      0        0
104 Operational    1       1      0        0
105 Operational    1       1      0        0
106 Operational    1       1      0        0
112 Operational    1       1      0        0
ws_def_acl Operational 1 1 0 0
7 ACLs, 7 active, 1 builds, 7 entries, 1408 ms last compile
1 history updates, 2000 history entries
0 mem limits, 65 Mb limit, 256 Mb default limit, 1 Mb max memory
0 compile failures, 0 priming failures
Overflows: L1 0, L2 0, L3 0
Table expands:[9]=0 [10]=0 [11]=0 [12]=0 [13]=0 [14]=0 [15]=0
L0: 1803Kb 2/3 8/9 3/4 2/3 2/3 2/3 2/3 2/3
L1: 5Kb 3/27 3/12 2/9 2/9
L2: 4Kb 3/150 2/81
```

```
L3: 7Kb 3/250
Ex: 8Kb
Tl: 1828Kb 41 equivs (18 dynamic)
Compiled ACL statistics for Data-Link:
ACL       State       Entries Config Fragment Redundant
int-l2-0  Operational    1       1      0        0
int-l2-1  Operational    2       2      0        0
int-l2-2  Operational    3       3      0        0
int-l2-3  Operational    4       4      0        0
int-l2-4  Operational    1       1      0        0
int-l2-5  Operational  199     199      0        0
int-l2-6  Operational  200     200      0        0
int-l2-8  Operational    3       3      0        0
int-l2-10 Operational    2       2      0        0
int-l2-15 Operational    1       1      0        0
int-l2-16 Operational    2       2      0        0
int-l2-17 Operational    3       3      0        0
int-l2-18 Operational    1       1      0        0
19 ACLs, 13 active, 22 builds, 422 entries, 832 ms last compile
0 history updates, 524288 history entries
0 mem limits, 128 Mb limit, 128 Mb default limit, 0 Mb max memory
0 compile failures, 0 priming failures
Overflows: L1 3
Table expands:[3]=3
L0: 593Kb 1013/1014 2/3
L1: 86Kb 1013/1518
Ex: 191Kb
Tl: 871Kb 2028 equivs (1013 dynamic)
```

Additional References

The following sections provide references related to this feature.

Related Documents

Related Topic	Document Title
Access Lists	"IP Access Lists" section of *Cisco IOS IP Application Services Configuration Guide, Release 12.4*
Network Services Engines	*Cisco 7304 Network Services Engine Installation and Configuration Guide*
PXF	*PXF Information for the Cisco 7304 Router*
Turbo Access Control Lists	*Turbo Access Control Lists*

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and technical documentation. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features

- **access-list compiled data-link limit memory**
- **access-list compiled ipv4 limit memory**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Feature Information for Turbo ACL Scalability Enhancements on the NSEs

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Turbo ACL Scalability Enhancements on the NSEs

Feature Name	Releases	Feature Information
Turbo ACL Scalability Enhancements on the NSEs	12.2(31)SB2	This feature was introduced.
Performance Enhancements for IOS ACL	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Glossary

Access Control List—A list kept by routers to control access to or from the router for a number of services.

NSE—network services engine. The Cisco 7304 router has two types of processor, the NSE and the network processing engine (NPE). Two versions of the NSE exist, the NSE-100 and the NSE-150.

RP—Route Processor. One of two processing paths on a Cisco 7304 router using an NSE, with the Parallel eXpress Forwarding path being the other path. All traffic not supported in the PXF path on a Cisco 7304 router using an NSE is forwarded using the RP path.

Turbo Access Control Lists—A Turbo Access Control list is an access list that more expediently processes traffic by compiling the ACLs into a set of lookup tables while still maintaining the match requirements.

PXF—Parallel eXpress Forwarding. One of two processing paths on a Cisco 7304 router using an NSE, with the Route Processor (RP) path being the other path. The PXF processing path is used to accelerate the performance for certain supported features.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Network Admission Control

First Published: May 27, 2004
Last Updated: October 31, 2008

The Network Admission Control feature addresses the increased threat and impact of worms and viruses have on business networks. This feature is part of the Cisco Self-Defending Network Initiative that helps customers identify, prevent, and adapt to security threats.

In its initial phase, the Cisco Network Admission Control (NAC) functionality enables Cisco routers to enforce access privileges when an endpoint attempts to connect to a network. This access decision can be on the basis of information about the endpoint device, such as its current antivirus state. The antivirus state includes information such as version of antivirus software, virus definitions, and version of scan engine.

Network admission control systems allow noncompliant devices to be denied access, placed in a quarantined area, or given restricted access to computing resources, thus keeping insecure nodes from infecting the network.

The key component of the Cisco Network Admission Control program is the Cisco Trust Agent, which resides on an endpoint system and communicates with Cisco routers on the network. The Cisco Trust Agent collects security state information, such as what antivirus software is being used, and communicates this information to Cisco routers. The information is then relayed to a Cisco Secure Access Control Server (ACS) where access control decisions are made. The ACS directs the Cisco router to perform enforcement against the endpoint.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Network Admission Control" section on page 29.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Prerequisites for Network Admission Control

- The Cisco IOS router must be running Cisco IOS software Release 12.3(8)T or later.
- The Cisco Trust Agent must be installed on the endpoint devices (for example, on PCs and laptops).
- A Cisco Secure ACS is required for authentication, authorization, and accounting (AAA).
- A proficiency with configuring access control lists (ACLs) and AAA is necessary.

Restrictions for Network Admission Control

- This feature is available only on Cisco IOS firewall feature sets.

Information About Network Admission Control

Before configuring the Network Admission Control feature, the following concepts need to be understood:

Virus Infections and Their Effect on Networks

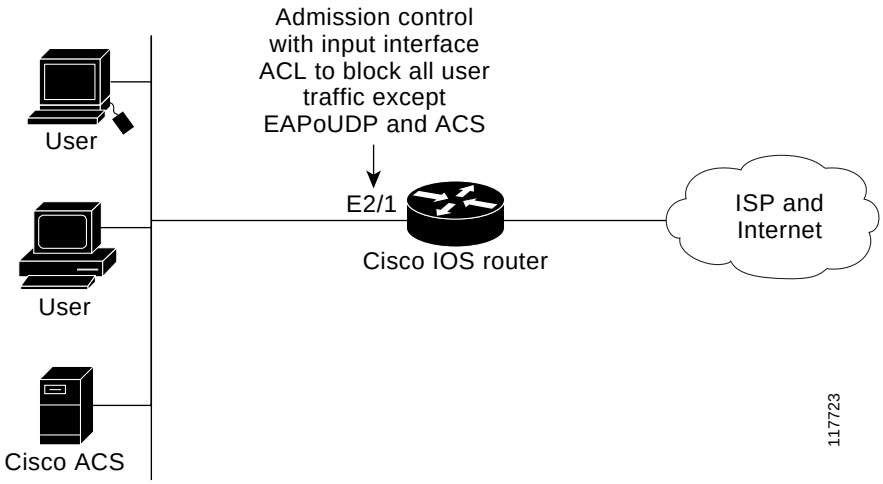
Virus infections are the single largest cause of serious security breaches for networks and often result in huge financial losses. Sources of virus infections are insecure endpoints (for example, PCs, laptops, and servers). Although the endpoints may have antivirus software installed, the software is often disabled. Even if the software is enabled, the endpoints may not have the latest virus definitions and scan engines. A larger security risk is from devices that do not have any antivirus software installed. Although antivirus vendors today are making it more difficult to disable the antivirus software, they are not addressing the risk of outdated virus definitions and scan engines.

How Network Admission Control Works

Endpoint systems, or clients, are normally hosts on the network, such as PCs, laptops, workstations, and servers. The endpoint systems are a potential source of virus infections, and their antivirus states have to be validated before they are granted network access. When an endpoint attempts an IP connection to a network through an upstream Cisco network access device (typically a Cisco IOS router), the router challenges the endpoint for its antivirus state. The endpoint systems run a client called Cisco Trust Agent, which collects antivirus state information from the end device and transports the information to the Cisco network access device. This information is then communicated to a Cisco Secure ACS where the antivirus state of the endpoint is validated and access control decisions are made and returned to Cisco network access devices. The network devices either permit, deny, or quarantine the end device. The Cisco Secure ACS may in turn use back-end antivirus vendor-specific servers for evaluating the antivirus state of the endpoint.

Figure 1 illustrates how Cisco Network Admission Control works.

Figure 1 Cisco IOS Network Admission Control System



Network Access Device

A network access device (NAD) is typically a Cisco IOS router (a Layer 3 Extensible Authentication Protocol over User Datagram Protocol [EAPoUDP] access point) that provides connectivity to external networks, such as the Internet or remote enterprise networks. Cisco Network Admission Control functionality may have an Intercept ACL, which determines connections that are intercepted for network

admission. Connections from endpoints that match the access list are intercepted by Network Admission Control and are challenged for their antivirus states over a Layer 3 association before they are granted network access.

Cisco Trust Agent

Cisco Trust Agent is a specialized software that runs on endpoint systems. Cisco Trust Agent responds to challenges from the router about the antivirus state of an endpoint system. If an endpoint system is not running the Cisco Trust Agent, the network access device (router) classifies the endpoint system as "clientless." The network access device uses the EOU clientless username and EOU clientless password that are configured on the network access device as the credentials of the endpoint system for validation with Cisco Secure ACS. The policy attributes that are associated with this username are enforced against the endpoint system.

Cisco Secure ACS

Cisco Secure ACS provides authentication, authorization, and accounting services for network admission control using industry-standard RADIUS authentication protocol. Cisco Secure ACS returns access control decisions to the network access device on the basis of the antivirus credentials of the endpoint system.

Using RADIUS cisco_av_pair vendor-specific attributes (VSAs), the following attribute-value pairs (AV pairs) can be set on the Cisco Secure ACS. These AV pairs are sent to the network access device along with other access-control attributes.

- url-redirect—Enables the AAA client to intercept an HTTP request and redirect it to a new URL. This redirection is especially useful if the result of posture validation indicates that the network access control endpoint requires an update or patch to be made available on a remediation web server. For example, a user can be redirected to a remediation web server to download and apply a new virus Directory Administration Tool (DAT) file or an operating system patch. (See the following example.)

  ```
  url-redirect=http://10.1.1.1
  ```

- posture-token—Enables Cisco Secure ACS to send a text version of a system posture token (SPT) that is derived by posture validation. The SPT is always sent in numeric format, and using the posture-token AV pair makes it easier to view the result of a posture validation request on the AAA client. (See the following example.)

  ```
  posture-token=Healthy
  ```

  Valid SPTs, in order of best to worst, are as follows:

  - Healthy
  - Checkup
  - Quarantine
  - Infected
  - Unknown

- status-query-timeout—Overrides the status-query default value of the AAA client with the user specified value, in seconds. (See the following example.)

  ```
  status-query-timeout=150
  ```

For more information about AV pairs that are supported by Cisco IOS software, see the documentation for the releases of Cisco IOS software that are implemented on your AAA clients.

Remediation

Network Admission Control supports HTTP redirection that redirects any HTTP request from the endpoint device to a specified redirect address. This support mechanism redirects all HTTP requests from a source to a specified web page (URL) to which the latest antivirus files can be downloaded. For the HTTP redirection to work, the value must be set for the "url-redirect" VSA on the ACS and, correspondingly, associate an access control entry in the downloadable ACL that permits the access of the endpoint system to the redirect URL address. After the value of the url-redirect VSA has been set and the access control entry has been associated, any HTTP request that matches the IP admission Intercept ACL are redirected to the specified redirect URL address.

Network Admission Control and Authentication Proxy

It is possible that network admission control and authentication proxy can be configured for the same set of hosts on a given interface. In each case, the Intercept ACL should be the same for IP admission EAPoUDP and authentication proxy. IP admission proxy with proxy authentication should be configured first, followed by IP admission control.

NAC MIB

The NAC MIB feature adds Simple Network Management Protocol (SNMP) support for the NAC subsystem. Using SNMP commands (get and set operations), an administrator can monitor and control NAC sessions on the network access device (NAD).

For more information about SNMP get and set operations, see the subsection "Related Documents" in the section "Additional References."

Correlation Between SNMP Get and Set Operations and the Cisco CLI

Most of the objects in the object tables in the NAC MIB (CISCO-NAC-NAD-MIB.my) describe various EAPoUDP and session parameters that are applicable to the setup of a NAD. These properties can be viewed and modified by performing various SNMP get and set operations. Many of the values of the table objects can also be viewed or modified by configuring corresponding command-line interface (CLI) commands on a router. For example, an SNMP get operation can be performed on the cnnEOUGlobalObjectsGroup table or the **show eou** command can be configured on a router. The parameter information obtained from the SNMP get operation is the same as the output from the **show eou** command. Similarly, performing an SNMP get operation on the table cnnEouIfConfigTable provides interface-specific parameters that can also be viewed in output from the **show eou** command.

SNMP set operations are allowed for table objects that have corresponding CLI commands, which can be used to modify table object values. For example, to change the value range for the cnnEouHostValidateAction object in the cnnEouHostValidateAction MIB table to 2, you can either perform the SNMP set operation or configure the **eou initialize all** command on a router.

For examples of NAC MIB output, see the subsection "NAC MIB Output: Examples" in the section "Configuration Examples for Network Admission Control."

Initializing and Revalidating Sessions

NAC allows administrators to initialize and revalidate sessions using the following CLI commands:

- **eou initialize all**
- **eou initialize authentication clientless**
- **eou initialize authentication eap**
- **eou initialize authentication static**
- **eou initialize ip** {*ip-address*}
- **eou initialize mac** {*mac-address*}
- **eou initialize posturetoken** {*string*}
- **eou revalidate all**
- **eou revalidate authentication clientless**
- **eou revalidate authentication eap**
- **eou revalidate authentication static**
- **eou revalidate ip** {*ip-address*}
- **eou revalidate mac** {*mac-address*}
- **eou revalidate posturetoken** {*string*}

The initialization and revalidation actions can also be accomplished by performing SNMP set operations on the objects of the cnnEouHostValidateAction table. For more information about initializing and revalidating sessions, see the section "CLI Commands That Correlate to cnnEouHostValidateAction Table Objects."

For examples of CLI commands that correlate to changes that can be made to cnnEouHostValidateAction table objects, see the subsection "NAC MIB Output: Examples" in the section "Configuration Examples for Network Admission Control."

Session-Specific Information

The NAC MIB provides a way to view session-specific details using the cnnEouHostQueryTable and cnnEouHostResultTable. The cnnEouHostQueryTable is used to build the query. The query is the same format as the **show eou ip** {*ip-address*} command (that is, the IP address would be shown as in the **show eou ip** command—for example, 10.1.1.1). Administrators must use the SNMP set operation on the objects of the cnnEouHostQueryTable to create the query. The results of the query are stored as a row in the cnnEouHostResultTable. For more information about viewing session-specific details, see the section "Viewing MIB Query Results."

Using show Commands to View MIB Object Information

The CLI commands **show eou**, **show eou all**, **show eou authentication**, **show eou initialize**, **show eou ip**, **show eou mac**, **show eou posturetoken, show eou revalidate**, and **show ip device tracking all** provide the same output information as that in the CISCO-NAC-NAD-MIB tables using SNMP get operations.

For examples of **show** command output information that can also be viewed in MIB object tables, see the subsection "NAC MIB Output: Examples" in the section "Configuration Examples for Network Admission Control."

How to Configure Network Admission Control

This section contains the following procedures:

Configuring the ACL and Admission Control

Network admission control is applied in the inbound direction at any interface. Applying network admission control inbound at an interface causes network admission control to intercept the initial IP connections of the intercept end system through the router.

Figure 1 shows that IP admission control is applied at the LAN interface. All network devices must be validated for their antivirus states upon their initial IP connections through the router. Until then, all traffic from endpoint systems (except for EAPoUDP and Cisco Secure ACS traffic) is blocked at the interface.

The endpoint system is then challenged for its antivirus state over an EAPoUDP association. The endpoint system gains access to the network if it complies with the network admission control policy as evaluated by the Cisco Secure ACS. If the endpoint system does not comply, the device is either denied access or quarantined.

To configure an intercept ACL, perform the DETAILED STEPS below.

In this configuration, an intercept ACL is defined as “101,” and the Intercept ACL is associated with the IP admission control rule “greentree.” Any IP traffic that is destined to the 192.50.0.0 network are subjected to validation. In addition, beginning with Step 5, an intercept ACL is applied inbound to the interface that is associated with network admission control. This ACL typically blocks access to endpoint systems until they are validated. This ACL is referred to as the default access list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**permit** | **deny**} *protocol source destination*
4. **ip admission name** *admission-name* [**eapoudp** | **proxy** {**ftp** | **http** | **telnet**}] [**list** {*acl* | *acl-name*}]
5. **interface** *type slot/port*
6. **ip address** *ip-address mask*
7. **ip admission** *admission-name*
8. **exit**

9. **access-list** *access-list-number* {**permit** | **deny**} *protocol source destination*
10. **ip access-group** {*access-list-number* | *access-list-name*} **in**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**permit** | **deny**} *protocol source destination* **Example:** `Router (config)# access-list 101 permit ip any 192.50.0.0 0.0.0.255`	Defines a numbered access list.
Step 4	**ip admission name** *admission-name* [**eapoudp** | **proxy** {**ftp** | **http** | **telnet**}] [**list** {*acl* | *acl-name*}] **Example:** `Router (config)# ip admission name greentree eapoudp list 101`	Creates IP network admission control rules. The rules define how you apply admission control. The rules are as follows: • **eapoudp**—Specifies IP network admission control using EAPoUDP. • **proxy ftp**—Specifies FTP to trigger authentication proxy. • **proxy http**—Specifies HTTP to trigger authentication proxy. • **proxy telnet**—Specifies Telnet to trigger authentication proxy. You can associate the named rule with an ACL, providing control over which hosts use the admission control feature. If no standard access list is defined, the named admission rule intercepts IP traffic from all hosts whose connection-initiating packets are received at the configured interface. The list option allows you to apply a standard, extended (1 through 199) or named access list to a named admission control rule. IP connections that are initiated by hosts in the access list are intercepted by the admission control feature.
Step 5	**interface** *type slot/port* **Example:** `Router (config)# interface ethernet 2/1`	Defines an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 6	**ip address** *ip-address mask* **Example:** `Router (config-if)# ip address 192.0.0.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 7	**ip admission** *admission-name* **Example:** `Router (config-if)# ip admission greentree`	Applies the named admission control rule at the interface.
Step 8	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 9	**access-list** *access-list-number* {**permit** | **deny**} *protocol source destination* **Example:** `Router (config)# access-list 105 permit udp any any` or `Router (config)# access-list 105 permit ip host 192.168.0.2 any` or `Router (config)# access-list 105 deny ip any any`	Defines a numbered access list. **Note** In the first two examples (under "Command or Action"), ACL "105" denies all IP traffic except UDP and access to 192.168.0.2 (Cisco Secure ACS). **Note** In the third example (under "Command or Action," ACL "105" is applied on the interface that is configured for network admission control, and access to endpoint systems (except for EAPoUDP traffic and access to Cisco Secure ACS [192.168.0.2 in the example] is blocked until their antivirus states are validated. This ACL ("105") is referred to as "Interface ACL."
Step 10	**ip access-group** {*access-list-number* | *access-list-name*} **in** **Example:** `Router (config)# ip access-group 105 in`	Controls access to an interface.

Configuring Global EAPoUDP Values

To configure global EAPoUDP values, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **eou {allow | clientless | default | initialize | logging | max-retry | port | rate-limit | revalidate | timeout}**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`eou {allow | clientless | default | initialize | logging | max-retry | port | rate-limit | revalidate | timeout}` **Example:** `Router (config)# eou initialize`	Specifies EAPoUDP values. • For a breakout of available keywords and arguments for the **eou** command, see the following commands: – **eou allow** – **eou clientless** – **eou default** – **eou initialize** – **eou logging** – **eou max-retry** – **eou port** – **eou rate-limit** – **eou revalidate** – **eou timeout**

Configuring an Interface-Specific EAPoUDP Association

To configure an EAPoUDP association that can be changed or customized for a specific interface that is associated with network admission control, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **eou** [**default** | **max-retry** | **revalidate** | **timeout**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Router (config)# interface ethernet 2/1`	Defines an interface and enters interface configuration mode.
Step 4	**eou** [**default** | **max-retry** | **revalidate** | **timeout**] **Example:** `Router (config-if)# eou revalidate`	Enables an EAPoUDP association for a specific interface. • For a breakout of available keywords and arguments for the **eou** command, see the following commands: – **eou default** – **eou max-retry** – **eou revalidate** – **eou timeout**

Configuring AAA for EAPoUDP

To set up AAA for EAPoUDP, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication eou default enable group radius**
5. **aaa authorization network default group radius**
6. **radius-server host** {*hostname* | *ip-address*}
7. **radius-server key** {**0** *string* | **7** *string* | *string*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa authentication eou default enable group radius** **Example:** `Router (config)# aaa authentication eou default enable group radius`	Sets authentication lists for an EAPoUDP association.
Step 5	**aaa authorization network default group radius** **Example:** `Router (config)# aaa authorization network default group radius`	Uses the list of all RADIUS servers for authentication.
Step 6	**radius-server host** {*hostname* | *ip-address*} **Example:** `Router (config)# radius-server host 192.0.0.40`	Specifies a RADIUS server host.
Step 7	**radius-server key** {**0** *string* | **7** *string* | *string*} **Example:** `Router (config)# radius-server key cisco`	Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.

Configuring the Identity Profile and Policy

Identity is a common infrastructure that is used to specify local profile and policy configurations. The identity profile allows you to statically authorize or validate individual devices on the basis of IP address, MAC address, or device type. Each statically authenticated device can be associated with a local policy that specifies the network access control attributes. Hosts are added to this “exception list” using the **identity profile** command, and corresponding policies are associated with these hosts using the **identity policy** command.

If the client is part of the identity (that is, the client is on the exception list), the status of the client is set on the basis of the identity configuration. The client does not have to go through the posture validation process, and the associated identity policy is applied for the client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **identity profile eapoudp**
4. **device** {**authorize** {**ip address** *ip-address* {**policy** *policy-name*} | **mac-address** *mac-address* | **type** {**cisco** | **ip** | **phone**}} | **not-authorize**}
5. **exit**
6. **identity policy** *policy-name* [**access-group** *group-name* | **description** *line-of-description* | **redirect** *url* | **template** [**virtual-template** *interface-name*]]
7. **access-group** *group-name*
8. **exit**
9. **exit**
10. **ip access-list extended** *access-list-name*
11. {**permit** | **deny**} *source source-wildcard destination destination-wildcard*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**identity profile eapoudp** **Example:** `Router (config)# identity profile eapoudp`	Creates an identity profile and enters identity profile configuration mode.
Step 4	**device** {**authorize** {**ip address** *ip-address* {**policy** *policy-name*} | **mac-address** *mac-address* | **type** {**cisco** | **ip** | **phone**}} | **not-authorize**} **Example:** `Router (config-identity-prof)# device authorize ip address 10.10.142.25 policy policyname1`	Statically authorizes an IP device and applies an associated policy to the device.
Step 5	**exit** **Example:** `Router (config-identity-prof)# exit`	Exits identity profile configuration mode.

	Command or Action	Purpose
Step 6	**identity policy** *policy-name* [**access-group** *group-name* | **description** *line-of-description* | **redirect** *url* | **template** [**virtual-template** *interface-name*]] **Example:** `Router (config-identity-prof)# identity policy policyname1`	Creates an identity policy and enters identity policy configuration mode.
Step 7	**access-group** *group-name* **Example:** `Router (config-identity-policy)# access-group exempt-acl`	Defines network access attributes for the identity policy.
Step 8	**exit** **Example:** `Router (config-identity-policy)# exit`	Exits identity policy configuration mode.
Step 9	**exit** **Example:** `Router (config-identity-prof)# exit`	Exits identity profile configuration mode.
Step 10	**ip access-list extended** *access-list-name* **Example:** `Router (config)# ip access-list extended exempt-acl`	Defines access control for statically authenticated devices (and enters network access control configuration mode).
Step 11	{**permit** | **deny**} *source source-wildcard destination destination-wildcard* **Example:** `Router (config-ext-nacl)# permit ip any 192.50.0.0. 0.0.0.255`	Set conditions to allow a packet to pass a named IP access list.

Clearing EAPoUDP Sessions That Are Associated with an Interface

To clear EAPoUDP sessions that are associated with a particular interface or that are on the NAD, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **clear eou all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`clear eou all` **Example:** `Router# clear eou all`	Clears all EAPoUDP sessions on the NAD.

Verifying Network Admission Control

To verify EAP and EAPoUDP messages or sessions, perform the following steps. The **show** commands may be used in any order or independent of the other **show** command.

SUMMARY STEPS

1. **enable**
2. **show eou all**
3. **show ip admission eapoudp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`show eou all` **Example:** `Router# show eou all`	Displays information about EAPoUDP sessions on the network access device.
Step 3	`show ip admission eapoudp` **Example:** `Router# show ip admission eapoudp`	Displays the network admission control configuration or network admission cache entries.

Troubleshooting Network Admission Control

The following commands may be used to display information about EAP and EAPoUDP messages or sessions. The **debug** commands may be used in any order or independent of the other **debug** commands.

SUMMARY STEPS

1. **enable**
2. **debug eap** {**all** | **errors** | **packets** | **sm**}
3. **debug eou** {**all** | **eap** | **errors** | **packets** | **sm**}
4. **debug ip admission eapoudp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`debug eap {all | errors | packets | sm}` **Example:** `Router# debug eap all`	Displays information about EAP messages.
Step 3	`debug eou {all | eap | errors | packets | sm}` **Example:** `Router# debug eou all`	Displays information about EAPoUDP messages.
Step 4	`debug ip admission eapoudp` **Example:** `Router# debug ip admission eapoudp`	Displays information about IP admission events.

Monitoring and Controlling NAC with the CISCO-NAC-NAD-MIB

This section includes the following tasks:

CLI Commands That Correlate to cnnEouGlobalObjectsGroup Table Objects

An SNMP get or set operation can be performed to obtain or change information about value ranges for objects in the cnnEouGlobalObjectsGroup table. The same information can be viewed in output from the **show eou** command. Table 1 displays examples of some global configuration objects and the SNMP get and set operations required to obtain or change their values.

For an example of **show eou** command output, see the section “show eou” section on page 24.

Table 1 Obtaining and Changing Global Configuration Values Using SNMP Get and Set Operations

Global Configuration Objects	SNMP Operation
EAPoUDP version	Performs a get operation on the cnnEouVersion object. (The object value is “1.”)
EAPoUDP port	Performs a get operation on the cnnEouPort object.
Enabling logging (enable EOU logging)	Sets the cnnEouLoggingEnable object. (The object value is “true.”)

CLI Commands That Correlate to cnnEoulfConfigTable Objects

An SNMP get operation is performed to obtain information about value ranges for objects in the cnnEouIfConfigTable. The same information can be viewed in output from the **show eou** command. Table 2 displays examples of some interface-specific configuration objects and the SNMP get operations required to obtain their values.

Table 2 Obtaining Interface-Specific Configuration Values Using SNMP Get Operations

Interface-Specific Object	SNMP Operation
AAA timeout	Performs a get operation on the cnnEouIfTimeoutAAA object. • Format: GET cnnEouIfTimeoutAAA.IfIndex • You must specify the corresponding index number of the specific interface.
Maximum retries	Performs a get operation on the cnnEouIfMaxRetry object. • Format: GET cnnEouIfMaxRetry.IfIndex

CLI Commands That Correlate to cnnEouHostValidateAction Table Objects

EOU sessions can be initialized or revalidated by the CLI or by using the SNMP set operation on the table cnnEouHostValidateAction.

Following are some examples (listed by CLI command) that correlate to MIB objects.

eou initialize all

EOU initialization can be accomplished for all sessions by using the **eou initialize all** command or by using an SNMP set operation on the object cnnEouHostValidateAction. This object must be set to the numeric value 2.

eou initialize authentication clientless

EOU initialization can be accomplished for sessions having an authentication type “clientless” using the **eou initialize authentication clientless** command or an SNMP set operation on the object cnnEouHostValidateAction. This object must be set to the numeric value 3.

eou initialize ip

EOU initialization can be accomplished for a particular session using the **eou initialize ip** {*ip-address*} command.

To achieve the same result using an SNMP operation, three objects have to be set in the cnnEouHostValidateAction MIB table:

- cnnEouHostValidateAction—The value range must be set.
- cnnEouHostValidateIpAddrType—The IP address type must be set. This value must be set to IPv4 because IPv4 is currently the only address type supported by NAC. (This value is the type of address being set for the cnnEouHostValidateIPAddr object.)
- cnnEouHostValidateIPAddr—The IP address must be set.

Note The three MIB objects should be set in a single SNMP set operation.

eou initialize posturetoken

All sessions having a particular posturetoken can be initialized using the **eou initialize posturetoken** {*string*} command. The default value range for this command is 8.

To achieve the same result using an SNMP set operation, you must set the following objects:

- cnnEouHostValidateAction—Set this value to 8.
- cnnEouHostValidatePostureTokenStr—Set the string value.

Note The two MIB objects should be set in a single SNMP set operation.

Creating MIB Query Tables

The MIB table cnnEouHostQueryTable is used to create, or build, MIB queries.

MIB Query Correlating to the CLI show eou all Command

To build a query that provides the same results as using the **show eou all** command, perform the following SNMP get operation.

The object cnnEouHostQueryMask in the table cnnEouHostQueryTable indicates the kind of query. The corresponding value of the cnnEouHostQueryMask object in output from the **show eou all** command is 8 (the integer value).

SUMMARY STEPS

1. Set the cnnEouHostQueryStatus object to createandgo.
2. Set the cnnEouHostQueryMask object to 8.
3. Set the cnnEouHostQueryStatus object to active to indicate that query creation is complete.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Set the cnnEouHostQueryStatus object to createandgo.	Creates a query row.
Step 2	Set the cnnEouHostQueryMask object to 8.	Corresponds in value to the **show eou all** command.
Step 3	Set the cnnEouHostQueryStatus object to active.	Indicates that you have finished building the query.

Note Examples are not shown in the previous table because the format differs depending on the software you are using.

What to Do Next

View the results. See the section "Viewing MIB Query Results Correlating to the show eou all Command."

Viewing MIB Query Results Correlating to the show eou all Command

After the MIB query has been built and you have indicated that you are finished (with the "active" status), the results can be viewed. A query in the cnnEouHostQueryTable is represented by a row. The row number is the Query Index. Similarly, the cnnEouHostResultTable is composed of result rows. Each row in the cnnEouHostResultTable is uniquely identified by a combination of Query Index and Result Index. The results of the cnnEouHostQueryTable index and the cnnEouHostResultTable have to be matched. Match one row in the Query table to one of the rows in the Result table. For example, if a query that corresponds to a **show** command results in ten sessions, the Result table has ten rows, each row corresponding to a particular session. The first row in the Result table is R1.1. The second row is R1.2, and so on to R1.10. If another query is created in the Query table, and it results in five sessions, five rows are created in the Result table (R2.1, R2.2, R2.3, R2.4, and R2.5).

Table 3 illustrates how the Query table sessions are mapped to Result table rows.

Table 3 Query Table-to-Result Table Mapping

Query Table	Result Table Rows
Q1 (10 sessions)	R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7, R1.8, R1.9, R1.10
Q2 (5 sessions)	R2.1, R2.2, R2.3, R2.4, R2.5

Creating the SNMP Query

To create an SNMP query that provides the same information as output from the **show eou ip {***ip-address***} command**, perform the following steps.

SUMMARY STEPS

1. Set cnnEouHostQueryStatus to createandgo.
2. Set cnnEouHostQueryIpAddrType to IPv4 and the IP address (for example, 10.2.3.4).
3. Set cnnEouHostQueryStatus to active.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Set cnnEouHostQueryStatus to createandgo.	Creates a query row.
Step 2	Set cnnEouHostQueryIpAddrType to IPv4 and the IP address (for example, 10.2.3.4).	Sets the address type. • The only address type currently supported by NAC is IPv4.
Step 3	Set cnnEouHostQueryStatus to active.	Indicates you have finished building the query.

Note Examples are not shown in the previous table because the format differs depending on the software you are using.

Viewing the Results

To view the results in the cnnEouHostResultTable, perform the following steps.

SUMMARY STEPS

1. Perform a get operation on cnnEouHostQueryRows.
2. Perform a get operation on the cnnEouHostResultTable objects in the format resultTableObjectName.QueryIndex.ResultIndex.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Perform a get operation on cnnEouHostQueryRows.	Finds how many rows are created in a Result table for a particular query. • If a query row is a negative number, the query is still being processed.
Step 2	Perform a get operation on the cnnEouHostResultTable objects in the format resultTableObjectName.QueryIndex.ResultIndex.	Finds the value of a particular object in a Result table that matches a particular query. • For multiple rows in the Result table for a single query, the ResultIndex ranges from 1 to the value of cnnEouHostQueryRows.

Note Examples are not shown in the above table because the format differs depending on the software you are using.

MIB Query Correlating to the show eou ip Command

To build a MIB query that provides the same results as the **show eou ip** {*ip-address*}command, perform the following SNMP get operation.

SUMMARY STEPS

1. Set the cnnEouHostQueryStatus object to createandgo.
2. Set the cnnEouHostQueryIpAddrType object to “IPv4”.
3. Set the cnnEouHostQueryIpAddr object to IP address (for example, 10.2.3.4).
4. Set the cnnEouHostQueryStatus object to active.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Set the cnnEouHostQueryStatus object to createandgo.	Sets the query status.
Step 2	Set the cnnEouHostQueryIpAddrType object to “IPv4”.	Sets the address type. **Note** The only address type currently supported by NAC is IPv4.
Step 3	Set the cnnEouHostQueryIpAddr object to IP address (for example, 10.2.3.4).	Sets the IP address.
Step 4	Set the cnnEouHostQueryStatus object to active.	Indicates that you have finished building the query.

Note Examples are not shown in the previous table because the format differs depending on the software you are using.

Viewing MIB Query Results

After the MIB query has been built, the results can be viewed in cnnEouHostResultTable. For information about how to review the results, see the subsection “Viewing MIB Query Results Correlating to the show eou all Command” in the previous section “Creating MIB Query Tables.”

Splitting a Query into Subqueries

If you are doing a MIB query that correlates to the **show eou all** command, there could possibly be as many as 2,000 rows of output. To ensure that you can view all the information in a MIB query, you can split the query into subqueries. For example, for a query having 2,000 rows of output, you could split the query into four subqueries to view the results in a page-by-page format. The first subquery would include rows 1 through 500 (the first 500 sessions); the second subquery would include rows 501 through 1,000; the third subquery would include rows 1,001 through 1,500; and the fourth subquery would include rows 1,501 through 2,000.

Note The cnnEouHostQueryTotalHosts object provides the total number of hosts (number of rows) that match a query criterion. By looking at this number, you can determine how many subqueries are necessary. However, you cannot get the cnnEouHostQueryTotalHosts object number until you have built your first query.

Build your query by performing the following steps.

SUMMARY STEPS

1. Set the cnnEouHostQueryStatus object to createandgo.
2. Set the cnnEouHostQueryMask object to 8.
3. Set cnnEouHostQueryRows to 500.
4. Set cnnEouHostQuerySkipNHosts to 0.
5. Set the cnnEouHostQueryStatus object to active.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Set the cnnEouHostQueryStatus object to createandgo.	Sets the query status.
Step 2	Set the cnnEouHostQueryMask object to 8.	Correlates to the default of the **show eou all** command.
Step 3	Set cnnEouHostQueryRows to 500.	Identifies the maximum number of rows to be built in the result table for this query.
Step 4	Set cnnEouHostQuerySkipNHosts to 0.	Corresponds to the result rows to be created.
Step 5	Set the cnnEouHostQueryStatus object to active.	Indicates that you have finished building the query.

Note Examples are not shown in the previous table because the format differs depending on the software you are using. The table is on the basis of a query having 2,000 sessions (rows).

What to Do Next

After the above task is performed, information for the first 500 hosts (rows) is queried. To view query information for the next 500 hosts (rows), perform the same five steps, with the exception of changing the cnnEouHostQuerySkipNHosts object value to 500 in Step 4. This task results in query information for rows 501 through 1000. In the same way, to obtain query information for the remaining hosts (through 2000), perform the same five steps again, with the exception of changing the cnnEouHostQuerySkipNHosts object values in Step 4 to 1000 and 1500, respectively.

Configuration Examples for Network Admission Control

This section includes the following example.

Network Admission Control: Example

The following output example shows that IP admission control has been configured on a Cisco IOS router:

```
Router# show running-config

Building configuration...

Current configuration: 1240 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
aaa new-model
!
!
aaa authentication eou default group radius
aaa session-id common
ip subnet-zero
ip cef
!
! The following line creates a network admission rule. A list is not specified; therefore,
! the rule intercepts all traffic on the applied interface.
ip admission name avrule eapoudp
!
eou logging
!
!
interface FastEthernet0/0
 ip address 10.13.11.106 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 10.0.0.1 255.255.255.0
 ip access-group 102 in
! The following line configures an IP admission control interface.
 ip admission avrule
 duplex auto
 speed auto
!
ip http server
no ip http secure-server
ip classless
!
```

```
!
! The following lines configure an interface access list that allows EAPoUDP traffic
! and blocks the rest of the traffic until it is validated.
access-list 102 permit udp any any eq 21862
access-list 102 deny   ip any any
!
!
! The following line configures RADIUS.
radius-server host 10.13.11.105 auth-port 1645 acct-port 1646 key cisco
!
control-plane
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
!
!
end
```

NAC MIB Output: Examples

The following are examples of **show** command output displaying MIB object information.

show eou

The **show eou** command provides output for information that can also be viewed in various CISCO-NAC-NAD-MIB tables. The information that follows the **show eou** command can also be found in the cnnEouGlobalObjectsGroup table and the information that follows the **show eou all** command can be found in the cnnEouIfConfigTable.

```
Router# show eou

Global EAPoUDP Configuration
----------------------------
EAPoUDP Version     = 1
EAPoUDP Port        = 0x5566
Clientless Hosts    = Enabled
IP Station ID       = Disabled
Revalidation        = Enabled
Revalidation Period = 36000 Seconds
ReTransmit Period   = 3 Seconds
StatusQuery Period  = 300 Seconds
Hold Period         = 30 Seconds
AAA Timeout         = 60 Seconds
Max Retries         = 3
EAP Rate Limit      = 20
EAPoUDP Logging     = Enabled
Clientless Host Username = clientless
Clientless Host Password = clientless

Router# show eou all

Interface Specific EAPoUDP Configurations
-----------------------------------------
Interface Vlan333
AAA Timeout         = 60 Seconds
```

```
Max Retries         = 3
eou initialize interface {interface-name}
eou revalidate interface {interface-name}
```

show ip device tracking all

The **show ip device tracking all** command provides output for information that can also be found in the cnnIpDeviceTrackingObjectsGroup MIB table. The following is an example of such **show** command output:

```
Router# show ip device tracking all

IP Device Tracking = Enabled
Probe Count: 2
Probe Interval: 10
```

Additional References

The following sections provide references related to Network Admission Control.

Related Documents

Related Topic	Document Title
Configuring ACLs	"Access Control Lists: Overview and Guidelines" chapter of the "Traffic Filtering and Firewalls" section of the *Cisco IOS Security Configuration Guide*, Release 12.3.
Authentication, authorization, and accounting	"Authentication, Authorization, and Accounting" section of *Cisco IOS Security Configuration Guide*, Release 12.3.
Interfaces, configuring	*Cisco IOS Interface and Hardware Component Configuration Guide*, Release 12.3.
SNMP and SNMP get and set operations	• "Simple Network Management Protocol" section of the *Internetworking Technology Handbook* • "Configuring SNMP Support" section of the *Cisco IOS Configuring Fundamentals Configuration Guide*, Release 12.4.

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **aaa authentication eou default enable group radius**
- **access-group (identity policy)**
- **auth-type**
- **clear eou**
- **clear ip admission cache**
- **debug eap**
- **debug eou**
- **debug ip admission eapoudp**
- **description (identity policy)**
- **description (identity profile)**
- **device (identity profile)**
- **eou allow**
- **eou clientless**
- **eou default**
- **eou initialize**
- **eou logging**
- **eou max-retry**
- **eou port**

- **eou rate-limit**
- **eou revalidate**
- **eou timeout**
- **identity policy**
- **identity profile eapoudp**
- **ip admission**
- **ip admission name**
- **redirect (identity policy)**
- **show eou**
- **show ip admission**
- **show ip device tracking**
- **template (identity policy)**

Feature Information for Network Admission Control

Table 4 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 4 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 4 Feature Information for Network Admission Control

Feature Name	Releases	Feature Information
Network Admission Control	12.3(8)T	The Network Admission Control feature addresses the increased threat and impact of worms and viruses to networked businesses. This feature is part of the Cisco Self-Defending Network Initiative that helps customers identify, prevent, and adapt to security threats. In its initial phase, the Cisco Network Admission Control functionality enables Cisco routers to enforce access privileges when an endpoint attempts to connect to a network. The following sections provide information about this feature: • Prerequisites for Network Admission Control, page 2 • Restrictions for Network Admission Control, page 2 • Information About Network Admission Control, page 2 • How to Configure Network Admission Control, page 7 • Configuration Examples for Network Admission Control, page 23 The following commands were introduced or modified by this feature: **aaa authentication eou default enable group radius, access-group (identity policy), auth-type, clear eou, clear ip admission cache, debug eap, debug eou, debug ip admission eapoudp, description (identity policy), description (identity profile), device (identity profile), eou allow, eou clientless, eou default, eou initialize, eou logging, eou max-retry, eou port, eou rate-limit, eou revalidate, eou timeout, identity policy, identity profile eapoudp, ip admission, ip admission name, redirect (identity policy), show eou, show ip admission, template (identity policy).**

Table 4 Feature Information for Network Admission Control (continued)

Feature Name	Releases	Feature Information
NAC MIB	12.4(15)T	Support was added for the CISCO-NAC-NAD-MIB. This MIB module is used to monitor and configure the NAD on the Cisco NAC system. The following sections provide information about this feature: • "NAC MIB" section on page 5 • "Monitoring and Controlling NAC with the CISCO-NAC-NAD-MIB" section on page 16 The following commands were introduced or modified by this feature: **show ip device tracking**.
	12.2(33)SXI	This feature was integrated into Cisco IOS Release 12.2(33)SXI.

Glossary

default access policy—Set of ACLs that are applied to a client device until its credentials are validated by the AAA server.

EAPoUDP—Extensible Authentication Protocol over User Datagram Protocol. EAP is a framework that supports multiple, optional authentication mechanisms for PPP, including clear-text passwords, challenge-response, and arbitrary dialogue sequences. UDP is a connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, and it requires that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.

ip admission rule—Named rule that defines how IP admission control is applied. The IP admission rule is associated with an Intercept ACL and provides control over which hosts can use the IP admission feature. To create an IP admission control rule, use the ip admission name command.

posture token—Status that is used to convey the result of the evaluation of posture credentials. The AAA server maps the posture token (its status can be Healthy, Checkup, Quarantine, Infected, or Unknown) to a network access policy (ACL, URL, redirect, or status query timer) for the peer that the client wants to reach.

Context-Based Access Control

Configuring Context-Based Access Control

This chapter describes how to configure Context-based Access Control (CBAC). CBAC provides advanced traffic filtering functionality and can be used as an integral part of your network's firewall. For more information regarding firewalls, refer to the chapter "Cisco IOS Firewall Overview."

For a complete description of the CBAC commands used in this chapter, refer to the "Context-Based Access Control Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the chapter "Identifying Supported Platforms" section in the "Using Cisco IOS Software."

In This Chapter

This chapter has the following sections:

About Context-Based Access Control

This section describes CBAC features and functions:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

What CBAC Does

CBAC works to provide network protection on multiple levels using the following functions:

Traffic Filtering

CBAC intelligently filters TCP and UDP packets based on application-layer protocol session information. You can configure CBAC to permit specified TCP and UDP traffic through a firewall only when the connection is initiated from within the network you want to protect. CBAC can inspect traffic for sessions that originate from either side of the firewall, and CBAC can be used for intranet, extranet, and Internet perimeters of your network.

Without CBAC, traffic filtering is limited to access list implementations that examine packets at the network layer, or at most, the transport layer. However, CBAC examines not only network layer and transport layer information but also examines the application-layer protocol information (such as FTP connection information) to learn about the state of the session. This allows support of protocols that involve multiple channels created as a result of negotiations in the control channel. Most of the multimedia protocols as well as some other protocols (such as FTP, RPC, and SQL*Net) involve multiple channels.

Using CBAC, Java blocking can be configured to filter HTTP traffic based on the server address or to completely deny access to Java applets that are not embedded in an archived or compressed file. With Java, you must protect against the risk of users inadvertently downloading destructive applets into your network. To protect against this risk, you could require all users to disable Java in their browser. If this is not an acceptable solution, you can create a CBAC inspection rule to filter Java applets at the firewall, which allows users to download only applets residing within the firewall and trusted applets from outside the firewall. For extensive content filtering of Java, Active-X, or virus scanning, you might want to consider purchasing a dedicated content filtering product.

Traffic Inspection

CBAC inspects traffic that travels through the firewall to discover and manage state information for TCP and UDP sessions. This state information is used to create temporary openings in the firewall's access lists to allow return traffic and additional data connections for permissible sessions.

Inspecting packets at the application layer, and maintaining TCP and UDP session information, provides CBAC with the ability to detect and prevent certain types of network attacks such as SYN-flooding. A SYN-flood attack occurs when a network attacker floods a server with a barrage of requests for connection and does not complete the connection. The resulting volume of half-open connections can overwhelm the server, causing it to deny service to valid requests. Network attacks that deny access to a network device are called denial-of-service (DoS) attacks.

CBAC helps to protect against DoS attacks in other ways. CBAC inspects packet sequence numbers in TCP connections to see if they are within expected ranges—CBAC drops any suspicious packets. You can also configure CBAC to drop half-open connections, which require firewall processing and memory resources to maintain. Additionally, CBAC can detect unusually high rates of new connections and issue alert messages.

CBAC can help by protecting against certain DoS attacks involving fragmented IP packets. Even though the firewall prevents an attacker from making actual connections to a given host, the attacker can disrupt services provided by that host. This is done by sending many non-initial IP fragments or by sending complete fragmented packets through a router with an ACL that filters the first fragment of a fragmented packet. These fragments can tie up resources on the target host as it tries to reassemble the incomplete packets.

Alerts and Audit Trails

CBAC also generates real-time alerts and audit trails. Enhanced audit trail features use SYSLOG to track all network transactions; recording time stamps, source host, destination host, ports used, and the total number of transmitted bytes, for advanced, session-based reporting. Real-time alerts send SYSLOG error messages to central management consoles upon detecting suspicious activity. Using CBAC inspection rules, you can configure alerts and audit trail information on a per-application protocol basis. For example, if you want to generate audit trail information for HTTP traffic, you can specify that in the CBAC rule covering HTTP inspection.

Intrusion Prevention

CBAC provides a limited amount of intrusion detection to protect against specific SMTP attacks. With intrusion detection, SYSLOG messages are reviewed and monitored for specific “attack signatures.” Certain types of network attacks have specific characteristics, or signatures. When CBAC detects an attacks, it resets the offending connections and sends SYSLOG information to the SYSLOG server. Refer to the section “Interpreting Syslog and Console Messages Generated by CBAC” later in this chapter for a list of supported signatures.

In addition to the limited intrusion detection offered by CBAC, the Cisco IOS Firewall feature set offers intrusion detection technology for mid-range and high-end router platforms using the Cisco IOS Intrusion Prevention System (IPS). Cisco IOS IPS restructures and replaces the existing Cisco IOS Intrusion Detection System (IDS). It is ideal for any network perimeter, and especially for locations in which a router is being deployed and additional security between network segments is required. It also can protect intranet and extranet connections where additional security is mandated, and branch-office sites connecting to the corporate office or Internet.

The Cisco IOS IPS acts as an in-line intrusion detection sensor, watching packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When Cisco IOS IPS detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or Security Device Event Exchange (SDEE).

For more information about Cisco IOS IPS, refer to the module “Configuring Cisco IOS Intrusion Prevention System (IPS).”

What CBAC Does Not Do

CBAC does not provide intelligent filtering for all protocols; it only works for the protocols that you specify. If you do not specify a certain protocol for CBAC, the existing access lists will determine how that protocol is filtered. No temporary openings will be created for protocols not specified for CBAC inspection.

CBAC does not protect against attacks originating from within the protected network unless that traffic travels through a router that has the Cisco IOS Firewall feature set deployed on it. CBAC only detects and protects against attacks that travel through the firewall. This is a scenario in which you might want to deploy CBAC on an intranet-based router.

CBAC protects against certain types of attacks, but not every type of attack. CBAC should not be considered a perfect, impenetrable defense. Determined, skilled attackers might be able to launch effective attacks. While there is no such thing as a perfect defense, CBAC detects and prevents most of the popular attacks on your network.

How CBAC Works

You should understand the material in this section before you configure CBAC. If you do not understand how CBAC works, you might inadvertently introduce security risks by configuring CBAC inappropriately. This section contains the following sections:

How CBAC Works—Overview

CBAC creates temporary openings in access lists at firewall interfaces. These openings are created when specified traffic exits your internal network through the firewall. The openings allow returning traffic (that would normally be blocked) and additional data channels to enter your internal network back through the firewall. The traffic is allowed back through the firewall only if it is part of the same session as the original traffic that triggered CBAC when exiting through the firewall.

Throughout this chapter, the terms “inbound” and “outbound” are used to describe the direction of traffic relative to the router interface on which CBAC is applied. For example, if a CBAC rule is applied inbound on interface E0, then packets entering interface E0 from the network will be inspected. If a CBAC rule is applied outbound on interface E0, then packets leaving interface E0 to the network will be inspected. This is similar to the way ACLs work.

For example, consider a CBAC inspection rule named *hqusers*, and suppose that rule is applied inbound at interface E0:

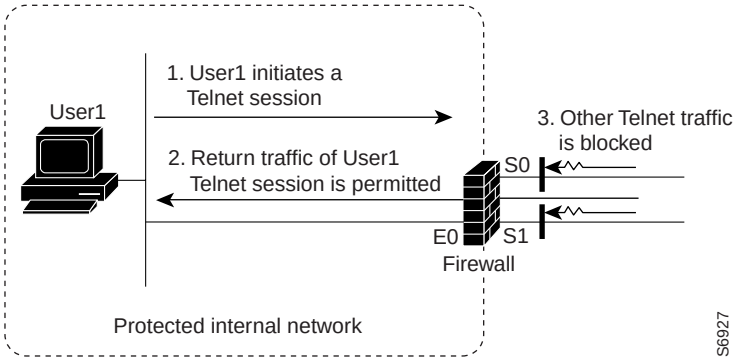
```
router (config-if)# ip inspect hqusers in
```

This command causes CBAC to inspect the packets coming into this interface from the network. If a packet is attempting to initiate a session, CBAC will then determine if this protocol is allowed, create a CBAC session, add the appropriate ACLs to allow return traffic and do any needed content inspection on any future packets for this session.

The terms “input” and “output” are used to describe the interfaces at which network traffic enters or exits the firewall router. A packet enters the firewall router via the input interface, is inspected by the firewall software and then exits the router via the output interface.

In Figure 20, the inbound access lists at S0 and S1 are configured to block Telnet traffic, and there is no outbound access list configured at E0. When the connection request for User1's Telnet session passes through the firewall, CBAC creates a temporary opening in the inbound access list at S0 to permit returning Telnet traffic for User1's Telnet session. (If the same access list is applied to both S0 and S1, the same opening would appear at both interfaces.) If necessary, CBAC would also have created a similar opening in an outbound access list at E0 to permit return traffic.

Figure 20 CBAC Opens Temporary Holes in Firewall Access Lists



How CBAC Works—Details

This section describes how CBAC inspects packets and maintains state information about sessions to provide intelligent filtering.

Packets Are Inspected

With CBAC, you specify which protocols you want to be inspected, and you specify an interface and interface direction (in or out) where inspection originates. Only specified protocols will be inspected by CBAC.

Packets entering the firewall are inspected by CBAC only if they first pass the inbound access list at the input interface and outbound access list at the output interface. If a packet is denied by the access list, the packet is simply dropped and not inspected by CBAC.

CBAC inspection tracks sequence numbers in all TCP packets, and drops those packets with sequence numbers that are not within expected ranges.

CBAC inspection recognizes application-specific commands (such as illegal SMTP commands) in the control channel, and detects and prevents certain application-level attacks.

When CBAC suspects an attack, the DoS feature can take several actions:

- Generate alert messages
- Protect system resources that could impede performance
- Block packets from suspected attackers

CBAC uses timeout and threshold values to manage session state information, helping to determine when to drop sessions that do not become fully established. Setting timeout values for network sessions helps prevent DoS attacks by freeing up system resources, dropping sessions after a specified amount of time. Setting threshold values for network sessions helps prevent DoS attacks by controlling the number of half-open sessions, which limits the amount of system resources applied to half-open sessions. When a

session is dropped, CBAC sends a reset message to the devices at both end points (source and destination) of the session. When the system under DoS attack receives a reset command, it releases, or frees up, processes and resources related to that incomplete session.

CBAC provides three thresholds against DoS attacks:

- The total number of half-open TCP or UDP sessions
- The number of half-open sessions based upon time
- The number of half-open TCP-only sessions per host

If a threshold is exceeded, CBAC has two options:

- Send a reset message to the end points of the oldest half-open session, making resources available to service newly arriving SYN packets.
- In the case of half open TCP only sessions, CBAC blocks all SYN packets temporarily for the duration configured by the threshold value. When the router blocks a SYN packet, the TCP three-way handshake is never initiated, which prevents the router from using memory and processing resources needed for valid connections.

DoS detection and prevention requires that you create a CBAC inspection rule and apply that rule on an interface. The inspection rule must include the protocols that you want to monitor against DoS attacks. For example, if you have TCP inspection enabled on the inspection rule, then CBAC can track all TCP connections to watch for DoS attacks. If the inspection rule includes FTP protocol inspection but not TCP inspection, CBAC tracks only FTP connections for DoS attacks.

For detailed information about setting timeout and threshold values in CBAC to detect and prevent DoS attacks, refer in the "Configuring Global Timeouts and Thresholds" section.

A State Table Maintains Session State Information

Whenever a packet is inspected, a state table is updated to include information about the state of the session.

Return traffic will only be permitted back through the firewall if the state table contains information indicating that the packet belongs to a permissible session. CBAC controls the traffic that belongs to a valid session. When return traffic is inspected, the state table information is updated as necessary.

UDP "Sessions" Are Approximated

With UDP—a connectionless service—there are no actual sessions, so the software approximates sessions by examining the information in the packet and determining if the packet is similar to other UDP packets (for example, same source/destination addresses and port numbers) and if the packet was detected soon after another similar UDP packet. "Soon" means within the configurable UDP idle timeout period.

Access List Entries Are Dynamically Created and Deleted to Permit Return Traffic and Additional Data Connections

CBAC dynamically creates and deletes access list entries at the firewall interfaces, according to the information maintained in the state tables. These access list entries are applied to the interfaces to examine traffic flowing back into the internal network. These entries create temporary openings in the firewall to permit only traffic that is part of a permissible session.

The temporary access list entries are never saved to NVRAM.

When and Where to Configure CBAC

Configure CBAC at firewalls protecting internal networks. Such firewalls should be Cisco routers with the Cisco IOS Firewall feature set configured as described previously in the section "Cisco IOS Firewall."

Use CBAC when the firewall will be passing traffic such as the following:

- Standard TCP and UDP Internet applications
- Multimedia applications
- Oracle support

Use CBAC for these applications if you want the application's traffic to be permitted through the firewall only when the traffic session is initiated from a particular side of the firewall (usually from the protected internal network).

In many cases, you will configure CBAC in one direction only at a single interface, which causes traffic to be permitted back into the internal network only if the traffic is part of a permissible (valid, existing) session. This is a typical configuration for protecting your internal networks from traffic that originates on the Internet.

You can also configure CBAC in two directions at one or more interfaces. CBAC is configured in two directions when the networks on both sides of the firewall should be protected, such as with extranet or intranet configurations, and to protect against DoS attacks. For example, if the firewall is situated between two partner companies' networks, you might wish to restrict traffic in one direction for certain applications, and restrict traffic in the opposite direction for other applications.

The CBAC Process

This section describes a sample sequence of events that occurs when CBAC is configured at an external interface that connects to an external network such as the Internet.

In this example, a TCP packet exits the internal network through the firewall's external interface. The TCP packet is the first packet of a Telnet session, and TCP is configured for CBAC inspection.

1. The packet reaches the firewall's external interface.
2. The packet is evaluated against the interface's existing outbound access list, and the packet is permitted. (A denied packet would simply be dropped at this point.)
3. The packet is inspected by CBAC to determine and record information about the state of the packet's connection. This information is recorded in a new state table entry created for the new connection.

   (If the packet's application—Telnet—was not configured for CBAC inspection, the packet would simply be forwarded out the interface at this point without being inspected by CBAC. See the section "Defining an Inspection Rule" later in this chapter for information about configuring CBAC inspection.)
4. Based on the obtained state information, CBAC creates a temporary access list entry which is inserted at the beginning of the external interface's inbound extended access list. This temporary access list entry is designed to permit inbound packets that are part of the same connection as the outbound packet just inspected.
5. The outbound packet is forwarded out the interface.
6. Later, an inbound packet reaches the interface. This packet is part of the same Telnet connection previously established with the outbound packet. The inbound packet is evaluated against the inbound access list, and it is permitted because of the temporary access list entry previously created.

7. The permitted inbound packet is inspected by CBAC, and the connection's state table entry is updated as necessary. Based on the updated state information, the inbound extended access list temporary entries might be modified in order to permit only packets that are valid for the current state of the connection.
8. Any additional inbound or outbound packets that belong to the connection are inspected to update the state table entry and to modify the temporary inbound access list entries as required, and they are forwarded through the interface.
9. When the connection terminates or times out, the connection's state table entry is deleted, and the connection's temporary inbound access list entries are deleted.

In the sample process just described, the firewall access lists are configured as follows:

- An outbound IP access list (standard or extended) is applied to the external interface. This access list permits all packets that you want to allow to exit the network, including packets you want to be inspected by CBAC. In this case, Telnet packets are permitted.
- An inbound extended IP access list is applied to the external interface. This access list denies any traffic to be inspected by CBAC—including Telnet packets. When CBAC is triggered with an outbound packet, CBAC creates a temporary opening in the inbound access list to permit only traffic that is part of a valid, existing session.

  If the inbound access list had been configured to permit *all* traffic, CBAC would be creating pointless openings in the firewall for packets that would be permitted anyway.

Supported Protocols

This section provides a list of CBAC supported protocols and includes a more detailed look at support for multimedia applications, specifically RTSP and H.323.

CBAC Supported Protocols

You can configure CBAC to inspect the following types of sessions:

- All TCP sessions, regardless of the application-layer protocol (sometimes called "single-channel" or "generic" TCP inspection)
- All UDP sessions, regardless of the application-layer protocol (sometimes called "single-channel" or "generic" UDP inspection)

You can also configure CBAC to specifically inspect certain application-layer protocols. The following application-layer protocols can all be configured for CBAC:

- CU-SeeMe (only the White Pine version)
- FTP
- H.323 (such as NetMeeting, ProShare)
- HTTP (Java blocking)
- Microsoft NetShow
- UNIX R-commands (such as rlogin, rexec, and rsh)
- RealAudio
- RTSP (Real Time Streaming Protocol)
- RPC (Sun RPC, not DCE RPC)

- SMTP (Simple Mail Transport Protocol)

Note CBAC can be configured to inspect SMTP but not ESMTP (Extended Simple Mail Transport Protocol). SMTP is described in RFC 821. CBAC SMTP inspect does not inspect the ESMTP session or command sequence. Configuring SMTP inspection is not useful for ESMTP, and it can cause problems.

To determine whether a mail-server is doing SMTP or ESMTP, contact your mail-server software vendor, or telnet to the mail-server port 25 and observe the banner to see if it reports SMTP or ESMTP.

- SQL*Net
- StreamWorks
- TFTP
- VDOLive

When a protocol is configured for CBAC, that protocol traffic is inspected, state information is maintained, and in general, packets are allowed back through the firewall only if they belong to a permissible session.

RTSP and H.323 Protocol Support for Multimedia Applications

CBAC supports a number of protocols for multimedia applications that require delivery of data with real-time properties such as audio and video conferencing. This support includes the following multimedia application protocols:

- Real Time Streaming Protocol (RTSP)
- H.323 Version 2 (H.323 V2)

RTSP and H.323 V2 inspection allows clients on a protected network to receive data associated with a multimedia session from a server on an unprotected network.

RTSP Support

RTSP is the IETF standards-based protocol (RFC 2326) for control over the delivery of data with real-time properties such as audio and video streams. It is useful for large-scale broadcasts and audio or video on demand streaming, and is supported by a variety of vendor products of streaming audio and video multimedia, including Cisco IP/TV, RealNetworks RealAudio G2 Player, and Apple QuickTime 4 software.

RFC 2326 allows RTSP to run over either UDP or TCP, though CBAC currently supports only TCP-based RTSP. RTSP establishes a TCP-based control connection, or channel, between the multimedia client and server. RTSP uses this channel to control commands such as “play” and “pause” between the client and server. These control commands and responses are text-based and are similar to HTTP.

RTSP typically relies on a UDP-based data transport protocol such as standard Real-Time Transport Protocol (RTP) to open separate channels for data and for RTP Control Protocol (RTCP) messages. RTP and RTCP channels occur in pairs, with RTP being an even numbered port and RTCP being the next consecutive port. Understanding the relationship of RTP and RTCP is important for verifying session information using CBAC **show** commands.

The RTSP client uses TCP port 554 or 8554 to open a multimedia connection with a server. The data channel or data control channel (using RTCP) between the client and the server is dynamically negotiated between the client and the server using any of the high UDP ports (1024 to 65536).

CBAC uses this port information along with connection information from the client to create dynamic access control list (ACL) entries in the firewall. As TCP or UDP connections are terminated, CBAC removes these dynamic entries from the appropriate ACLs.

CBAC support for RTSP includes the following data transport modes:

- Standard Real-Time Transport Protocol (RTP)

  RTP is an IETF standard (RFC 1889) supporting delivery of real-time data such as audio and video. RTP uses the RTP Control Protocol (RTCP) for managing the delivery of the multimedia data stream. This is the normal mode of operation for Cisco IP/TV and Apple QuickTime 4 software.

- RealNetworks Real Data Transport (RDT)

  RDT is a proprietary protocol developed by RealNetworks for data transport. This mode uses RTSP for communication control and uses RDT for the data connection and retransmission of lost packets. This is the normal mode of operation for the RealServer G2 from RealNetworks.

- Interleaved (Tunnel Mode)

  In this mode, RTSP uses the control channel to tunnel RTP or RDT traffic.

- Synchronized Multimedia Integration Language (SMIL)

  SMIL is a layout language that enables the creation of multimedia presentations consisting of multiple elements of music, voice, images, text, video and graphics. This involves multiple RTSP control and data streams between the player and the servers. This mode is available only using RTSP and RDT. SMIL is a proposed specification of the World Wide Web Consortium (W3C). The RealNetworks RealServer and RealServer G2 provide support for SMIL—Cisco IP/TV and Apple QuickTime 4 do not.

H.323 Support

CBAC support for H.323 inspection includes H.323 Version 2 and H.323 Version 1. H.323 V2 provides additional options over H.323 V1, including a “fast start” option. The fast start option minimizes the delay between the time that a user initiates a connection and the time that the user gets the data (voice, video). H.323 V2 inspection is backward compatible with H.323 V1.

With H.323 V1, after a TCP connection is established between the client and server (H.225 Channel), a separate channel for media control (H.245 Channel) is opened through which multimedia channels for audit and video are further negotiated.

The H.323 V2 client opens a connection to server which is listening on port 1720. The data channel between the client and the server is dynamically negotiated using any of the high UDP ports (1024 to 65536).

CBAC uses this port information along with connection information from the client to create dynamic access control list (ACL) entries in the firewall. As TCP or UDP connections are terminated, CBAC removes these dynamic entries from the appropriate ACLs.

Restrictions

CBAC has the following restrictions:

- CBAC is available only for IP protocol traffic. Only TCP and UDP packets are inspected. (Other IP traffic, such as ICMP, cannot be inspected with CBAC and should be filtered with basic access lists instead.)
- If you reconfigure your access lists when you configure CBAC, be aware that if your access lists block TFTP traffic into an interface, you will not be able to netboot over that interface. (This is not a CBAC-specific limitation, but is part of existing access list functionality.)
- Packets with the firewall as the source or destination address are not inspected by CBAC.
- CBAC ignores ICMP Unreachable messages.
- H.323 V2 and RTSP protocol inspection supports only the following multimedia client-server applications: Cisco IP/TV, RealNetworks RealAudio G2 Player, Apple QuickTime 4.

You can use CBAC together with all the other firewall features mentioned previously in the “Cisco IOS Firewall Overview” chapter.

CBAC works with fast switching and process switching.

This section also discusses restrictions concerning:

- FTP Traffic and CBAC
- IPSec and CBAC Compatibility

FTP Traffic and CBAC

- With FTP, CBAC does not allow third-party connections (three-way FTP transfer).
- When CBAC inspects FTP traffic, it only allows data channels with the destination port in the range of 1024 to 65535.
- CBAC will not open a data channel if the FTP client-server authentication fails.

IPSec and CBAC Compatibility

When CBAC and IPSec are enabled on the same router, and the firewall router is an endpoint for IPSec for the particular flow, then IPSec is compatible with CBAC (that is, CBAC can do its normal inspection processing on the flow).

If the router is not an IPSec endpoint, but the packet is an IPSec packet, then CBAC will not inspect the packets because the protocol number in the IP header of the IPSec packet is not TCP or UDP. CBAC only inspects UDP and TCP packets.

Memory and Performance Impact

CBAC uses less than approximately 600 bytes of memory per connection. Because of the memory usage, you should use CBAC only when you need to. There is also a slight amount of additional processing that occurs whenever packets are inspected.

Sometimes CBAC must evaluate long access lists, which might have presented a negative impact to performance. However, this impact is avoided, because CBAC evaluates access lists using an accelerated method (CBAC hashes access lists and evaluates the hash).

CBAC Configuration Task List

To configure CBAC, perform the tasks described in the following sections. The tasks in the first seven sections are required; the task of verifying the CBAC configuration is optional.

- Picking an Interface: Internal or External (Required)
- Configuring IP Access Lists at the Interface (Required)
- Configuring Global Timeouts and Thresholds (Required)
- Defining an Inspection Rule (Required)
- Applying the Inspection Rule to an Interface (Required)
- Configuring Logging and Audit Trail (Required)
- Other Guidelines for Configuring a Firewall (Required)
- Verifying CBAC (Optional)

Following CBAC configuration, you can monitor and maintain CBAC using the information in this section.

Note If you try to configure Context-based Access Control (CBAC) but do not have a good understanding of how CBAC works, you might inadvertently introduce security risks to the firewall and to the protected network. You should be sure you understand what CBAC does before you configure CBAC.

Note As with all networking devices, protect access into the firewall by configuring passwords as described in the "Configuring Passwords and Privileges" chapter. You should also consider configuring user authentication, authorization, and accounting as described in the "Authentication, Authorization, and Accounting (AAA)" part of this guide. Additional guidelines to help you establish a good security policy can be found in the "Cisco IOS Firewall Overview" chapter.

For CBAC configuration examples, refer to the "CBAC Configuration Examples" section at the end of this chapter.

Picking an Interface: Internal or External

You must decide whether to configure CBAC on an internal or external interface of your firewall.

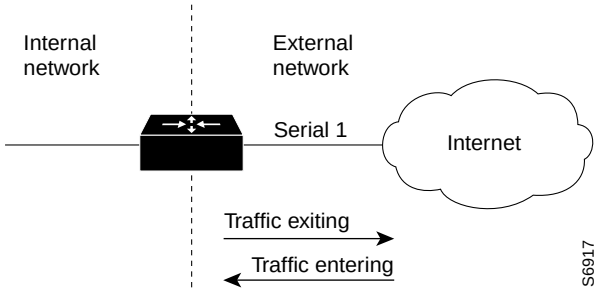
"Internal" refers to the side where sessions must originate for their traffic to be permitted through the firewall. "External" refers to the side where sessions cannot originate (sessions originating from the external side will be blocked).

If you will be configuring CBAC in two directions, you should configure CBAC in one direction first, using the appropriate "internal" and "external" interface designations. When you configure CBAC in the other direction, the interface designations will be swapped. (CBAC can be configured in two directions at one or more interfaces. Configure CBAC in two directions when the networks on both sides of the firewall require protection, such as with extranet or intranet configurations, and for protection against DoS attacks.)

The firewall is most commonly used with one of two basic network topologies. Determining which of these topologies is most like your own can help you decide whether to configure CBAC on an internal interface or on an external interface.

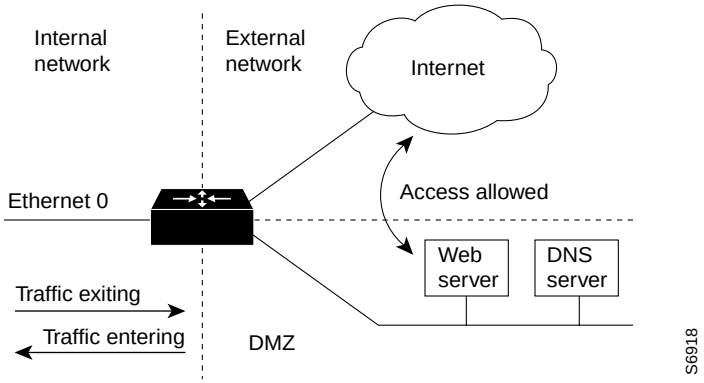
The first topology is shown in Figure 21. In this simple topology, CBAC is configured for the *external* interface Serial 1. This prevents specified protocol traffic from entering the firewall and the internal network, unless the traffic is part of a session initiated from within the internal network.

Figure 21 Simple Topology—CBAC Configured at the External Interface



The second topology is shown in Figure 22. In this topology, CBAC is configured for the *internal* interface Ethernet 0. This allows external traffic to access the services in the Demilitarized Zone (DMZ), such as DNS services, but prevents specified protocol traffic from entering your internal network—unless the traffic is part of a session initiated from within the internal network.

Figure 22 DMZ Topology—CBAC Configured at the Internal Interface



Using these two sample topologies, decide whether to configure CBAC on an internal or external interface.

To view various firewall configuration scenarios, see the "CBAC Configuration Examples" section at the end of this chapter.

Configuring IP Access Lists at the Interface

For CBAC to work properly, you need to make sure that you have IP access lists configured appropriately at the interface.

Follow these three general rules when evaluating your IP access lists at the firewall:

- Start with a basic configuration.

If you try to configure access lists without a good understanding of how access lists work, you might inadvertently introduce security risks to the firewall and to the protected network. You should be sure you understand what access lists do before you configure your firewall. For more information about access control lists, refer to the "Access Control Lists: Overview and Guidelines" chapter.

A basic initial configuration allows all network traffic to flow from the protected networks to the unprotected networks, while blocking network traffic from any unprotected networks.

- Permit CBAC traffic to leave the network through the firewall.

  All access lists that evaluate traffic leaving the protected network should permit traffic that will be inspected by CBAC. For example, if Telnet will be inspected by CBAC, then Telnet traffic should be permitted on all access lists that apply to traffic leaving the network.

- Use extended access lists to deny CBAC return traffic entering the network through the firewall.

  For temporary openings to be created in an access list, the access list must be an extended access list. So wherever you have access lists that will be applied to returning traffic, you must use extended access lists. The access lists should deny CBAC return traffic because CBAC will open up temporary holes in the access lists. (You want traffic to be normally blocked when it enters your network.)

Note If your firewall only has two connections, one to the internal network and one to the external network, using all inbound access lists works well because packets are stopped before they get a chance to affect the router itself.

This section contains the following sections:

Basic Configuration

The first time you configure the Cisco IOS Firewall, it is helpful to start with a basic access list configuration that makes the operation of the firewall easy to understand without compromising security. The basic configuration allows all network traffic from the protected networks access to the unprotected networks, while blocking all network traffic (with some exceptions) from the unprotected networks to the protected networks.

Any firewall configuration depends on your site security policy. If the basic configuration does not meet your initial site security requirements, configure the firewall to meet your policy. If you are unfamiliar with that policy or need help with the configuration, contact your network administration group for assistance. For additional guidelines on configuring a firewall, refer to the "Other Guidelines for Configuring a Firewall" section in this chapter.

Use the following guidelines for configuring the initial firewall access lists:

- Do not configure an access list for traffic from the protected networks to the unprotected networks, meaning that all traffic from the protected networks can flow through the interface.

  This helps to simplify firewall management by reducing the number of access lists applied at the interfaces. Of course this assumes a high level of trust for the users on the protected networks, and it assumes there are no malicious users on the protected networks who might launch attacks from the "inside." You can fine tune network access for users on the protected networks as you gain experience with access list configuration and the operation of the firewall.

- Configure an access list that includes entries permitting certain ICMP traffic from unprotected networks.

  While an access list that denies all IP traffic not part of a connection inspected by CBAC seems most secure, it is not practical for normal operation of the router. The router expects to see ICMP traffic from other routers in the network. Additionally, ICMP traffic is not inspected by CBAC, meaning specific entries are needed in the access list to permit return traffic for ICMP commands. For example, a user on a protected network uses the **ping** command to get the status of a host on an unprotected network; without entries in the access list that permit **echo reply** messages, the user on the protected network gets no response to the **ping** command.

  Include access list entries to permit the following ICMP messages:

Message	Description
echo reply	Outgoing ping commands require echo-reply messages to come back.
time-exceeded	Outgoing traceroute commands require time-exceeded messages to come back.
packet-too-big	Path MTU discovery requires “too-big” messages to come back.
traceroute	Allow an incoming traceroute.
unreachable	Permit all “unreachable” messages to come back. If a router cannot forward or deliver a datagram, it sends an ICMP unreachable message back to the source and drops the datagram.

- Add an access list entry denying any network traffic from a source address matching an address on the protected network.

  This is known as anti-spoofing protection because it prevents traffic from an unprotected network from assuming the identity of a device on the protected network.

- Add an entry denying broadcast messages with a source address of 255.255.255.255.

  This entry helps to prevent broadcast attacks.

- By default, the last entry in an extended access list is an implicit denial of all IP traffic not specifically allowed by other entries in the access list.

  Although this is the default setting, this final deny statement is not shown by default in an access list. Optionally, you can add an entry to the access list denying IP traffic with any source or destination address with no undesired effects.

For complete information about how to configure IP access lists, refer to the “Configuring IP Services” chapter of the *Cisco IOS IP Addressing Services Configuration Guide*.

For tips on applying access lists at an external or internal interface, review the sections “External Interface” and “Internal Interface” in this chapter.

External Interface

Here are some guidelines for your access lists when you will be configuring CBAC on an external interface:

- If you have an outbound IP access list at the external interface, the access list can be a standard or extended access list. This outbound access list should permit traffic that you want to be inspected by CBAC. If traffic is not permitted, it will not be inspected by CBAC, but will be simply dropped.

- The inbound IP access list at the external interface must be an extended access list. This inbound access list should deny traffic that you want to be inspected by CBAC. (CBAC will create temporary openings in this inbound access list as appropriate to permit only return traffic that is part of a valid, existing session.)
- For complete information about how to configure IP access lists, refer to the "Configuring IP Services" chapter of the *Cisco IOS IP Addressing Services Configuration Guide*.

Internal Interface

Here are some tips for your access lists when you will be configuring CBAC on an internal interface:

- If you have an inbound IP access list at the internal interface or an outbound IP access list at external interface(s), these access lists can be either a standard or extended access list. These access lists should permit traffic that you want to be inspected by CBAC. If traffic is not permitted, it will not be inspected by CBAC, but will be simply dropped.
- The outbound IP access list at the internal interface and the inbound IP access list at the external interface must be extended access lists. These outbound access lists should deny traffic that you want to be inspected by CBAC. (CBAC will create temporary openings in these outbound access lists as appropriate to permit only return traffic that is part of a valid, existing session.) You do not necessarily need to configure an extended access list at both the outbound internal interface and the inbound external interface, but at least one is necessary to restrict traffic flowing through the firewall into the internal protected network.
- For complete information about how to configure IP access lists, refer to the "Configuring IP Services" chapter of the *Cisco IOS IP Addressing Services Configuration Guide*.

Configuring Global Timeouts and Thresholds

CBAC uses timeouts and thresholds to determine how long to manage state information for a session, and to determine when to drop sessions that do not become fully established. These timeouts and thresholds apply globally to all sessions.

You can use the default timeout and threshold values, or you can change to values more suitable to your security requirements. You should make any changes to the timeout and threshold values before you continue configuring CBAC.

Note If you want to enable the more aggressive TCP host-specific denial-of-service prevention that includes the blocking of connection initiation to a host, you must set the **block-time** specified in the **ip inspect tcp max-incomplete host** command (see the last row in Table 24).

All the available CBAC timeouts and thresholds are listed in Table 24, along with the corresponding command and default value. To change a global timeout or threshold listed in the "Timeout of Threshold Value to Change" column, use the global configuration command in the "Command" column:

Table 24 Timeout and Threshold Values

Timeout or Threshold Value to Change	Command	Default
The length of time the software waits for a TCP session to reach the established state before dropping the session.	**ip inspect tcp synwait-time** *seconds*	30 seconds
The length of time a TCP session will still be managed after the firewall detects a FIN-exchange.	**ip inspect tcp finwait-time** *seconds*	5 seconds
The length of time a TCP session will still be managed after no activity (the TCP idle timeout).[1]	**ip inspect tcp idle-time** *seconds*	3600 seconds (1 hour)
The length of time a UDP session will still be managed after no activity (the UDP idle timeout).[1]	**ip inspect udp idle-time** *seconds*	30 seconds
The length of time a DNS name lookup session will still be managed after no activity.	**ip inspect dns-timeout** *seconds*	5 seconds
The number of existing half-open sessions that will cause the software to start deleting half-open sessions.[2]	**ip inspect max-incomplete high** *number*	500 existing half-open sessions
The number of existing half-open sessions that will cause the software to stop deleting half-open sessions.[2]	**ip inspect max-incomplete low** *number*	400 existing half-open sessions
The rate of new sessions that will cause the software to start deleting half-open sessions.[2]	**ip inspect one-minute high** *number*	500 half-open sessions per minute
The rate of new sessions that will cause the software to stop deleting half-open sessions.[2]	**ip inspect one-minute low** *number*	400 half-open sessions per minute
The number of existing half-open TCP sessions with the same destination host address that will cause the software to start dropping half-open sessions to the same destination host address.[3]	**ip inspect tcp max-incomplete host** *number* **block-time** *minutes*	50 existing half-open TCP sessions; 0 minutes

1. The global TCP and UDP idle timeouts can be overridden for specified application-layer protocols' sessions as described in the **ip inspect name** (global configuration) command description, found in the "Context-Based Access Control Commands" chapter of the *Cisco IOS Security Command Reference*.
2. See the following section, "Half-Open Sessions," for more information.
3. Whenever the **max-incomplete host** threshold is exceeded, the software will drop half-open sessions differently depending on whether the **block-time** timeout is zero or a positive non-zero number. If the **block-time** timeout is zero, the software will delete the oldest existing half-open session for the host for every new connection request to the host and will let the SYN packet through. If the **block-time** timeout is greater than zero, the software will delete all existing half-open sessions for the host, and then block all new connection requests to the host. The software will continue to block all new connection requests until the **block-time** expires.

To reset any threshold or timeout to the default value, use the **no** form of the command in Table 24.

Half-Open Sessions

An unusually high number of half-open sessions (either absolute or measured as the arrival rate) could indicate that a denial-of-service attack is occurring. For TCP, "half-open" means that the session has not reached the established state—the TCP three-way handshake has not yet been completed. For UDP, "half-open" means that the firewall has detected no return traffic.

CBAC measures both the total number of existing half-open sessions and the rate of session establishment attempts. Both TCP and UDP half-open sessions are counted in the total number and rate measurements. Rate measurements are made several times per minute.

When the number of existing half-open sessions rises above a threshold (the **max-incomplete high** number), the software will delete half-open sessions as required to accommodate new connection requests. The software will continue to delete half-open requests as necessary, until the number of existing half-open sessions drops below another threshold (the **max-incomplete low** number).

When the rate of new connection attempts rises above a threshold (the **one-minute high** number), the software will delete half-open sessions as required to accommodate new connection attempts. The software will continue to delete half-open sessions as necessary, until the rate of new connection attempts drops below another threshold (the **one-minute low** number). The rate thresholds are measured as the number of new session connection attempts detected in the last one-minute sample period.

Defining an Inspection Rule

After you configure global timeouts and thresholds, you must define an inspection rule. This rule specifies what IP traffic (which application-layer protocols) will be inspected by CBAC at an interface.

Normally, you define only one inspection rule. The only exception might occur if you want to enable CBAC in two directions as described earlier in the section "When and Where to Configure CBAC." For CBAC configured in both directions at a single firewall interface, you should configure two rules, one for each direction.

An inspection rule should specify each desired application-layer protocol as well as generic TCP or generic UDP if desired. The inspection rule consists of a series of statements each listing a protocol and specifying the same inspection rule name.

Inspection rules include options for controlling alert and audit trail messages and for checking IP packet fragmentation.

To define an inspection rule, follow the instructions in the following sections:

- Configuring Application-Layer Protocol Inspection
- Configuring Generic TCP and UDP Inspection

Configuring Application-Layer Protocol Inspection

This section provides instructions for configuring CBAC with the following inspection information:

- Configuring Application-Layer Protocols
- Configuring Java Blocking
- Configuring IP Packet Fragmentation Inspection

Note For CBAC inspection to work with NetMeeting 2.0 traffic (an H.323 application-layer protocol), you must also configure inspection for TCP, as described later in the "Configuring Generic TCP and UDP Inspection" section. This requirement exists because NetMeeting 2.0 uses an additional TCP channel not defined in the H.323 specification.

Configuring Application-Layer Protocols

To configure CBAC inspection for an application-layer protocol, use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ip inspect name inspection-name protocol [alert {on | off}] [audit-trail {on | off}] [timeout seconds]`	Configures CBAC inspection for an application-layer protocol (except for RPC and Java). Use one of the protocol keywords defined in Table 25. Repeat this command for each desired protocol. Use the same *inspection-name* value to create a single inspection rule.
`Router(config)# ip inspect name inspection-name rpc program-number number [wait-time minutes] [alert {on | off}] [audit-trail {on | off}] [timeout seconds]`	Enables CBAC inspection for the RPC application-layer protocol. You can specify multiple RPC program numbers by repeating this command for each program number. Use the same *inspection-name* value to create a single inspection rule.

Refer to the description of the **ip inspect name** global configuration command in the "Context-Based Access Control Commands" chapter of the *Cisco IOS Security Command Reference* for more information about how the command works with each application-layer protocol.

To enable CBAC inspection for Java blocking, see the following section, "Configuring Java Blocking."

Table 25 identifies application protocol keywords for the **ip inspect name** command.

Table 25 Application Protocol Keywords for the ip inspect name Command

Application Protocol	Protocol Keyword
CU-SeeMe	**cuseeme**
FTP	**ftp**
H.323	**h323**
Microsoft NetShow	**netshow**
UNIX R commands (rlogin, rexec, rsh)	**rcmd**
RealAudio	**realaudio**
SMTP	**smtp**
SQL*Net	**sqlnet**
StreamWorks	**streamworks**
TFTP	**tftp**
VDOLive	**vdolive**

Configuring Java Blocking

Java applet filtering distinguishes between trusted and untrusted applets by relying on a list of external sites that you designate as "friendly." If an applet is from a friendly site, the firewall allows the applet through. If the applet is not from a friendly site, the applet will be blocked. (Alternately, you could permit applets from all external sites except for those you specifically designate as hostile.)

Note Java blocking forces a strict order on TCP packets. To properly verify that Java applets are not in the response, a firewall will drop any TCP packet that is out of order. Because the network—not the firewall—determines how packets are routed, the firewall cannot control the order of the packets; the firewall can only drop and retransmit all TCP packets that are not in order.

To block all Java applets except for applets from friendly locations, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip access-list standard` *name* `  permit ...` `  deny ... (Use permit and deny statements as appropriate.)` or `Router(config)# access-list` *access-list-number* {`deny` | `permit`} *protocol source* [*source-wildcard*]`eq www` *destination* [*destination-wildcard*]	Creates a standard access list that permits traffic only from friendly sites, and denies traffic from hostile sites. Use the **any** keyword for the destination as appropriate—but be careful to not misuse the **any** keyword to inadvertently allow all applets through.
Step 2	`Router(config)# ip inspect name inspection-name http [java-list` *access-list*`] [alert {on | off}] [audit-trail {on | off}] [timeout` *seconds*`]`	Blocks all Java applets except for applets from the friendly sites defined previously in the access list. Java blocking only works with numbered standard access lists. To create a single inspection rule, use the same *inspection-name* value as when you specified other protocols.

Caution CBAC does not detect or block encapsulated Java applets. Therefore, Java applets that are wrapped or encapsulated, such as applets in .zip or .jar format, are *not* blocked at the firewall. CBAC also does not detect or block applets loaded from FTP, gopher, HTTP on a nonstandard port, and so forth.

Configuring IP Packet Fragmentation Inspection

CBAC inspection rules can help protect hosts against certain DoS attacks involving fragmented IP packets.

Using fragmentation inspection, the firewall maintains an *interfragment state* (structure) for IP traffic. Non-initial fragments are discarded unless the corresponding initial fragment was permitted to pass through the firewall. Non-initial fragments received before the corresponding initial fragments are discarded.

Note Fragmentation inspection can have undesirable effects in certain cases, because it can result in the firewall discarding any packet whose fragments arrive out of order. There are many circumstances that can cause out-of-order delivery of legitimate fragments. Applying fragmentation inspection in situations where legitimate fragments, which are likely to arrive out of order, might have a severe performance impact.

Because routers running Cisco IOS software are used in a large variety of networks, and because the CBAC feature is often used to isolate parts of internal networks from one another, the fragmentation inspection feature is disabled by default. Fragmentation detection must be explicitly enabled for an inspection rule using the **ip inspect name** command. Unfragmented traffic is never discarded because it lacks a fragment state. Even when the system is under heavy attack with fragmented packets, legitimate fragmented traffic, if any, gets some fraction of the firewall's fragment state resources, and legitimate, unfragmented traffic can flow through the firewall unimpeded.

Configuring Generic TCP and UDP Inspection

You can configure TCP and UDP inspection to permit TCP and UDP packets to enter the internal network through the firewall, even if the application-layer protocol is not configured to be inspected. However, TCP and UDP inspection do not recognize application-specific commands, and therefore might not permit all return packets for an application, particularly if the return packets have a different port number than the previous exiting packet.

Any application-layer protocol that is inspected will take precedence over the TCP or UDP packet inspection. For example, if inspection is configured for FTP, all control channel information will be recorded in the state table, and all FTP traffic will be permitted back through the firewall if the control channel information is valid for the state of the FTP session. The fact that TCP inspection is configured is irrelevant to the FTP state information.

With TCP and UDP inspection, packets entering the network must exactly match the corresponding packet that previously exited the network. The entering packets must have the same source/destination addresses and source/destination port numbers as the exiting packet (but reversed); otherwise, the entering packets will be blocked at the interface. Also, all TCP packets with a sequence number outside of the window are dropped.

With UDP inspection configured, replies will only be permitted back in through the firewall if they are received within a configurable time after the last request was sent out. (This time is configured with the **ip inspect udp idle-time** command.)

To configure CBAC inspection for TCP or UDP packets, use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ip inspect name` *inspection-name* `tcp [alert {on | off}] [audit-trail {on | off}] [timeout` *seconds*`]`	Enables CBAC inspection for TCP packets. To create a single inspection rule, use the same *inspection-name* value as when you specified other protocols.
`Router(config)# ip inspect name` *inspection-name* `udp [alert {on | off}] [audit-trail {on | off}] [timeout` *seconds*`]`	Enables CBAC inspection for UDP packets. To create a single inspection rule, use the same *inspection-name* value as when you specified other protocols.

Applying the Inspection Rule to an Interface

After you define an inspection rule, you apply this rule to an interface.

Normally, you apply only one inspection rule to one interface. The only exception might occur if you want to enable CBAC in two directions as described earlier in the section “When and Where to Configure CBAC.” For CBAC configured in both directions at a single firewall interface, you should apply two rules, one for each direction.

If you are configuring CBAC on an external interface, apply the rule to outbound traffic.

If you are configuring CBAC on an internal interface, apply the rule to inbound traffic.

To apply an inspection rule to an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip inspect` *inspection-name* `{in | out}`	Applies an inspection rule to an interface.

Configuring Logging and Audit Trail

Turn on logging and audit trail to provide a record of network access through the firewall, including illegitimate access attempts, and inbound and outbound services. To configure logging and audit trail functions, enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# service timestamps log datetime`	Adds the date and time to syslog and audit trail messages.
Step 2	`Router(config)# logging` *host*	Specifies the host name or IP address of the host where you want to send syslog messages.
Step 3	`Router(config)# logging facility` *facility-type*	Configures the syslog facility in which error messages are sent.
Step 4	`Router(config)# logging trap` *level*	(Optional) Uses this command to limit messages logged to the syslog servers based on severity. The default is level 7 (informational).
Step 5	`Router(config)# ip inspect audit-trail`	Turns on CBAC audit trail messages.

For information on how to interpret the syslog and audit trail messages, refer to the "Interpreting Syslog and Console Messages Generated by CBAC" section.

To configure audit trail functions on a per-application basis, refer to the "Defining an Inspection Rule" section for more information.

For complete information about how to configure logging, refer to the "Troubleshooting the Router" chapter of the *Cisco IOS Network Management Configuration Guide*.

Other Guidelines for Configuring a Firewall

As with all networking devices, you should always protect access into the firewall by configuring passwords as described in the module "Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices." You should also consider configuring user authentication, authorization, and accounting as described in the "Authentication, Authorization, and Accounting (AAA)" part of this guide.

You should also consider the following recommendations:

- When setting passwords for privileged access to the firewall, use the **enable secret** command rather than the **enable password** command, which does not have as strong an encryption algorithm.

- Put a password on the console port. In authentication, authorization, and accounting (AAA) environments, use the same authentication for the console as for elsewhere. In a non-AAA environment, at a minimum configure the **login** and **password** *password* commands.
- Think about access control *before* you connect a console port to the network in any way, including attaching a modem to the port. Be aware that a *break* on the console port might give total control of the firewall, even with access control configured.
- Apply access lists and password protection to all virtual terminal ports. Use access lists to limit who can Telnet into your router.
- Do not enable any local service (such as SNMP or NTP) that you do not use. Cisco Discovery Protocol (CDP) and Network Time Protocol (NTP) are on by default, and you should turn these off if you do not need them.

  To turn off CDP, enter the **no cdp run** global configuration command. To turn off NTP, enter the **ntp disable** interface configuration command on each interface not using NTP.

  If you must run NTP, configure NTP only on required interfaces, and configure NTP to listen only to certain peers.

  Any enabled service could present a potential security risk. A determined, hostile party might be able to find creative ways to misuse the enabled services to access the firewall or the network.

  For local services that are enabled, protect against misuse. Protect by configuring the services to communicate only with specific peers, and protect by configuring access lists to deny packets for the services at specific interfaces.
- Protect against spoofing: protect the networks on both sides of the firewall from being spoofed from the other side. You could protect against spoofing by configuring input access lists at all interfaces to pass only traffic from expected source addresses, and to deny all other traffic.

  You should also disable source routing. For IP, enter the **no ip source-route** global configuration command. Disabling source routing at *all* routers can also help prevent spoofing.

  You should also disable minor services. For IP, enter the **no service tcp-small-servers** and **no service udp-small-servers** global configuration commands. In Cisco IOS Release 12.0 and later, these services are disabled by default.
- Prevent the firewall from being used as a relay by configuring access lists on any asynchronous Telnet ports.
- Normally, you should disable directed broadcasts for all applicable protocols on your firewall and on all your other routers. For IP, use the **no ip directed-broadcast** command. Rarely, some IP networks do require directed broadcasts; if this is the case, do not disable directed broadcasts.

  Directed broadcasts can be misused to multiply the power of denial-of-service attacks, because every denial-of-service packet sent is broadcast to every host on a subnet. Furthermore, some hosts have other intrinsic security risks present when handling broadcasts.
- Configure the **no proxy-arp** command to prevent internal addresses from being revealed. (This is important to do if you do not already have NAT configured to prevent internal addresses from being revealed.)
- Keep the firewall in a secured (locked) room.

Verifying CBAC

You can view and verify CBAC configuration, status, statistics, and session information by using one or more of the following commands in EXEC mode:

Command	Purpose
`Router# show ip access-lists`	Displays the contents of all current IP access lists.
`Router# show ip inspect name inspection-name`	Shows a particular configured inspection rule.
`Router# show ip inspect config`	Shows the complete CBAC inspection configuration.
`Router# show ip inspect interfaces`	Shows interface configuration with regards to applied inspection rules and access lists.
`Router# show ip inspect session [detail]`	Shows existing sessions that are currently being tracked and inspected by CBAC.
`Router# show ip inspect all`	Shows all CBAC configuration and all existing sessions that are currently being tracked and inspected by CBAC.

In most cases, you can tell whether CBAC is inspecting network traffic properly because network applications are working as expected. In some cases, however, you might want to verify CBAC operation. For example, to verify RTSP or H.323 inspection, initiate an RTSP- or H.323-based application through the firewall. Use the **show ip inspect session** and **show ip access lists** commands to verify CBAC operation. These commands display the dynamic ACL entries and the established connections for a multimedia session.

In the case of RTSP inspection, session output can vary based on the multimedia protocol and the transport mode. This section uses examples of RTSP and H.323 V2 sessions to illustrate verification procedures and to illustrate how session information, and the interpretation of that session information, varies based on the protocol being inspected. This section provides the following sample session output:

- RTSP with RDT
- RTSP with TCP Only (Interleaved Mode)
- RTSP with SMIL
- RTSP with RTP (IP/TV)
- H.323 V2

RTSP with RDT

The following example illustrates the result of the **show ip inspect session** command. It shows that a control channel (rtsp) and data channel (rtsp-data) are open between hosts 192.168.155.2 and 192.168.35.1.

```
router# show ip inspect session
Established Sessions
 Session 616B4F1C (192.168.155.2:7548)=>(192.168.35.1:6970) rtsp-data SIS_OPEN
 Session 611E2904 (192.168.35.1:1221)=>(192.168.155.2:554) rtsp SIS_OPEN
```

The following example illustrates the result of the **show ip access-list** command. It shows that two dynamic entries (permit statements) were added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 554 (RTSP protocol port) on the client and port 1221 on the server. The UDP entry creates a dynamic opening between data port 7548 on the client and data port 6970 on the server.

```
router# show ip access-list
Extended IP access list 100
 permit udp host 192.168.155.2 eq 7548 host 192.168.35.1 eq 6970 (31 matches)
 permit tcp host 192.168.155.2 eq 554 host 192.168.35.1 eq 1221 (27 matches)
```

After closing the multimedia session, review the session output using the **show** commands to verify the firewall software has removed the dynamic entries from the configuration.

RTSP with TCP Only (Interleaved Mode)

The following example illustrates the result of the **show ip inspect session** command. It shows that only a single control channel (rtsp) is open between hosts 192.168.155.2 and 192.168.35.1. In this mode, data is tunneled through the firewall using the TCP connection to interleave RDT or RTP data.

```
router# show ip inspect session
Established Sessions
 Session 611E2904 (192.168.35.1:1228)=>(192.168.155.2:554) rtsp SIS_OPEN
```

The following example illustrates the result of the **show ip access-list** command. It shows that a single dynamic entry (permit statement) was added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 554 (RTSP protocol port) on the client and port 1228 on the server.

```
router# show ip access-lists
Extended IP access list 100
 permit tcp host 192.168.155.2 eq 554 host 192.168.35.1 eq 1228 (391 matches)
```

After closing the multimedia session, review the session output using the **show** commands to verify the firewall software has removed the dynamic entries from the configuration.

RTSP with SMIL

The following example illustrates the result of the **show ip inspect session** command for RTSP using Synchronized Multimedia Integration Language (SMIL). It shows that a single control channel (rtsp) and multiple data channels (rtsp-data) are open between hosts 192.168.155.2 and 192.168.35.1. The data channels appear as half open sessions because the UDP data flows in one direction only, which is from the server to the client.

```
router# show ip inspect session
Established Sessions
 Session 616CA914 (192.168.155.2:30616)=>(192.168.35.1:6974) rtsp-data SIS_OPEN
 Session 616B4E78 (192.168.35.1:1230)=>(192.168.155.2:554) rtsp SIS_OPEN
 Session 614AB61C (192.168.155.2:29704)=>(192.168.35.1:6976) rtsp-data SIS_OPEN
 Session 616CAA88 (192.168.155.2:26764)=>(192.168.35.1:6972) rtsp-data SIS_OPEN
Half-open Sessions
 Session 614AAEF0 (192.168.155.2:15520)=>(192.168.35.1:6970) rtsp-data SIS_OPENING
```

The following example illustrates the result of the **show ip access-lists** command. It shows that multiple dynamic entries (permit statements) were added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 554 (RTSP protocol port) on the client and port 1230 on the server. The UDP entries create dynamic openings between negotiated data ports on the client (192.168.155.2) and the server (192.168.35.1).

```
router# show ip access-list
Extended IP access list 100
 permit udp host 192.168.155.2 eq 29704 host 192.168.35.1 eq 6976 (182 matches)
 permit udp host 192.168.155.2 eq 30616 host 192.168.35.1 eq 6974 (268 matches)
 permit udp host 192.168.155.2 eq 26764 host 192.168.35.1 eq 6972 (4 matches)
 permit udp host 192.168.155.2 eq 15520 host 192.168.35.1 eq 6970 (12 matches)
 permit tcp host 192.168.155.2 eq 554 host 192.168.35.1 eq 1230 (41 matches)
```

After closing the multimedia session, review the session output using the **show** commands to verify the firewall software has removed the dynamic entries from the configuration.

RTSP with RTP (IP/TV)

The following example illustrates the result of the **show ip inspect session** command for RTSP with the Cisco IP/TV application. The output shows that a single control channel (rtsp) and multiple data channels (rtsp-data) are open between hosts 192.168.2.15 and 192.168.102.23. The data channels appear as half-open sessions because the UDP data flows in one direction only, which is from the server to the client.

```
router# show ip inspect session
Established Sessions
 Session 611493C0 (192.168.2.15:2571)=>(192.168.102.23:8554) rtsp SIS_OPEN
Half-open Sessions
 Session 6114A22C (192.168.102.23:2428)=>(192.168.2.15:20112) rtsp-data SIS_OPENING
 Session 61149F44 (192.168.102.23:2428)=>(192.168.2.15:20113) rtsp-data SIS_OPENING
 Session 6114A0B8 (192.168.102.23:2429)=>(192.168.2.15:20115) rtsp-data SIS_OPENING
 Session 6114A3A0 (192.168.102.23:2429)=>(192.168.2.15:20114) rtsp-data SIS_OPENING
```

The following example illustrates the result of the **show ip access-lists** command. It shows that multiple dynamic entries (permit statements) were added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 554 (RTSP protocol port) on the client and port 1230 on the server. The UDP entries create dynamic openings between negotiated data ports on the client (192.168.2.15) and the server (192.168.102.23).

```
router# show ip access-lists
Extended IP access list 100
 permit udp host 192.168.102.23 eq 2428 host 192.168.2.15 eq 20113 (11 matches)
 permit udp host 192.168.102.23 eq 2428 host 192.168.2.15 eq 20112 (256 matches)
 permit udp host 192.168.102.23 eq 2429 host 192.168.2.15 eq 20115 (11 matches)
 permit udp host 192.168.102.23 eq 2429 host 192.168.2.15 eq 20114 (4598 matches)
 permit tcp host 192.168.102.23 eq 8554 host 192.168.2.15 eq 2571 (22 matches)
```

After closing the multimedia session, review the session output using the **show** commands to verify that the firewall software has removed the dynamic entries from the configuration.

H.323 V2

The following example illustrates the result of the **show ip inspect session** command for H.323 V2. It shows a single H.323 control channel, an RTP Control Protocol channel for both audio and video data, and an RTP data channel between hosts 192.168.155.2 and 192.168.35.1.

```
Session 615E2688 (192.168.35.1:49609)=>(192.168.155.1:49609) H323-RTCP-audio SIS_OPEN
Session 615E2688 (192.168.35.1:49508)=>(192.168.155.1:49508) H323-RTP-audio SIS_OPEN
Session 615E2688 (192.168.35.1:49410)=>(192.168.155.1:49410) H323-RTP-video SIS_OPEN
Session 615E2688 (192.168.35.1:49611)=>(192.168.155.1:49611) H323-RTCP-video SIS_OPEN
Session 615E1640 (192.168.35.1:4414)=>(192.168.155.1:1720) H323 SIS_OPEN
```

The following example illustrates the result of the **show ip access-lists** command. It shows that multiple dynamic entries (permit statements) were added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 1720 (H.323 V2 protocol port) on the client and port 4414 on the server. The UDP entries create dynamic openings between negotiated data ports on the client (192.168.155.1) and the server (192.168.35.1).

```
router# show ip access-lists
Extended IP access list 100
 permit udp host 192.168.155.1 eq 49609 host 192.168.35.1 eq 49609 (11 matches)
 permit udp host 192.168.155.1 eq 49508 host 192.168.35.1 eq 49508 (256 matches)
 permit udp host 192.168.155.1 eq 49411 host 192.168.35.1 eq 49411 (11 matches)
 permit udp host 192.168.155.1 eq 49610 host 192.168.35.1 eq 49610 (4598 matches)
 permit tcp host 192.168.155.1 eq 1720 host 192.168.35.1 eq 4414 (22 matches)
```

Monitoring and Maintaining CBAC

You can watch for network attacks and investigate network problems using debug commands and system messages. This section has the following sections:

Debugging Context-Based Access Control

To assist CBAC debugging, you can turn on audit trail messages that will be displayed on the console after each CBAC session closes. Audit trail information is also configurable on a per-application basis using the CBAC inspection rules.

To turn on audit trail messages, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip inspect audit-trail`	Turns on CBAC audit trail messages.

If required, you can also use the CBAC **debug** commands listed in this section. (Debugging can be turned off for each of the commands in this section by using the **no** form of the command. To disable all debugging, use the privileged EXEC commands **no debug all** or **undebug all**.)

The following **debug** commands are available:

For a complete description of the debug commands, refer to the *Cisco IOS Debug Command Reference*.

Note Effective with Cisco IOS Release 12.4(20)T, the **debug ip inspect** command is replaced by the **debug policy-firewall** command. See the *Cisco IOS Debug Command Reference* for more information.

Generic Debug Commands

You can use the following generic **debug** commands, entered in privileged EXEC mode:

Command	Purpose
`Router# debug ip inspect function-trace`	Displays messages about software functions called by CBAC.
`Router# debug ip inspect object-creation`	Displays messages about software objects being created by CBAC. Object creation corresponds to the beginning of CBAC-inspected sessions.
`Router# debug ip inspect object-deletion`	Displays messages about software objects being deleted by CBAC. Object deletion corresponds to the closing of CBAC-inspected sessions.

Command	Purpose
`Router# debug ip inspect events`	Displays messages about CBAC software events, including information about CBAC packet processing.
`Router# debug ip inspect timers`	Displays messages about CBAC timer events such as when a CBAC idle timeout is reached.
`Router# debug ip inspect detail`	Enables the detailed option, which can be used in combination with other options to get additional information.

Transport Level Debug Commands

You can use the following transport-level **debug** commands, entered in privileged EXEC mode:

Command	Purpose
`Router# debug ip inspect tcp`	Displays messages about CBAC-inspected TCP events, including details about TCP packets.
`Router# debug ip inspect udp`	Displays messages about CBAC-inspected UDP events, including details about UDP packets.

Application Protocol Debug Commands

You can use the following application protocol **debug** command, entered in privileged EXEC mode:

Command	Purpose
`Router# debug ip inspect` *protocol*	Displays messages about CBAC-inspected protocol events, including details about the protocol's packets. Refer to Table 26 to determine the protocol keyword.

Table 26 identifies application protocol keywords for the **debug ip inspect** command.

Table 26 Application Protocol Keywords for the debug ip inspect Command

Application Protocol	Protocol Keyword
CU-SeeMe	**cuseeme**
FTP commands and responses	**ftp-cmd**
FTP token (enables tracing of the FTP tokens parsed)	**ftp-token**
H.323	**h323**
HTTP (Java applets)	**http**
Microsoft NetShow	**netshow**
UNIX R commands (rlogin, rexec, rsh)	**rcmd**
RealAudio	**realaudio**
RPC	**rpc**
SMTP	**smtp**

Table 26 Application Protocol Keywords for the debug ip inspect Command (continued)

Application Protocol	Protocol Keyword
SQL*Net	**sqlnet**
StreamWorks	**streamworks**
TFTP	**tftp**
VDOLive	**vdolive**

Interpreting Syslog and Console Messages Generated by CBAC

CBAC provides syslog messages, console alert messages, and audit trail messages. These messages are useful because they can alert you to network attacks and because they provide an audit trail that provides details about sessions inspected by CBAC. While they are generally referred to as error messages, not all error messages indicate problems with your system.

Audit trail and alert information is configurable on a per-application basis using the CBAC inspection rules.

The following types of messages can be generated by CBAC:

- Denial-of-Service Attack Detection Error Messages
- SMTP Attack Detection Error Messages
- Java Blocking Error Messages
- FTP Error Messages
- Audit Trail Messages

For explanations and recommended actions related to the error messages mentioned in this section, refer to the *Cisco IOS System Error Messages*.

Denial-of-Service Attack Detection Error Messages

CBAC detects and blocks denial-of-service attacks and notifies you when denial-of-service attacks occur. Error messages such as the following may indicate that denial-of-service attacks have occurred:

```
%FW-4-ALERT_ON: getting aggressive, count (550/500) current 1-min rate: 250
%FW-4-ALERT_OFF: calming down, count (0/400) current 1-min rate: 0
```

When `%FW-4-ALERT_ON and %FW-4-ALERT_OFF` error messages appear together, each "aggressive/calming" pair of messages indicates a separate attack. The preceding example shows one separate attack.

Error messages such as the following may indicate that a denial-of-service attack has occurred on a specific TCP host:

```
%FW-4-HOST_TCP_ALERT_ON: Max tcp half-open connections (50) exceeded for host
172.21.127.242.
%FW-4-BLOCK_HOST: Blocking new TCP connections to host 172.21.127.242 for 2 minutes
(half-open count 50 exceeded)
%FW-4-UNBLOCK_HOST: New TCP connections to host 172.21.127.242 no longer blocked
```

SMTP Attack Detection Error Messages

CBAC detects and blocks SMTP attacks (illegal SMTP commands) and notifies you when SMTP attacks occur. Error messages such as the following may indicate that an SMTP attack has occurred:

```
%FW-4-SMTP_INVALID_COMMAND: Invalid SMTP command from initiator (192.168.12.3:52419)
```

CBAC also detects a limited number of SMTP attack signatures. A signature in a SYSLOG message indicates a possible attack against the protected network, such as the detection of illegal SMTP commands in a packet. Whenever a signature is detected, the connection will be reset.

The Cisco IOS Firewall supports the following SMTP attack signatures:

Signature	Description
Mail: bad rcpt	Triggers on any mail message with a “pipe” (|) symbol in the recipient field.
Mail: bad from	Triggers on any mail message with a “pipe” (|) symbol in the “From:” field.
Mail: old attack	Triggers when “wiz” or “debug” commands are sent to the SMTP port.
Mail: decode	Triggers on any mail message with a “:decode@” in the header.
Majordomo	A bug in the Majordomo program will allow remote users to execute arbitrary commands at the privilege level of the server.

The following is a sample SMTP attack signature message:

```
02:04:55: %FW-4-TCP_MAJORDOMO_EXEC_BUG: Sig:3107:Majordomo Execute Attack - from
192.168.25.1 to 192.168.205.1:
```

Java Blocking Error Messages

CBAC detects and selectively blocks Java applets and notifies you when a Java applet has been blocked. Error messages such as the following may indicate that a Java applet has been blocked:

```
%FW-4-HTTP_JAVA_BLOCK: JAVA applet is blocked from (172.21.127.218:80) to
(172.16.57.30:44673).
```

FTP Error Messages

CBAC detects and prevents certain FTP attacks and notifies you when this occurs. Error messages such as the following may appear when CBAC detects these FTP attacks:

```
%FW-3-FTP_PRIV_PORT: Privileged port 1000 used in PORT command -- FTP client 10.0.0.1  FTP
server 10.1.0.1
%FW-3-FTP_SESSION_NOT_AUTHENTICATED: Command issued before the session is authenticated
-- FTP client 10.0.0.1
%FW-3-FTP_NON_MATCHING_IP_ADDR: Non-matching address 172.19.148.154 used in PORT command
-- FTP client 172.19.54.143  FTP server 172.16.127.242
```

Audit Trail Messages

CBAC provides audit trail messages to record details about inspected sessions. Audit trail information is configurable on a per-application basis using the CBAC inspection rules. To determine which protocol was inspected, use the responder's port number. The port number follows the responder's address. The following are sample audit trail messages:

```
%FW-6-SESS_AUDIT_TRAIL: tcp session initiator (192.168.1.13:33192) sent 22 bytes --
responder (192.168.129.11:25) sent 208 bytes
%FW-6-SESS_AUDIT_TRAIL: http session initiator (172.16.57.30:44673) sent 1599 bytes --
responder (172.21.127.218:80) sent 93124 bytes
```

Turning Off CBAC

You can turn off CBAC using the **no ip inspect** global configuration command.

Note The **no ip inspect** command removes all CBAC configuration entries and resets all CBAC global timeouts and thresholds to the defaults. All existing sessions are deleted and their associated access lists removed.

In most situations, turning off CBAC has no negative security impact because CBAC creates "permit" access lists. Without CBAC configured, no "permit" access lists are maintained. Therefore, no derived traffic (returning traffic or traffic from the data channels) can go through the firewall. The exception is SMTP and Java blocking. With CBAC turned off, unacceptable SMTP commands or Java applets may go through the firewall.

CBAC Configuration Examples

The following sections provide CBAC configuration examples:

- Ethernet Interface Configuration Example
- ATM Interface Configuration Example
- Remote Office to ISP Configuration Example
- Remote Office to Branch Office Configuration Example
- Two-Interface Branch Office Configuration Example
- Multiple-Interface Branch Office Configuration Example

The first example develops a CBAC inspection rule for specific protocols and a supporting access control list (ACL). This example focuses how to configure CBAC; it does not provide a complete router configuration and does not describe other elements of the configuration.

The next example develops a CBAC inspection rule for sites that might have remote traffic through an ATM interface. This example further illustrates on how to configure CBAC and emphasizes the application of the configuration rule at the interface, whatever that interface might be. This example does not provide a complete router configuration and does not describe other elements of the configuration.

The remote-office examples also focus on the firewall configuration but do not provide detailed descriptions of other configuration elements, such as the Basic Rate Interface (BRI) and dialer interface configurations.

Other examples provide more complete firewall configurations, further illustrating ways in which to apply CBAC.

In each example, configuring protocol inspection using CBAC has four components:

- Defining an access list with the appropriate permissions.
- Applying the ACL at an interface where you want to control access.
- Defining an inspection rule that includes the protocol that you want to inspect.
- Applying the inspection rule at an interface where you want to inspect traffic.

Ethernet Interface Configuration Example

This example looks at each of these four components. For this example, CBAC is being configured to inspect RTSP and H.323 protocol traffic inbound from the protected network on a router with two Ethernet interfaces. Interface Ethernet1/0 is the protected network and interface Ethernet1/1 is the unprotected network. The security policy for the protected site uses access control lists (ACLs) to restrict inbound traffic on the unprotected interface to specific ICMP protocol traffic, denying inbound access for TCP and UDP protocol traffic. Inbound access for specific protocol traffic is provided through dynamic access lists, which are generated according to CBAC inspection rules.

ACL 100 denies TCP and UDP traffic from any source or destination while permitting specific ICMP protocol traffic. The final deny statement is not required, but is included for explicitness—the final entry in any ACL is an implicit denial of all IP protocol traffic.

```
Router(config)# access-list 100 deny tcp any any
Router(config)# access-list 100 deny udp any any
Router(config)# access-list 100 permit icmp any any echo-reply
Router(config)# access-list 100 permit icmp any any time-exceeded
Router(config)# access-list 100 permit icmp any any packet-too-big
Router(config)# access-list 100 permit icmp any any traceroute
Router(config)# access-list 100 permit icmp any any unreachable
Router(config)# access-list 100 deny ip any any
```

ACL 100 is applied inbound at interface Ethernet1/1 to block all access from the unprotected network to the protected network.

```
Router(config)# interface Ethernet1/1
Router(config-if)# ip access-group 100 in
```

An inspection rule is created for “hqusers” that covers two protocols: RTSP and H.323.

```
Router(config)# ip inspect name hqusers rtsp
Router(config)# ip inspect name hqusers h323
```

The inspection rule is applied inbound at interface Ethernet1/0 to inspect traffic from users on the protected network. When CBAC detects multimedia traffic from the protected network, CBAC creates dynamic entries in access list 100 to allow return traffic for multimedia sessions.

```
Router(config)# interface Ethernet1/0
Router(config-if)# ip inspect hqusers in
```

ATM Interface Configuration Example

In this example, CBAC inspection (firewall protection) is required against inbound traffic on an ATM interface. This example might apply to sites where local hosts require access to hosts or services on a remote network. The security policy for this site uses access control lists (ACLs) to restrict inbound

traffic on the ATM interface to specific ICMP protocol traffic, denying inbound access for TCP and UDP protocol traffic. Inbound access for specific TCP and UDP protocol traffic is provided through dynamic access lists, which are generated according to CBAC inspection rules.

For information on how to select the interface on which to apply CBAC, refer to the "Picking an Interface: Internal or External" section.

Note For Frame Relay or ATM interfaces, you can apply CBAC inspection rules separately on each sub-interface, even though the subinterfaces are physically connected through one interface.

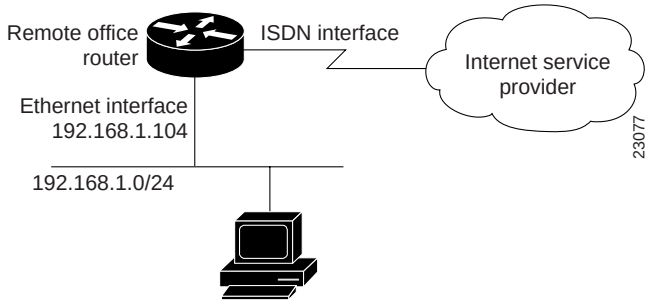
```
! -------------------------
! Create the Inspection Rule
! -------------------------
!
! Create the CBAC inspection rule "test", allowing inspection of the protocol traffic
! specified by the rule. This inspection rule sets the timeout value to 30 seconds for
! each protocol (except for RPC). The timeout value defines the maximum time that a
! connection for a given protocol can remain active without any traffic passing through
! the router. When these timeouts are reached, the dynamic ACLs that are inserted to
! permit the returning traffic are removed, and subsequent packets (possibly even valid
! ones) are not permitted.
ip inspect name test cuseeme timeout 30
ip inspect name test ftp timeout 30
ip inspect name test h323 timeout 30
ip inspect name test realaudio timeout 30
ip inspect name test rpc program-number 100000
ip inspect name test streamworks timeout 30
ip inspect name test vdolive timeout 30
!
! ------------------------------
! Create the Access Control List
! ------------------------------
!
! In this example, ACL 105 denies all TCP and UDP protocol traffic. ICMP traffic from
! subnet 192.168.1.0 is permitted to allow access for routing and control traffic.
! ACL 105 specifies that only the return traffic for protocols defined in the
! inspection rule is allow access through the interface where this rule is applied. The
! final deny statement is added for explicitness.
access-list 105 deny TCP any any
access-list 105 deny UDP any any
access-list 105 permit icmp any any echo-reply
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 time-exceeded
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 packet-too-big
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 traceroute
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 unreachable
access-list 105 deny ip any any
!
! --------------------------------
! Apply the Inspection Rule and ACL
! --------------------------------
!
! In this example, the inspection rule "test" is applied to traffic at interface ATM3/0
! for connections initiated in the outbound direction; that is, from hosts that are
! located on a local network. CBAC creates dynamic access list entries for traffic
! initiated by local hosts. These dynamic entries allow inbound (returning) traffic for
! that connection. ACL 105 is applied at interface ATM3/0 in the inbound direction to
! block traffic initiated from hosts on a remote network that is not part of an
! existing connection.
interface ATM3/0
    ip address 10.1.10.1 255.0.0.0
    ip access-group 105 in
```

```
no ip directed-broadcast
ip inspect test out
no shutdown
atm clock INTERNAL
atm pvc 7 7 7 aal5snap
map-group atm
```

Remote Office to ISP Configuration Example

This example describes one possible Cisco IOS Firewall configuration for a remote office router connected to an Internet service provider (ISP). In this configuration, the site security policy allows hosts on the local network to initiate traffic to the ISP while traffic inbound to the router from the ISP is blocked at the ISDN interface. Specific ICMP control message traffic is permitted through the firewall. No mail or Web services are available from the local network. Figure 23 illustrates this example.

Figure 23 Remote Office to ISP Sample Configuration



The firewall has two interfaces:

- An Ethernet interface connects to the internal protected network.

  Interface Ethernet0 has no ACL applied to it, meaning that all traffic initiated on the LAN is allowed access to the ISP. In this configuration example, Network Address Translation (NAT) is not turned on, and the addresses on interface Ethernet0 are reserved IP addresses. In a production environment, addresses on Ethernet0 either must be registered network addresses, or you must turn on NAT to hide these inside addresses from being visible on the Internet.

- An ISDN Basic Rate Interface (BRI) connects the router to the ISP. In this example, a dialer profile is used to control the BRI interface. This means that the ACL and CBAC inspection rules are applied at the dialer interface, not directly at the physical ISDN (BRI) interface using a dialer map.

```
! -------------------------------------
! General Cisco IOS Firewall Guidelines
! -------------------------------------
! The following global configuration entries illustrate good security practices.
enable secret 5 <elided>
no ip source-route
no cdp run
!
! -------------------------------
! Create the CBAC inspection rule
! -------------------------------
! Create the CBAC inspection rule STOP to allow inspection of the protocol traffic
! specified by the rule.
ip inspect name STOP tcp
ip inspect name STOP ftp
ip inspect name STOP smtp
```

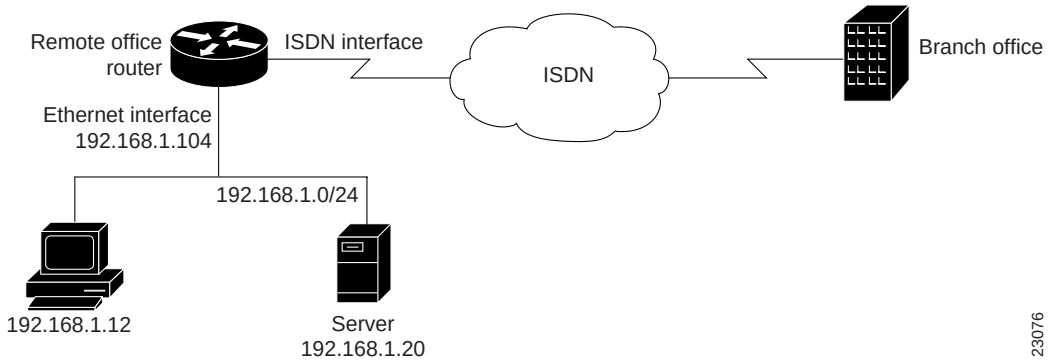
```
ip inspect name STOP h323
ip inspect name STOP rcmd
!
! --------------------------------
! Create Access Control List 105
! --------------------------------
! ACL 105 denies all IP protocol traffic except for specific ICMP control traffic.
! This means that only the return traffic for protocols defined in the
! inspection rule and the specified ICMP traffic is allowed access through the
! interface where this rule is applied.
!
! Deny broadcast messages with a source address of 255.255.255.255; this helps to
! prevent broadcast attacks.
access-list 105 deny ip host 255.255.255.255 any
!
! Add anti-spoofing protection by denying traffic with a source address matching a host
! on the Ethernet interface.
acl 105 deny ip 192.168.1.0 0.0.0.255 any
!
! ICMP traffic is not inspected by CBAC. To control the type of ICMP traffic at the
! interface, add static access list entries. This example has the following ICMP
! requirements: outgoing ping commands require echo-reply messages to come back,
! outgoing traceroute commands require time-exceeded messages to come back, path MTU
! discovery requires “too-big” messages to come back, and incoming traceroute
! messages must be allowed. Additionally, permit all “unreachable” messages to come
! back; that is, if a router cannot forward or deliver a datagram, it sends an ICMP
! unreachable message back to the source and drops the datagram.
access-list 105 permit icmp any any echo-reply
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 time-exceeded
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 packet-too-big
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 traceroute
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 unreachable
!
! Final deny for explicitness. This entry is not required but helps complete the access
! list picture. By default, the final entry in any access list is an implicit deny of
! IP protocol traffic. This ensures that the firewall blocks any traffic not explicitly
! permitted by the access list.
access-list 105 deny ip any any
!
! ------------------------------------------------------------
! Configure the interface
! ------------------------------------------------------------
! In this example, no ACLs or inspection rules are applied at interface Ethernet0,
! meaning that all traffic on the local network is allowed to go out. This assumes a
! high-level of trust for the users on the local network.
interface Ethernet0
    ip address 192.168.1.104 255.255.255.0
!
no ip directed-broadcast
!
! This example uses a dialer profile, so the ACL and CBAC inspection rules are applied
! at the dialer interface, not the physical BRI interface. The dialer pool-member
! command is used to associate the physical interface with a dialer profile.
interface BRI0
    no ip address
    no ip directed-broadcast
    encapsulation ppp
    dialer pool-member 1
    isdn switch-type basic-5ess
!
! ------------------------------------------------------------------
! Create the dialer profile.
! ------------------------------------------------------------------
! Through the dialer profile, the ACL and CBAC inspection rules are
```

```
! applied to every pool member. In this example, the ACL is applied in, meaning that it
! applies to traffic inbound from the ISP. The CBAC inspection rule STOP is applied
! out, meaning that CBAC monitors the traffic through the interface and controls return
! traffic to the router for an existing connection.
interface Dialer0
    ip address negotiated
    ip access-group 105 in
    no ip directed-broadcast
    ip inspect STOP out
    encapsulation ppp
    dialer remote-name <ISP router>
    dialer idle-timeout 500
    dialer string <elided>
    dialer pool 1
    dialer-group 1
    ppp authentication callin
!
! ---------------------------------------------------------
! Additional entries
! ---------------------------------------------------------
! Configure the router to forward packets destined for an unrecognized subnet of
! a directly connected network.
ip classless
! Route traffic to the dialer interface.
ip route 0.0.0.0 0.0.0.0 Dialer0
! Include a dialer list protocol entry to specify the protocol that triggers dialing.
dialer-list 1 protocol ip permit
! Add a user name (name of the router your are configuring) and password for caller
! identification and password authentication with the ISP router.
username <router host name> password 5 <elided>
```

Remote Office to Branch Office Configuration Example

This example describes one possible Cisco IOS Firewall configuration for a remote office router connected to a branch office. In this configuration, the site security policy allows hosts on the local network to initiate traffic to the branch office. Mail or Web services are available from a server on the local network, and access to these services is available from the branch office. Traffic from the branch office, except for mail and Web traffic, is blocked at the outside interface. Specific ICMP control message traffic is permitted through the firewall. Figure 24 illustrates this example.

Figure 24 Remote Office to Branch Office Sample Configuration



The firewall has two interfaces:

- An Ethernet interface connects to the internal protected network.

Interface Ethernet0 has no ACL applied to it, meaning that all traffic initiated from the LAN is allowed access through the firewall.

- An ISDN Basic Rate Interface (BRI) connects the router to the branch office. In this example, a dialer profile is used to control the BRI interface. This means that the ACL and CBAC inspection rules are applied at dialer interface, not directly at the physical ISDN (BRI) interface.

```
! ----------------------------------------
! General firewall configuration guidelines
! ----------------------------------------
! The following global configuration entries illustrate good security practices.
enable secret 5 <elided>
no ip source-route
no cdp run
!
! --------------------------
! Create the Inspection Rule
! --------------------------
! Create the CBAC inspection rule STOP to allow inspection of the specified protocol
! traffic. Create the inspection rule GO to allow inspection of SMTP traffic.
ip inspect name STOP tcp
ip inspect name STOP ftp
ip inspect name STOP smtp
ip inspect name STOP h323
ip inspect name GO smtp
!
! ------------------------------------------------------------
! Create Access Control Lists 106 and 51
! ------------------------------------------------------------
! ACL 106 permits mail and Web traffic from any host to the specified server. ACL 106
! denies all other ip protocol traffic except for specific ICMP control traffic.
! This means that only the return traffic for protocols defined in the
! inspection rule and the specified ICMP traffic is allowed access through the
! interface where this rule is applied.
!
! Deny broadcast messages with a source address of 255.255.255.255; this helps to
! prevent broadcast attacks.
access-list 106 deny ip host 255.255.255.255 any
!
! Add anti-spoofing protection by denying traffic with a source address matching a host
! on the Ethernet interface.
access-list 106 deny ip 192.168.1.0 0.0.0.255 any
!
! ICMP traffic is not inspected by CBAC. To control the type of ICMP traffic at the
! interface, add static access list entries. This example has the following ICMP
! requirements: outgoing ping commands require echo-reply messages to come back,
! outgoing traceroute commands require time-exceeded messages to come back, path MTU
! discovery requires “too-big” messages to come back, and incoming traceroute must be
! allowed. Additionally, permit all “unreachable” messages to come back; that is, if a
! router cannot forward or deliver a datagram, it sends an ICMP unreachable message
! back to the source and drops the datagram.
access-list 106 permit icmp any any echo-reply
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 time-exceeded
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 packet-too-big
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 traceroute
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 unreachable
!
! Permit mail and Web access to a specific server.
access-list 106 permit tcp any host 192.168.1.20 eq smtp
access-list 106 permit tcp any host 192.168.1.20 eq www
!
! Final deny for explicitness. This entry is not required but helps complete the access
! list picture. By default, the final entry in any access list is an implicit deny of
! IP protocol traffic. This ensures that the firewall blocks any traffic not explicitly
```

```
! permitted by the access list.
access-list 106 deny ip any any
!
! ------------------------------------------------------
! Configure the interface.
! ------------------------------------------------------
! In this example, no ACLs or inspection rules are applied at interface Ethernet0,
! meaning that all traffic on the local network is allowed to go out. This assumes a
! high-level of trust for the users on the local network.
interface Ethernet0
    ip address 192.168.1.104 255.255.255.0
    no ip directed-broadcast
!
! This example uses a dialer profile, so the ACL and CBAC inspection rules are applied
! at the dialer interface, not the physical BRI interface. The dialer pool-member
! command is used to associate the physical interface with a dialer profile.
interface BRI0
    no ip address
    no ip directed-broadcast
    encapsulation ppp
    dialer pool-member 1
    isdn switch-type basic-5ess
!
! ---------------------------------------------------------------------
! Apply the ACL and CBAC inspection rules at the dialer interface.
! ---------------------------------------------------------------------
! Through the dialer profile, the ACL and CBAC inspection rules are
! applied to every pool member. In this example, the ACL is applied in, meaning that it
! applies to traffic inbound from the branch office. The CBAC inspection rule STOP is
! applied out, meaning that CBAC monitors the traffic and controls return traffic to
! the router for an existing connection. The CBAC inspection rule GO is applied in,
! protecting against certain types of DoS attacks as described in this document. Note
! that the GO inspection rule does not control return traffic because there is no ACL
! blocking traffic in that direction; however, it does monitor the connections.
interface Dialer0
    ip address <ISDN interface address>
    ip access-group 106 in
    no ip directed-broadcast
    ip inspect STOP out
    ip inspect GO in
    encapsulation ppp
    dialer remote-name <branch office router>
    dialer idle-timeout 500
    dialer string <elided>
    dialer pool 1
    dialer-group 1
    ppp authentication
!
! ------------------------------------------------------
! Additional entries
! ------------------------------------------------------
! Configure the router to forward packets destined for an unrecognized subnet of
! a directly connected network.
ip classless
! Route traffic to the dialer interface.
ip route 0.0.0.0 0.0.0.0 Dialer0
! Include a dialer list protocol entry to specify the protocol that triggers dialing.
dialer-list 1 protocol ip permit
! Add a user name (name of the router your are configuring) and password for caller
! identification and password authentication with the ISP router.
username <router host name> password 5 <elided>
```

Two-Interface Branch Office Configuration Example

This sample configuration file describes a firewall configured with CBAC. The firewall is positioned between a protected field office's internal network and a WAN connection to the corporate headquarters. CBAC is configured on the firewall in order to protect the internal network from potential network threats coming from the WAN side.

The firewall has two interfaces configured:

- Interface Ethernet0 connects to the internal protected network
- Interface Serial0 connects to the WAN with Frame Relay

```
! ------------------------------------------------------------------------
! This first section contains some configuration that is not required for CBAC,
! but illustrates good security practices. Note that there are no
! services on the Ethernet side. Email is picked up via POP from a server on the
! corporate side.
! ------------------------------------------------------------------------
!
hostname user1-examplecorp-fr
!
boot system flash c1600-fw1600-l
enable secret 5 <elided>
!
username user1 password <elided>
ip subnet-zero
no ip source-route
ip domain-name example.com
ip name-server 172.19.2.132
ip name-server 198.92.30.32
!
!
! ------------------------------------------------------------------------
! The next section includes configuration required specifically for CBAC.
! ------------------------------------------------------------------------
!
! The following commands define the inspection rule “myfw”, allowing
! the specified protocols to be inspected. Note that Java applets will be permitted
! according to access list 51, defined later in this configuration.
ip inspect name myfw cuseeme timeout 3600
ip inspect name myfw ftp timeout 3600
ip inspect name myfw http java-list 51 timeout 30
ip inspect name myfw rcmd timeout 3600
ip inspect name myfw realaudio timeout 3600
ip inspect name myfw smtp timeout 3600
ip inspect name myfw tftp timeout 30
ip inspect name myfw udp timeout 15
ip inspect name myfw tcp timeout 3600
!
! The following interface configuration applies the “myfw” inspection rule to
! inbound traffic at Ethernet 0. Since this interface is on the internal network
! side of the firewall, traffic entering Ethernet 0 is actually
! exiting the internal network. Applying the inspection rule to this interface causes
! inbound traffic (which is exiting the network) to be inspected; return traffic will
! only be permitted back through the firewall if part of a session which began from
! within the network.
! Also note that access list 101 is applied to inbound traffic at Ethernet 0.
! (Traffic blocked by the access list will not be inspected.)
interface Ethernet0
description ExampleCorp Ethernet chez user1
ip address 172.19.139.1 255.255.255.248
ip broadcast-address 172.19.131.7
```

```
no ip directed-broadcast
no ip proxy-arp
ip inspect myfw in
ip access-group 101 in
no cdp enable
!
interface Serial0
description Frame Relay (Telco ID 22RTQQ062438-001) to ExampleCorp HQ
no ip address
ip broadcast-address 0.0.0.0
encapsulation frame-relay IETF
no arp frame-relay
bandwidth 56
service-module 56k clock source line
service-module 56k network-type dds
frame-relay lmi-type ansi
!
! Note that the following interface configuration applies access list 111 to
! inbound traffic at the external serial interface. (Inbound traffic is
! entering the network.) When CBAC inspection occurs on traffic exiting the
! network, temporary openings will be added to access list 111 to allow returning
! traffic that is part of existing sessions.
!
interface Serial0.1 point-to-point
ip unnumbered Ethernet0
ip access-group 111 in
bandwidth 56
no cdp enable
frame-relay interface-dlci 16
!
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0.1
!
! The following access list defines “friendly” and “hostile” sites for Java
! applet blocking. Because Java applet blocking is defined in the inspection
! rule “myfw” and references access list 51, applets will be actively denied
! if they are from any of the “deny” addresses and allowed only if they are from
! either of the two “permit” networks.
!
access-list 51 deny   172.19.1.203
access-list 51 deny   172.19.2.147
access-list 51 permit 172.18.0.0 0.1.255.255
access-list 51 permit 192.168.1.0 0.0.0.255
access-list 51 deny   any
!
! The following access list 101 is applied to interface Ethernet 0 above.
! This access list permits all traffic that should be CBAC inspected, and also
! provides anti-spoofing. The access list is deliberately set up to deny unknown
! IP protocols, because no such unknown protocols will be in legitimate use.
!
access-list 101 permit tcp 172.19.139.0 0.0.0.7 any
access-list 101 permit udp 172.19.139.0 0.0.0.7 any
access-list 101 permit icmp 172.19.139.0 0.0.0.7 any
access-list 101 deny   ip any any
!
! The following access list 111 is applied to interface Serial 0.1 above.
! This access list filters traffic coming in from the external side. When
! CBAC inspection occurs, temporary openings will be added to the beginning of
! this access list to allow return traffic back into the internal network.
! This access list should restrict traffic that will be inspected by
! CBAC. (Remember that CBAC will open holes as necessary to permit returning traffic.)
! Comments precede each access list entry. These entries are not all specifically
! related to CBAC, but are created to provide general good security.
!
```

```
! Anti-spoofing.
access-list 111 deny   ip 172.19.139.0 0.0.0.7 any
! Sometimes EIGRP is run on the Frame Relay link. When you use an
! input access list, you have to explicitly allow even control traffic.
! This could be more restrictive, but there would have to be entries
! for the EIGRP multicast as well as for the office's own unicast address.
access-list 111 permit igrp any any
!
! These are the ICMP types actually used...
! administratively-prohibited is useful when you are trying to figure out why
! you cannot reach something you think you should be able to reach.
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 administratively-prohibited
!
! This allows network admins at headquarters to ping hosts at the field office:
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 echo
!
! This allows the field office to do outgoing pings
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 echo-reply
!
! Path MTU discovery requires too-big messages
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 packet-too-big
!
! Outgoing traceroute requires time-exceeded messages to come back
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 time-exceeded
!
! Incoming traceroute
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 traceroute
!
! Permits all unreachables because if you are trying to debug
! things from the remote office, you want to see them. If nobody ever did
! any debugging from the network, it would be more appropriate to permit only
! port unreachables or no unreachables at all.
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 unreachable
!
!
! These next two entries permit users on most ExampleCorp networks to Telnet to
! a host in the field office. This is for remote administration by the network admins.
access-list 111 permit tcp 172.18.0.0 0.1.255.255 host 172.19.139.1 eq telnet
access-list 111 permit tcp 192.168.1.0 0.0.0.255 host 172.19.139.1 eq telnet
!
! Final deny for explicitness
access-list 111 deny ip any any
!
no cdp run
snmp-server community <elided> RO
!
line con 0
exec-timeout 0 0
password <elided>
login local
line vty 0
exec-timeout 0 0
password <elided>
login local
length 35
line vty 1
exec-timeout 0 0
password 7 <elided>
login local
line vty 2
exec-timeout 0 0
password 7 <elided>
login local
line vty 3
```

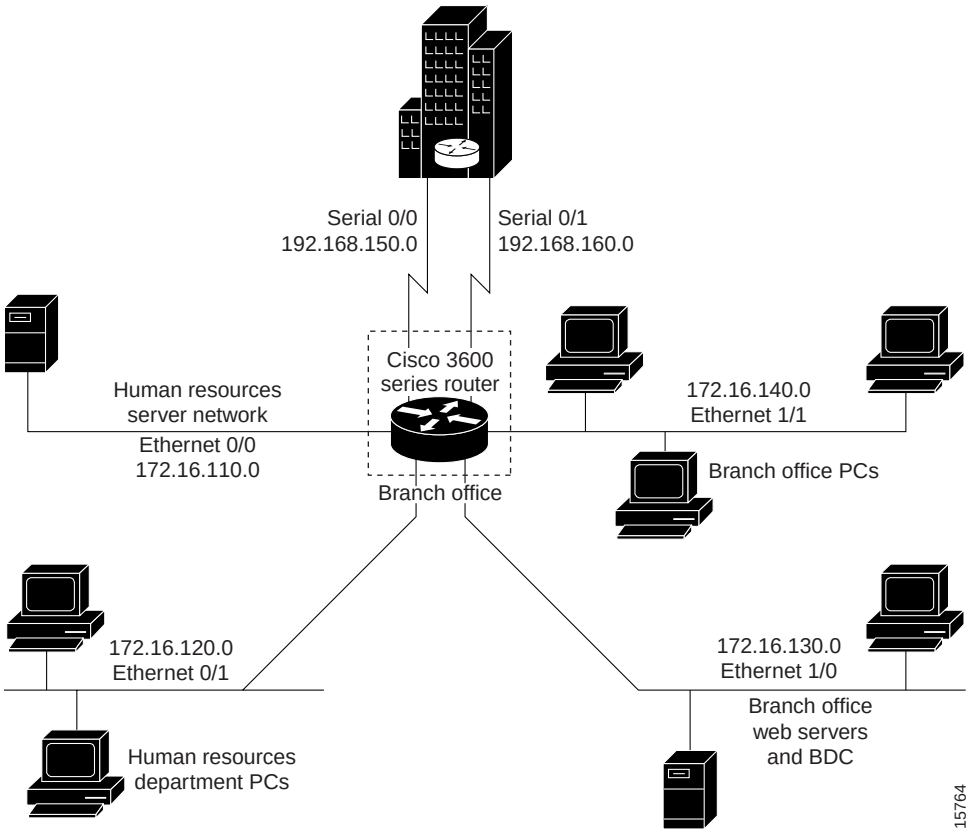
```
exec-timeout 0 0
password 7 <elided>
login local
line vty 4
exec-timeout 0 0
password 7 <elided>
login local
!
scheduler interval 500
end
```

Multiple-Interface Branch Office Configuration Example

In this configuration example, a single Cisco 3600 series firewall router is positioned at a branch office. It has four internal networks and two WAN connections to the corporate headquarters. CBAC is configured on the firewall to protect two of the internal networks from potential network threats coming from the WAN side and from less secure internal networks. Anti-spoofing protection is added at each interface with client systems. Figure 25 illustrates this configuration.

Note This example shows a moderately high level of trust by the administrators toward the expected users. Additional protection could be added to this configuration for a situation in a lower level of trust. That configuration would include ICMP filtering statements, significantly more protocol and address control through the use of more restrictive access control lists, and anti-spoofing applied everywhere. This configuration does not contain those additional restrictions because that would detract from the CBAC example.

Figure 25 Sample Cisco IOS Firewall Application Environment



The branch office has this sample network configuration:

- Ethernet interface 0/0 supports the Human Resources department servers. This network includes an email (SMTP and POP3) host and a Windows NT server. The Windows NT server is the Primary Domain Controller (PDC) for the Human Resources domain and has a trust relationship with the rest of the company; however, it contains applications and databases that must not be accessed by the rest of the company or the other groups in the branch office. The devices on this LAN are accessible only by users in the Human Resources department on Ethernet interface 0/1. The Mail server must be able to send and receive email (through SMTP sessions) with all other devices. The Windows 95 machines can use this machine as their email server (for sending email through SMTP sessions) and as a repository for accumulating email that they can then download through POP3 sessions. No one else in the company is allowed to form POP3 sessions to any machine on this LAN.
- Ethernet interface 0/1 supports the Windows 95 computers in the Human Resources department. These users must have access to the Human Resources mail servers located on Ethernet interface 0/0 as well as access to the rest of the company. Access to the Windows NT server resources are controlled through the Windows NT permissions assigned to each user in the Windows NT domain.
- Ethernet interface 1/0 supports the branch office web servers, which can be accessed by everyone in the company. These servers use TCP ports 80 (HTTP) and 443 (SHTTP) for inbound Web access. This network also includes a backup domain controller (BDC) for the overall domain that is also used as file, print, and service server.

Ethernet interface 1/1 supports all users who are not in the Human Resources department. These users have no access to the Human Resources department servers, but they can access the other network interfaces and the serial interfaces for WAN connectivity. Serial interface 0/0 and 0/1 connect to the WAN with T1 links (links to corporate headquarters). In this sample configuration, the Domain Name System (DNS) servers are located somewhere within the rest of the company.

Additionally, network management (SNMP) and Telnet sessions are limited to the management network (192.168.55.0), which is located somewhere within the rest of the company across the serial interface.

```
! --------------------------------------------------------------------
! This first section contains some configuration that is not required
! for CBAC, but illustrates good security practices.
! --------------------------------------------------------------------
! Add this line to get timestamps on the syslog messages.
service timestamps log datetime localtime show-timezone
!
hostname Router1
!
boot system flash c3600-fw3600-l
!
! Configure AAA user authentication.
aaa new-model
aaa authentication login lista group tacacs+ enable
!
enable secret 5 <elided>
ip subnet-zero
!
! Disable source routing to help prevent spoofing.
no ip source-route
!
! Set up the domain name and server IP addresses.
ip domain-name example.com
ip name-server 192.168.55.132
ip name-server 192.168.27.32
!
! The audit-trail command enables the delivery of specific CBAC messages
! through the syslog notification process.
ip inspect audit-trail
!
! Establish the time-out values for DNS queries. When this idle-timer expires,
! the dynamic ACL entries that were created to permit the reply to a DNS request
! will be removed and any subsequent packets will be denied.
ip inspect dns-timeout 10
!
! ------------------------------------------------------------------------------
! The next section includes configuration statements required specifically for CBAC.
! ------------------------------------------------------------------------------
! Define the CBAC inspection rule “inspect1”, allowing the specified protocols to be
! inspected. The first rule enables SMTP specific inspection. SMTP inspection causes
! the exchange of the SMTP session to be inspected for illegal commands. Any packets
! with illegal commands are dropped, and the SMTP session will hang and eventually
! time out.
ip inspect name inspect1 smtp timeout 30
!
! In the next two lines of inspect1, define the maximum time that each of the UDP and
! TCP sessions are allowed to continue without any traffic passing
! through the router. When these timeouts are reached, the dynamic ACLs that
! are inserted to permit the returning traffic are removed and subsequent packets
! (possibly even valid ones) will not be permitted.
ip inspect name inspect1 udp timeout 30
ip inspect name inspect1 tcp timeout 30
!
```

```
! Define the CBAC inspection rule “inspect2”, allowing the specified protocols to be
! inspected. These rules are similar to those used in the inspection rule “inspect1,”
! except that on the interfaces where this rule is applied, SMTP sessions are not
! expected to go through; therefore, the SMTP rule element is not applied here.
ip inspect name inspect2 udp timeout 30
ip inspect name inspect2 tcp timeout 30
!
! ------------------------------------------------------------------------
! The next section shows the Ethernet interface configuration statements for each
! interface, including access lists and inspections rules.
! ------------------------------------------------------------------------
! Apply the “inspect1” inspection rule to sessions that are initiated in the outbound
! direction (toward the LAN) at Ethernet interface 0/0. All packets in these sessions
! will be inspected by CBAC. Provided that network traffic passes the Access Control
! List (ACL) restrictions, traffic is then inspected by CBAC for access through the
! Cisco Secure Integrated Software. Traffic blocked by the access list is not inspected
! by CBAC. Access list 110 is applied to outbound traffic on this interface.
interface Ethernet0/0
    description HR_Server Ethernet
    ip address 172.16.110.1 255.255.255.0
    ip access-group 110 out
    no ip directed-broadcast
    no ip proxy-arp
    ip inspect inspect1 out
    no cdp enable
!
! Apply access list 120 to inbound traffic on Ethernet interface 0/1.
! Applying access list 120 to inbound traffic provides anti-spoofing on this interface
! by dropping traffic with a source address matching the IP address on a network other
! than Ethernet 0/1. The IP helper address lists the IP address of the DHCP server on
! Ethernet interface 1/0.
interface Ethernet0/1
    description HR_client Ethernet
    ip address 172.16.120.1 255.255.255.0
    ip access-group 120 in
    ip helper-address 172.16.130.66
    no ip directed-broadcast
    no ip proxy-arp
    no cdp enable
!
! Apply the “inspect2” inspection rule to sessions that are initiated in the outbound
! direction (toward the LAN) at Ethernet interface 1/0. Provided that network traffic
! passes the Access Control List (ACL) restrictions, traffic is then inspected by CBAC
! through the Cisco Secure Integrated Software. Traffic blocked by the access list is
! not inspected
! by CBAC. Access list 130 is applied to outbound traffic on this interface.
interface Ethernet1/0
    description Web_server Ethernet
    ip address 172.16.130.1 255.255.255.0
    ip access-group 130 out
    no ip directed-broadcast
    no ip proxy-arp
    ip inspect inspect2 out
    no cdp enable
!
! Apply access list 140 to inbound traffic at Ethernet interface 1/1. This
! provides anti-spoofing on the interface by dropping traffic with a source address
! matching the IP address of a network other than Ethernet 1/1. The IP helper address
! lists the IP address of the DHCP server on Ethernet interface 1/0.
interface Ethernet1/1
    description Everyone_else Ethernet
    ip address 172.16.140.1 255.255.255.0
    ip access-group 140 in
    ip helper-address 172.16.130.66
```

```
    no ip directed-broadcast

    no ip proxy-arp
    no cdp enable
!
! -------------------------------------------------------------------------
! The next section configures the serial interfaces, including access lists.
! -------------------------------------------------------------------------
! Apply access list 150 to Serial interfaces 0/0. This provides anti-spoofing on the
! serial interface by dropping traffic with a source address matching the IP address
! of a host on Ethernet interface 0/0, 0/1, 1/0, or 1/1.
interface Serial0/0
    description T1 to HQ
    ip address 192.168.150.1 255.255.255.0
    ip access-group 150 in
    bandwidth 1544
!
interface Serial1/1
    description T1 to HQ
    ip address 192.168.160.1 255.255.255.0
    ip access-group 150 in
    bandwidth 1544
!
! ------------------------------
! Configure routing information.
! ------------------------------
router igrp 109
network 172.16.0.0
network 192.168.150.0
network 192.168.160.0
!
! Define protocol forwarding on the firewall. When you turn on a related command,
! ip helper-address, you forward every IP broadcast in the ip forward protocol
! command list, including several which are on by default: TFTP (port 69),
! DNS (port 53), Time service (port 37), NetBIOS Name Server (port 137),
! NetBIOS Datagram Server (port 138), BOOTP client and server datagrams
! (ports 67 and 68), and TACACS service (port 49). One common
! application that requires helper addresses is Dynamic Host Configuration
! Protocol (DHCP). DHCP information is carried inside of BOOTP packets. The
! “no ip forward protocol” statements turn off forwarding for the specified protocols.
no ip forward-protocol udp netbios-ns
no ip forward-protocol udp netbios-dgm
no ip forward-protocol udp tacacs
no ip forward-protocol udp tftp
ip forward-protocol udp bootpc
!
! Add this line to establish where router SYSLOG messages are sent. This includes the
! CBAC messages.
logging 192.168.55.131
!
! ---------------------------------------------------------------
! Define the configuration of each access list.
! ---------------------------------------------------------------
! Defines Telnet controls in access list 12.
access-list 12 permit 192.168.55.0 0.0.0.255
!
! Defines SNMP controls in access list 13.
access-list 13 permit 192.168.55.12
access-list 13 permit 192.168.55.19
!
! Access list 110 permits TCP and UDP protocol traffic for specific ports and with a
! source address on Ethernet interface 0/1. The access list denies IP protocol traffic
! with any other source and destination address. The access list permits ICMP access
! for any source and destination address. Access list 110 is deliberately set up to
```

```
! deny unknown IP protocols because no such unknown protocols will be in legitimate
! use. Access list 110 is applied to outbound traffic at Ethernet interface 0/0. In ACL
! 110, network traffic is being allowed access to the ports on any server on the HR
! server network. In less trusted environments, this can be a security problem;
! however, you can limit access more severely by specifying specific destination
! addresses in the ACL statements.
access-list 110 permit tcp 172.16.120.0 0.0.0.255 any eq smtp
access-list 110 permit tcp 172.16.120.0 0.0.0.255 any eq pop3
access-list 110 permit tcp 172.16.120.0 0.0.0.255 any eq 110
access-list 110 permit udp any any eq 137
access-list 110 permit udp any any eq 138
access-list 110 permit udp any any eq 139
access-list 110 permit icmp any any
access-list 110 deny ip any any!
!
! Access-list 120 permits TCP, UDP, and ICMP protocol traffic with a source address
! on Ethernet interface 0/1, but denies all other IP protocol traffic. Access list
! 120 is applied to inbound traffic on Ethernet interface 0/1.
access-list 120 permit tcp 172.16.120.0 0.0.0.255 any
access-list 120 permit udp 172.16.120.0 0.0.0.255 any
access-list 120 permit icmp 172.16.120.0 0.0.0.255 any
access-list 120 deny ip any any
!
! Access list 130 permits TCP, UDP, and ICMP protocol traffic for specific ports and
! with any source and destination address. It opens access to the web server and to
! all NBT services to the rest of the company, which can be controlled through the
! trust relations on the Windows NT servers. The bootpc entry permits access to the
! DHCP server. Access list 130 denies all other IP protocol traffic. Access list 130 is
! applied to outbound traffic at Ethernet interface 1/0.
access-list 130 permit tcp any any eq www
access-list 130 permit tcp any any eq 443
access-list 130 permit tcp any any eq 110
access-list 130 permit udp any any eq 137
access-list 130 permit udp any any eq 138
access-list 130 permit udp any any eq 139
access-list 130 permit udp any any eq bootpc
access-list 130 permit icmp any any
access-list 130 deny ip any any
!
! Access list 140 permits TCP, UDP, and ICMP protocol traffic with a source address on
! Ethernet interface 1/1, and it denies all other IP protocol traffic. Access list 140
! is applied to inbound traffic at Ethernet interface 1/1.
access-list 140 permit tcp 172.16.140.0 0.0.0.255 any
access-list 140 permit udp 172.16.140.0 0.0.0.255 any
access-list 140 permit icmp 172.16.140.0 0.0.0.255 any
access-list 140 deny ip any any
!
! Access list 150 denies IP protocol traffic with a source address on Ethernet
! interfaces 0/0, 0/1, 1/0, and 1/1, and it permits IP protocol traffic with any other
! source and destination address. Access list 150 is applied to inbound traffic
! on each of the serial interfaces.
access-list 150 deny ip 172.16.110.0 0.0.0.255 any
access-list 150 deny ip 172.16.120.0 0.0.0.255 any
access-list 150 deny ip 172.16.130.0 0.0.0.255 any
access-list 150 deny ip 172.16.140.0 0.0.0.255 any
access-list 150 permit ip any any
!
! Disable Cisco Discovery Protocol.
no cdp run
!
snmp-server community <elided> ro 13
tacacs-server host 192.168.55.2
tacacs-server key <elided>
!
```

```
! --------------------------------------------------------------------------------
! Configures the router console port and the virtual terminal line interfaces,
! including AAA authentication at login. Authentication is required for users defined
! in “lista.” Access-class 12 is applied on each line, restricting Telnet access to
! connections with a source address on the network management network.
! --------------------------------------------------------------------------------
line console 0
exec-timeout 3 00
login authentication lista
line aux 0
exec-timeout 3 00
login authentication lista
line vty 0
    exec-timeout 1 30
    login authentication lista
    access-class 12 in
line vty 1
    exec-timeout 1 30
    login authentication lista
    access-class 12 in
line vty 2
    exec-timeout 1 30
    login authentication lista
    access-class 12 in
line vty 3
    exec-timeout 1 30
    login authentication lista
    access-class 12 in
line vty 4
    exec-timeout 1 30
    login authentication lista
    access-class 12 in
!
end
```

Cisco IOS Firewall

Configuring Cisco IOS Firewall Intrusion Detection System

This chapter describes the Cisco IOS Firewall Intrusion Detection System (IDS) feature. Intrusion detection systems provide a level of protection beyond the firewall by protecting the network from internal and external attacks and threats. Cisco IOS Firewall IDS technology enhances perimeter firewall protection by taking appropriate action on packets and flows that violate the security policy or represent malicious network activity.

For a complete description of the Cisco IOS Firewall IDS commands in this chapter, refer to the "Cisco IOS Firewall IDS Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the chapter "Identifying Supported Platforms" section in the "Using Cisco IOS Software."

In This Chapter

This chapter has the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

About the Firewall Intrusion Detection System

The Cisco IOS Firewall IDS feature supports intrusion detection technology for midrange and high-end router platforms with firewall support. It is ideal for any network perimeter, and especially for locations in which a router is being deployed and additional security between network segments is required. It also can protect intranet and extranet connections where additional security is mandated, and branch-office sites connecting to the corporate office or Internet.

The Cisco IOS Firewall IDS feature identifies 59 of the most common attacks using "signatures" to detect patterns of misuse in network traffic. The intrusion-detection signatures included in the Cisco IOS Firewall were chosen from a broad cross-section of intrusion-detection signatures. The signatures represent severe breaches of security and the most common network attacks and information-gathering scans. For a description of Cisco IOS Firewall IDS signatures, refer to the "Cisco IOS Firewall IDS Signature List" section.

The Cisco IOS Firewall IDS acts as an in-line intrusion detection sensor, watching packets and sessions as they flow through the router, scanning each to match any of the IDS signatures. When it detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog or the Cisco Secure Intrusion Detection System (Cisco Secure IDS, formerly known as NetRanger) Post Office Protocol. The network administrator can configure the IDS system to choose the appropriate response to various threats. When packets in a session match a signature, the IDS system can be configured to take these actions:

- Send an alarm to a syslog server or a Cisco Secure IDS Director (centralized management interface)
- Drop the packet
- Reset the TCP connection

Cisco developed its Cisco IOS software-based intrusion-detection capabilities in Cisco IOS Firewall with flexibility in mind, so that individual signatures could be disabled in case of false positives. Also, while it is preferable to enable both the firewall and intrusion detection features of the CBAC security engine to support a network security policy, each of these features may be enabled independently and on different router interfaces. Cisco IOS software-based intrusion detection is part of the Cisco IOS Firewall.

This section has the following sections:

Interaction with Cisco IOS Firewall Default Parameters

When Cisco IOS IDS is enabled, Cisco IOS Firewall is automatically enabled. Thus, IDS uses Cisco IOS Firewall default parameter values to inspect incoming sessions. Default parameter values include the following:

- The rate at which IDS starts deleting half-open sessions (modified via the **ip inspect one-minute high** command)

- The rate at which IDS stops deleting half-open sessions (modified via the **ip inspect one-minute low** command)
- The maximum incomplete sessions (modified via the **ip inspect max-incomplete high** and the **ip inspect max-incomplete low** commands)

After the incoming TCP session setup rate crosses the one-minute high water mark, the router will reset the oldest half-open session, which is the default behavior of the Cisco IOS Firewall. Cisco IOS IDS cannot modify this default behavior. Thus, after a new TCP session rate crosses the one-minute high water mark and a router attempts to open new connections by sending SYN packets at the same time, the latest SYN packet will cause the router to reset the half-open session that was opened by the earlier SYN packet. Only the last SYN request will survive.

Compatibility with Cisco Secure Intrusion Detection

Cisco IOS Firewall is compatible with the Cisco Secure Intrusion Detection System (formally known as NetRanger). The Cisco Secure IDS is an enterprise-scale, real-time, intrusion detection system designed to detect, report, and terminate unauthorized activity throughout a network.

The Cisco Secure IDS consists of three components:

- Sensor
- Director
- Post Office

Cisco Secure IDS Sensors, which are high-speed network appliances, analyze the content and context of individual packets to determine if traffic is authorized. If a network's data stream exhibits unauthorized or suspicious activity, such as a SATAN attack, a ping sweep, or the transmission of a secret research project code word, Cisco Secure IDS Sensors can detect the policy violation in real time, forward alarms to a Cisco Secure IDS Director management console, and remove the offender from the network.

The Cisco Secure IDS Director is a high-performance, software-based management system that centrally monitors the activity of multiple Cisco Secure IDS Sensors located on local or remote network segments.

The Cisco Secure IDS Post Office is the communication backbone that allows Cisco Secure IDS services and hosts to communicate with each other. All communication is supported by a proprietary, connection-based protocol that can switch between alternate routes to maintain point-to-point connections.

Cisco Secure IDS customers can deploy the Cisco IOS Firewall IDS signatures to complement their existing IDS systems. This allows an IDS to be deployed to areas that may not be capable of supporting a Cisco Secure IDS Sensor. Cisco IOS Firewall IDS signatures can be deployed alongside or independently of other Cisco IOS Firewall features.

The Cisco IOS Firewall IDS can be added to the Cisco Secure IDS Director screen as an icon to provide a consistent view of all intrusion detection sensors throughout a network. The Cisco IOS Firewall intrusion detection capabilities have an enhanced reporting mechanism that permits logging to the Cisco Secure IDS Director console in addition to Cisco IOS syslog.

For additional information about Cisco Secure IDS (NetRanger), refer to the *NetRanger User Guide*.

Functional Description

The Cisco IOS Firewall IDS acts as an in-line intrusion detection sensor, watching packets as they traverse the router's interfaces and acting upon them in a definable fashion. When a packet, or a number of packets in a session, match a signature, the Cisco IOS Firewall IDS may perform the following configurable actions:

- Alarm—Sends an alarm to a syslog server or Cisco Secure IDS Director
- Drop—Drops the packet
- Reset—Resets the TCP connection

The following describes the packet auditing process with Cisco IOS Firewall IDS:

- You create an audit rule, which specifies the signatures that should be applied to packet traffic and the actions to take when a match is found. An audit rule can apply informational and attack signatures to network packets. The signature list can have just one signature, all signatures, or any number of signatures in between. Signatures can be disabled in case of false positives or the needs of the network environment.
- You apply the audit rule to an interface on the router, specifying a traffic direction (*in* or *out*).
- If the audit rule is applied to the *in* direction of the interface, packets passing through the interface are audited before the inbound ACL has a chance to discard them. This allows an administrator to be alerted if an attack or information-gathering activity is underway even if the router would normally reject the activity.
- If the audit rule is applied to the *out* direction on the interface, packets are audited after they enter the router through another interface. In this case, the inbound ACL of the other interface may discard packets before they are audited. This may result in the loss of Cisco IOS Firewall IDS alarms even though the attack or information-gathering activity was thwarted.
- Packets going through the interface that match the audit rule are audited by a series of modules, starting with IP; then either ICMP, TCP, or UDP (as appropriate); and finally, the Application level.
- If a signature match is found in a module, then the following user-configured action(s) occur:
  - If the action is **alarm**, then the module completes its audit, sends an alarm, and passes the packet to the next module.
  - If the action is **drop**, then the packet is dropped from the module, discarded, and not sent to the next module.
  - If the action is **reset**, then the packets are forwarded to the next module, and packets with the reset flag set are sent to both participants of the session, if the session is TCP.

Note It is recommended that you use the **drop** and **reset** actions together.

If there are multiple signature matches in a module, only the first match fires an action. Additional matches in other modules fire additional alarms, but only one per module.

Note This process is different than on the Cisco Secure IDS Sensor appliance, which identifies all signature matches for each packet.

When to Use Cisco IOS Firewall IDS

Cisco IOS Firewall IDS capabilities are ideal for providing additional visibility at intranet, extranet, and branch-office Internet perimeters. Network administrators enjoy more robust protection against attacks on the network and can automatically respond to threats from internal or external hosts.

The Cisco IOS Firewall with intrusion detection is intended to satisfy the security goals of all of our customers, and is particularly appropriate for the following scenarios:

- Enterprise customers that are interested in a cost-effective method of extending their perimeter security across all network boundaries, specifically branch-office, intranet, and extranet perimeters.
- Small and medium-sized businesses that are looking for a cost-effective router that has an integrated firewall with intrusion-detection capabilities.
- Service provider customers that want to set up managed services, providing firewalling and intrusion detection to their customers, all housed within the necessary function of a router.

Memory and Performance Impact

The performance impact of intrusion detection will depend on the configuration of the signatures, the level of traffic on the router, the router platform, and other individual features enabled on the router such as encryption, source route bridging, and so on. Enabling or disabling individual signatures will not alter performance significantly, however, signatures that are configured to use Access Control Lists will have a significant performance impact.

Because this router is being used as a security device, no packet will be allowed to bypass the security mechanisms. The IDS process in the Cisco IOS Firewall router sits directly in the packet path and thus will search each packet for signature matches. In some cases, the entire packet will need to be searched, and state information and even application state and awareness must be maintained by the router.

For auditing atomic signatures, there is no traffic-dependent memory requirement. For auditing compound signatures, CBAC allocates memory to maintain the state of each session for each connection. Memory is also allocated for the configuration database and for internal caching.

Cisco IOS Firewall IDS Signature List

The following is a complete list of Cisco IOS Firewall IDS signatures. A signature detects patterns of misuse in network traffic. In Cisco IOS Firewall IDS, signatures are categorized into four types:

- Info Atomic
- Info Compound
- Attack Atomic
- Attack Compound

An info signature detects information-gathering activity, such as a port sweep.

An attack signature detects attacks attempted into the protected network, such as denial-of-service attempts or the execution of illegal commands during an FTP session.

Info and attack signatures can be either atomic or compound signatures. Atomic signatures can detect patterns as simple as an attempt to access a specific port on a specific host. Compound signatures can detect complex patterns, such as a sequence of operations distributed across multiple hosts over an arbitrary period of time.

The intrusion-detection signatures included in the Cisco IOS Firewall were chosen from a broad cross-section of intrusion-detection signatures as representative of the most common network attacks and information-gathering scans that are not commonly found in an operational network.

The following signatures are listed in numerical order by their signature number in the Cisco Secure IDS Network Security Database. After each signature's name is an indication of the type of signature (info or attack, atomic or compound).

Note Atomic signatures marked with an asterisk (Atomic*) are allocated memory for session states by CBAC.

1000 IP options-Bad Option List (Info, Atomic)

Triggers on receipt of an IP datagram where the list of IP options in the IP datagram header is incomplete or malformed. The IP options list contains one or more options that perform various network management or debugging tasks.

1001 IP options-Record Packet Route (Info, Atomic)

Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 7 (Record Packet Route).

1002 IP options-Timestamp (Info, Atomic)

Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 4 (Timestamp).

1003 IP options-Provide s,c,h,tcc (Info, Atomic)

Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 2 (Security options).

1004 IP options-Loose Source Route (Info, Atomic)

Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 3 (Loose Source Route).

1005 IP options-SATNET ID (Info, Atomic)

Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 8 (SATNET stream identifier).

1006 IP options-Strict Source Route (Info, Atomic)

Triggers on receipt of an IP datagram in which the IP option list for the datagram includes option 2 (Strict Source Routing).

1100 IP Fragment Attack (Attack, Atomic)

Triggers when any IP datagram is received with the "more fragments" flag set to 1 or if there is an offset indicated in the offset field.

1101 Unknown IP Protocol (Attack, Atomic)

Triggers when an IP datagram is received with the protocol field set to 101 or greater. These protocol types are undefined or reserved and should not be used.

1102 Impossible IP Packet (Attack, Atomic)

This triggers when an IP packet arrives with source equal to destination address. This signature will catch the so-called Land Attack.

2000 ICMP Echo Reply (Info, Atomic)

Triggers when a IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 0 (Echo Reply).

2001 ICMP Host Unreachable (Info, Atomic)

Triggers when an IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 3 (Host Unreachable).

2002 ICMP Source Quench (Info, Atomic)

Triggers when an IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 4 (Source Quench).

2003 ICMP Redirect (Info, Atomic)

Triggers when an IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 5 (Redirect).

2004 ICMP Echo Request (Info, Atomic)

Triggers when an IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 8 (Echo Request).

2005 ICMP Time Exceeded for a Datagram (Info, Atomic)

Triggers when an IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 11(Time Exceeded for a Datagram).

2006 ICMP Parameter Problem on Datagram (Info, Atomic)

Triggers when an IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 12 (Parameter Problem on Datagram).

2007 ICMP Timestamp Request (Info, Atomic)

Triggers when an IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 13 (Timestamp Request).

2008 ICMP Timestamp Reply (Info, Atomic)

Triggers when an IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 14 (Timestamp Reply).

2009 ICMP Information Request (Info, Atomic)

Triggers when an IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 15 (Information Request).

2010 ICMP Information Reply (Info, Atomic)

Triggers when an IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 16 (ICMP Information Reply).

2011 ICMP Address Mask Request (Info, Atomic)

Triggers when an IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 17 (Address Mask Request).

2012 ICMP Address Mask Reply (Info, Atomic)

Triggers when an IP datagram is received with the “protocol” field in the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 18 (Address Mask Reply).

2150 Fragmented ICMP Traffic (Attack, Atomic)

Triggers when an IP datagram is received with the protocol field in the IP header set to 1 (ICMP) and either the more fragments flag is set to 1 (ICMP) or there is an offset indicated in the offset field.

2151 Large ICMP Traffic (Attack, Atomic)

Triggers when an IP datagram is received with the protocol field in the IP header set to 1 (ICMP) and the IP length is greater than 1024.

2154 Ping of Death Attack (Attack, Atomic)

Triggers when an IP datagram is received with the protocol field in the IP header set to 1 (ICMP), the Last Fragment bit is set, and

```
( IP offset * 8 ) + (IP data length)  > 65535
```

In other words, the IP offset (which represents the starting position of this fragment in the original packet, and which is in 8-byte units) plus the rest of the packet is greater than the maximum size for an IP packet.

3040 TCP - no bits set in flags (Attack, Atomic)

Triggers when a TCP packet is received with no bits set in the flags field.

3041 TCP - SYN and FIN bits set (Attack, Atomic)

Triggers when a TCP packet is received with both the SYN and FIN bits set in the flag field.

3042 TCP - FIN bit with no ACK bit in flags (Attack, Atomic)

Triggers when a TCP packet is received with the FIN bit set but with no ACK bit set in the flags field.

3050 Half-open SYN Attack/SYN Flood (Attack, Compound)

Triggers when multiple TCP sessions have been improperly initiated on any of several well-known service ports. Detection of this signature is currently limited to FTP, Telnet, HTTP, and e-mail servers (TCP ports 21, 23, 80, and 25 respectively).

3100 Smail Attack (Attack, Compound)

Triggers on the very common “smail” attack against SMTP-compliant e-mail servers (frequently sendmail).

3101 Sendmail Invalid Recipient (Attack, Compound)

Triggers on any mail message with a “pipe” (|) symbol in the recipient field.

3102 Sendmail Invalid Sender (Attack, Compound)

Triggers on any mail message with a “pipe” (|) symbol in the “From:” field.

3103 Sendmail Reconnaissance (Attack, Compound)

Triggers when “expn” or “vrfy” commands are issued to the SMTP port.

3104 Archaic Sendmail Attacks (Attack, Compound)

Triggers when “wiz” or “debug” commands are issued to the SMTP port.

3105 Sendmail Decode Alias (Attack, Compound)

Triggers on any mail message with “: decode@” in the header.

3106 Mail Spam (Attack, Compound)

Counts number of Rcpt to: lines in a single mail message and alarms after a user-definable maximum has been exceeded (default is 250).

3107 Majordomo Execute Attack (Attack, Compound)

A bug in the Majordomo program will allow remote users to execute arbitrary commands at the privilege level of the server.

3150 FTP Remote Command Execution (Attack, Compound)

Triggers when someone tries to execute the FTP SITE command.

3151 FTP SYST Command Attempt (Info, Compound)

Triggers when someone tries to execute the FTP SYST command.

3152 FTP CWD ~root (Attack, Compound)

Triggers when someone tries to execute the CWD ~root command.

3153 FTP Improper Address Specified (Attack, Atomic*)

Triggers if a port command is issued with an address that is not the same as the requesting host.

3154 FTP Improper Port Specified (Attack, Atomic*)

Triggers if a port command is issued with a data port specified that is less than 1024 or greater than 65535.

4050 UDP Bomb (Attack, Atomic)

Triggers when the UDP length specified is less than the IP length specified.

4100 Tftp Passwd File (Attack, Compound)

Triggers on an attempt to access the passwd file (typically /etc/passwd) via TFTP.

6100 RPC Port Registration (Info, Atomic*)

Triggers when attempts are made to register new RPC services on a target host.

6101 RPC Port Unregistration (Info, Atomic*)

Triggers when attempts are made to unregister existing RPC services on a target host.

6102 RPC Dump (Info, Atomic*)

Triggers when an RPC dump request is issued to a target host.

6103 Proxied RPC Request (Attack, Atomic*)

Triggers when a proxied RPC request is sent to the portmapper of a target host.

6150 ypserv Portmap Request (Info, Atomic*)

Triggers when a request is made to the portmapper for the YP server daemon (ypserv) port.

6151 ypbind Portmap Request (Info, Atomic*)

Triggers when a request is made to the portmapper for the YP bind daemon (ypbind) port.

6152 yppasswdd Portmap Request (Info, Atomic*)

Triggers when a request is made to the portmapper for the YP password daemon (yppasswdd) port.

6153 ypupdated Portmap Request (Info, Atomic*)

Triggers when a request is made to the portmapper for the YP update daemon (ypupdated) port.

6154 ypxfrd Portmap Request (Info, Atomic*)

Triggers when a request is made to the portmapper for the YP transfer daemon (ypxfrd) port.

6155 mountd Portmap Request (Info, Atomic*)

Triggers when a request is made to the portmapper for the mount daemon (mountd) port.

6175 rexd Portmap Request (Info, Atomic*)

Triggers when a request is made to the portmapper for the remote execution daemon (rexd) port.

6180 rexd Attempt (Info, Atomic*)

Triggers when a call to the rexd program is made. The remote execution daemon is the server responsible for remote program execution. This may be indicative of an attempt to gain unauthorized access to system resources.

6190 statd Buffer Overflow (Attack, Atomic*)

Triggers when a large statd request is sent. This could be an attempt to overflow a buffer and gain access to system resources.

8000 FTP Retrieve Password File (Attack, Atomic*)

SubSig ID: 2101

Triggers on string “passwd” issued during an FTP session. May indicate someone attempting to retrieve the password file from a machine in order to crack it and gain unauthorized access to system resources.

Cisco IOS Firewall IDS Configuration Task List

See the following sections for configuration tasks for the Cisco IOS Firewall Intrusion Detection System feature. Each task in the list is identified as optional or required:

- Initializing Cisco IOS Firewall IDS (Required)
- Initializing the Post Office (Required)
- Configuring and Applying Audit Rules (Required)

- Verifying the Configuration (Optional)

For examples using the commands in this chapter, see the "Cisco IOS Firewall IDS Configuration Examples" section at the end of this chapter.

Initializing Cisco IOS Firewall IDS

To initialize Cisco IOS Firewall IDS on a router, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip audit smtp spam recipients`	Sets the threshold beyond which spamming in e-mail messages is suspected. Here, *recipients* is the maximum number of recipients in an e-mail message. The default is 250.
Step 2	`Router(config)# ip audit po max-events number_events`	Sets the threshold beyond which queued events are dropped from the queue for sending to the Cisco Secure IDS Director. Here, *number_events* is the number of events in the event queue. The default is 100. Increasing this number may have an impact on memory and performance, as each event in the event queue requires 32 KB of memory.
Step 3	`Router(config)# exit`	Exits global configuration mode.

Initializing the Post Office

Note You must reload the router every time you make a Post Office configuration change.

To initialize the Post Office system, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip audit notify nr-director` or `Router(config)#ip audit notify log`	Sends event notifications (alarms) to either a Cisco Secure IDS Director, a syslog server, or both. For example, if you are sending alarms to a Cisco Secure IDS Director, use the **nr-director** keyword in the command syntax. If you are sending alarms to a syslog server, use the **log** keyword in the command syntax.
Step 2	`router(config)# ip audit po local hostid` *host-id* `orgid` *org-id*	Sets the Post Office parameters for both the router (using the **ip audit po local** command) and the Cisco Secure IDS Director (using the **ip audit po remote** command). Here, *host-id* is a unique number between 1 and 65535 that identifies the router, and *org-id* is a unique number between 1 and 65535 that identifies the organization to which the router and Director both belong.
Step 3	`Router(config)# ip audit po remote hostid` *host-id* `orgid` *org-id* `rmtaddress` *ip-address* `localaddress` *ip-address* `port` *port-number* `preference` *preference-number* `timeout` *seconds* `application` *application-type*	Sets the Post Office parameters for both the Cisco Secure IDS Director (using the **ip audit po remote** command). • *host-id* is a unique number between 1 and 65535 that identifies the Director. • *org-id* is a unique number between 1 and 65535 that identifies the organization to which the router and Director both belong. • **rmtaddress** *ip-address* is the Director's IP address. • **localaddress** *ip-address* is the router's interface IP address. • *port-number* identifies the UDP port on which the Director is listening for alarms (45000 is the default). • *preference-number* is the relative priority of the route to the Director (1 is the default)—if more than one route is used to reach the same Director, then one must be a primary route (preference 1) and the other a secondary route (preference 2). • *seconds* is the number of seconds the Post Office waits before it determines that a connection has timed out (5 is the default). • *application-type* is either **director** or **logger**. **Note** If you are sending Post Office notifications to a Sensor, use **logger** instead of **director** as your application. Sending to a logging application means that no alarms are sent to a GUI; instead, the Cisco Secure IDS alarm data is written to a flat file, which can then be processed with filters, such as **perl** and **awk**, or staged to a database. Use **logger** only in advanced applications where you want the alarms only to be logged and not displayed.

	Command	Purpose
Step 4	`Router(config)# logging console info`	Displays the syslog messages on the router console if you are sending alarms to the syslog console.
Step 5	`Router(config)# exit`	Exits global configuration mode.
Step 6	`Router# write memory`	Saves the configuration.
Step 7	`Router# reload`	Reloads the router.

After you have configured the router, add the Cisco IOS Firewall IDS router's Post Office information to the */usr/nr/etc/hosts* and */usr/nr/etc/routes* files on the Cisco Secure IDS Sensors and Directors communicating with the router. You can do this with the nrConfigure tool in Cisco Secure IDS. For more information, refer to the *NetRanger User Guide*.

Configuring and Applying Audit Rules

To configure and apply audit rules, use the following commands starting in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip audit info {action [alarm] [drop] [reset]}` and `Router(config)# ip audit attack {action [alarm] [drop] [reset]}`	Sets the default actions for info and attack signatures. Both types of signatures can take any or all of the following actions: alarm, drop, and reset. The default action is **alarm**.
Step 2	`Router(config)# ip audit name audit-name {info | attack} [list standard-acl] [action [alarm] [drop] [reset]]`	Creates audit rules, where *audit-name* is a user-defined name for an audit rule. For example: **ip audit name** *audit-name* **info** **ip audit name** *audit-name* **attack** The default action is **alarm**. **Note** Use the same name when you assign attack and info type signatures. You can also use the **ip audit name** command to attach access control lists to an audit rule for filtering out sources of false alarms. In this case *standard-acl* is an integer representing an ACL. If you attach an ACL to an audit rule, the ACL must be defined as well: **ip audit name** *audit-name* **{info|attack} list** *acl-list* In the following example, ACL 99 is attached to the audit rule INFO, and ACL 99 is defined: `ip audit name INFO info list 99` `access-list 99 deny 10.1.1.0 0.0.0.255` `access-list 99 permit any` **Note** The ACL in the preceding example is *not* denying traffic from the 10.1.1.0 network (as expected if it were applied to an interface). Instead, the hosts on that network are not filtered through the audit process because they are trusted hosts. On the other hand, all other hosts, as defined by **permit any**, are processed by the audit rule.

	Command	Purpose
Step 3	`Router(config)# ip audit signature signature-id {disable | list acl-list}`	Disables individual signatures. Disabled signatures are not included in audit rules, as this is a global configuration change: `ip audit signature signature-number disable` To re-enable a disabled signature, use the **no ip audit signature** command, where *signature-number* is the number of the disabled signature. You can also use the **ip audit signature** command to apply ACLs to individual signatures for filtering out sources of false alarms. In this case *signature-number* is the number of a signature, and *acl-list* is an integer representing an ACL: `ip audit signature signature-number list acl-list` For example, ACL 35 is attached to the 1234 signature, and then defined: `ip audit signature 1234 list 35` `access-list 35 deny 10.1.1.0 0.0.0.255` `access-list 35 permit any` **Note** The ACL in the preceding example is *not* denying traffic from the 10.1.1.0 network (as expected if it were applied to an interface). Instead, the hosts on that network are not filtered through the signature because they are trusted hosts or are otherwise causing false positives to occur. On the other hand, all other hosts, as defined by **permit any**, are processed by the signature.
Step 4	`Router(config-if)# interface interface-number`	Enters interface configuration mode.
Step 5	`Router(config-if)# ip audit audit-name {in | out}`	Applies an audit rule at an interface. With this command, *audit-name* is the name of an existing audit rule, and *direction* is either **in** or **out**.
Step 6	`Router(config-if)# exit`	Exits interface configuration mode.
Step 7	`Router(config)# ip audit po protected ip-addr [to ip-addr]`	Configures which network should be protected by the router. Here, *ip_addr* is the IP address to protect.
Step 8	`Router(config)# exit`	Exits global configuration mode.

Verifying the Configuration

You can verify that Cisco IOS Firewall IDS is properly configured with the **show ip audit configuration** command (see Example 1).

Example 1 Output from show ip audit configuration Command

```
ids2611# show ip audit configuration
```

```
Event notification through syslog is enabled
Event notification through Net Director is enabled
Default action(s) for info signatures is alarm
Default action(s) for attack signatures is alarm drop reset
Default threshold of recipients for spam signature is 25
PostOffice:HostID:55 OrgID:123 Msg dropped:0
          :Curr Event Buf Size:100  Configured:100
HID:14 OID:123 S:1 A:2 H:82 HA:49 DA:0 R:0 Q:0
 ID:1 Dest:10.1.1.99:45000 Loc:172.16.58.99:45000 T:5 S:ESTAB *

Audit Rule Configuration
 Audit name AUDIT.1
  info actions alarm
  attack actions alarm drop reset
```

You can verify which interfaces have audit rules applied to them with the **show ip audit interface** command (see Example 2).

Example 2 Output from show ip audit interface Command

```
ids2611# show ip audit interface

Interface Configuration
 Interface Ethernet0
 Inbound IDS audit rule is AUDIT.1
  info actions alarm
  attack actions alarm drop reset
 Outgoing IDS audit rule is not set
Interface Ethernet1
 Inbound IDS audit rule is AUDIT.1
  info actions alarm
  attack actions alarm drop reset
 Outgoing IDS audit rule is not set
```

Monitoring and Maintaining Cisco IOS Firewall IDS

This section describes the EXEC commands used to monitor and maintain Cisco IOS Firewall IDS.

Command	Purpose
`Router# clear ip audit configuration`	Disables Cisco IOS Firewall IDS, removes all intrusion detection configuration entries, and releases dynamic resources.
`Router# clear ip audit statistics`	Resets statistics on packets analyzed and alarms sent.
`Router# show ip audit statistics`	Displays the number of packets audited and the number of alarms sent, among other information.

The following display provides sample output from the **show ip audit statistics** command:

```
Signature audit statistics [process switch:fast switch]
 signature 2000 packets audited: [0:2]
 signature 2001 packets audited: [9:9]
 signature 2004 packets audited: [0:2]
 signature 3151 packets audited: [0:12]
Interfaces configured for audit 2
Session creations since subsystem startup or last reset 11
Current session counts (estab/half-open/terminating) [0:0:0]
```

```
Maxever session counts (estab/half-open/terminating) [2:1:0]
Last session created 19:18:27
Last statistic reset never

HID:1000 OID:100 S:218 A:3 H:14085 HA:7114 DA:0 R:0
```

Cisco IOS Firewall IDS Configuration Examples

The following sections provide Cisco IOS Firewall IDS configuration examples:

- Cisco IOS Firewall IDS Reporting to Two Directors Example
- Adding an ACL to the Audit Rule Example
- Disabling a Signature Example
- Adding an ACL to Signatures Example
- Dual-Tier Signature Response Example

Cisco IOS Firewall IDS Reporting to Two Directors Example

In the following example, Cisco IOS Firewall IDS is initialized. Notice that the router is reporting to two Directors. Also notice that the AUDIT.1 audit rule will apply both info and attack signatures.

```
ip audit smtp spam 25
ip audit notify nr-director
ip audit notify log
ip audit po local hostid 55 orgid 123
ip audit po remote hostid 14 orgid 123 rmtaddress 10.1.1.99 localaddress 10.1.1.1
ip audit po remote hostid 15 orgid 123 rmtaddress 10.1.2.99 localaddress 10.1.1.1

ip audit name AUDIT.1 info action alarm
ip audit name AUDIT.1 attack action alarm drop reset

interface e0
 ip address 10.1.1.1 255.0.0.0
 ip audit AUDIT.1 in

interface e1
 ip address 172.16.57.1 255.255.255.0
 ip audit AUDIT.1 in
```

Adding an ACL to the Audit Rule Example

In the following example, an ACL is added to account for a Cisco Secure IDS Scanner (172.16.59.16) that scans for all types of attacks. As a result, no packets originating from the device will be audited.

```
ip audit smtp spam 25
ip audit notify nr-director
ip audit notify log
ip audit po local hostid 55 orgid 123
ip audit po remote hostid 14 orgid 123 rmtaddress 10.1.1.99 localaddress 10.1.1.1
ip audit po remote hostid 15 orgid 123 rmtaddress 10.1.2.99 localaddress 10.1.1.1

ip audit name AUDIT.1 info list 90 action alarm
ip audit name AUDIT.1 attack list 90 action alarm drop reset
```

```
interface e0
 ip address 10.1.1.1 255.0.0.0
 ip audit AUDIT.1 in

interface e1
 ip address 172.16.57.1 255.255.255.0
 ip audit AUDIT.1 in

access-list 90 deny 172.16.59.16
access-list 90 permit any
```

Disabling a Signature Example

The security administrator notices that the router is generating a lot of false positives for signatures 1234, 2345, and 3456. The system administrator knows that there is an application on the network that is causing signature 1234 to fire, and it is not an application that should cause security concerns. This signature can be disabled, as illustrated in the following example:

```
ip audit smtp spam 25
ip audit notify nr-director
ip audit notify log
ip audit po local hostid 55 orgid 123
ip audit po remote hostid 14 orgid 123 rmtaddress 10.1.1.99 localaddress 10.1.1.1
ip audit po remote hostid 15 orgid 123 rmtaddress 10.1.2.99 localaddress 10.1.1.1


ip audit signature 1234 disable

ip audit name AUDIT.1 info list 90 action alarm
ip audit name AUDIT.1 attack list 90 action alarm drop reset

interface e0
 ip address 10.1.1.1 255.0.0.0
 ip audit AUDIT.1 in

interface e1
 ip address 172.16.57.1 255.255.255.0
 ip audit AUDIT.1 in

access-list 90 deny 172.16.59.16
access-list 90 permit any
```

Adding an ACL to Signatures Example

After further investigation, the security administrator discovers that the false positives for signatures 2345 and 3456 are caused by specific applications on hosts 10.4.1.1 and 10.4.1.2, as well as by some workstations using DHCP on the 172.16.58.0 subnetwork. Attaching an ACL that denies processing of these hosts stops the creation of false positive alarms, as illustrated in the following example:

```
ip audit smtp spam 25
ip audit notify nr-director
ip audit notify log
ip audit po local hostid 55 orgid 123
ip audit po remote hostid 14 orgid 123 rmtaddress 10.1.1.99 localaddress 10.1.1.1
ip audit po remote hostid 15 orgid 123 rmtaddress 10.1.2.99 localaddress 10.1.1.1

ip audit signature 1234 disable
ip audit signature 2345 list 91
```

```
ip audit signature 3456 list 91

ip audit name AUDIT.1 info list 90 action alarm
ip audit name AUDIT.1 attack list 90 action alarm drop reset

 interface e0
ip address 10.1.1.1 255.0.0.0
 ip audit AUDIT.1 in

interface e1
 ip address 172.16.57.1 255.255.255.0
 ip audit AUDIT.1 in

access-list 90 deny 172.16.59.16
access-list 90 permit any
access-list 91 deny host 10.4.1.1
access-list 91 deny host 10.4.1.2
access-list 91 deny 172.16.58.0 0.0.0.255
access-list 91 permit any
```

Dual-Tier Signature Response Example

The company has now reorganized and has placed only trusted people on the 172.16.57.0 network. The work done by the employees on these networks must not be disrupted by Cisco IOS Firewall IDS, so attack signatures in the AUDIT.1 audit rule now will only alarm on a match.

For sessions that originate from the outside network, any attack signature matches (other than the false positive ones that are being filtered out) are to be dealt with in the following manner: send an alarm, drop the packet, and reset the TCP session.

This dual-tier method of signature response is accomplished by configuring two different audit specifications and applying each to a different ethernet interface, as illustrated in the following example:

```
ip audit smtp spam 25
ip audit notify nr-director
ip audit notify log
ip audit po local hostid 55 orgid 123
ip audit po remote hostid 14 orgid 123 rmtaddress 10.1.1.99 localaddress 10.1.1.1
ip audit po remote hostid 15 orgid 123 rmtaddress 10.1.2.99 localaddress 10.1.1.1

ip audit signature 1234 disable
ip audit signature 2345 list 91
ip audit signature 3456 list 91

ip audit name AUDIT.1 info list 90 action alarm
ip audit name AUDIT.1 attack list 90 action alarm
ip audit name AUDIT.2 info action alarm
ip audit name AUDIT.2 attack alarm drop reset

interface e0
    ip address 10.1.1.1 255.0.0.0
    ip audit AUDIT.2 in

interface e1
    ip address 172.16.57.1 255.255.255.0
    ip audit AUDIT.1 in

access-list 90 deny host 172.16.59.16
access-list 90 permit any
access-list 91 deny host 10.4.1.1
access-list 91 deny host 10.4.1.2
```

```
access-list 91 deny 172.16.58.0 0.0.0.255
access-list 91 permit any
```

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Authentication Proxy

Configuring Authentication Proxy

This chapter describes the Cisco IOS Firewall Authentication Proxy feature. Authentication proxy provides dynamic, per-user authentication and authorization, authenticating users against industry standard TACACS+ and RADIUS authentication protocols. Authenticating and authorizing connections by users provides more robust protection against network attacks.

For a complete description of the authentication proxy commands in this chapter, refer to the "Authentication Proxy Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the chapter "Using Cisco IOS Software."

In This Chapter

This chapter contains the following sections:

About Authentication Proxy

The Cisco IOS Firewall authentication proxy feature allows network administrators to apply specific security policies on a per-user basis. Previously, user identity and related authorized access were associated with a user IP address, or a single security policy had to be applied to an entire user group or subnetwork. Now, users can be identified and authorized on the basis of their per-user policy. Tailoring of access privileges on an individual basis is possible, as opposed to applying a general policy across multiple users.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

With the authentication proxy feature, users can log in to the network or access the Internet via HTTP, and their specific access profiles are automatically retrieved and applied from a CiscoSecure ACS, or other RADIUS, or TACACS+ authentication server. The user profiles are active only when there is active traffic from the authenticated users.

The authentication proxy is compatible with other Cisco IOS security features such as Network Address Translation (NAT), Context-based Access Control (CBAC), IP Security (IPSec) encryption, and Cisco Secure VPN Client (VPN client) software.

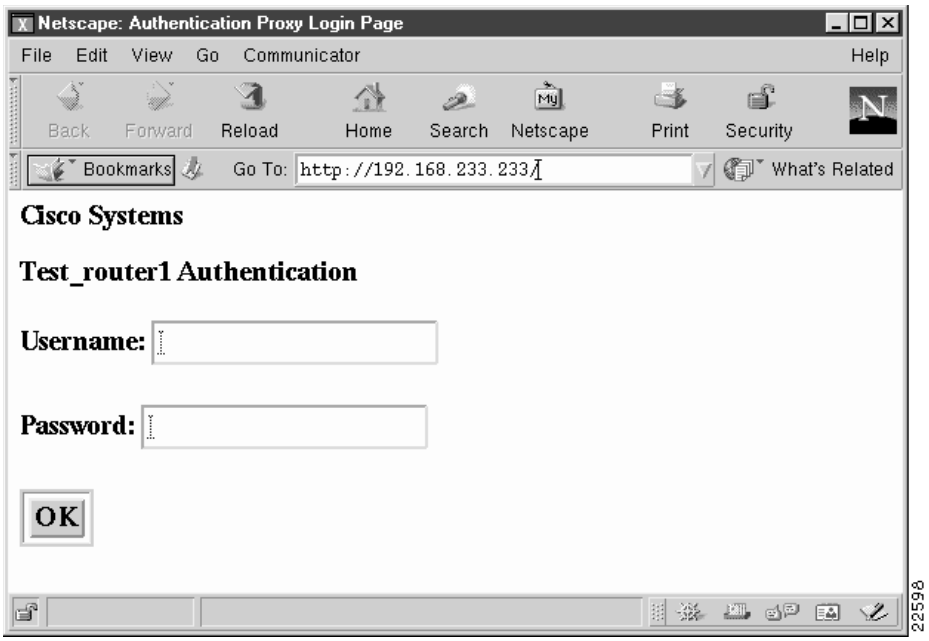
This section contains the following sections:

How the Authentication Proxy Works

When a user initiates an HTTP session through the firewall, the authentication proxy is triggered. The authentication proxy first checks to see if the user has been authenticated. If a valid authentication entry exists for the user, the connection is completed with no further intervention by the authentication proxy. If no entry exists, the authentication proxy responds to the HTTP connection request by prompting the user for a username and password.

Figure 42 illustrates the authentication proxy HTML login page.

Figure 42 Authentication Proxy Login Page



Users must successfully authenticate themselves with the authentication server by entering a valid username and password.

If the authentication succeeds, the user's authorization profile is retrieved from the AAA server. The authentication proxy uses the information in this profile to create dynamic access control entries (ACEs) and add them to the inbound (input) access control list (ACL) of an input interface and to the outbound (output) ACL of an output interface, if an output ACL exists at the interface. This process enables the firewall to allow authenticated users access to the network as permitted by the authorization profile. For example, a user can initiate a Telnet connection through the firewall if Telnet is permitted in the user's profile.

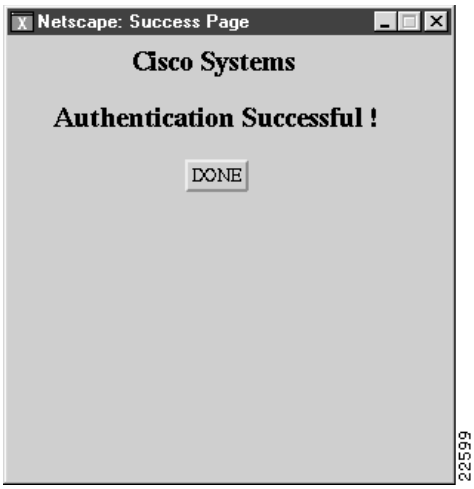
If the authentication fails, the authentication proxy reports the failure to the user and prompts the user with multiple retries. If the user fails to authenticate after five attempts, the user must wait two minutes and initiate another HTTP session to trigger authentication proxy.

The login page is refreshed each time the user makes requests to access information from a web server.

The authentication proxy customizes each of the access list entries in the user profile by replacing the source IP addresses in the downloaded access list with the source IP address of the authenticated host.

At the same time that dynamic ACEs are added to the interface configuration, the authentication proxy sends a message to the user confirming that the login was successful. Figure 43 illustrates the login status in the HTML page.

Figure 43 Authentication Proxy Login Status Message



The authentication proxy sets up an inactivity (idle) timer for each user profile. As long as there is activity through the firewall, new traffic initiated from the user's host does not trigger the authentication proxy, and authorized user traffic is permitted access through the firewall.

If the idle timer expires, the authentication proxy removes the user's profile information and dynamic access lists entries. When this happens, traffic from the client host is blocked. The user must initiate another HTTP connection to trigger the authentication proxy.

Secure Authentication

The authentication proxy uses JavaScript to help achieve secure authentication using the client browser. Secure authentication prevents a client from mistakenly submitting a username and password to a network web server other than the authentication proxy router.

This section contains the following sections:

- Operation with JavaScript
- Operation Without JavaScript

Operation with JavaScript

Users should enable JavaScript on the browser prior to initiating an HTTP connection. With JavaScript enabled on the browser, secure authentication is done automatically, and the user sees the authentication message shown in Figure 43. The HTTP connection is completed automatically for the user.

Operation Without JavaScript

If the client browser does not support JavaScript, or if site security policy prevents users from enabling JavaScript, any login attempt generates a popup window with instructions for manually completing the connection. Figure 44 illustrates the authentication proxy login status message with JavaScript disabled on the browser.

Figure 44 Authentication Proxy Login Status Message with JavaScript Disabled

To close this window, click Close on the browser File menu.

After closing the popup window, the user should click Reload (Refresh for Internet Explorer) in the browser window in which the authentication login page is displayed. If the user's last authentication attempt succeeds, clicking Reload brings up the web page the user is trying to retrieve. If the user's last attempt fails, clicking Reload causes the authentication proxy to intercept the client HTTP traffic again, prompting the user with another login page that solicits the username and password.

If JavaScript is not enabled, it is strongly recommended that site administrators advise users of the correct procedure for closing the popup window as described in the section "Establishing User Connections Without JavaScript."

Using the Authentication Proxy

Unlike some Cisco IOS Firewall features that operate transparently to the user, the authentication proxy feature requires some user interaction on the client host. Table 40 describes the interaction of the authentication proxy with the client host.

Table 40 Authentication Proxy Interaction with the Client Host

Authentication Proxy Action with Client	Description
Triggering on HTTP connections	If a user is not currently authenticated at the firewall router, any HTTP connection initiated by the user triggers the authentication proxy. If the user is already authenticated, the authentication proxy is transparent to the user.
Logging in using the login page	Triggering the authentication proxy generates an HTML-based login page.The user must enter a username and password to be authenticated with the AAA server. Figure 42 illustrates the authentication proxy login page.
Authenticating the user at the client	Following the login attempt, the authentication proxy action can vary depending on whether JavaScript is enabled in the browser. If JavaScript is enabled, and authentication is successful, the authentication proxy displays a message indicating the status of the authentication as shown in Figure 43. After the authentication status is displayed, the proxy automatically completes the HTTP connection. If JavaScript is disabled, and authentication is successful, the authentication proxy generates a popup window with additional instructions for completing the connection. See Figure 44. If authentication is unsuccessful in any case, the user must log in again from the login page.

When to Use the Authentication Proxy

Here are examples of situations in which you might use the authentication proxy:

- You want to manage access privileges on an individual (per-user) basis using the services provided by the authentication servers instead of configuring access control based on host IP address or global access policies. Authenticating and authorizing users from any host IP address also allows network administrators to configure host IP addresses using DHCP.
- You want to authenticate and authorize local users before permitting access to intranet or Internet services or hosts through the firewall.
- You want to authenticate and authorize remote users before permitting access to local services or hosts through the firewall.
- You want to control access for specific extranet users. For example, you might want to authenticate and authorize the financial officer of a corporate partner with one set of access privileges while authorizing the technology officer for that same partner to use another set of access privileges.
- You want to use the authentication proxy in conjunction with VPN client software to validate users and to assign specific access privileges.
- You want to use the authentication proxy in conjunction with AAA accounting to generate “start” and “stop” accounting records that can be used for billing, security, or resource allocation purposes, thereby allowing users to track traffic from the authenticated hosts.

Applying the Authentication Proxy

Apply the authentication proxy in the inbound direction at any interface on the router where you want per-user authentication and authorization. Applying the authentication proxy inbound at an interface causes it to intercept a user's initial connection request before that request is subjected to any other processing by the firewall. If the user fails to gain authentication with the AAA server, the connection request is dropped.

How you apply the authentication proxy depends on your security policy. For example, you can block all traffic through an interface and enable the authentication proxy feature to require authentication and authorization for all user initiated HTTP connections. Users are authorized for services only after successful authentication with the AAA server.

The authentication proxy feature also allows you to use standard access lists to specify a host or group of hosts whose initial HTTP traffic triggers the proxy.

Figure 45 shows the authentication proxy applied at the LAN interface with all network users required to be authenticated upon the initial connection (all traffic is blocked at each interface).

Figure 45 Applying the Authentication Proxy at the Local Interface

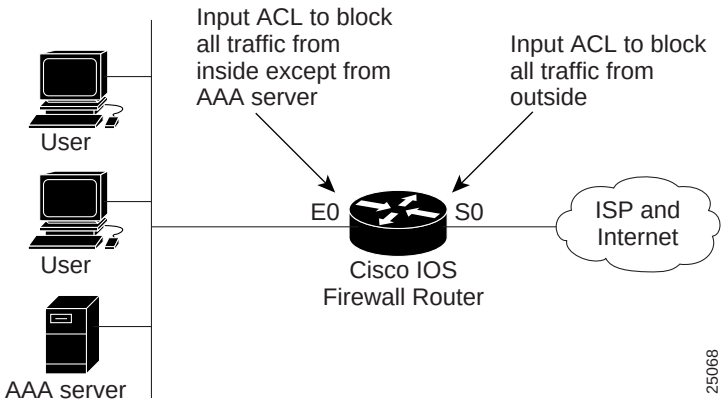
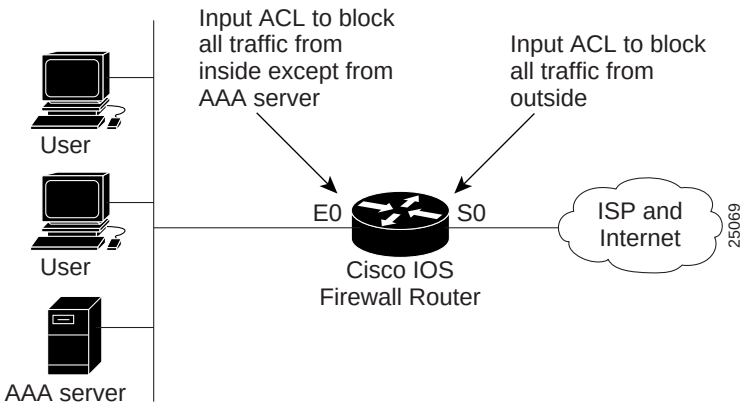


Figure 46 shows the authentication proxy applied at the dial-in interface with all network traffic blocked at each interface.

Figure 46 Applying the Authentication Proxy at an Outside Interface



Operation with One-Time Passwords

Given a one-time password, the user enters the username and one-time password in the HTML login page as usual.

The user must enter the correct token password within the first three attempts. After three incorrect entries, the user must enter two valid token passwords in succession before authentication is granted by the AAA server.

Compatibility with Other Security Features

The authentication proxy is compatible with Cisco IOS software and with Cisco IOS security features:

- Cisco IOS Firewall Intrusion Detection System (IDS)
- NAT
- CBAC
- IPSec encryption
- VPN client software

The authentication proxy works transparently with the Cisco IOS Firewall IDS and IPSec encryption features. The following sections describe the relationship of the NAT, CBAC, and VPN client software features with the authentication proxy:

- NAT Compatibility
- CBAC Compatibility
- VPN Client Compatibility

NAT Compatibility

The authentication proxy feature is compatible with NAT only if the ACL and authentication are completed prior to the NAT translation. Although NAT is compatible with the authentication proxy feature, NAT is not a requirement of the feature.

CBAC Compatibility

Although authentication proxy is compatible with CBAC security functions, CBAC is not required to use the authentication proxy feature.

Authentication proxy's authorization returns Access Control Entries (ACEs) that are dynamically prepended into a manually created ACL. Thereafter, apply the ACL to the "protected side" inbound interface, allowing or disallowing an authorized user's source IP address access to the remote networks.

VPN Client Compatibility

Using the authentication proxy, network administrators can apply an extra layer of security and access control for VPN client traffic. If a VPN client initiates an HTTP connection, the authentication proxy first checks for prior client authentication. If the client is authenticated, authorized traffic is permitted. If the client is not authenticated, the HTTP request triggers the authentication proxy, and the user is prompted for a username and password.

If the user authentication is successful, the authentication proxy retrieves the user profile from the AAA server. The source address in the user profile entries is replaced with the IP address of the authenticated VPN client from the decrypted packet.

Compatibility with AAA Accounting

Using the authentication proxy, you can generate "start" and "stop" accounting records with enough information to be used for billing and security auditing purposes. Thus, you can monitor the actions of authenticated hosts that use the authentication proxy service.

When an authentication proxy cache and associated dynamic access control lists are created, the authentication proxy will start to track the traffic from the authenticated host. Accounting saves data about this event in a data structure stored with the data of other users. If the accounting start option is enabled, you can generate an accounting record (a "start" record) at this time. Subsequent traffic from the authenticated host will be recorded when the dynamic ACL created by the authentication proxy receives the packets.

When an authentication proxy cache expires and is deleted, additional data, such as elapsed time, is added to the accounting information and a "stop" record is sent to the server. At this point, the information is deleted from the data structure.

The accounting records for the authentication proxy user session are related to the cache and the dynamic ACL usage.

Note The accounting records must include RADIUS attributes 42, 46, and 47 for both RADIUS and TACACS+.

For more information on RADIUS attributes, refer to the appendix "RADIUS Attributes."

Protection Against Denial-of-Service Attacks

The authentication proxy monitors the level of incoming HTTP requests. For each request, the authentication proxy prompts the user's for login credentials. A high number of open requests could indicate that the router is the subject of a denial-of-service (DoS) attack. The authentication proxy limits the level of open requests and drops additional requests until the number of open requests has fallen below 40.

If the firewall is experiencing a high level of connection requests requiring authentication, legitimate network users may experience delays when making connections, or the connection may be rejected and the user must try the connection again.

Risk of Spoofing with Authentication Proxy

When the authentication proxy is triggered, it creates a dynamic opening in the firewall by temporarily reconfiguring an interface with user access privileges. While this opening exists, another host might spoof the authenticated users address to gain access behind the firewall. The authentication proxy does not cause the address spoofing problem; the problem is only identified here as a matter of concern to the user. Spoofing is a problem inherent to all access lists, and the authentication proxy does not specifically address this problem.

Comparison with the Lock-and-Key Feature

Lock-and-key is another Cisco IOS Firewall feature that uses authentication and dynamic access list to provide user access through the firewall. Table 41 compares the authentication proxy and lock-and-key features.

Table 41 Comparison of the Authentication Proxy and Lock-and-Key Features

Lock-and-Key	Authentication Proxy
Triggers on Telnet connection requests.	Triggers on HTTP connection requests.
TACACS+, RADIUS, or local authentication.	TACACS+ or RADIUS authentication and authorization.
Access lists are configured on the router only.	Access lists are retrieved from the AAA server only.
Access privileges are granted on the basis of the user's host IP address.	Access privileges are granted on a per-user and host IP address basis.
Access lists are limited to one entry for each host IP address.	Access lists can have multiple entries as defined by the user profiles on the AAA server.
Associates a fixed IP addresses with a specific user. Users must log in from the host with that IP address.	Allows DHCP-based host IP addresses, meaning that users can log in from any host location and obtain authentication and authorization.

Use the authentication proxy in any network environment that provides a per-user security policy. Use lock-and-key in network environments that might benefit from local authentication and a limited number of router-based access control policies based on host addresses. Use lock-and-key in environments not using the Cisco Secure Integrated Software.

Restrictions

- The authentication proxy triggers only on HTTP connections.
- HTTP services must be running on the standard (well-known) port, which is port 80 for HTTP.
- Client browsers must enable JavaScript for secure authentication.
- The authentication proxy access lists apply to traffic passing through the router. Traffic destined to the router is authenticated by the existing authentication methods provided by Cisco IOS software.
- The authentication proxy does not support concurrent usage; that is, if two users try to log in from the same host at the same time, authentication and authorization applies only to the user who first submits a valid username and password.
- Load balancing using multiple or different AAA servers is not supported.

Prerequisites to Configuring Authentication Proxy

Prior to configuring authentication proxy, review the following:

- For the authentication proxy to work properly, the client host must be running the following browser software:
  - Microsoft Internet Explorer 3.0 or later
  - Netscape Navigator 3.0 or later
- The authentication proxy has an option to use standard access lists. You must have a solid understanding of how access lists are used to filter traffic before you attempt to configure the authentication proxy. For an overview of how to use access lists with the Cisco IOS Firewall, refer to the chapter “Access Control Lists: Overview and Guidelines.”
- The authentication proxy employs user authentication and authorization as implemented in the Cisco authentication, authorization, and accounting (AAA) paradigm. You must understand how to configure AAA user authentication, authorization, and accounting before you configure the authentication proxy. User authentication, authorization, and accounting are explained in the chapter “Authentication, Authorization, and Accounting (AAA).”
- To run the authentication proxy successfully with Cisco IOS Firewall, configure CBAC on the firewall. For complete information on the CBAC feature, refer to the chapter “Configuring Context-Based Access Control.”

Authentication Proxy Configuration Task List

To configure the authentication proxy feature, perform the following tasks:

- Configuring AAA (Required)
- Configuring the HTTP Server (Required)
- Configuring the Authentication Proxy (Required)
- Verifying the Authentication Proxy (Optional)

For authentication proxy configuration examples using the commands in this chapter, refer to the section “Authentication Proxy Configuration Examples” at the end of this chapter.

Configuring AAA

You must configure the authentication proxy for AAA services. Use the following commands in global configuration mode to enable authorization and to define the authorization methods:

	Command	Purpose
Step 1	`router(config)# aaa new-model`	Enables the AAA functionality on the router.
Step 2	`router(config)# aaa authentication login default` *TACACS+ RADIUS*	Defines the list of authentication methods at login.
Step 3	`router(config)# aaa authorization auth-proxy default` [*method1* [*method2*...]]	Uses the **auth-proxy** keyword to enable authentication proxy for AAA methods.
Step 4	`router(config)# aaa accounting auth-proxy default start-stop group tacacs+`	Uses the **auth-proxy** keyword to set up the authorization policy as dynamic ACLs that can be downloaded. This command activates authentication proxy accounting.
Step 5	`router(config)# tacacs-server host` *hostname*	Specifies an AAA server. For RADIUS servers, use the **radius server host** command.
Step 6	`router(config)# tacacs-server key` *key*	Sets the authentication and encryption key for communications between the router and the AAA server. For RADIUS servers use the **radius server key** command.
Step 7	`router(config)# access-list` *access-list-number* `permit tcp host` *source* `eq` *tacacs* `host` *destination*	Creates an ACL entry to allow the AAA server to return traffic to the firewall. The source address is the IP address of the AAA server, and the destination is the IP address of the router interface where the AAA server resides.

In addition to configuring AAA on the firewall router, the authentication proxy requires a per-user access profile configuration on the AAA server. To support the authentication proxy, configure the AAA authorization service **auth-proxy** on the AAA server as outlined here:

- Define a separate section of authorization for the **auth-proxy** keyword to specify the downloadable user profiles. This keyword does not interfere with other type of services, such as EXEC. The following example shows a user profile on a TACACS server:

```
default authorization = permit
key = cisco
user = newuser1 {
login = cleartext cisco
service = auth-proxy
{
priv-lvl=15
proxyacl#1="permit tcp any any eq 26"
proxyacl#2="permit icmp any host 60.0.0.2″
proxyacl#3="permit tcp any any eq ftp"
proxyacl#4="permit tcp any any eq ftp-data"
proxyacl#5="permit tcp any any eq smtp"
proxyacl#6="permit tcp any any eq telnet"
}
}
```

- The only supported attribute in the AAA server user configuration is proxyacl#n. Use the proxyacl#n attribute when configuring the access lists in the profile. The attribute proxyacl#n is for both RADIUS and TACACS+ attribute-value (AV) pairs.
- The privilege level must be set to 15 for all users.

- The access lists in the user profile on the AAA server must have access commands that contain only the **permit** keyword.
- Set the source address to the **any** keyword in each of the user profile access list entries. The source address in the access lists is replaced with the source address of the host making the authentication proxy request when the user profile is downloaded to the firewall.
- The supported AAA servers are:
  - CiscoSecure ACS 2.1.x for Windows NT
  - CiscoSecure ACS 2.3 for Windows NT
  - CiscoSecure ACS 2.2.4 for UNIX
  - CiscoSecure ACS 2.3 for UNIX
  - TACACS+ server (vF4.02.alpha)
  - Ascend RADIUS server radius-980618 (required attribute-value pair patch)
  - Livingston RADIUS server (v1.16)

Refer to the section "AAA Server User Profile Example" for sample AAA server configurations.

Configuring the HTTP Server

To use authentication proxy, you must also enable the HTTP server on the firewall and set the HTTP server authentication method to use AAA. Enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`router(config)# ip http server`	Enables the HTTP server on the router. The authentication proxy uses the HTTP server to communicate with the client for user authentication.
Step 2	`router(config)# ip http access-class` *access-list-number*	Specifies the access list for the HTTP server. Use the standard access list number configured in the section "Interface Configuration Example."

Configuring the Authentication Proxy

Note Set the **auth-cache-time** option for any authentication proxy rule to a higher value than the idle timeout value for any CBAC inspection rule. When the authentication proxy removes an authentication cache along with its associated dynamic user ACL, there may be some idle connections monitored by CBAC, and removal of user-specific ACLs could cause those idle connections to hang. If CBAC has a shorter idle timeout, CBAC resets these connections when the idle timeout expires; that is, before the authentication proxy removes the user profile.

To configure the authentication proxy, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`router(config)# ip auth-proxy auth-cache-time min`	Sets the global authentication proxy idle timeout value in minutes. If the timeout expires, user authentication entries are removed, along with any associated dynamic access lists. The default value is 60 minutes.
Step 2	`router(config)# ip auth-proxy auth-proxy-banner`	(Optional) Displays the name of the firewall router in the authentication proxy login page. The banner is disabled by default.
Step 3	`router(config)# ip auth-proxy name auth-proxy-name http [auth-cache-time min] [list {acl |acl-name}]`	Creates authentication proxy rules. The rules define how you apply authentication proxy. This command associates connections initiating HTTP protocol traffic with an authentication proxy name. You can associate the named rule with an access control list (ACL), providing control over which hosts use the authentication proxy feature. If no standard access list is defined, the named authentication proxy rule intercepts HTTP traffic from all hosts whose connection initiating packets are received at the configured interface. (Optional) The **auth-cache-time** option overrides the global authentication proxy cache timer. This option provides more control over timeout values for a specific authentication proxy rule. If no value is specified, the proxy rule assumes the value set with the **ip auth-proxy auth-cache-time** command. (Optional) The **list** option allows you to apply a standard, extended (1-199), or named access list to a named authentication proxy rule. HTTP connections initiated by hosts in the access list are intercepted by the authentication proxy.
Step 4	`router(config)# interface type`	Enters interface configuration mode by specifying the interface type on which to apply the authentication proxy.
Step 5	`router(config-if)# ip auth-proxy auth-proxy-name`	In interface configuration mode, applies the named authentication proxy rule at the interface. This command enables the authentication proxy rule with that name.

Verifying the Authentication Proxy

Verifying the authentication proxy configuration can have several components:

- Checking the Authentication Proxy Configuration (Optional)
- Establishing User Connections with JavaScript (Optional)
- Establishing User Connections Without JavaScript (Optional)

Checking the Authentication Proxy Configuration

To check the current authentication proxy configuration, use the **show ip auth-proxy configuration** command in privileged EXEC mode:

Command	Purpose
`router# show ip auth-proxy configuration`	Displays the authentication proxy configuration.

In the following example, the global authentication proxy idle timeout value is set to 60 minutes, the named authentication proxy rule is "pxy", and the idle timeout value for this named rule is one minute. The display shows that no host list is specified, meaning that all connections initiating HTTP traffic at the interface are subject to the authentication proxy rule.

```
router# show ip auth-proxy configuration
Authentication cache time is 60 minutes
Authentication Proxy Rule Configuration
Auth-proxy name pxy
http list not specified auth-cache-time 1 minutes
```

To verify that the authentication proxy is successfully configured on the router, ask a user to initiate an HTTP connection through the router. The user must have authentication and authorization configured at the AAA server. If the user authentication is successful, the firewall completes the HTTP connection for the user. If the authentication is unsuccessful, check the access list and the AAA server configurations.

Display the user authentication entries using the **show ip auth-proxy cache** command in privileged EXEC mode:

Command	Purpose
`router# show ip auth-proxy cache`	Displays the list of user authentication entries.

The authentication proxy cache lists the host IP address, the source port number, the timeout value for the authentication proxy, and the state of the connection. If the authentication proxy state is HTTP_ESTAB, the user authentication was successful.

```
router# show ip auth-proxy cache
Authentication Proxy Cache
 Client IP 192.168.25.215 Port 57882, timeout 1, state HTTP_ESTAB
```

Wait for one minute, which is the timeout value for this named rule, and ask the user to try the connection again. After one minute, the user connection is denied because the authentication proxy has removed the user's authentication entry and any associated dynamic ACLs. The user is presented with a new authentication login page and must log in again to gain access through the firewall.

Establishing User Connections with JavaScript

To verify client connections using the authentication proxy with JavaScript enabled on the client browser, follow this procedure:

Step 1 From a client host, initiate an HTTP connection through the firewall. This generates the authentication proxy login page.

Step 2 At the authentication proxy login page, enter a username and password.

Step 3 Click **OK** to submit the username and password to the AAA server.

A popup window appears indicating whether the login attempt succeeded or failed. If the authentication is successful, the connection is completed automatically. If the authentication fails, the authentication proxy reports the failure to the user and prompts the user with multiple retries.

Note If the authentication attempt is unsuccessful after five attempts, the user must wait two minutes and initiate another HTTP session to trigger authentication proxy.

Establishing User Connections Without JavaScript

To ensure secure authentication, the authentication proxy design requires JavaScript. You can use the authentication proxy without enabling JavaScript on the browser, but this poses a potential security risk if users do not properly establish network connections. The following procedure provides the steps to properly establish a connection with JavaScript disabled. Network administrators are strongly advised to instruct users on how to properly establish connections using the procedure in this section.

Note Failure to follow this procedure can cause user credentials to be passed to a network web server other than the authentication proxy or can cause the authentication proxy to reject the login attempt.

To verify client connections using the authentication proxy when JavaScript is not enabled on the client browser, follow this procedure:

Step 1 Initiate an HTTP connection through the firewall.

This generates the authentication proxy login page.

Step 2 From the authentication proxy login page at the client, enter the username and password.

Step 3 Click **OK** to submit the username and password to the AAA server.

A popup window appears indicating whether the login attempt succeeded or failed. If the popup window indicates successful authentication, go to Step 7.

Step 4 If the popup window displays a failed authentication message, click **Close** on the browser **File** menu.

Note Do not click **Reload** (**Refresh** for Internet Explorer) to close the popup window.

Step 5 From the original authentication login page, click **Reload** (**Refresh** for Internet Explorer) on the browser toolbar. The user login credentials are cleared from the form.

Note Do not click **OK**. You must click **Reload** or **Refresh** to clear the username and password and to reload the form before attempting to log in again.

Step 6 Enter the username and password again.

If the authentication is successful, a window appears displaying a successful authentication message. If the window displays a failed authentication message, go to Step 4.

Step 7 Click **Close** on the browser **File** menu.

Step 8 From the original authentication proxy login page, click **Reload** (**Refresh** for Internet Explorer) on the browser toolbar.

The authentication proxy completes the authenticated connection with the web server.

Monitoring and Maintaining the Authentication Proxy

This section describes how to view dynamic access list entries and how to manually remove authentication entries. This section contains the following sections:

- Displaying Dynamic ACL Entries
- Deleting Authentication Proxy Cache Entries

Displaying Dynamic ACL Entries

You can display dynamic access list entries when they are in use. After an authentication proxy entry is cleared by you or by the idle timeout parameter, you can no longer display it. The number of matches displayed indicates the number of times the access list entry was hit.

To view dynamic access lists and any temporary access list entries that are currently established by the authentication proxy, use the **show ip access-lists** command in privileged EXEC mode:

Command	Purpose
`router# show ip access-lists`	Displays the standard and extended access lists configured on the firewall, including dynamic ACL entries.

Consider the following example where ACL 105 is applied inbound at the input interface where you configure authentication proxy. The initial display shows the contents of the ACLs prior to authentication. The second display shows the same displays after user authentication with the AAA server.

Note If NAT is configured, the **show ip access list** command might display the translated host IP address for the dynamic ACL entry or the IP address of the host initiating the connection. If the ACL is applied on the NAT outside interface, the translated address is displayed. If the ACL is applied on the NAT inside interface, the IP address of the host initiating the connection is displayed. The **show ip auth-proxy cache** command always displays the IP address of the host initiating the connection.

For example, the following is a list of ACL entries prior to the authentication proxy:

```
Router# show ip access-lists
    .
    .
    .
Extended IP access list 105
 deny tcp any any eq telnet
 deny udp any any
 permit tcp any any (28 matches)
 permit ip any any
```

The following sample output shows a list of ACL entries following user authentication:

```
Router# show ip access-lists
    .
    .
    .
Extended IP access list 105
! The ACL entries following user authentication are shown below.
 permit tcp host 192.168.25.215 any eq 26
 permit icmp host 192.168.25.215 host 60.0.0.2
 permit tcp host 192.168.25.215 any eq telnet
 permit tcp host 192.168.25.215 any eq ftp
 permit tcp host 192.168.25.215 any eq ftp-data
 permit tcp host 192.168.25.215 any eq smtp
 deny tcp any any eq telnet
 deny udp any any
 permit tcp any any (76 matches)
 permit ip any any
```

Deleting Authentication Proxy Cache Entries

When the authentication proxy is in use, dynamic access lists dynamically grow and shrink as authentication entries are added and deleted. To display the list of authentication entries, use the **show ip auth-proxy cache** command. To manually delete an authentication entry, use the **clear ip auth-proxy cache** command in privileged EXEC mode:

Command	Purpose
`router# clear ip auth-proxy cache {* | host ip address}`	Deletes authentication proxy entries from the firewall before they time out. Use an asterisk to delete all authentication cache entries. Enter a specific IP address to delete an entry for a single host.

Authentication Proxy Configuration Examples

Configuring the authentication proxy feature requires configuration changes on both the router and the AAA server. The following sections provide authentication proxy configuration examples:

- Authentication Proxy Configuration Example
- Authentication Proxy, IPSec, and CBAC Configuration Example
- Authentication Proxy, IPSec, NAT, and CBAC Configuration Example
- AAA Server User Profile Example

Throughout these examples, the exclamation point (!) indicates a comment line. Comment lines precede the configuration entries being described.

Authentication Proxy Configuration Example

The following examples highlight the specific authentication proxy configuration entries. These examples do not represent a complete router configuration. Complete router configurations using the authentication proxy are included later in this chapter.

This section contains the following examples:

- AAA Configuration Example
- HTTP Server Configuration Example
- Authentication Proxy Configuration Example
- Interface Configuration Example

AAA Configuration Example

```
aaa new-model
aaa authentication login default group tacacs group radius
! Set up the aaa new model to use the authentication proxy.
aaa authorization auth-proxy default group tacacs group radius
! Define the AAA servers used by the router.
aaa accounting auth-proxy default start-stop group tacacs+
! Set up authentication proxy with accounting.
tacacs-server host 172.31.54.143
tacacs-server key cisco
radius-server host 172.31.54.143
radius-server key cisco
```

HTTP Server Configuration Example

```
! Enable the HTTP server on the router.
ip http server
! Set the HTTP server authentication method to AAA.
ip http authentication aaa
! Define standard access list 61 to deny any host.
access-list 61 deny any
! Use ACL 61 to deny connections from any host to the HTTP server.
ip http access-class 61
```

Authentication Proxy Configuration Example

```
! Set the global authentication proxy timeout value.
ip auth-proxy auth-cache-time 60
! Apply a name to the authentication proxy configuration rule.
ip auth-proxy name HQ_users http
```

Interface Configuration Example

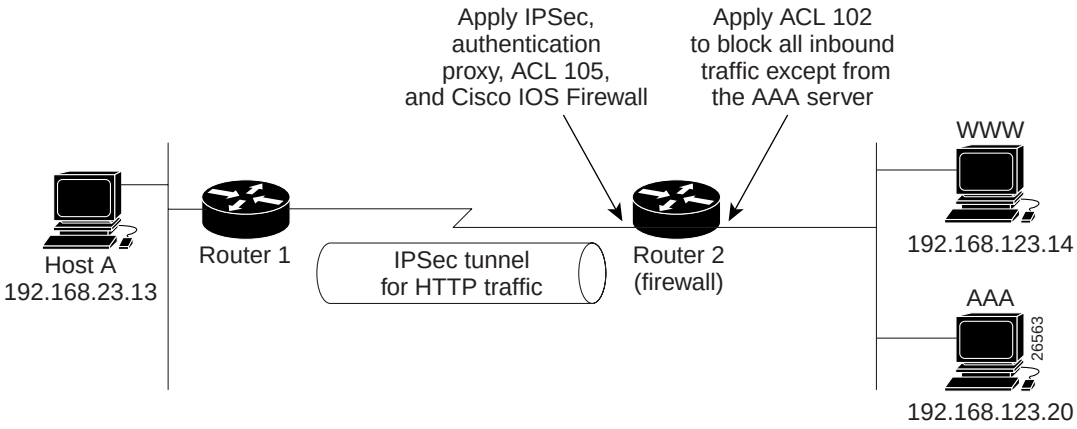
```
! Apply the authentication proxy rule at an interface.
interface e0
    ip address 10.1.1.210 255.255.255.0
    ip auth-proxy HQ_users
```

Authentication Proxy, IPSec, and CBAC Configuration Example

The following example shows a router configuration with the authentication proxy, IPSec, and CBAC features. Figure 47 illustrates the configuration.

Note If you are using this feature with Cisco IOS software release 12.3(8)T or later, see the document Crypto Access Check on Clear-Text Packets (feature module, release 12.3(8)T).

Figure 47 Authentication Proxy, IPSec, and CBAC Configuration Example



In this example, Host A initiates an HTTP connection with the web server (WWW). The HTTP traffic between Router 1 and Router 2 is encrypted using IPSec. The authentication proxy, IPSec, and CBAC are configured at interface Serial0 on Router 2, which is acting as the firewall. ACL 105 blocks all traffic at interface Serial0. ACL 102 is applied at interface Ethernet0 on Router 2 to block all traffic on that interface except traffic from the AAA server.

When Host A initiates an HTTP connection with the web server, the authentication proxy prompts the user at Host A for a username and password. These credentials are verified with the AAA server for authentication and authorization. If authentication is successful, the per-user ACLs are downloaded to the firewall to permit services.

The following examples provide both the Router 1 and Router 2 configurations for completeness:

- Router 1 Configuration Example
- Router 2 Configuration Example

Router 1 Configuration Example

```
! Configure Router 1 for IPSec.
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router1
!
logging buffered 4096 debugging
no logging console
enable secret 5 $1$E0OB$AQF1vFZM3fLr3LQAOsudL/
enable password junk
!
username Router2 password 0 welcome
crypto isakmp policy 1
 authentication pre-share
```

```
crypto isakmp key cisco1234 address 10.0.0.2
!
crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
!
!
 crypto map testtag 10 ipsec-isakmp
 set peer 10.0.0.2
 set transform-set rule_1
 match address 155
!
interface Ethernet0/0
 ip address 192.168.23.2 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Serial3/1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 encapsulation PPP
 ip route-cache
 no ip mroute-cache
 no keepalive
 no fair-queue
 clockrate 56000
 crypto map testtag
!
!
ip classless
ip route 192.168.123.0 255.255.255.0 10.0.0.2
! Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.23.13 host 192.168.123.14 eq www
access-list 155 permit tcp host 192.168.23.13 eq www host 192.168.123.14
```

Router 2 Configuration Example

```
! Configure Router 2 as the firewall, using the authentication proxy, IPSec, and CBAC.
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router2
!
logging buffered 4096 debugging
aaa new-model
aaa authentication login default group tacacs
aaa authentication login console_line none
aaa authentication login special none
aaa authentication ppp default group tacacs
aaa authorization exec default group tacacs
! Configure AAA for the authentication proxy.
aaa authorization auth-proxy default group tacacs+
enable password junk
!
! Create the CBAC inspection rule HTTP_TEST.
ip inspect name rule22 http
ip inspect name rule22 tcp
ip inspect name rule22 ftp
ip inspect name rule22 smtp
!
! Create the authentication proxy rule PXY.
```

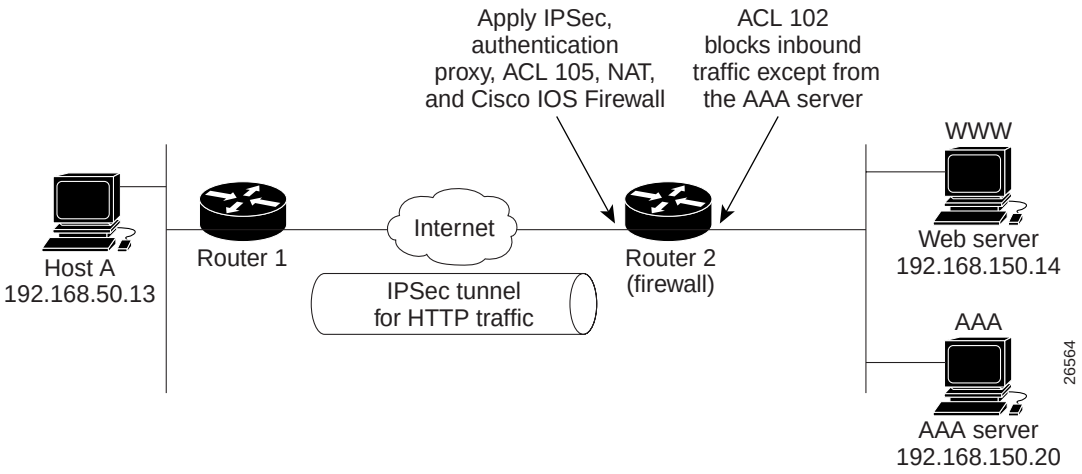
```
ip auth-proxy name pxy http
! Turn on display of the router name in the authentication proxy login page.
ip auth-proxy auth-proxy-banner
ip audit notify log
ip audit po max-events 100
!
! Configure IPSec.
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco1234 address 10.0.0.1
!
crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
!
 crypto map testtag 10 ipsec-isakmp
 set peer 10.0.0.1
 set transform-set rule_1
 match address 155
!
! Apply the CBAC inspection rule and the authentication proxy rule at interface
! Serial0/0.
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
 ip access-group 105 in
 no ip directed-broadcast
 ip inspect rule22 in
 ip auth-proxy pxy
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 no keepalive
 no fair-queue
crypto map testtag
!
interface Ethernet0/1
 ip address 192.168.123.2 255.255.255.0
 ip access-group 102 in
 no ip directed-broadcast
 ip route-cache
 no ip mroute-cache
!
no ip classless
ip route 192.168.23.0 255.255.255.0 10.0.0.1
ip route 192.168.50.0 255.255.255.0 16.0.0.1
! Configure the HTTP server.
ip http server
ip http access-class 15
ip http authentication aaa
!
! Create ACL 15 to block all traffic for the http server.
access-list 15 deny any
! Create ACL 102 to block all traffic inbound on interface Ethernet0/1 except for
! traffic from the AAA server.
access-list 102 permit tcp host 192.168.123.20 eq tacacs host 192.168.123.2
access-list 102 deny   tcp any any
access-list 102 deny   udp any any
access-list 102 permit ip any any
! Create ACL 105 to block all traffic inbound on interface Serial0/0. Permit only IP
! protocol traffic.
access-list 105 deny   tcp any any
access-list 105 deny   udp any any
access-list 105 permit ip any any
! Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.123.14 host 192.168.23.13 eq www
access-list 155 permit tcp host 192.168.123.14 eq www host 192.168.23.13
```

```
!
! Define the AAA server host and encryption key.
tacacs-server host 192.168.123.14
tacacs-server key cisco
!
line con 0
 exec-timeout 0 0
 login authentication special
 transport input none
line aux 0
 transport input all
 speed 38400
 flowcontrol hardware
line vty 0 4
 password lab
```

Authentication Proxy, IPSec, NAT, and CBAC Configuration Example

The following example provides a router configuration with the authentication proxy, IPSec, NAT, and CBAC features. Figure 48 illustrates the configuration.

Figure 48 Authentication Proxy, IPSec, and CBAC Configuration Example



In this example, Host A initiates an HTTP connection with the web server (WWW). The HTTP traffic between router 1 (interface BRI0) and router 2 (interface Serial2) is encrypted using IPSec. The authentication proxy is configured on router 2, which is acting as the firewall. The authentication proxy, NAT, and CBAC are configured at interface Serial2, which is acting as the firewall. ACL 105 blocks all traffic at interface Serial2. ACL 102 is applied at interface Ethernet0 on router 2 to block all traffic on that interface except traffic from the AAA server. In this example, the authentication proxy uses standard ACL 10 to specify the hosts using the authentication proxy feature.

When any host in ACL 10 initiates an HTTP connection with the web server, the authentication proxy prompts the user at that host for a username and password. These credentials are verified with AAA server for authentication and authorization. If authentication is successful, the per-user ACLs are downloaded to the firewall to permit services.

The following examples provide both the router 1 and router 2 configurations for completeness:

- Router 1 Configuration Example

- Router 2 Configuration Example

Router 1 Configuration Example

```
! Configure router 1 for IPSec.
version 12.0
 service timestamps debug uptime
 service timestamps log uptime
 no service password-encryption
 !
 hostname Router1
 !
 logging buffered 4096 debugging
 no logging console
 !
isdn switch-type basic-5ess
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco1234 address 16.0.0.2
 crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
 !
  !
  crypto map testtag 10 ipsec-isakmp
  set peer 16.0.0.2
  set transform-set rule_1
  match address 155
 !
 !
 process-max-time 200
 !
 interface BRI0
  ip address 16.0.0.1 255.0.0.0
  no ip directed-broadcast
  encapsulation ppp
  dialer idle-timeout 5000
  dialer map ip 16.0.0.2 name router2 broadcast 50006
  dialer-group 1
  isdn switch-type basic-5ess
  crypto map testtag
 !
 interface FastEthernet0
  ip address 192.168.50.2 255.255.255.0
  no ip directed-broadcast
 !
 ip classless
 ip route 192.168.150.0 255.255.255.0 16.0.0.2
 no ip http server
! Identify the IPSec specific traffic.
 access-list 155 permit tcp host 192.168.50.13 host 192.168.150.100 eq www
 access-list 155 permit tcp host 192.168.50.13 eq www host 192.168.150.100
 dialer-list 1 protocol ip permit
 !
 line con 0
  exec-timeout 0 0
  transport input none
 line aux 0
 line vty 0 4
  password lab
  login
```

Router 2 Configuration Example

```
! Configure router 2 as the firewall, using the authentication proxy, IPSec, NAT, and
! CBAC.
 version 12.0
 service timestamps debug uptime
 service timestamps log uptime
 no service password-encryption
 !
 hostname router2
 !
 logging buffered 4096 debugging
 aaa new-model
 aaa authentication login default group tacacs+
 aaa authentication login console_line none
 aaa authorization exec default group tacacs+
 ! Configure AAA for the authentication proxy.
 aaa authorization auth-proxy default group tacacs+
!
 ! Create the CBAC inspection rule “rule44.”
 ip inspect name rule44 http java-list 5
 ip inspect name rule44 tcp
 ip inspect name rule44 ftp
 ip inspect name rule44 smtp
 !
 ! Create the authentication proxy rule “pxy.” Set the timeout value for rule
 ! pxy to three minutes. Standard ACL 10 is applied to the rule.
 ip auth-proxy name pxy http list 10 auth-cache-time 3
 isdn switch-type primary-5ess
 !
 ! Configure IPSec.
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco1234 address 16.0.0.1
 !
 !
 crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
 !
  !
  crypto map testtag 10 ipsec-isakmp
  set peer 16.0.0.1
  set transform-set rule_1
  match address 155
!
 controller T1 2/0
  framing esf
  linecode b8zs
  pri-group timeslots 1-24
!
! Apply ACL 102 inbound at interface Ethernet0/1 and configure NAT.
interface Ethernet0/1
 ip address 192.168.150.2 255.255.255.0
 ip access-group 102 in
 no ip directed-broadcast
 ip nat inside
 no ip mroute-cache
!
! Apply the authentication proxy rule PXY, CBAC inspection rule HTTP_TEST, NAT, and
! and ACL 105 at interface Serial2/0:23.
interface Serial2/0:23
 ip address 16.0.0.2 255.0.0.0
 ip access-group 105 in
 no ip directed-broadcast
```

```
 ip nat outside
 ip inspect rule44 in
 ip auth-proxy pxy
 encapsulation ppp
 ip mroute-cache
 dialer idle-timeout 5000
 dialer map ip 16.0.0.1 name router1 broadcast 71011
 dialer-group 1
 isdn switch-type primary-5ess
 fair-queue 64 256 0
 crypto map testtag
!
! Use NAT to translate the Web server address.
ip nat inside source static 192.168.150.14 192.168.150.100
ip classless
ip route 192.168.50.0 255.255.255.0 16.0.0.1
! Configure the HTTP server.
ip http server
ip http access-class 15
ip http authentication aaa
!
! Create standard ACL 5 to specify the list of hosts from which to accept java applets.
! ACL 5 is used to block Java applets in the CBAC inspection rule named “rule44,” which
! is applied at interface Serial2/0:23.
access-list 5 permit any
! Create standard ACL 10 to specify the hosts using the authentication proxy. This ACL
! used in the authentication proxy rule named “PXY”, which is applied at interface
! Serial2/0:23.
access-list 10 permit any
! Create ACL 15 to block all traffic for the http server.
access-list 15 deny any
! Create extended ACL 102 to block all traffic inbound on interface Ethernet0/1
! except for traffic from the AAA server.
access-list 102 permit tcp host 192.168.150.20 eq tacacs 192.168.150.2
access-list 102 deny   tcp any any
access-list 102 deny   udp any any
access-list 102 permit ip any any
! Create extended ACL 105 to block all TCP and UDP traffic inbound on interface
! Serial2/0:23.
access-list 105 deny   tcp any any
access-list 105 deny   udp any any
access-list 105 permit ip any any
! Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.150.100 host 192.168.50.13 eq www
access-list 155 permit tcp host 192.168.150.100 eq www host 192.168.50.13
dialer-list 1 protocol ip permit
! Define the AAA server host and encryption key.
tacacs-server host 192.168.126.14
tacacs-server key cisco
!
line con 0
 exec-timeout 0 0
! Define the AAA server host and encryption key.
 login authentication console_line
 transport input none
 line aux 0
 line vty 0 4
 password lab
!
!
 end
```

AAA Server User Profile Example

This section includes examples of the authentication proxy user profile entries on the AAA servers. The "proxyacl" entries define the user access privileges. After the user has successfully used the authentication proxy to log in, these entries are transferred to the firewall router. Each entry in the profile must specify "permit" access for the service or application. The source address in each entry is set to "any", which is replaced with the IP address of the authenticating host when the profile is downloaded to the firewall. The privilege level must be set to 15 for all AAA users.

This section contains the following sections:

CiscoSecure ACS 2.3 for Windows NT

This section describes how to configure authentication proxy on CiscoSecure ACS 2.3 for Windows NT. For detailed information about CiscoSecure ACS, refer to the documentation for that product.

The following sample configuration is for the TACACS+ service of CiscoSecure ACS for Windows NT.

Step 1 Click the Interface Configuration icon and click **TACACS+** (Cisco).

a. Scroll down to New Services.

b. Add a new service, "auth-proxy", in the Service field. Leave the Protocol field empty.

c. Select both the User and Group check boxes for the new service.

d. Scoll down to Advance Configuration Options and check the Per-user Advance TACACS+ features.

e. Click **Submit**.

Step 2 Click the Network Configuration icon.

a. Click the Add Entry icon for Network Access Servers and fill in the Network Access Server Hostname, IP address, and key (the key configured on the router) fields.

b. Select TACACS+ (Cisco) for the Authenticate Using option.

c. Click the Submit + Restart icon.

Step 3 Click the Group Setup icon.

a. Select a user group from the drop-down menu.

b. Select the Users in Group check box.

c. Select a user from the user list.

d. In the User Setup list, scroll down to TACACS+ Settings and select the "auth-proxy" check box.

e. Select the Custom Attributes check box.

f. Add the profile entries (do not use single or double quotes around the entries) and set the privilege level to 15.

```
priv-lvl=15
proxyacl#1=permit tcp any any eq 26
```

```
proxyacl#2=permit icmp any host 60.0.0.2
proxyacl#3=permit tcp any any eq ftp
proxyacl#4=permit tcp any any eq ftp-data
proxyacl#5=permit tcp any any eq smtp
proxyacl#6=permit tcp any any eq telnet
```

g. Click **Submit**.

Step 4 Click the User Setup icon.

a. Click **List All Users**.

b. Add a username.

c. Scoll down to User Setup Password Authentication.

d. Select SDI SecurID Token Card from the Password Authentication drop-down menu.

e. Select the previous configured user group 1.

f. Click **Submit**.

Step 5 Click Group Setup icon again.

a. Select the user group 1.

b. Click **Users in Group**.

c. Click **Edit Settings**.

d. Click the Submit + Restart icon to make sure the latest configuration is updated and sent to the AAA server.

CiscoSecure ACS 2.3 for UNIX

This section describes how to configure authentication proxy on CiscoSecure ACS 2.3 for UNIX. For detailed information regarding CiscoSecure ACS, refer to the documentation for that product.

To manage the CiscoSecure ACS using the Administrator program, you need a web browser that supports Java and JavaScript. You must enable Java in the browser application. You can start the Java-based CiscoSecure Administrator advanced configuration program from any of the CiscoSecure ACS Administrator web pages.

The following sample configuration procedure is for the TACACS+ service of CiscoSecure ACS 2.3 for UNIX.

Step 1 On the CiscoSecure ACS web menu bar of the CiscoSecure ACS web interface, click **Advanced** and then click **Advanced** again.

The Java-based CiscoSecure Administrator advanced configuration program appears. It might require a few minutes to load.

Step 2 In the CiscoSecure Administrator advanced configuration program, locate and deselect Browse in the Navigator pane of the tabbed Members page.

This displays the Create New Profile icon.

Step 3 In the Navigator pane, do one of the following:

- Locate and click the group to which the user will belong.
- If you do not want the user to belong to a group, click the [Root] folder icon.

Step 4 Click **Create Profile** to display the New Profile dialog box.

Step 5 Make sure the Group check box is cleared.

Step 6 Enter the name of the user you want to create and click **OK**. The new user appears in the tree.

Step 7 Click the icon for the group or user profile in the tree that is displayed in the Navigator pane of the tabbed Members page.

Step 8 If necessary, in the Profile pane, click the Profile icon to expand it.

A list or dialog box that contains attributes applicable to the selected profile or service appears in the window at the bottom right of the screen. The information in this window changes depending on what you have selected in the Profile pane.

Step 9 Click **Service-String**.

Step 10 Click **string**, enter **auth-proxy** in the text field, and click **Apply**.

Step 11 Select the **Option** menu.

Step 12 On the **Option** menu, click **Default Attributes**.

Step 13 Change the attribute from Deny to **Permit**.

Step 14 Click **Apply**.

Step 15 On the **Option** menu, click **Attribute** and enter the privilege level in the text field:

```
priv-lvl=15
```

Step 16 On the **Option** menu, click **Attribute** and enter the **proxyacl** entries in the text field:

```
proxyacl#1=”permit tcp any any eq 26”
```

Repeat this step for each additional service or protocol to add:

```
proxyacl#2=”permit icmp any host 60.0.0.2”
proxyacl#3=”permit tcp any any eq ftp”
proxyacl#4=”permit tcp any any eq ftp-data”
proxyacl#5=”permit tcp any any eq smtp”
proxyacl#6=”permit tcp any any eq telnet”
```

Step 17 When you have finished making all your changes, click **Submit**.

TACACS+ Server

```
default authorization = permit
key = cisco
user = Brian {
login = cleartext cisco
service = auth-proxy
 {
  priv-lvl=15
  proxyacl#1="permit tcp any any eq 26"
  proxyacl#2="permit icmp any host 60.0.0.2
  proxyacl#3="permit tcp any any eq ftp"
  proxyacl#4="permit tcp any any eq ftp-data"
  proxyacl#5="permit tcp any any eq smtp"
  proxyacl#6="permit tcp any any eq telnet"
 }
}
```

Livingston Radius Server

```
Bob Password = "cisco" User-Service-Type=Outbound-User
cisco-avpair = "auth-proxy:priv-lvl=15",
cisco-avpair = "auth-proxy:proxyacl#1=permit tcp any any eq 26",
cisco-avpair = "auth-proxy:proxyacl#2=permit icmp any host 60.0.0.2",
cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq ftp",
cisco-avpair = "auth-proxy:proxyacl#4=permit tcp any any eq ftp-data",
cisco-avpair = "auth-proxy:proxyacl#5=permit tcp any any eq smtp",
cisco-avpair = "auth-proxy:proxyacl#6=permit tcp any any eq telnet"
```

Ascend Radius Server

```
Alice Password = "cisco" User-Service = Dialout-Framed-User
cisco-avpair = "auth-proxy:priv-lvl=15",
cisco-avpair = "auth-proxy:proxyacl#1=permit tcp any any eq 26",
cisco-avpair = "auth-proxy:proxyacl#2=permit icmp any host 60.0.0.2",
cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq ftp",
cisco-avpair = "auth-proxy:proxyacl#4=permit tcp any any eq ftp-data",
cisco-avpair = "auth-proxy:proxyacl#5=permit tcp any any eq smtp",
cisco-avpair = "auth-proxy:proxyacl#6=permit tcp any any eq telnet"
```

Configuring Port to Application Mapping

This chapter describes the Cisco IOS Firewall Port to Application Mapping (PAM) feature. PAM enables CBAC-supported applications to be run on nonstandard ports. Using PAM, network administrators can customize access control for specific applications and services to meet the distinct needs of their networks.

For a complete description of the PAM commands in this chapter, refer to the chapter "Port to Application Mapping Commands" of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the chapter "Using Cisco IOS Software."

In This Chapter

This chapter contains the following sections:

About Port to Application Mapping

Port to Application Mapping (PAM) is a feature of the Cisco IOS Firewall feature set. PAM allows you to customize TCP or UDP port numbers for network services or applications. PAM uses this information to support network environments that run services using ports that are different from the registered or well-known ports associated with an application.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Using the port information, PAM establishes a table of default port-to-application mapping information at the firewall. The information in the PAM table enables Context-based Access Control (CBAC) supported services to run on nonstandard ports. Previously, CBAC was limited to inspecting traffic using only the well-known or registered ports associated with an application. Now, PAM allows network administrators to customize network access control for specific applications and services.

PAM also supports host or subnet specific port mapping, which allows you to apply PAM to a single host or subnet using standard access control lists (ACLs). Host or subnet specific port mapping is done using standard ACLs.

This section contains the following sections:

How PAM Works

PAM generates a table of information that identifies specific applications with specific TCP or UDP port information. When the firewall router first starts up, the PAM table is populated with system-defined mapping information. As you customize the mapping information, the PAM table is modified with the new information. The information in the PAM table serves as the default port mapping for traffic passing through the firewall.

PAM works with CBAC to identify the applications associated with various port numbers, including services running on non-standard ports, as it inspect traffic passing through the firewall. Previously, CBAC was limited to inspecting traffic using only the well-known or registered ports associated with an application.

Entries in the PAM table provide three types of mapping information:

- System-Defined Port Mapping
- User-Defined Port Mapping
- Host-Specific Port Mapping

System-Defined Port Mapping

PAM creates a table, or database, of system-defined mapping entries using the well-known or registered port mapping information set up during the system start-up. The system-defined entries comprise all the services supported by CBAC, which requires the system-defined mapping information to function properly. The system-defined mapping information cannot be deleted or changed; that is, you cannot map HTTP services to port 21 (FTP) or FTP services to port 80 (HTTP).

Note You can override the system-defined entries for specific hosts using the PAM host-specific option. Refer to the section "Host-Specific Port Mapping" in this chapter.

Table 42 lists the default system-defined services and applications in the PAM table.

Table 42 System-Defined Port Mapping

Application Name	Well-Known or Registered Port Number	Protocol Description
cuseeme	7648	CU-SeeMe Protocol
exec	512	Remote Process Execution
ftp	21	File Transfer Protocol (control port)
http	80	Hypertext Transfer Protocol
h323	1720	H.323 Protocol (for example, MS NetMeeting, Intel Video Phone)
login	513	Remote login
mgcp	2427	Media Gateway Control Protocol
msrpc	135	Microsoft Remote Procedure Call
netshow	1755	Microsoft NetShow
real-audio-video	7070	RealAudio and RealVideo
rtsp	8559	Real Time Streaming Protocol
shell	514	Remote command
sip	5060	Session Initiation Protocol
smtp	25	Simple Mail Transfer Protocol
sqlnet	1521	SQL-NET
streamworks	1558	StreamWorks Protocol
sunrpc	111	SUN Remote Procedure Call
telnet	23	Telnet
tftp	69	Trivial File Transfer Protocol
vdolive	7000	VDOLive Protocol

This section has the following sections:

User-Defined Port Mapping

Network services or applications that use non-standard ports require user-defined entries in the PAM table. For example, your network might run HTTP services on the non-standard port 8000 instead of on the system-defined default port (port 80). In this case, you can use PAM to map port 8000 with HTTP services. If HTTP services run on other ports, use PAM to create additional port mapping entries. After you define a port mapping, you can overwrite that entry at a later time by simply mapping that specific port with a different application.

Note If you try to map an application to a system-defined port, a message appears that warns you of a mapping conflict.

User-defined port mapping information can also specify a range of ports for an application by establishing a separate entry in the PAM table for each port number in the range.

User-defined entries are saved with the default mapping information when you save the router configuration.

Host-Specific Port Mapping

User-defined entries in the mapping table can include host-specific mapping information, which establishes port mapping information for specific hosts or subnets. In some environments, it might be necessary to override the default port mapping information for a specific host or subnet.

With host-specific port mapping, you can use the same port number for different services on different hosts. This means that you can map port 8000 with HTTP services for one host, while mapping port 8000 with Telnet services for another host.

Host-specific port mapping also allows you to apply PAM to a specific subnet when that subnet runs a service that uses a port number that is different from the port number defined in the default mapping information. For example, hosts on subnet 192.168.21.0 might run HTTP services on non-standard port 8000, while other traffic through the firewall uses the default port for HTTP services, which is port 80.

Host-specific port mapping allows you to override a system-defined entry in the PAM table. For example, if CBAC finds an entry in the PAM table that maps port 25 (the system-defined port for SMTP) with HTTP for a specific host, CBAC identifies port 25 as HTTP protocol traffic on that host.

Note If the host-specific port mapping information is the same as an existing system-defined or user-defined default entries, host-specific port changes have no effect.

PAM and CBAC

CBAC uses the information in the PAM table to identify a service or application from traffic flowing through the firewall. With PAM, CBAC can associate non-standard port numbers with specific protocols. For example, if you use PAM to map port 8000 with HTTP services, CBAC can determine that traffic using port 8000 is an HTTP application.

When to Use PAM

Here are a few examples of when you might want to use PAM:

- Use PAM to apply a non-standard port numbers for a service or application.
- Use PAM when a specific host or subnet uses a port number for an application that is different than the default port number established in the PAM table.
- Use PAM when different hosts use the same port number for different applications.

PAM Configuration Task List

See the following sections for PAM configuration tasks. Each task in the list indicates if it is optional or required:

- Configuring Standard ACLs (Optional)
- Configuring PAM (Required)
- Verifying PAM (Optional)

Configuring Standard ACLs

If you require PAM for a specific host or subnet, use the **access-list** (standard) command in global configuration mode to define an ACL:

Command	Purpose
`Router(config)# access-list access-list-number permit source [source-wildcard]`	(Optional) Creates a standard ACL that defines the specific host or subnet for host-specific PAM. For complete information on access-list command, refer to the *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services*.

Configuring PAM

To configure PAM, use the **ip port-map** command in global configuration mode:

Command	Purpose
`Router(config)# ip port-map appl_name port port_num [list acl_num]`	Establishes a port mapping entry using the TCP or UDP port number and the application name. (Optional) Use the list option to associate this port mapping to the specific hosts in the ACL. (PAM uses standard access lists only.) If an access list is included, the hosts defined in that ACL have the application *appl_name* running on port *port_num*.

Verifying PAM

To verify the port mapping information, enter the **show ip port-map** command in privileged EXEC mode and review the entries:

```
Router# show ip port-map
```

This command displays all entries in the PAM table, including the system-defined entries.

For PAM configuration examples using the commands in this chapter, refer to the "PAM Configuration Examples" section at the end of this chapter.

Monitoring and Maintaining PAM

The following commands can be used to monitor and maintain PAM:

Command	Purpose
`Router# show ip port-map [appl_name | port port_num]`	Displays the port mapping information, including the system-defined entries. Include the application name to display a list of entries by application. Include the port number to display the entries by port.
`Router(config)# no ip port-map appl_name port port_num [list acl_num]`	Deletes user-defined port mapping information. This command has no effect on the system-defined port mapping information.

PAM Configuration Examples

The following sections provide PAM configuration examples:

- Mapping an Application to a Non-Standard Port Example
- Mapping an Application with a Port Range Example
- Invalid Port Mapping Entry Example
- Mapping an Application to a Port for a Specific Host Example
- Mapping an Application to a Port for a Subnet Example
- Overriding a System-Defined Port Mapping Example
- Mapping Different Applications to the Same Port Example

Mapping an Application to a Non-Standard Port Example

In this example, non-standard port 8000 is established as the user-defined default port mapping for HTTP services:

```
ip port-map http port 8000
```

Mapping an Application with a Port Range Example

The following PAM entries establish a range of non-standard ports for HTTP services:

```
ip port-map http 8001
ip port-map http 8002
ip port-map http 8003
ip port-map http 8004
```

Invalid Port Mapping Entry Example

This example is not valid because it tries to establish port 21, which is the system-defined default port for FTP, as the user-defined port for HTTP services:

```
ip port-map http port 21
```

Mapping an Application to a Port for a Specific Host Example

In this example, a specific host uses port 8000 for FTP services. ACL 10 identifies the server address (192.168.32.43), while port 8000 is mapped with FTP services.

```
access-list 10 permit 192.168.32.43
ip port-map ftp port 8000 list 10
```

Mapping an Application to a Port for a Subnet Example

In this example, a specific subnet runs HTTP services on port 8080. ACL 50 identifies the subnet, while port 8080 is mapped with HTTP services.

```
access-list 50 permit 192.168.92.0 0.0.0.255
ip port-map http 8080 list 50
```

Overriding a System-Defined Port Mapping Example

In this example, a specific host runs HTTP services on port 25, which is the system-defined port number for SMTP services. This requires a host-specific PAM entry that overrides the system-defined default port mapping for HTTP, which is port 80. ACL 15 identifies the host address (192.168.33.33), while port 25 is mapped with HTTP services.

```
access-list 15 permit 192.168.33.33
ip port-map http port 25 list 15
```

Mapping Different Applications to the Same Port Example

In this example, the same port number is required by different services running on different hosts. Port 8000 is required for HTTP services for host 192.168.3.4, while port 8000 is also required for FTP services for host 192.168.5.6. ACL 10 and ACL 20 identify the specific hosts, while the PAM entries map the ports with the services for each ACL.

```
access-list 10 permit 192.168.3.4
access-list 20 permit 192.168.5.6
ip port-map http port 8000 list 10
ip port-map http ftp 8000 list 20
```

coincidental.

IPSec and IKE

Internet Key Exchange for IPSec VPNs

Configuring Internet Key Exchange for IPSec VPNs

This module describes how to configure the Internet Key Exchange (IKE) protocol for basic IP Security (IPSec) virtual private networks (VPNs). IKE is a key management protocol standard that is used in conjunction with the IPSec standard. IPSec is an IP security feature that provides robust authentication and encryption of IP packets.

IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard.

IKE is a hybrid protocol, which implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association Key Management Protocol (ISAKMP) framework. (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.)

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for Configuring IKE for IPSec VPNs" section on page 24.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for IKE Configuration

- You should you should be familiar with the concepts and tasks explained in the module "Configuring Security for VPNs with IPSec."
- Ensure that your access control lists (ACLs) are compatible with IKE. Because IKE negotiation uses User Datagram Protocol (UDP) on port 500, your ACLs must be configured so that UDP port 500 traffic is not blocked at interfaces used by IKE and IPSec. In some cases you might need to add a statement to your ACLs to explicitly permit UDP port 500 traffic.

Restrictions for IKE Configuration

The following restrictions are applicable when configuring IKE negotiation:

- The initiating router *must not* have a certificate associated with the remote peer.
- The preshared key *must* be by fully qualified domain name (FQDN) on both peers. (To configure the preshared key, enter the **crypto isakmp key** command.)
- The communicating routers *must* have a FQDN host entry for each other in their configurations.
- The communicating routers *must* be configured to authenticate by hostname, *not* by IP address; thus, you should use the **crypto isakmp identity hostname** command.

Information About Configuring IKE for IPSec VPNs

To configure IKE for IPSec VPNs, you should understand the following concepts:

- Supported Standards for Use with IKE, page 2
- IKE Benefits, page 4
- IKE Main Mode and Aggressive Mode, page 4

Supported Standards for Use with IKE

Cisco implements the following standards:

- IPSec—IP Security Protocol. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer; it uses IKE to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.
- ISAKMP—Internet Security Association and Key Management Protocol. A protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.
- Oakley—A key exchange protocol that defines how to derive authenticated keying material.
- Skeme—A key exchange protocol that defines how to derive authenticated keying material, with rapid key refreshment.

The component technologies implemented for use by IKE include the following:

- AES—Advanced Encryption Standard. A cryptographic algorithm that protects sensitive, unclassified information. AES is privacy transform for IPSec and IKE and has been developed to replace the DES. AES is designed to be more secure than DES: AES offers a larger key size, while ensuring that the only known approach to decrypt a message is for an intruder to try every possible key. AES has a variable key length—the algorithm can specify a 128-bit key (the default), a 192-bit key, or a 256-bit key.
- DES—Data Encryption Standard. An algorithm that is used to encrypt packet data. IKE implements the 56-bit DES-CBC with Explicit IV standard. Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPSec packet.

  Cisco IOS software also implements Triple DES (168-bit) encryption, depending on the software versions available for a specific platform. Triple DES (3DES) is a strong form of encryption that allows sensitive information to be transmitted over untrusted networks. It enables customers, particularly in the finance industry, to utilize network-layer encryption.

  Note Cisco IOS images that have strong encryption (including, but not limited to, 56-bit data encryption feature sets) are subject to United States government export controls, and have a limited distribution. Images that are to be installed outside the United States require an export license. Customer orders might be denied or subject to delay because of United States government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- SEAL—Software Encryption Algorithm. An alternative algorithm to software-based DES, 3DES, and AES. SEAL encryption uses a 160-bit encryption key and has a lower impact to the CPU when compared to other software-based algorithms.
- Diffie-Hellman—A public-key cryptography protocol that allows two parties to establish a shared secret over an unsecure communications channel. Diffie-Hellman is used within IKE to establish session keys. 768-bit (the default), 1024-bit, and 1536-bit Diffie-Hellman groups are supported.
- MD5 (HMAC variant)—Message Digest 5. A hash algorithm used to authenticate packet data. HMAC is a variant that provides an additional level of hashing.
- SHA (HMAC variant)—Secure Hash Algorithm. A hash algorithm used to authenticate packet data. HMAC is a variant that provides an additional level of hashing.
- RSA signatures and RSA encrypted nonces—RSA is the public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. RSA signatures provide nonrepudiation, and RSA encrypted nonces provide repudiation. (Repudation and nonrepudation have to do with traceability.)

IKE interoperates with the following standard:

X.509v3 certificates—Used with the IKE protocol when authentication requires public keys. This certificate support allows the protected network to scale by providing the equivalent of a digital ID card to each device. When two devices intend to communicate, they exchange digital certificates to prove their identity (thus removing the need to manually exchange public keys with each peer or to manually specify a shared key at each peer).

IKE Benefits

IKE automatically negotiates IPSec security associations (SAs) and enables IPSec secure communications without costly manual preconfiguration. Specifically, IKE provides the following benefits:

- Eliminates the need to manually specify all the IPSec security parameters in the crypto maps at both peers.
- Allows you to specify a lifetime for the IPSec SA.
- Allows encryption keys to change during IPSec sessions.
- Allows IPSec to provide anti-replay services.
- Permits certification authority (CA) support for a manageable, scalable IPSec implementation.
- Allows dynamic authentication of peers.

IKE Main Mode and Aggressive Mode

IKE has two phases of key negotiation: phase 1 and phase 2. Phase 1 negotiates a security association (a key) between two IKE peers. The key negotiated in phase 1 enables IKE peers to communicate securely in phase 2. During phase 2 negotiation, IKE establishes keys (security associations) for other applications, such as IPSec.

Phase 1 negotiation can occur using main mode or aggressive mode. Main mode tries to protect all information during the negotiation, meaning that no information is available to a potential attacker. When main mode is used, the identities of the two IKE peers are hidden. Although this mode of operation is very secure, it is relatively costly in terms of the time it takes to complete the negotiation. Aggressive mode takes less time to negotiate keys between peers; however, it gives up some of the security provided by main mode negotiation. For example, the identities of the two parties trying to establish a security association are exposed to an eavesdropper.

The two modes serve different purposes and have different strengths. Main mode is slower than aggressive mode, but main mode is more secure and more flexible because it can offer an IKE peer more security proposals than aggressive mode. Aggressive mode is less flexible and not as secure, but much faster.

In Cisco IOS software, the two modes are not configurable. The default action for IKE authentication (rsa-sig, rsa-encr, or preshared) is to initiate main mode; however, in cases where there is no corresponding information to initiate authentication, and there is a preshared key associated with the hostname of the peer, Cisco IOS software can initiate aggressive mode. Cisco IOS software will respond in aggressive mode to an IKE peer that initiates aggressive mode.

How to Configure IKE for IPSec VPNs

If you do not want IKE to be used with your IPSec implementation, you can disable it at all IPSec peers via the **no crypto isakmp** command, skip the rest of this chapter, and begin your IPSec VPN.

Note If you disable IKE, you will have to manually specify all the IPSec SAs in the crypto maps at all peers, the IPSec SAs of the peers will never time out for a given IPSec session, the encryption keys will never change during IPSec sessions between the peers, anti-replay services will not be available between the peers, and public key infrastructure (PKI) support cannot be used.

IKE is enabled by default. IKE does not have to be enabled for individual interfaces, but it is enabled globally for all interfaces at the router.

Perform the following tasks to provide authentication of IPSec peers, negotiate IPSec SAs, and establish IPSec keys:

- Creating IKE Policies: Security Parameters for IKE Negotiation, page 5 (required)
- Configuring IKE Authentication, page 9 (required)
- Configuring IKE Mode Configuration, page 17

Creating IKE Policies: Security Parameters for IKE Negotiation

An IKE policy defines a combination of security parameters to be used during the IKE negotiation. You must create an IKE policy at each peer participating in the IKE exchange.

If you do not configure any IKE policies, your router will use the default policy, which is always set to the lowest priority and which contains the default value of each parameter.

About IKE Policies

Because IKE negotiations must be protected, each IKE negotiation begins by agreement of both peers on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how the peers are authenticated.

After the two peers agree upon a policy, the security parameters of the policy are identified by an SA established at each peer, and these SAs apply to all subsequent IKE traffic during the negotiation.

You can configure multiple, prioritized policies on each peer—each with a different combination of parameter values. However, at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer. For each policy that you create, you assign a unique priority (1 through 10,000, with 1 being the highest priority).

Tip If you are interoperating with a device that supports only one of the values for a parameter, your choice is limited to the value supported by the other device. Aside from this limitation, there is often a trade-off between security and performance, and many of these parameter values represent such a trade-off. You should evaluate the level of security risks for your network and your tolerance for these risks.

IKE Peers Agreeing Upon a Matching IKE Policy

When the IKE negotiation begins, IKE searches for an IKE policy that is the same on both peers. The peer that initiates the negotiation will send all its policies to the remote peer, and the remote peer will try to find a match. The remote peer looks for a match by comparing its own highest priority policy against the policies received from the other peer. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

A match is made when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values, and when the remote peer's policy specifies a lifetime that is less than or equal to the lifetime in the policy being compared. (If the lifetimes are not identical, the shorter lifetime—from the remote peer's policy—will be used.)

If a match is found, IKE will complete negotiation, and IPSec security associations will be created. If no acceptable match is found, IKE refuses negotiation and IPSec will not be established.

Note Depending on which authentication method is specified in a policy, additional configuration might be required (as described in the section "Configuring IKE Authentication"). If a peer's policy does not have the required companion configuration, the peer will not submit the policy when attempting to find a matching policy with the remote peer.

Restrictions

If you are configuring an AES IKE policy, note the following restrictions:

- Your router must support IPSec and long keys (the "k9" subsystem).
- AES cannot encrypt IPSec and IKE traffic if an acceleration card is present.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp policy** *priority*
4. **encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**}
5. **hash** {**sha** | **md5**}
6. **authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
7. **group** {**1** | **2** | **5**}
8. **lifetime** *seconds*
9. **exit**
10. **exit**
11. **show crypto isakmp policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 10`	Defines an IKE policy and enters config-isakmp configuration mode. • *priority*—Uniquely identifies the IKE policy and assigns a priority to the policy. Valid values: 1 to 10,000; 1 is the highest priority.
Step 4	**encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**} **Example:** `Router(config-isakmp)# encryption aes 256`	Specifies the encryption algorithm. By default, the **des** keyword is used. • **des**—56-bit DES-CBC • **3des**—168-bit DES • **aes**—128-bit AES • **aes 192**—192-bit AES • **aes 256**—256-bit AES
Step 5	**hash** {**sha** | **md5**} **Example:** `Router(config-isakmp)# hash sha`	Specifies the hash algorithm. By default, SHA-1 (**sha**) is the used. **Note** MD5 has a smaller digest and is considered to be slightly faster than SHA-1.
Step 6	**authentication** {**rsa-sig** | **rsa-encr** | **pre-share**} **Example:** `Router(config-isakmp)# authentication pre-share`	Specifies the authentication method. By default, RSA signatures are used. • **rsa-sig**—RSA signatures require that you configure your peer routers to obtain certificates from a CA. • **rsa-encr**—RSA encrypted nonces require that you ensure each peer has the other peer's RSA public keys. • **pre-share**—Preshared keys require that you separately configure these preshared keys.

	Command or Action	Purpose
Step 7	**group** {**1** | **2** | **5**} **Example:** `Router(config-isakmp)# group 1`	Specifies the Diffie-Hellman group identifier. By default, D-H group 1 is used. • **1**—768-bit Diffie-Hellman • **2**—1024-bit Diffie-Hellman • **5**—1536-bit Diffie-Hellman **Note** The 1024-bit and 1536-bit Diffie-Hellman options are harder to "crack," but require more CPU time to execute.
Step 8	**lifetime** *seconds* **Example:** `Router(config-isakmp)# lifetime 180`	Specifies the lifetime of the IKE SA. • *seconds*—Time, in seconds, before each SA expires. Valid values: 60 to 86,400 seconds; default value: 86,400. **Note** The shorter the lifetime (up to a point), the more secure your IKE negotiations will be. However, with longer lifetimes, future IPSec SAs can be set up more quickly.
Step 9	**exit** **Example:** `Router(config-isakmp)# exit`	Exits config-isakmp configuration mode.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits the global configuration mode.
Step 11	**show crypto isakmp policy** **Example:** `Router# show crypto isakmp policy`	(Optional) Displays all existing IKE policies.
Step 12	—	Repeat these steps for each policy you want to create.

Note These parameters apply to the IKE negotiations after the IKE SA is established.

Examples

The following sample output from the **show crypto isakmp policy** command displays a warning message after a user tries to configure an IKE encryption method that the hardware does not support:

```
Router# show crypto isakmp policy

Protection suite of priority 1
        encryption algorithm:  AES - Advanced Encryption Standard (256 bit keys).
WARNING:encryption hardware does not support the configured
encryption method for ISAKMP policy 1
        hash algorithm:        Secure Hash Standard
        authentication method: Pre-Shared Key
```

```
Diffie-Hellman group:  #1 (768 bit)
lifetime:              3600 seconds, no volume limit
```

Troubleshooting Tips

- Clear (and reinitialize) IPSec SAs by using the **clear crypto sa** EXEC command.

  Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information,see the **clear crypto sa** command in the *Cisco IOS Security Command Reference*, Release 12.4.

- The default policy and default values for configured policies do not show up in the configuration when you issue the **show running-config** command. To see the default policy and any default values within configured policies, use the **show crypto isakmp policy** command.
- Any IPSec transforms or IKE encryption methods that the current hardware does not support should be disabled; they are ignored whenever an attempt to negotiate with the peer is made.

  If a user enters an IPSec transform or an IKE encryption method that the hardware does not support, a warning message will be generated. These warning messages are also generated at boot time. When an encrypted card is inserted, the current configuration is scanned. If any IPSec transforms or IKE encryption methods are found that are not supported by the hardware, a warning message will be generated.

What to Do Next

Depending on which authentication method you specified in your IKE policies (RSA signatures, RSA encrypted nonces, or preshared keys), you must do certain additional configuration tasks before IKE and IPSec can successfully use the IKE policies. For information on completing these additional tasks, refer to the following section "Configuring IKE Authentication."

To configure an AES-based transform set, see the module "Configuring Security for VPNs with IPSec."

Configuring IKE Authentication

After you have created at least one IKE policy in which you specified an authentication method (or accepted the default method), you need to configure an authentication method. IKE policies cannot be used by IPSec until the authentication method is successfully configured.

To configure IKE authentication, you should perform one of the following tasks, as appropriate:

- Configuring RSA Keys Manually for RSA Encrypted Nonces, page 11
- Configuring Preshared Keys, page 13
- Configuring RSA Keys to Obtain Certificates from a CA. For information on completing this task, see the module "Deploying RSA Keys Within a PKI."

IKE Authentication Methods: Overview

IKE authentication consists of three options—RSA signatures, RSA encrypted nonces, and preshared keys. Each authentication method requires additional configuration as follows:

RSA Signatures

With RSA signatures, you can configure the peers to obtain certificates from a CA. (The CA must be properly configured to issue the certificates.) Using a CA can dramatically improve the manageability and scalability of your IPSec network. Additionally, RSA signature-based authentication uses only two public key operations, whereas RAS encryption uses four public key operations, making it costlier in terms of overall performance. To properly configure CA support, see the chapter "Implementing and Managing a PKI."

The certificates are used by each peer to exchange public keys securely. (RSA signatures requires that each peer has the public signature key of the remote peer.) When both peers have valid certificates, they will automatically exchange public keys with each other as part of any IKE negotiation in which RSA signatures are used.

You can also exchange the public keys manually, as described in the section "Configuring RSA Keys Manually for RSA Encrypted Nonces."

RSA signatures provide nonrepudiation for the IKE negotiation. And, you can prove to a third party after the fact that you did indeed have an IKE negotiation with the remote peer.

RSA Encrypted Nonces

With RSA encrypted nonces, you must ensure that each peer has the public keys of the other peers.

Unlike RSA signatures, the RSA encrypted nonces method can not use certificates to exchange public keys. Instead, you ensure that each peer has the others' public keys by one of the following methods:

- Manually configuring RSA keys as described in the section "Configuring RSA Keys Manually for RSA Encrypted Nonces."

  or

- Ensuring that an IKE exchange using RSA signatures with certificates has already occurred between the peers. (The peers' public keys are exchanged during the RSA-signatures-based IKE negotiations if certificates are used.)

  To make that the IKE exchange happens, specify two policies: a higher-priority policy with RSA encrypted nonces and a lower-priority policy with RSA signatures. When IKE negotiations occur, RSA signatures will be used the first time because the peers do not yet have each other's public keys. Then future IKE negotiations can use RSA encrypted nonces because the public keys will have been exchanged.

Note This alternative requires that you already have CA support configured.

RSA encrypted nonces provide repudiation for the IKE negotiation; however, unlike RSA signatures, you cannot prove to a third party that you had an IKE negotiation with the remote peer.

Preshared Keys

With preshared keys, you must configure them as described in the section "Configuring Preshared Keys."

Preshared keys are clumsy to use if your secured network is large, and they do not scale well with a growing network. However, they do not require use of a CA, as do RSA signatures, and might be easier to set up in a small network with fewer than ten nodes. RSA signatures also can be considered more secure when compared with preshared key authentication.

Note If RSA encryption is configured and signature mode is negotiated (and certificates are used for signature mode), the peer will request both signature and encryption keys. Basically, the router will request as many keys as the configuration will support. If RSA encryption is not configured, it will just request a signature key.

Prerequisites

You must have configured at least one IKE policy, which is where the authentication method was specified (or RSA signatures was accepted by default).

Configuring RSA Keys Manually for RSA Encrypted Nonces

To manually configure RSA keys, perform this task for each IPSec peer that uses RSA encrypted nonces in an IKE policy.

Note This task can be performed only if a CA is not in use.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** {**general-keys** | **usage-keys**} [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*]
4. **exit**
5. **show crypto key mypubkey rsa**
6. configure terminal
7. **crypto key pubkey-chain rsa**
8. named-key *key-name* [encryption | signature]

   or

   addressed-key *key-address* [**encryption** | **signature**]
9. address *ip-address*
10. key-string *key-string*
11. quit
12. Repeat these steps at each peer that uses RSA encrypted nonces in an IKE policy.
13. exit
14. exit
15. **show crypto key pubkey-chain rsa** [**name** *key-name* | **address** *key-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa** {**general-keys** | **usage-keys**} [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] **Example:** `Router(config)# crypto key generate rsa general-keys modulus 360`	Generates RSA keys. • If a *key-label* argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the router, is used.
Step 4	**exit** **Example:** `Router(config)# exit`	(Optional) Exits global configuration mode.
Step 5	`show crypto key mypubkey rsa` **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the generated RSA public keys.
Step 6	**configure terminal** **Example:** `Router# configure terminal`	Returns to global configuration mode.
Step 7	`crypto key pubkey-chain rsa` **Example:** `Router(config)# crypto key pubkey-chain rsa`	Enters public key configuration mode (so you can manually specify the RSA public keys of other devices).
Step 8	**named-key** *key-name* [**encryption** | **signature**] **Example:** `Router(config-pubkey-chain)# named-key otherpeer.example.com` or **addressed-key** *key-address* [encryption | signature] **Example:** `Router(config-pubkey-chain)# addressed-key 10.1.1.2 encryption`	Indicates which remote peer's RSA public key you are going to specify and enters public key configuration mode. If the remote peer uses its host name as its ISAKMP identity, use the **named-key** command and specify the remote peer's FQDN, such as somerouter.example.com, as the *key-name*. If the remote peer uses its IP address as its ISAKMP identity, use the **addressed-key** command and specify the remote peer's IP address as the *key-address*.

	Command or Action	Purpose
Step 9	**address** *ip-address* **Example:** `Router(config-pubkey-key)# address 10.5.5.1`	Specifies the IP address of the remote peer. If you use the **named-key** command, you need to use this command to specify the IP address of the peer.
Step 10	`key-string` *key-string* **Example:** `Router(config-pubkey-key)# key-string` `Router(config-pubkey)# 00302017 4A7D385B 1234EF29 335FC973` `Router(config-pubkey)# 2DD50A37 C4F4B0FD 9DADE748 429618D5` `Router(config-pubkey)# 18242BA3 2EDFBDD3 4296142A DDF7D3D8` `Router(config-pubkey)# 08407685 2F2190A0 0B43F1BD 9A8A26DB` `Router(config-pubkey)# 07953829 791FCDE9 A98420F0 6A82045B` `Router(config-pubkey)# 90288A26 DBC64468 7789F76E EE21`	Specifies the RSA public key of the remote peer. (This key was previously viewed by the administrator of the remote peer when the RSA keys of the remote router were generated.)
Step 11	`quit` **Example:** `Router(config-pubkey-k)# quit`	Returns to public key chain configuration mode.
Step 12	—	Repeat these steps at each peer that uses RSA encrypted nonces in an IKE policy.
Step 13	`exit` **Example:** `Router(config-pubkey-c)# exit`	Returns to global configuration mode.
Step 14	`exit` **Example:** `Router(config)# exit`	Returns to EXEC mode.
Step 15	`show crypto key pubkey-chain rsa [name` *key-name* `| address` *key-address*`]` **Example:** `Router# show crypto key pubkey-chain rsa`	(Optional) Displays either a list of all RSA public keys that are stored on your router or details of a particular RSA key that is stored on your router.

Configuring Preshared Keys

To configure preshared keys, perform these steps at each peer that uses preshared keys in an IKE policy.

Setting ISAKMP Identity for Preshared Keys

You should set the ISAKMP identity for each peer that uses preshared keys in an IKE policy.

When two peers use IKE to establish IPSec SAs, each peer sends its identity to the remote peer. Each peer sends either its host name or its IP address, depending on how you have set the ISAKMP identity of the router.

By default, a peer's ISAKMP identity is the IP address of the peer. If appropriate, you could change the identity to be the peer's host name instead. As a general rule, set the identities of all peers the same way—either all peers should use their IP addresses or all peers should use their hostnames. If some peers use their host names and some peers use their IP addresses to identify themselves to each other, IKE negotiations could fail if the identity of a remote peer is not recognized and a DNS lookup is unable to resolve the identity.

Mask Preshared Keys

A mask preshared key allows a group of remote users with the same level of authentication to share an IKE preshared key. The preshared key of the remote peer must match the preshared key of the local peer for IKE authentication to occur.

A mask preshared key is usually distributed through a secure out-of-band channel. In a remote peer-to-local peer scenario, any remote peer with the IKE preshared key configured can establish IKE SAs with the local peer.

If you specify the **mask** keyword with the **crypto isakmp key** command, it is up to you to use a subnet address, which will allow more peers to share the same key. That is, the preshared key is no longer restricted to use between two users.

Note Using 0.0.0.0 as a subnet address is not recommended because it encourages group preshared keys, which allow all peers to have the same group key, thereby reducing the security of your user authentication.

Disable Xauth on a Specific IPSec Peer

Disabling Extended Authentication (Xauth) for static IPSec peers prevents the routers from being prompted for Xauth information—username and password.

Without the ability to disable Xauth, a user cannot select which peer on the same crypto map should use Xauth. That is, if a user has router-to-router IPSec on the same crypto map as a VPN-client-to-Cisco-IOS IPSec, both peers are prompted for a username and password. In addition, a remote static peer (a Cisco IOS router) cannot establish an IKE SA with the local Cisco IOS router. (Xauth is not an optional exchange, so if a peer does not respond to an Xauth request, the IKE SA is deleted.) Thus, the same interface cannot be used to terminate IPSec to VPN clients (that need Xauth) as well as other Cisco IOS routers (that cannot respond to Xauth) unless this feature is implemented.

Note Xauth can be disabled only if preshared keys are used as the authentication mechanism for the given crypto map.

Restrictions

- Preshared do not scale well with a growing network.
- Mask preshared keys have the following restrictions:
  - The SA cannot be established between the IPSec peers until all IPSec peers are configured for the same preshared key.

- The mask preshared key must be distinctly different for remote users requiring varying levels of authorization. You must configure a new preshared key for each level of trust and assign the correct keys to the correct parties. Otherwise, an untrusted party may obtain access to protected data.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp identity** {**address** | **hostname**}
4. **ip host** *hostname address1* [*address2...address8*]
5. **crypto isakmp key** *keystring* **address** *peer-address* [**mask**] [**no-xauth**]

   or

   crypto isakmp key *keystring* **hostname** *hostname* [**no-xauth**]
6. **crypto isakmp key** *keystring* **address** *peer-address* [**mask**] [**no-xauth**]

   or

   crypto isakmp key *keystring* **hostname** *hostname* [**no-xauth**]
7. Repeat these steps for each peer that uses preshared keys.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp identity** {**address** | **hostname**} **Example:** `Router(config)# crypto isakmp identity address`	Specifies the peer's ISAKMP identity by IP address or by hostname at the local peer. • **address**—Typically used when there is only one interface (and therefore only one IP address) that will be used by the peer for IKE negotiations, and the IP address is known. • **hostname**—Should be used if there is more than one interface on the peer that might be used for IKE negotiations, or if the interface's IP address is unknown (such as with dynamically assigned IP addresses).
Step 4	**ip host** *hostname address1* [*address2...address8*] **Example:** `Router(config)# ip host RemoteRouter.example.com 192.168.0.1`	If the local peer's ISAKMP identity was specified using a hostname, maps the peer's host name to its IP address(es) at all the remote peers. (This step might be unnecessary if the hostname or address is already mapped in a DNS server.)
Step 5	**crypto isakmp key** *keystring* **address** *peer-address* [**mask**] [**no-xauth**] **Example:** `Router(config)# crypto isakmp key sharedkeystring address 192.168.1.33 no-xauth` or **crypto isakmp key** *keystring* **hostname** *hostname* [**no-xauth**] **Example:** `Router(config) crypto isakmp key sharedkeystring hostname RemoteRouter.example.com`	Specifies at the local peer the shared key to be used with a particular remote peer. If the remote peer specified its ISAKMP identity with an address, use the **address** keyword in this step; otherwise use the **hostname** keyword in this step. • **no-xauth—**Prevents the router from prompting the peer for Xauth information. Use this keyword if router-to-router IPSec is on the same crypto map as VPN-client-to-Cisco IOS IPSec. **Note** According to the design of preshared key authentication in IKE main mode, preshared keys must be based on the IP address of the peers. Although you can send hostname as the identity of preshared key authentication, the key is searched on the IP address of the peer; if the key is not found (based on the IP address) the negotiation will fail.

	Command or Action	Purpose
Step 6	**crypto isakmp key** *keystring* **address** *peer-address* [**mask**] [**no-xauth**] **Example:** `Router(config) crypto isakmp key sharedkeystring address 10.0.0.1` or **crypto isakmp key** *keystring* **hostname** *hostname* [**no-xauth**] **Example:** `Router(config) crypto isakmp key sharedkeystring hostname LocalRouter.example.com`	Specifies at the remote peer the shared key to be used with the local peer. This is the same key you just specified at the local peer. If the local peer specified its ISAKMP identity with an address, use the **address** keyword in this step; otherwise use the **hostname** keyword in this step.
Step 7	—	Repeat these steps at each peer that uses preshared keys in an IKE policy.

Configuring IKE Mode Configuration

Perform the following task to configure IKE mode configuration.

About IKE Mode Configuration

IKE mode configuration, as defined by the Internet Engineering Task Force (IETF) , allows a gateway to download an IP address (and other network level configuration) to the client as part of an IKE negotiation. Using this exchange, the gateway gives IP addresses to the IKE client to be used as an “inner” IP address encapsulated under IPSec. This method provides a known IP address for the client that can be matched against IPSec policy.

To implement IPSec VPNs between remote access clients that have dynamic IP addresses and a corporate gateway, you have to dynamically administer scalable IPSec policy on the gateway once each client is authenticated. With IKE Mode Configuration, the gateway can set up scalable policy for a very large set of clients irrespective of the IP addresses of those clients.

There are two types of IKE Mode Configuration:

- Gateway initiation—Gateway initiates the configuration mode with the client. Once the client responds, the IKE modifies the identity of the sender, the message is processed, and the client receives a response.
- Client initiation—Client initiates the configuration mode with the gateway. The gateway responds with an IP address that it has allocated for the client.

Restrictions

IKE Mode Configuration has the following restrictions:

- Interfaces with crypto maps that are configured for IKE Mode Configuration may experience a slightly longer connection setup time, which is true even for IKE peers that refuse to be configured or do not respond to the configuration mode request. In both cases, the gateway initiates the configuration of the client.
- This feature was not designed to enable the configuration mode for every IKE connection by default. Configure this feature at the global crypto map level.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip local pool** *pool-name start-addr end-addr*
4. **crypto isakmp client configuration address-pool local** *pool-name*
5. **crypto map** *tag* **client configuration address** [**initiate** | **respond**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip local pool** *pool-name start-addr end-addr* **Example:** `Router(config) ip local pool ire 172.16.23.0 172.16.23.255`	Defines an existing local address pool that defines a set of addresses.
Step 4	**crypto isakmp client configuration address-pool local** *pool-name* **Example:** `Router(config) crypto isakmp client configuration address-pool local ire`	References the local address pool in the IKE configuration.
Step 5	**crypto map** *tag* **client configuration address** [**initiate** | **respond**] **Example:** `Router(config)# crypto map dyn client configuration address initiate`	Configures IKE Mode Configuration in global crypto map configuration mode.

Configuration Examples for an IKE Configuration

This section contains the following configuration examples:

- Creating IKE Policies: Examples, page 19
- Configuring IKE Authentication: Example, page 21

Creating IKE Policies: Examples

This section contains the following examples, which show how to configure a 3DES IKE policy and an AES IKE policy:

- Creating 3DES IKE Policies: Example, page 20
- Creating an AES IKE Policy: Example, page 20

Creating 3DES IKE Policies: Example

This example creates two IKE policies, with policy 15 as the highest priority, policy 20 as the next priority, and the existing default priority as the lowest priority. It also creates a preshared key to be used with policy 20 with the remote peer whose IP address is 192.168.224.33.

```
crypto isakmp policy 15
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
!
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
!
crypto isakmp key 1234567890 address 192.168.224.33
```

In the example, the encryption des of policy 15 would not appear in the written configuration because this is the default value for the encryption algorithm parameter.

If the **show crypto isakmp policy** command is issued with this configuration, the output is as follows:

```
Protection suite priority 15
encryption algorithm:3DES - Triple Data Encryption Standard (168 bit keys)
hash algorithm:Message Digest 5
authentication method:Rivest-Shamir-Adleman Signature
Diffie-Hellman group:#2 (1024 bit)
lifetime:5000 seconds, no volume limit
Protection suite priority 20
encryption algorithm:DES - Data Encryption Standard (56 bit keys)
hash algorithm:Secure Hash Standard
authentication method:preshared Key
Diffie-Hellman group:#1 (768 bit)
lifetime:10000 seconds, no volume limit
Default protection suite
encryption algorithm:DES - Data Encryption Standard (56 bit keys)
hash algorithm:Secure Hash Standard
authentication method:Rivest-Shamir-Adleman Signature
Diffie-Hellman group:#1 (768 bit)
lifetime:86400 seconds, no volume limit
```

Note that although the output shows “no volume limit” for the lifetimes, you can configure only a time lifetime (such as 86,400 seconds); volume-limit lifetimes are not configurable.

Creating an AES IKE Policy: Example

The following example is sample output from the **show running-config** command. In this example, the AES 256-bit key is enabled.

```
Current configuration : 1665 bytes
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname "Router1"
!
!
ip subnet-zero
!
```

```
!
no ip domain lookup
!
ip audit notify log
ip audit po max-events 100
!
crypto isakmp policy 10
 encryption aes 256
 authentication pre-share
 lifetime 180
crypto isakmp key cisco123 address 10.0.110.1
!
!
crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac
 mode transport
!
crypto map aesmap 10 ipsec-isakmp
 set peer 10.0.110.1
 set transform-set aesset
 match address 120
!
.
.
.
```

Configuring IKE Authentication: Example

The following example shows how to manually specify the RSA public keys of two IPSec peer— the peer at 10.5.5.1 uses general-purpose keys, and the other peer uses special-usage keys:

```
crypto key pubkey-chain rsa
 named-key otherpeer.example.com
 address 10.5.5.1
 key-string
 005C300D 06092A86 4886F70D 01010105
 00034B00 30480241 00C5E23B 55D6AB22
 04AEF1BA A54028A6 9ACC01C5 129D99E4
 64CAB820 847EDAD9 DF0B4E4C 73A05DD2
 BD62A8A9 FA603DD2 E2A8A6F8 98F76E28
 D58AD221 B583D7A4 71020301 0001
 quit
 exit
 addressed-key 10.1.1.2 encryption
 key-string
 00302017 4A7D385B 1234EF29 335FC973
 2DD50A37 C4F4B0FD 9DADE748 429618D5
 18242BA3 2EDFBDD3 4296142A DDF7D3D8
 08407685 2F2190A0 0B43F1BD 9A8A26DB
 07953829 791FCDE9 A98420F0 6A82045B
 90288A26 DBC64468 7789F76E EE21
 quit
 exit
 addressed-key 10.1.1.2 signature
 key-string
 0738BC7A 2BC3E9F0 679B00FE 53987BCC
 01030201 42DD06AF E228D24C 458AD228
 58BB5DDD F4836401 2A2D7163 219F882E
 64CE69D4 B583748A 241BED0F 6E7F2F16
 0DE0986E DF02031F 4B0B0912 F68200C4
 C625C389 0BFF3321 A2598935 C1B1
 quit
 exit
```

```
exit
```

Where to Go Next

After you have successfully configured IKE negotiation, you can begin configuring IPSec. For information on completing these tasks, see the module “Configuring Security for VPNs With IPSec.”

Additional References

The following sections provide references related to configuring IKE for IPSec VPNs.

Related Documents

Related Topic	Document Title
IPSec configuration	"Configuring Security for VPNs with IPSec" module
Configuring RSA keys to obtain certificates from a CA	"Deploying RSA Keys Within a PKI" module
IKE, IPSec, and PKI configuration commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*, Release 12.4

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2408	*Internet Security Association and Key Management Protocol (ISAKMP)*
RFC 2409	*The Internet Key Exchange (IKE)*
RFC 2412	*The OAKLEY Key Determination Protocol*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Glossary

anti-replay—Security service in which the receiver can reject old or duplicate packets to protect itself against replay attacks. IPSec provides optional anti-replay services by use of a sequence number and the use of authentication.

data authentication—Verification of the integrity and origin of the data.

Data authentication can refer either to integrity alone or to both of these concepts (although data origin authentication is dependent upon data integrity).

peer—In the context of this chapter, a "peer"is a router or other device that participates in IPSec and IKE.

PFS—perfect forward secrecy. Cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not also compromised, because subsequent keys are not derived from previous keys.

repudiation—Quality that prevents a third party from being able to prove that a communication between two other parties ever took place. Repudiation is a desirable quality if you do not want your communications to be traceable.

nonrepudiation—Quality that allows a third party to prove that a communication between two other parties took place. Nonrepudiation is desirable if you want to be able to trace your communications and prove that they occurred.

SA—security association. How two or more entities utilize security services to communicate securely.

For example, an IPSec SA defines the encryption algorithm (if used), the authentication algorithm, and the shared session key to be used during the IPSec connection. Both IPSec and IKE require and use SAs to identify the parameters of their connections. IKE can negotiate and establish its own SA. The IPSec SA is established either by IKE or by manual user configuration.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Feature Information for Configuring IKE for IPSec VPNs

Table 43 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 43 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 43 Feature Information for Configuring IKE for IPSec VPNs

Feature Name	Software Releases	Feature Configuration Information
Ability to Disable Extended Authentication for Static IPSec Peers	12.2(4)T	This feature allows a user to disable Xauth while configuring the preshared key for router-to-router IPSec. Thus, the router will not prompt the peer for a username and password, which are transmitted when Xauth occurs for VPN-client-to-Cisco-IOS IPSec. The following section provides information about this feature: • Configuring Preshared Keys The following command was modified by this feature: **crypto isakmp key**
Advanced Encryption Standard (AES)	12.2(8)T	This feature adds support for the new encryption standard AES, which is a privacy transform for IPSec and IKE and has been developed to replace DES. The following sections provide information about this feature: • Supported Standards for Use with IKE • Creating IKE Policies: Security Parameters for IKE Negotiation The following commands were modified by this feature: **crypto ipsec transform-set**, **encryption (IKE policy)**, **show crypto isakmp policy**, **show crypto ipsec transform-set**
SEAL Encryption	12.3(7)T	This feature adds support for SEAL encryption in IPSec. The following section provides information about this feature: • Supported Standards for Use with IKE The following command was modified by this feature: **crypto ipsec transform-set**
IKE Extended Authentication (Xauth)	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Wildcard Pre-Shared Key	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
IKE - Diffie-Hellman (768 Bit or 1024 Bit) PKCS #3 Support	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Table 43 Feature Information for Configuring IKE for IPSec VPNs (continued)

Feature Name	Software Releases	Feature Configuration Information
IKE Phase 1 Main Mode and Phase 1 Aggressive Mode	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
IKE - RSA Signature	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Call Admission Control for IKE

First Published: May 17, 2004
Last Updated: August 04, 2008

The Call Admission Control for IKE feature describes the application of Call Admission Control (CAC) to the Internet Key Exchange (IKE) protocol in Cisco IOS. CAC limits the number of simultaneous IKE security associations (SAs) (that is, calls to CAC) that a router can establish.

History for the Call Admission Control for IKE Feature

Release	Modification
12.3(8)T	This feature was introduced.
12.2(18)SXD1	This feature was integrated into Cisco IOS Release 12.2(18)SXD1 on the Cisco 6500 and Cisco 7600.
12.4(6)T	This feature was integrated into Cisco IOS Release 12.4(6)T. The ability to configure a limit on the number of in-negotiation IKE connections was added only to this and subsequent T-train releases.
12.2(33)SRA	This feature was integrated into Cisco IOS Release 12.2(33)SRA on the Cisco 7600. The in-negotiation IKE connection feature was not added to this SRA release.
12.2(33)SXH	This feature was integrated into Cisco IOS Release 12.2(33)SXH. The in-negotiation IKE connection feature was not added to this SXH release
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

Prerequisites for Call Admission Control for IKE

- Configure IKE on the router. Refer to the *Cisco IOS Security Configuration Guide*, Release 12.3.

Information About Call Admission Control for IKE

To configure CAC for IKE, you need to understand the following concepts:

IKE Session

There are two ways to limit the number of IKE SAs that a router can establish to or from another router:

- Configure the absolute IKE SA limit by entering the **crypto call admission limit** command. The router drops new IKE SA requests when the value has been reached.
- Configure the system resource limit by entering the **call admission limit** command. The router drops new IKE SA requests when the level of system resources that are configured in the unit of charge is being used.

For information about using these commands, see the "Command Reference" section on page 8.

CAC is applied only to new SAs (that is, when an SA does not already exist between the peers). Every effort is made to preserve existing SAs. Only new SA requests will ever be denied due to a lack of system resources or because the configured IKE SA limit has been reached.

Security Association Limit

An SA is a description of how two or more entities will utilize security services to communicate securely on behalf of a particular data flow. IKE requires and uses SAs to identify the parameters of its connections. IKE can negotiate and establish its own SA. An IKE SA is used by IKE only, and it is bidirectional. An IKE SA cannot limit IPsec.

IKE drops SA requests based on a user-configured SA limit. To configure an IKE SA limit, enter the **crypto call admission limit** command. When there is a new SA request from a peer router, IKE determines if the number of active IKE SAs plus the number of SAs being negotiated meets or exceeds the configured SA limit. If the number is greater than or equal to the limit, the new SA request is rejected and a syslog is generated. This log contains the source destination IP address of the SA request.

Limit on Number of In-negotiation IKE Connections

Effective with Cisco IOS Release 12.4(6)T, a limit on the number of in-negotiation IKE connections can be configured. This type of IKE connection represents either an aggressive mode IKE SA or a main mode IKE SA prior to its authentication and actual establishment.

Using the **crypto call admission limit ike in-negotiation-sa** {*number*} command allows the configured number of in-negotiation IKE SAs to start negotition without contributing to the maximum number of IKE SAs allowed.

System Resource Usage

CAC polls a global resource monitor so that IKE knows when the router is running short of CPU cycles or memory buffers. You can configure a resource limit, from 1 to 100000, that represents the level of system resources that are configured in the unit of charge. When that level of resources is being used, IKE drops (will not accept new) SA requests. To configure the system resource usage, enter the **call admission control** command.

How to Configure Call Admission Control for IKE

This section contains the following procedures:

- Configure the IKE Security Association Limit, page 3 (optional)
- Configure the System Resource Limit, page 4 (optional)

Note You must perform one of the procedures.

Configure the IKE Security Association Limit

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto call admission limit** {**ike** {**in-negotiation-sa** *number* | **sa** *number* }}
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto call admission limit** {**ike** {**in-negotiation-sa** *number* | **sa** *number*}} **Example:** `Router(config)# crypto call admission limit ike sa 25`	Specifies the maximum number of IKE SAs that the router can establish before IKE begins rejecting new SA requests. **Note** An ISAKMP connection needs to be built in two directions. If you have 500 spokes in your network, you should set this value at a minimum of 1000 (500 x 2).
Step 4	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.

Configure the System Resource Limit

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call admission limit** *charge*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call admission limit** *charge* **Example:** `Router(config)# call admission limit 90000`	Sets the level of the system resources that, when used, causes IKE to stop accepting new SA requests. • *charge*—Valid values are 1 to 100000.
Step 4	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.

Verifying the Call Admission Control for IKE Configuration

To verify the CAC for IKE configuration, perform the following steps.

SUMMARY STEPS

1. **show call admission statistics**
2. **show crypto call admission statistics**

DETAILED STEPS

Note For detailed field descriptions of the command output, see the "Command Reference" section on page 8.

Step 1 **show call admission statistics**

Use this command to monitor the global CAC configuration parameters and the behavior of CAC.

```
Router# show call admission statistics

Total Call admission charges: 0, limit 25
Total calls rejected 12, accepted 51
Load metric: charge 0, unscaled 0
```

Step 2 **show crypto call admission statistics**

Use this command to monitor Crypto CAC statistics.

```
Router# show crypto call admission statistics

----------------------------------------------------------
               Crypto Call Admission Control Statistics
----------------------------------------------------------
System Resource Limit: 0    Max IKE SAs 0
Total IKE SA Count:    0    active:     0   negotiating: 0
Incoming IKE Requests: 0    accepted:   0   rejected:    0
Outgoing IKE Requests: 0    accepted:   0   rejected:    0
Rejected IKE Requests: 0    rsrc low:   0   SA limit:    0
```

Configuration Examples for Call Admission Control for IKE

This section provides the following configuration examples:

- Configuring the IKE Security Association Limit: Example, page 6
- Configuring the System Resource Limit: Example, page 6

Configuring the IKE Security Association Limit: Example

The following example shows how to specify that there can be a maximum of 25 SAs before IKE starts rejecting new SA requests:

```
Router(config)# crypto call admission limit ike sa 25
```

Configuring the System Resource Limit: Example

The following example shows how to specify that IKE should drop SA requests when the level of system resources that are configured in the unit of charge reaches 90000:

```
Router(config)# call admission limit 90000
```

Additional References

The following sections provide references related to Call Admission Control for IKE.

Related Documents

Related Topic	Document Title
IKE commands	• *Cisco IOS Security Command Reference*

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC #2409	*The Internet Key Exchange*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **call admission limit**
- **clear crypto call admission statistics**
- **crypto call admission limit**
- **show call admission statistics**
- **show crypto call admission statistics**

Certificate to ISAKMP Profile Mapping

First Published: May 17, 2004
Last Updated: August 21, 2007

The Certificate to ISAKMP Profile Mapping feature enables you to assign an Internet Security Association and Key Management Protocol (ISAKMP) profile to a peer on the basis of the contents of arbitrary fields in the certificate. In addition, this feature allows you to assign a group name to those peers that are assigned an ISAKMP profile.

History for Certificate to ISAKMP Profile Mapping Feature

Release	Modification
12.3(8)T	This feature was introduced.
12.2(33)SRA	This feature was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This feature was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Certificate to ISAKMP Profile Mapping

- You should be familiar with configuring certificate maps.
- You should be familiar with configuring ISAKMP profiles.

Restrictions for Certificate to ISAKMP Profile Mapping

This feature will not be applicable if you use Rivest, Shamir, and Adelman- (RSA-) signature or RSA-encryption authentication without certificate exchange. ISAKMP peers must be configured to do RSA-signature or RSA-encryption authentication using certificates.

Information About Certificate to ISAKMP Profile Mapping

To configure the Certificate to ISAKMP Profile Mapping feature, you should understand the following concepts:

Certificate to ISAKMP Profile Mapping Overview

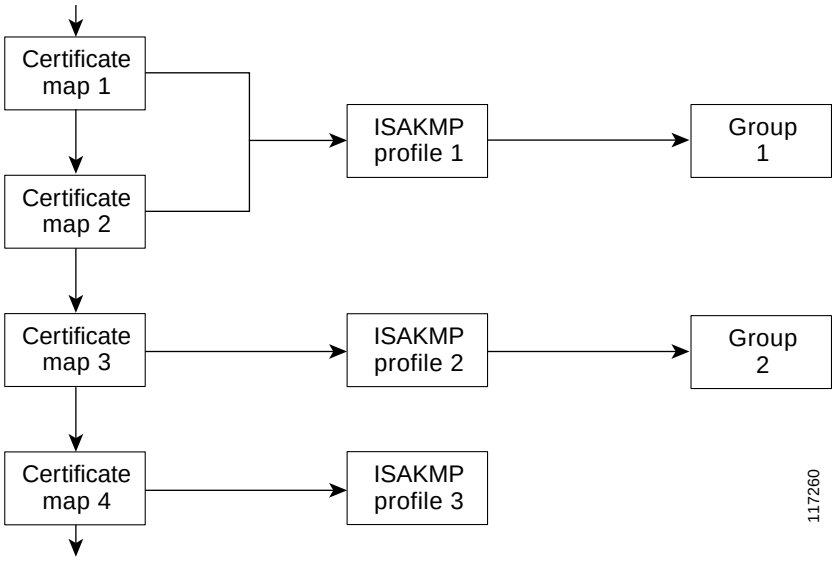
Prior to Cisco IOS Release 12.3(8)T, the only way to map a peer to an ISAKMP profile was as follows. The ISAKMP identity field in the ISAKMP exchange was used for mapping a peer to an ISAKMP profile. When certificates were used for authentication, the ISAKMP identity payload contained the subject name from the certificate. If a certificate authority (CA) did not provide the required group value in the first Organizational Unit (OU) field of a certificate, an ISAKMP profile could not be assigned to a peer.

Effective with Cisco IOS Release 12.3(8)T, a peer can still be mapped as explained above. However, the Certificate to ISAKMP Profile Mapping feature enables you to assign an ISAKMP profile to a peer on the basis of the contents of arbitrary fields in the certificate. You are no longer limited to assigning an ISAKMP profile on the basis of the subject name of the certificate. In addition, this feature allows you to assign a group to a peer to which an ISAKMP profile has been assigned.

How Certificate to ISAKMP Profile Mapping Works

Figure 1 illustrates how certificate maps may be attached to ISAKMP profiles and assigned group names.

Figure 1 Certificate Maps Mapped for Profile Group Assignment



A certificate map can be attached to only one ISAKMP profile although an ISAKMP profile can have several certificate maps attached to it.

Certificate maps provide the ability for a certificate to be matched with a given set of criteria. ISAKMP profiles can bind themselves to certificate maps, and if the presented certificate matches the certificate map present in an ISAKMP profile, the peer will be assigned the ISAKMP profile. If the ISAKMP profile contains a client configuration group name, the same group name will be assigned to the peer. This ISAKMP profile information will override the information in the ID_KEY_ID identity or in the first OU field of the certificate.

Assigning an ISAKMP Profile and Group Name to a Peer

To assign an ISAKMP profile to a peer on the basis of arbitrary fields in the certificate, use the **match certificate** command after the ISAKMP profile has been defined.

To associate a group name with an ISAKMP profile that will be assigned to a peer, use the **client configuration group** command, also after the ISAKMP profile has been defined.

How to Configure Certificate to ISAKMP Profile Mapping

This section contains the following procedures:

- Mapping the Certificate to the ISAKMP Profile, page 4 (required)
- Verifying That the Certificate Has Been Mapped, page 4 (optional)
- Assigning the Group Name to the Peer, page 5 (required)
- Monitoring and Maintaining Your Certificate to ISAKMP Profile Mapping, page 6 (optional)

Mapping the Certificate to the ISAKMP Profile

To map the certificate to the ISAKMP profile, perform the following steps. This configuration will enable you to assign the ISAKMP profile to a peer on the basis of the contents of arbitrary fields in the certificate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name*
4. **match certificate** *certificate-map*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure** *terminal* **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp profile** *profile-name* **Example:** `Router (config)# crypto isakmp profile vpnprofile`	Defines an ISAKMP profile and enters into crypto ISAKMP profile configuration mode.
Step 4	**match certificate** *certificate-map* **Example:** `Router (conf-isa-prof)# match certificate map1`	Accepts the name of a certificate map.

Verifying That the Certificate Has Been Mapped

The following **show** command may be used to verify that the subject name of the certificate map has been properly configured.

SUMMARY

1. **enable**
2. **show crypto ca certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`show crypto ca certificates` **Example:** `Router# show crypto ca certificates`	Displays information about your certificate.

Assigning the Group Name to the Peer

To associate a group name with a peer when the peer is mapped to an ISAKMP profile, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name*
4. **client configuration group** *group-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp profile** *profile-name* **Example:** `Router (config)# crypto isakmp profile vpnprofile`	Defines an ISAKMP profile and enters into isakmp profile configuration mode.
Step 4	**client configuration group** *group-name* **Example:** `Router (conf-isa-prof)# client configuration group group1`	Accepts the name of a group that will be assigned to a peer when the peer is assigned this crypto ISAKMP profile.

Monitoring and Maintaining Your Certificate to ISAKMP Profile Mapping

To monitor and maintain your certificate to ISAKMP profile mapping, you may use the following **debug** command.

SUMMARY STEPS

1. **enable**
2. **debug crypto isakmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`debug crypto isakmp` **Example:** `Router# debug crypto isakmp`	Displays output showing that the certificate has gone through certificate map matching and that the certificate matches the ISAKMP profile. The command may also be used to verify that the peer has been assigned a group.

Configuration Examples for Certificate to ISAKMP Profile Mapping

This section contains the following configuration examples:

Certificates Mapped to the ISAKMP Profile on the Basis of Arbitrary Fields: Example

The following configuration example shows that whenever a certificate contains “ou = green,” the ISAKMP profile “cert_pro” will be assigned to the peer:

```
crypto pki certificate map cert_map 10
 subject-name co ou = green
!
!
crypto isakmp identity dn
crypto isakmp profile cert_pro
   ca trust-point 2315
   ca trust-point LaBcA
   initiate mode aggressive
   match certificate cert_map
```

Group Name Assigned to a Peer That Is Associated with an ISAKMP Profile: Example

The following example shows that the group “some_group” is to be associated with a peer that has been assigned an ISAKMP profile:

```
crypto isakmp profile id_profile
   ca trust-point 2315
```

```
   match identity host domain cisco.com
 client configuration group some_group
```

Mapping a Certificate to an ISAKMP Profile Verification: Example

The following examples show that a certificate has been mapped to an ISAKMP profile. The examples include the configurations for the responder and initiator, **show command** output verifying that the subject name of the certificate map has been configured, and **debug** command output showing that the certificate has gone through certificate map matching and been matched to the ISAKMP profile.

Responder Configuration

```
crypto pki certificate map cert_map 10
! The above line is the certificate map definition.
 subject-name co ou = green
! The above line shows that the subject name must have “ou = green.”
!
crypto isakmp profile certpro
! The above line shows that this is the ISAKMP profile that will match if the certificate
of the peer matches cert_map (shown on third line below).
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
   initiate mode aggressive
```

Initiator Configuration

```
crypto ca trustpoint LaBcA
 enrollment url http://10.76.82.20:80/cgi-bin/openscep
 subject-name ou=green,c=IN
! The above line ensures that the subject name “ou = green” is set.
 revocation-check none
```

show crypto ca certificates Command Output for the Initiator

```
Router# show crypto ca certificates

Certificate
  Status: Available
  Certificate Serial Number: 21
  Certificate Usage: General Purpose
  Issuer:
    cn=blue-lab CA
    o=CISCO
    c=IN
  Subject:
    Name: Router1.cisco.com
    c=IN
    ou=green
! The above line is a double check that “ou = green” has been set as the subject name.
    hostname=Router1.cisco.com
  Validity Date:
    start date: 14:34:30 UTC Mar 31 2004
    end   date: 14:34:30 UTC Apr 1 2009
    renew date: 00:00:00 UTC Jan 1 1970
  Associated Trustpoints: LaBcA
```

debug crypto isakmp Command Output for the Responder

```
Router# debug crypto isakmp
```

```
6d23h: ISAKMP (0:268435460): received packet from 192.0.0.2 dport 500 sport 500 Global (R)
MM_KEY_EXCH
6d23h: ISAKMP: Main Mode packet contents (flags 1, len 892):
6d23h:           ID payload
6d23h:             FQDN <Router1.cisco.com> port 500 protocol 17
6d23h:           CERT payload
6d23h:           SIG payload
6d23h:           KEEPALIVE payload
6d23h:           NOTIFY payload
6d23h: ISAKMP:(0:4:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
6d23h: ISAKMP:(0:4:HW:2):Old State = IKE_R_MM4  New State = IKE_R_MM5

6d23h: ISAKMP:(0:4:HW:2): processing ID payload. message ID = 0
6d23h: ISAKMP (0:268435460): ID payload
        next-payload : 6
        type         : 2
        FQDN name    : Router1.cisco.com
        protocol     : 17
        port         : 500
        length       : 28
6d23h: ISAKMP:(0:4:HW:2):: peer matches *none* of the profiles
6d23h: ISAKMP:(0:4:HW:2): processing CERT payload. message ID = 0
6d23h: ISAKMP:(0:4:HW:2): processing a CT_X509_SIGNATURE cert
6d23h: ISAKMP:(0:4:HW:2): peer's pubkey isn't cached
6d23h: ISAKMP:(0:4:HW:2): OU = green
6d23h: ISAKMP:(0:4:HW:2): certificate map matches certpro profile
! The above line shows that the certificate has gone through certificate map matching and
that it matches the “certpro” profile.
6d23h: ISAKMP:(0:4:HW:2): Trying to re-validate CERT using new profile
6d23h: ISAKMP:(0:4:HW:2): Creating CERT validation list: 2315, LaBcA,
6d23h: ISAKMP:(0:4:HW:2): CERT validity confirmed.
```

Group Name Assigned to a Peer Verfication: Example

The following configuration and debug output show that a group has been assigned to a peer.

Initiator Configuration

```
crypto isakmp profile certpro
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
   client configuration group new_group
! The statement on the above line will assign the group “new_group” to any peer that
matches the ISAKMP profile “certpro.”
   initiate mode aggressive
!
```

debug crypto isakmp profile Command Output for the Responder

The following debug output example shows that the peer has been matched to the ISAKMP profile named “certpro” and that it has been assigned a group named “new_group.”

```
Router# debug crypto isakmp profile
6d23h: ISAKMP (0:268435461): received packet from 192.0.0.2 dport 500 sport 500 Global (R)
MM_KEY_EXCH
6d23h: ISAKMP: Main Mode packet contents (flags 1, len 892):
6d23h:           ID payload
6d23h:             FQDN <Router1.cisco.com> port 500 protocol 17
6d23h:           CERT payload
6d23h:           SIG payload
6d23h:           KEEPALIVE payload
```

```
6d23h:           NOTIFY payload
6d23h: ISAKMP:(0:5:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
6d23h: ISAKMP:(0:5:HW:2):Old State = IKE_R_MM4  New State = IKE_R_MM5

6d23h: ISAKMP:(0:5:HW:2): processing ID payload. message ID = 0
6d23h: ISAKMP (0:268435461): ID payload
        next-payload : 6
        type         : 2
        FQDN name    : Router1.cisco.com
        protocol     : 17
        port         : 500
        length       : 28
6d23h: ISAKMP:(0:5:HW:2):: peer matches *none* of the profiles
6d23h: ISAKMP:(0:5:HW:2): processing CERT payload. message ID = 0
6d23h: ISAKMP:(0:5:HW:2): processing a CT_X509_SIGNATURE cert
6d23h: ISAKMP:(0:5:HW:2): peer's pubkey isn't cached
6d23h: ISAKMP:(0:5:HW:2): OU = green
6d23h: ISAKMP:(0:5:HW:2): certificate map matches certpro profile
6d23h: ISAKMP:(0:5:HW:2): Trying to re-validate CERT using new profile
6d23h: ISAKMP:(0:5:HW:2): Creating CERT validation list: 2315, LaBcA,
6d23h: ISAKMP:(0:5:HW:2): CERT validity confirmed.
6d23h: ISAKMP:(0:5:HW:2):Profile has no keyring, aborting key search
6d23h: ISAKMP:(0:5:HW:2): Profile certpro assigned peer the group named new_group
```

Additional References

The following sections provide references related to Certificate to ISAKMP Profile Mapping.

Related Documents

Related Topic	Document Title
Configuring certificate maps	*Certificate Security Attribute-Based Access Control*, Release 12.2 T
Configuring ISAKMP profiles	*VRF-Aware IPSec*, *Release 12.2 T*
Security commands	*Cisco IOS Security Command Reference*, Release 12.4 T

Standards

Standards	Title
There are no new or modified standards associated with this feature.	—

MIBs

MIBs	MIBs Link
There are no new or modified MIBs associated with this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
There are no new or modified RFCs associated with this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **client configuration group**
- **match certificate (ISAKMP)**

Encrypted Preshared Key

The Encrypted Preshared Key feature allows you to securely store plain text passwords in type 6 (encrypted) format in NVRAM.

Feature History for Encrypted Preshared Key

Release	Modification
12.3(2)T	This feature was introduced.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Encrypted Preshared Key

- Old ROM monitors (ROMMONs) and boot images cannot recognize the new type 6 passwords. Therefore, errors are expected if you boot from an old ROMMON.
- For Cisco 836 routers, please note that support for Advanced Encryption Standard (AES) is available only on IP plus images.

Information About Encrypted Preshared Key

Before Using the Encrypted Preshared Key feature, you should understand the following concepts:

- Using the Encrypted Preshared Key Feature to Securely Store Passwords, page 2
- How to Configure an Encrypted Preshared Key, page 3

Using the Encrypted Preshared Key Feature to Securely Store Passwords

Using the Encrypted Preshared Key feature, you can securely store plain text passwords in type 6 format in NVRAM using a command-line interface (CLI). Type 6 passwords are encrypted. Although the encrypted passwords can be seen or retrieved, it is difficult to decrypt them to find out the actual password. Use the **key config-key password-encryption** command with the **password encryption aes** command to configure and enable the password (symmetric cipher AES is used to encrypt the keys). The password (key) configured using the **config-key password-encryption** command is the master encryption key that is used to encrypt all other keys in the router.

If you configure the **password encryption aes** command without configuring the **key config-key password-encryption** command, the following message is printed at startup or during any nonvolatile generation (NVGEN) process, such as when the **show running-config** or **copy running-config startup-config** commands have been configured:

```
“Can not encrypt password. Please configure a configuration-key with ‘key config-key’”
```

Changing a Password

If the password (master key) is changed, or reencrypted, using the **key config-key password-encryption** command), the list registry passes the old key and the new key to the application modules that are using type 6 encryption.

Deleting a Password

If the master key that was configured using the **key config-key password-encryption** command is deleted from the system, a warning is printed (and a confirm prompt is issued) that states that all type 6 passwords will become useless. As a security measure, after the passwords have been encrypted, they will never be decrypted in the Cisco IOS software. However, passwords can be reencrypted as explained in the previous paragraph.

Caution If the password configured using the **key config-key password-encryption** command is lost, it cannot be recovered. The password should be stored in a safe location.

Unconfiguring Password Encryption

If you later unconfigure password encryption using the **no password encryption aes** command, all existing type 6 passwords are left unchanged, and as long as the password (master key) that was configured using the **key config-key password-encryption** command exists, the type 6 passwords will be decrypted as and when required by the application.

Storing Passwords

Because no one can "read" the password (configured using the **key config-key password-encryption** command), there is no way that the password can be retrieved from the router. Existing management stations cannot "know" what it is unless the stations are enhanced to include this key somewhere, in which case the password needs to be stored securely within the management system. If configurations are stored using TFTP, the configurations are not standalone, meaning that they cannot be loaded onto a router. Before or after the configurations are loaded onto a router, the password must be manually added (using the **key config-key password-encryption** command). The password can be manually added to the stored configuration but is not recommended because adding the password manually allows anyone to decrypt all passwords in that configuration.

Configuring New or Unknown Passwords

If you enter or cut and paste cipher text that does not match the master key, or if there is no master key, the cipher text is accepted or saved, but an alert message is printed. The alert message is as follows:

```
"ciphertext>[for username bar>] is incompatible with the configured master key."
```

If a new master key is configured, all the plain keys are encrypted and made type 6 keys. The existing type 6 keys are not encrypted. The existing type 6 keys are left as is.

If the old master key is lost or unknown, you have the option of deleting the master key using the **no key config-key password-encryption** command. Deleting the master key using the **no key config-key password-encryption** command causes the existing encrypted passwords to remain encrypted in the router configuration. The passwords will not be decrypted.

Enabling the Encrypted Preshared Key

The **password encryption aes** command is used to enable the encrypted password.

How to Configure an Encrypted Preshared Key

This section contains the following procedures:

- Configuring an Encrypted Preshared Key, page 4 (required)
- Monitoring Encrypted Preshared Keys, page 5 (optional)
- Configuring an ISAKMP Preshared Key, page 6 (optional)
- Configuring an ISAKMP Preshared Key in ISAKMP Keyrings, page 7 (optional)
- Configuring ISAKMP Aggressive Mode, page 8 (optional)
- Configuring a Unity Server Group Policy, page 9 (optional)

- Configuring an Easy VPN Client, page 10 (optional)

Configuring an Encrypted Preshared Key

To configure an encrypted preshared key, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key config-key password-encryption** [*text*]
4. **password encryption aes**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**key config-key password-encryption** [*text*] **Example:** `Router (config)# key config-key password-encryption`	Stores a type 6 encryption key in private NVRAM. • If you want to key in interactively (using the enter key) and an encrypted key already exists, you will be prompted for the following: Old key, New key, and Confirm key. • If you want to key in interactively but an encryption key is not present, you will be prompted for the following: New key and Confirm key. • If you want to remove the password that is already encrypted, you will see the following prompt: "WARNING: All type 6 encrypted keys will become unusable. Continue with master key deletion? [yes/no]:".
Step 4	**password encryption aes** **Example:** `Router (config)# password-encryption aes`	Enables the encrypted preshared key.

Troubleshooting Tips

If you see the warning message "ciphertext >[for username bar>] is incompatible with the configured master key," you have entered or cut and pasted cipher text that does not match the master key or there is no master key. (The cipher text will be accepted or saved.) The warning message will allow you to locate the broken configuration line or lines.

Monitoring Encrypted Preshared Keys

To get logging output for encrypted preshared keys, perform the following steps.

1. **enable**
2. **password logging**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`password logging` **Example:** `Router# password logging`	Provides a log of debugging output for a type 6 password operation.

Examples

The following **password logging** debug output shows that a new master key has been configured and that the keys have been encrypted with the new master key:

```
Router (config)# key config-key password-encrypt
New key:
Confirm key:
Router (config)#
01:40:57: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master keypas

Router (config)# key config-key password-encrypt
Old key:
New key:
Confirm key:
Router (config)#
01:42:11: TYPE6_PASS: Master key change heralded, re-encrypting the keys
with the new master key
01:42:11: TYPE6_PASS: Mac verification successful
01:42:11: TYPE6_PASS: Mac verification successful
01:42:11: TYPE6_PASS: Mac verification successful
```

What To Do Next

You can perform any of the following procedures. Each procedure is independent of the others.

- Configuring an ISAKMP Preshared Key, page 6
- Configuring an ISAKMP Preshared Key in ISAKMP Keyrings, page 7
- Configuring ISAKMP Aggressive Mode, page 8
- Configuring a Unity Server Group Policy, page 9
- Configuring an Easy VPN Client, page 10

Configuring an ISAKMP Preshared Key

To configure an ISAKMP preshared key, perform the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp key** *keystring* **address** *peer-address*
4. **crypto isakmp key** *keystring* **hostname** *hostname*

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**crypto isakmp key** *keystring* **address** *peer-address* **Example:** `Router (config)# crypto isakmp key cisco address 10.2.3.4`	Configures a preshared authentication key. • The *peer-address* argument specifies the IP address of the remote peer.
Step 4	**crypto isakmp key** *keystring* **hostname** *hostname* **Example:** `Router (config)# crypto isakmp key foo hostname foo.com`	Configures a preshared authentication key. • The *hostname* argument specifies the fully qualified domain name (FQDN) of the peer.

Example

The following sample output shows that an encrypted preshared key has been configured:

```
crypto isakmp key 6 _Hg[^^ECgLGGPF^RXTQfDDWQ][YAAB address 10.2.3.4
crypto isakmp key 6 `eR\eTRaKCUZPYYQfDgXRWi_AAB hostname foo.com
```

Configuring an ISAKMP Preshared Key in ISAKMP Keyrings

To configure an ISAKMP preshared key in ISAKMP keyrings, which are used in IPSec Virtual Route Forwarding (VRF) configurations, perform the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto keyring** *keyring-name*
4. **pre-shared-key address** *address* **key** *key*
5. **pre-shared-key hostname** *hostname* **key** *key*

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**crypto keyring** *keyring-name* **Example:** `Router (config)# crypto keyring foo`	Defines a crypto keyring to be used during Internet Key Exchange (IKE) authentication and enters keyring configuration mode.
Step 4	**pre-shared-key address** *address* **key** *key* **Example:** `Router (config-keyring)# pre-shared-key address 10.2.3.5 key cisco`	Defines a preshared key to be used for IKE authentication. • The *address* argument specifies the IP address of the remote peer.
Step 5	**pre-shared-key hostname** *hostname* **key** *key* **Example:** `Router (config-keyring)# pre-shared-key hostname foo.com key cisco`	Defines a preshared key to be used for IKE authentication. • The *hostname* argument specifies the FQDN of the peer.

Example

The following s**how-running-config** sample output shows that an encrypted preshared key in ISAKMP keyrings has been configured.

```
crypto keyring foo
  pre-shared-key address 10.2.3.5 key 6 `WHCJYR_Z]GRPF^RXTQfDcfZ]GPAAB
  pre-shared-key hostname foo.com key 6 aE_REHDcOfYCPF^RXTQfDJYVVNSAAB
```

Configuring ISAKMP Aggressive Mode

To configure ISAKMP aggressive mode, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp peer ip-address** *ip-address*
4. **set aggressive-mode client-endpoint** *client-endpoint*
5. **set aggressive-mode password** *password*

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**crypto isakmp peer ip-address** *ip-address* **Example:** `Router (config)# crypto isakmp peer ip-address 10.2.3.4`	To enable an IP Security (IPSec) peer for IKE querying of authentication, authorization, and accounting (AAA) for tunnel attributes in aggressive mode and to enter ISAKMP peer configuration mode.
Step 4	**set aggressive-mode client-endpoint** *client-endpoint* **Example:** `Router (config-isakmp-peer)# set aggressive-mode client-endpoint fqdn cisco.com`	Specifies the Tunnel-Client-Endpoint attribute within an ISAKMP peer configuration.
Step 5	**set aggressive-mode password** *password* **Example:** `Router (config-isakmp-peer)# set aggressive-mode password cisco`	Specifies the Tunnel-Password attribute within an ISAKMP peer configuration.

Example

The following s**how-running-config** sample output shows that an encrypted preshared key in ISAKMP aggressive mode has been configured.

```
crypto isakmp peer address 10.2.3.4
 set aggressive-mode password 6 ^aKPIQ_KJE_PPF^RXTQfDTIaLNeAAB
 set aggressive-mode client-endpoint fqdn cisco.com
```

Configuring a Unity Server Group Policy

To configure a unity server group policy, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** *group-name*
4. **pool** *name*
5. **domain** *name*
6. **key** *name*

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**crypto isakmp client configuration group** *group-name* **Example:** `Router (config)# crypto isakmp client configuration group foo`	Specifies the policy profile of the group that will be defined and enters ISAKMP group configuration mode.
Step 4	**pool** *name* **Example:** `Router (config-isakmp-group)# pool foopool`	Defines a local pool address.

	Command	Description
Step 5	**domain name** **Example:** `Router (config-isakmp-group)# domain cisco.com`	Specifies the Domain Name Service (DNS) domain to which a group belongs.
Step 6	**key** *name* **Example:** `Router (config-isakmp-group)# key cisco`	Specifies the IKE preshared key for group policy attribute definition.

Example

The following **show-running-config** sample output shows that an encrypted key has been configured for a unity server group policy:

```
crypto isakmp client configuration group foo
 key 6 cZZgDZPOE\dDPF^RXTQfDTIaLNeAAB
 domain cisco.com
 pool foopool
```

Configuring an Easy VPN Client

To configure an Easy VPN client, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. crypto ipsec client ezvpn *name*
4. peer *ipaddress*
5. mode client
6. group *group-name* key *group-key*
7. connect manual

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.

	Command	Description
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn foo`	Creates a Cisco Easy VPN remote configuration and enters Cisco Easy VPN remote configuration mode.
Step 4	**peer** *ipaddress* **Example:** `Router (config-isakmp-peer)# peer 10.2.3.4`	Sets the peer IP address for the VPN connection.
Step 5	**mode client** **Example:** `Router (config-isakmp-ezpvy)# mode client`	Automatically configures the router for Cisco Easy VPNclient mode operation, which uses Network Address Translation (NAT) or Peer Address Translation (PAT) address translations.
Step 6	**group** *group-name* **key** *group-key* **Example:** `Router (config-isakmp-ezvpn)# group foo key cisco`	Specifies the group name and key value for the VPN connection.
Step 7	**connect manual** **Example:** `Router (config-isakmp-ezvpn)# connect manual`	Specifies the manual setting for directing the Cisco Easy VPN remote client to wait for a command or application program interface (API) call before attenpting to establish the Cisco Easy VPN remote connection.

Example

The following **show-running-config** sample output shows that an Easy VPN client has been configured. The key has been encrypted.

```
crypto ipsec client ezvpn foo
 connect manual
 group foo key 6 gdMI`S^^[GIcPF^RXTQfDFKEO\RAAB
 mode client
 peer 10.2.3.4
```

Configuration Examples for Encrypted Preshared Key

This section provides the following configuration examples:

Encrypted Preshared Key: Example

The following is an example of a configuration for which a type 6 preshared key has been encrypted. It includes the prompts and messages that a user might see.

```
Router (config)# crypto isakmp key cisco address 10.0.0.2
Router (config)# exit
Router# show runnning-config | include crypto isakmp key
 crypto isakmp key cisco address 10.0.0.2
Router#
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router (config)# password encryption aes
Router (config)# key config-key password-encrypt
New key:
Confirm key:
Router (config)#
01:46:40: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master key
Router (config)# exit
Router # show running-config | include crypto isakmp key
crypto isakmp key 6 CXWdhVTZYB_Vcd^`cIHDOahiFTa address 10.0.0.2
```

No Previous Key Present: Example

In the following configuration example, no previous key is present:

```
Router (config)# key config-key password-encryption testkey 123
```

Key Already Exists: Example

In the following configuration example, a key already exists:

```
Router (config)# key config-key password-encryption testkey123
Old key:
Router (config)#
```

Key Already Exists But the User Wants to Key In Interactively: Example

In the following configuration example, the user wants to key in interactively, but a key already exists. The Old key, New key, and Confirm key prompts will show on your screen if you enter the **key config-key password-encryption** command and press the enter key to get into interactive mode.

```
Router (config)# key config-key password-encryption
Old key:
New key:
Confirm key:
```

No Key Present But the User Wants to Key In Interactively: Example

In the following example, the user wants to key in interactively, but no key is present. The New key and Confirm key prompts will show on your screen if you are in interactive mode.

```
Router (config)# key config-key password-encryption
New key:
```

```
Confirm key:
```

Removal of the Password Encryption: Example

In the following configuration example, the user wants to remove the encrypted password. The "WARNING: All type 6 encrypted keys will become unusable. Continue with master key deletion? [yes/no]:" prompt will show on your screen if you are in interactive mode.

```
Router (config)# no key config-key password-encryption

WARNING: All type 6 encrypted keys will become unusable. Continue with master key
deletion ? [yes/no]: y
```

Where to Go Next

Configure any other preshared keys.

Additional References

The following sections provide references related to Encrypted Preshared Key.

Related Documents

Related Topic	Document Title
Configuring passwords	*The section "Part 4: IP Security and Encryption" of the Cisco IOS Security Configuration Guide* *Cisco IOS Security Command Reference, Release 12.3 T*

Standards

Standards	Title
This feature has no new or modified standards.	—

MIBs

MIBs	MIBs Link
This feature has no new or modified MIBs.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
This feature has no new or modified RFCs.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **crypto ipsec client ezvpn (global)**
- **crypto isakmp client configuration group**
- **crypto isakmp key**
- **key config-key password-encryption**
- **password encryption aes**
- **password logging**
- **pre-shared-key**
- **set aggressive-mode password**

IKE: Initiate Aggressive Mode

Feature History

Release	Modification
12.2(8)T	This feature was introduced.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This document describes the IKE: Initiate Aggressive Mode feature in Cisco IOS Release 12.2(8)T. It includes the following sections:

Feature Overview

The IKE: Initiate Aggressive Mode feature allows you to configure Internet Key Exchange (IKE) preshared keys as RADIUS tunnel attributes for IP Security (IPSec) peers. Thus, you can scale your IKE preshared keys in a hub-and-spoke topology.

Although IKE preshared keys are simple to understand and easy to deploy, they do not scale well with an increasing number of users and are therefore prone to security threats. Instead of keeping your preshared keys on the hub router, this feature allows you to scale your preshared keys by storing and retrieving them from an authentication, authorization, and accounting (AAA) server. The preshared keys are stored in the AAA server as Internet Engineering Task Force (IETF) RADIUS tunnel attributes and are retrieved when a user tries to “speak” to the hub router. The hub router retrieves the preshared key

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

from the AAA server and the spokes (the users) initiate aggressive mode to the hub by using the preshared key that is specified in the Internet Security Association Key Management Policy (ISAKMP) peer policy as a RADIUS tunnel attribute.

RADIUS Tunnel Attributes

To initiate an IKE aggressive mode negotiation, the Tunnel-Client-Endpoint (66) and Tunnel-Password (69) attributes must be configured in the ISAKMP peer policy. The Tunnel-Client-Endpoint attribute will be communicated to the server by encoding it in the appropriate IKE identity payload; the Tunnel-Password attribute will be used as the IKE preshared key for the aggressive mode negotiation.

Benefits

The IKE: Initiate Aggressive Mode feature allows you to specify RADIUS tunnel attributes for an IPSec peer and to initiate an IKE aggressive mode negotiation with the tunnel attributes. This feature is best implemented in a crypto hub-and-spoke scenario, by which the spokes initiate IKE aggressive mode negotiation with the hub by using the preshared keys that are specified as tunnel attributes and stored on the AAA server. This scenario is scalable because the preshared keys are kept at a central repository (the AAA server) and more than one hub router and one application can use the information.

Restrictions

TED Restriction

This feature is not intended to be used with a dynamic crypto map that uses Tunnel Endpoint Discovery (TED) to initiate tunnel setup. TED is useful in configuring a full mesh setup, which requires an AAA server at each site to store the preshared keys for the peers; this configuration is not practical for use with this feature.

Tunnel-Client-Endpoint ID Types

Only the following ID types can be used in this feature:

- ID_IPV4 (IPV4 address)
- ID_FQDN (fully qualified domain name, for example “foo.cisco.com”)
- ID_USER_FQDN (e-mail address)

Related Documents

- *Cisco IOS Security Configuration Guide*, Release 12.2
- *Cisco IOS Security Command Reference*, Release 12.2

Supported Platforms

This feature runs on all platforms that support IPSec and public key infrastructure (PKI).

- Cisco 800 series

- Cisco 805
- Cisco 806
- Cisco 828
- Cisco 1400 series
- Cisco 1600 series
- Cisco 1600-R series
- Cisco 1710
- Cisco 1720
- Cisco 1750
- Cisco 1751
- Cisco 2400 series
- Cisco 2600 series
- Cisco 3620
- Cisco 3640
- Cisco 3660
- Cisco 3725
- Cisco 3745
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series
- Cisco 7700 series
- Cisco MC3810
- Route Processor Module (RPM)
- Universal Route Module (URM)

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

- RFC 2409, *The Internet Key Exchange*
- RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

Prerequisites

Before configuring the Initiate Aggressive Mode IKE feature, you must perform the following tasks:

- Configure AAA
- Configure an IPSec Transform
- Configure a Static Crypto Map
- Configure an ISAKMP Policy
- Configure a Dynamic Crypto Map

For information on completing these tasks, refer to the chapters “Configuring Authentication,” “Configuring IPSec Network Security,” and “Configuring Internet Key Exchange Security Protocol” in the *Cisco IOS Security Configuration Guide*, Release 12.2.

Configuration Tasks

See the following sections for configuration tasks for the IKE: Initiate Aggressive Mode feature. Each task in the list is identified as either required or optional.

- Configuring RADIUS Tunnel Attributes (required)
- Verifying RADIUS Tunnel Attribute Configurations (optional)

Configuring RADIUS Tunnel Attributes

To configure the Tunnel-Client-Endpoint and Tunnel-Password attributes within the ISAKMP peer configuration, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name isakmp authorization list list-name`	Enables IKE querying of AAA for tunnel attributes in aggressive mode.
Step 2	`Router(config)# crypto isakmp peer {ip-address ip-address | fqdn fqdn}`	Enables an IPSec peer for IKE querying of AAA for tunnel attributes in aggressive mode and enters ISAKMP policy configuration mode.
Step 3	`Router(config-isakmp)# set aggressive-mode client-endpoint client-endpoint`	Specifies the Tunnel-Client-Endpoint attribute within an ISAKMP peer configuration.
Step 4	`Router(config-isakmp)# set aggressive-mode password password`	Specifies the Tunnel-Password attribute within an ISAKMP peer configuration.

Verifying RADIUS Tunnel Attribute Configurations

To verify that the Tunnel-Client-Endpoint and Tunnel-Password attributes have been configured within the ISAKMP peer policy, use the **show running-config** global configuration command.

Troubleshooting Tips

To troubleshoot the IKE: Initiate Aggressive Mode feature, use the following debug commands in EXEC mode:

Command	Purpose
`Router# debug aaa authorization`	Displays information on AAA authorization.
`Router# debug crypto isakmp`	Displays messages about IKE events.
`Router# debug radius`	Displays information associated with the RADIUS.

Configuration Examples

This section provides the following configuration examples:

- Hub Configuration Example
- Spoke Configuration Example
- RADIUS User Profile Example

Hub Configuration Example

The following example shows how to configure a hub for a hub-and-spoke topology that supports aggressive mode using RADIUS tunnel attributes:

```
!The AAA configurations are as follows:
aaa new-model
aaa authorization network ike group radius
aaa authentication login default group radius
!
! The Radius configurations are as follows:
radius-server host 1.1.1.1 auth-port 1645 acct-port 1646
radius-server key rad123
!
! The IKE configurations are as follows:
crypto isakmp policy 1
 authentication pre-share
!
! The IPSec configurations are as follows:
crypto ipsec transform-set trans1 esp-3des esp-sha-hmac
!
crypto dynamic-map Dmap 10
 set transform-set trans1
!
crypto map Testtag isakmp authorization list ike
crypto map Testtag 10 ipsec-isakmp dynamic Dmap
!
interface Ethernet0
 ip address 4.4.4.1 255.255.255.0
 crypto map Testtag
!
interface Ethernet1
 ip address 2.2.2.1 255.255.255.0
```

Spoke Configuration Example

The following example shows how to configure a spoke for a hub-and-spoke topology that supports aggressive mode using RADIUS tunnel attributes:

```
!The IKE configurations are as follows:
crypto isakmp policy 1
 authentication pre-share
!
! The IPSec configurations are as follows:
crypto ipsec transform-set trans1 esp-3des esp-sha-hmac
 access-list 101 permit ip 3.3.3.0 0.0.0.255 2.2.2.0 0.0.0.255
!
! Initiate aggressive mode using Radius tunnel attributes
crypto isakmp peer address 4.4.4.1
 set aggressive-mode client-endpoint user-fqdn user@cisco.com
 set aggressive-mode password cisco123
!
crypto map Testtag 10 ipsec-isakmp
 set peer 4.4.4.1
 set transform-set trans1
 match address 101
!
interface Ethernet0
 ip address 5.5.5.1 255.255.255.0
 crypto map Testtag
!
```

```
interface Ethernet1
 ip address 3.3.3.1 255.255.255.0
```

RADIUS User Profile Example

The following is an example of a user profile on a RADIUS server that supports the Tunnel-Client-Endpoint and Tunnel-Password attributes:

```
user@cisco.com  Password = "cisco", Service-Type = Outbound
    Tunnel-Medium-Type = :1:IP,
    Tunnel-Type = :1:ESP,
    Cisco:Avpair = "ipsec:tunnel-password=cisco123",
    Cisco:Avpair = "ipsec:key-exchange=ike"
```

Command Reference

The following commands are introduced or modified in the feature or features

- **crypto isakmp peer**
- **set aggressive-mode client-endpoint**
- **set aggressive-mode password**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Multi-ISA

This feature module describes the Multi-ISA feature and includes the following sections:

Feature Overview

The Multi-ISA feature allows a Cisco IOS router to accommodate more than one hardware crypto engine at a time. This feature allows users to increase the capacity of their routers with multiple Integrated Services Adapters (ISAs) and Integrated Services Module (ISMs).

Note ISAs are used on Cisco 7200 routers and ISMs are used on Cisco 7100 routers. Hereafter, for purposes of this document and unless otherwise noted, the term ISA will denote both Integrated Services Adapters and Integrated Services Modules.

The multi-ISA layer provides a single interface, which Cisco IOS software can use to send commands to different hardware crypto engines. The multi-ISA layer accepts all commands and packets on behalf of all underlying hardware crypto engines; it distributes all commands and packets in a predefined manner. That is, when you request an Internet Key Exchange-security association (IKE-SA) session, the multi-ISA layer determines which of the two hardware crypto engine contains fewer IKE-SAs, and it assigns the next session to the hardware crypto engine that has fewer IKE-SAs.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

How Multi-ISA Works

When your router has only one ISA in an active state, all IKE and IP Security-SA sessions go to this one ISA. Once you have inserted the second ISA into your router and it becomes active, subsequent IKE-SAs will flow to the second ISA until the first and second ISAs have an equal number of IKE-SA sessions. For example, if ISA-1 has 10 IKE sessions, and then ISA-2 becomes active, the router will send the following 11 through 20 IKE sessions to ISA-2. Thereafter, the multi-ISA layer will maintain a balance of IKE-SA sessions on both ISAs.

Note The second ISA becomes active through online insertion and removal (OIR) or micro reload.

Benefits

The Multi-ISA feature provides the following benefits:

- Load-sharing of IKE-SAs
- Increased IPSec traffic throughput of your router
- OIR support

Restrictions

System Requirements

Your system should contain at least 128 megabytes of memory to run a single ISA and 256 megabytes of memory to run two ISAs. You need more than 128 megabytes of memory to cross 2,000 bidirectional IPSec tunnels.

Note All tunnels referenced to in this document are defined as bidirectional IPSec tunnels.

Crypto Capabilities

All ISAs supported by the multi-ISA layer must have the same crypto capabilities. For example, if one ISA supports 3DES, another ISA that does not support 3DES cannot be in the same router under the multi-ISA layer.

Failover Limitations

Keepalives are needed to achieve failover in your router. If you turn on keepalives, you cannot exceed 500 tunnels because of current IKE keepalive limitations in the Cisco IOS software.

Note This restriction will be lifted in a future release.

MPPE

Multiple Microsoft Point-to-Point Encryption (MPPE) ISA support is not available.

Related Documents

The following documents provide information related to the Multi-ISA feature:

- *Cisco IOS Security Configuration Guide*, Release 12.1
- *Cisco IOS Security Command Reference*, Release 12.1
- ISA and ISM Installation and Configuration
- The chapter "Basic System Management" in *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.1
- The chapter "Performing Basic System Management" in *Cisco IOS Configuration Fundamentals Configuration Guide*

Supported Platforms

The following platforms support the Multi-ISA feature:

- Cisco 7140
- Cisco 7200 with NP300 service module

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

IPSec Encryption

To use the Multi-ISA feature, you must configure your ISAs to provide IPSec encryption services by performing the following tasks:

- Configure IKE policies.
- Configure IPSec.
- Create crypto map entries.
- Apply a crypto map set to each interface through which IPSec traffic flows.

For information on completing these tasks, refer to *ISA and ISM Installation and Configuration*.

Manual Buffer Tunning

In the 7200 platform, it is recommended that you manually finetune packet buffers to establish more IKE-SA or IPSec tunnels.

To manually configure buffer tuning, enter the following global configuration command:

Command	Purpose
`Router# (config) buffers {small | middle | big | verybig | large | huge | type number} {permanent | max-free | min-free | initial} number`	Makes adjustments to initial buffer pool settings and to the limits at which temporary buffers are created and destroyed.

For buffer tuning examples, see the "Configuration Examples" section later in this document.

Configuration Tasks

None

Verifying Multi-ISA

To verify how many active IKE and IPSec sessions are on each hardware crypto engine, how many Diffie-Hellman (DH) keys are in use, and how far your ISA is from reaching its maximum limit, use the **show crypto eli** command in EXEC mode.

Monitoring and Maintaining Multi-ISA

To obtain a snapshot of how many IKE-SAs and IPSec sessions are active and how many DH keys are in use, use the following command in EXEC mode:

Command	Purpose
`Router# show crypto eli`	Displays a snapshot of how many IKE-SAs and IPSec sessions are active and how many DH keys are in use for each hardware crypto engine.

Configuration Examples

This section provides the following configuration examples:

- Single ISA Scenario Example
- Load-Balancing with Multi-ISA Example
- Manual Buffer Tuning Examples

Single ISA Scenario Example

The following example is sample output from the **show crypto eli** command. In this example, a router has established 1246 IKE sessions; however, all IKE sessions flow to a single ISA because only one ISA is in an active state.

```
P0# show crypto eli

Encryption Layer :  ACTIVE
 Number of crypto engines = 1 .

 Slot-3 crypto engine details.
 Capability-IPSec :No-IPPCP, 3DES, NoRSA

 IKE-Session   : 1246 active,  2029 max, 0 failed
 DH-Key        :    0 active,  1014 max, 0 failed
 IPSec-Session : 2712 active,  4059 max, 0 failed
```

Load-Balancing with Multi-ISA Example

The following example is sample output from the **show crypto eli** command. In this example, a router has established 2492 IKE sessions; the IKE sessions are equally distributed between two ISAs (each ISA contains 1246 IKE sessions), allowing you to increase the capacity of your router.

```
P0# show crypto eli

Encryption Layer :  ACTIVE
 Number of crypto engines = 2 .

 Slot-3 crypto engine details.
 Capability-IPSec :No-IPPCP, 3DES, NoRSA

 IKE-Session   : 1246 active,  2029 max, 0 failed
 DH-Key        :    0 active,  1014 max, 0 failed
 IPSec-Session : 2676 active,  4059 max, 0 failed


 Slot-5 crypto engine details.
 Capability-IPSec :No-IPPCP, 3DES, NoRSA

 IKE-Session   : 1246 active,  2029 max, 0 failed
 DH-Key        :    0 active,  1014 max, 0 failed
 IPSec-Session : 2678 active,  4059 max, 0 failed
```

Table 1 describes significant fields shown in the display.

Table 1 show crypto eli summary Field Descriptions

Field	Description
active	The number of sessions that are active on a given hardware crypto engine.
max	The maximum number of sessions allowed for any given IKE, DH, or IPSec entry.
failed	The number of times that Cisco IOS software attempted to create more sessions than the number specified in “max.”

Manual Buffer Tuning Examples

The following example shows recommended numbers to use, based on available memory, when manually configuring packet buffers on the 7200 platform using the **buffers** command:

```
Memory (MB)       Non-HUGE Buffers
             Min-free  Permanent     Max-free
-----------  -------   --------    --------
    32          64        256        1280
    64         128        512        2560
    96         192        768        3840
   128         256       1024        5120
   160         320       1280        6400
   192         384       1536        7680
   224         448       1792        8960
   256         512       2048       10240

Memory (MB)         HUGE Buffers
             Min-free  Permanent     Max-free
-----------  -------   --------    --------
    32           4         16          64
    64           8         32         128
    96          12         48         192
   128          16         64         256
   160          20         80         320
   192          24         96         384
   224          28        112         448
   256          32        128         512
```

Table 2 decribes significant fields shown in the display.

Table 2 Manual Buffer Tuning Example Descriptions

Field	Description
Memory (MB)	Available memory on your 7200 platform.
Min-free	Minimum number of free or unallocated buffers in a buffer pool.
Permanent	Number of permanent buffers that the system tries to create and keep.
Max-free	Maximum number of free or unallocated buffers in a buffer pool.

The following example shows sample output from the **show buffers** command:

```
P0# show buffers

Buffer elements:
     500 in free list (500 max allowed)
     12666974 hits, 0 misses, 0 created

Public buffer pools:
Small buffers, 104 bytes (total 2048, permanent 2048):
     2039 in free list (512 min, 10240 max allowed)
     293 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
Middle buffers, 600 bytes (total 2048, permanent 2048, peak 5000 @ 16:34:14):
     2048 in free list (512 min, 10240 max allowed)
     849 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
Big buffers, 1524 bytes (total 2048, permanent 2048):
     2048 in free list (512 min, 10240 max allowed)
     84 hits, 0 misses, 0 trims, 0 created
```

```
     0 failures (0 no memory)
VeryBig buffers, 4520 bytes (total 2048, permanent 2048):
     2048 in free list (512 min, 10240 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
Large buffers, 5024 bytes (total 2048, permanent 2048, peak 5000 @ 16:34:25):
     2047 in free list (512 min, 10240 max allowed)
     18 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
Huge buffers, 18024 bytes (total 128, permanent 128):
     128 in free list (32 min, 512 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
```

Command Reference

The following commands are introduced or modified in the feature or features

- **show crypto eli**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Glossary

crypto map—A Cisco IOS software configuration entity that performs two primary functions: (1) selecting data flows that need security processing and (2) defining the policy for these flows and the crypto peer to which traffic needs to go. A crypto map is applied to an interface. The concept of a crypto map was introduced in classic crypto but was expanded for IPSec.

DH—See Diffie-Hellman.

Diffie-Hellman—A public-key cryptography protocol that allows two parties to establish a shared secret over an insecure communications channel. Diffie-Hellman is used within IKE to establish session keys and is a component of Oakley.

IKE—Internet Key Exchange. A hybrid protocol that implements Oakley key exchange and Skeme key exchange inside the Internet Security Association and Key Management Protocol framework. IKE can be used with other protocols, but its initial implementation is with IPSec. IKE provides authentication of the IPSec peers, negotiates IPSec keys, and negotiates IPSec security associations.

IKE is used to establish a shared security policy and authenticated keys for services (such as IPSec) that require keys. Before any IPSec traffic can be passed, each router, firewall, or host must be able to verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a certification authority (CA) service.

Integrated Services Adapter—See ISA.

Internet Key Exchange—See IKE.

IPSec—IP Security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

IP Security—See IPSec.

ISA—Integrated Services Adapter. Provides high-performance, hardware-assisted tunneling and encryption services suitable for private WAN and virtual private network (VPN) applications. Within this feature module, ISA inlcudes ISM.

ISM—Integrated Services Module.

OIR—online insertion and removal. A feature that allows you to add, replace, or remove a card from your router without interrupting the system power, entering console commands, or causing other software or interfaces to shut down. This feature is sometimes referred to as “hot swapping” or “power-on servicing.”

online insertion and removal—See OIR.

Security for VPNs with IPSec

Configuring Security for VPNs with IPSec

This module describes how to configure basic IP Security (IPSec) virtual private networks (VPNs). IPSec is a framework of open standards developed by the Internet Engineering Task Force (IETF). It provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices (“peers”), such as Cisco routers.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the “Feature Information for Security for VPNs with IPSec” section on page 35.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring Security for VPNs with IPSec

IKE Configuration

You must configure Internet Key Exchange (IKE) as described in the module "Configuring Internet Key Exchange Security for IPSec VPNs."

Even if you decide to not use IKE, you still must disable it as described in the module "Configuring Internet Key Exchange for IPSec VPNs."

Ensure Access Lists Are Compatible with IPSec

IKE uses UDP port 500. The IPSec Encapsulating Security Payload (ESP) and Authentication Header (AH) protocols use protocol numbers 50 and 51. Ensure that your access lists are configured so that protocol 50, 51, and User Datagram Protocol (UDP) port 500 traffic is not blocked at interfaces used by IPSec. In some cases you might need to add a statement to your access lists to explicitly permit this traffic.

Restrictions for Configuring Security for VPNs with IPSec

Unicast IP Datagram Application Only

At this time, IPSec can be applied to unicast IP datagrams only. Because the IPSec Working Group has not yet addressed the issue of group key distribution, IPSec does not currently work with multicasts or broadcast IP datagrams.

NAT Configuration

If you use Network Address Translation (NAT), you should configure static NAT translations so that IPSec works properly. In general, NAT translation should occur before the router performs IPSec encapsulation; in other words, IPSec should be working with global addresses.

Information About Configuring Security for VPNs with IPSec

To configure basic IPSec VPNs, you should understand the following concepts:

- Supported Standards, page 2
- Supported Hardware, Switching Paths, and Encapsulation, page 4
- IPSec Functionality Overview, page 6
- IPSec Traffic Nested to Multiple Peers, page 8

Supported Standards

Cisco implements the following standards with this feature:

- IPSec—IP Security Protocol. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer; it uses IKE to handle negotiation of protocols and algorithms

based on local policy, and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

Note The term IPSec is sometimes used to describe the entire protocol of IPSec data services and IKE security protocols and is also sometimes used to describe only the data services.

IPSec is documented in a series of Internet Drafts, all available at http://www.ietf.org/html.charters/ipsec-charter.html.

- IKE—A hybrid protocol that implements Oakley and SKEME key exchanges inside the ISAKMP framework. While IKE can be used with other protocols, its initial implementation is with the IPSec protocol. IKE provides authentication of the IPSec peers, negotiates IPSec security associations, and establishes IPSec keys.

The component technologies implemented for IPSec include:

- AES—Advanced Encryption Standard. A cryptographic algorithm that protects sensitive, unclassified information. AES is privacy transform for IPSec and IKE and has been developed to replace the DES. AES is designed to be more secure than DES: AES offers a larger key size, while ensuring that the only known approach to decrypt a message is for an intruder to try every possible key. AES has a variable key length—the algorithm can specify a 128-bit key (the default), a 192-bit key, or a 256-bit key.
- DES—Data Encryption Standard. An algorithm that is used to encrypt packet data. Cisco IOS implements the mandatory 56-bit DES-CBC with Explicit IV. Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPSec packet. For backwards compatibility, Cisco IOS IPSec also implements the RFC 1829 version of ESP DES-CBC.

  Cisco IOS also implements Triple DES (168-bit) encryption, depending on the software versions available for a specific platform. Triple DES (3DES) is a strong form of encryption that allows sensitive information to be transmitted over untrusted networks. It enables customers to utilize network layer encryption.

  Note Cisco IOS images with strong encryption (including, but not limited to, 56-bit data encryption feature sets) are subject to United States government export controls, and have a limited distribution. Images to be installed outside the United States require an export license. Customer orders might be denied or subject to delay due to United States government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- SEAL—Software Encryption Algorithm. An alternative algorithm to software-based DES, 3DES, and AES. SEAL encryption uses a 160-bit encryption key and has a lower impact to the CPU when compared to other software-based algorithms.
- MD5 (HMAC variant)—MD5 (Message Digest 5) is a hash algorithm. HMAC is a keyed hash variant used to authenticate data.
- SHA (HMAC variant)—SHA (Secure Hash Algorithm) is a hash algorithm. HMAC is a keyed hash variant used to authenticate data.

IPSec as implemented in Cisco IOS software supports the following additional standards:

- AH—Authentication Header. A security protocol which provides data authentication and optional anti-replay services. AH is embedded in the data to be protected (a full IP datagram).

- ESP—Encapsulating Security Payload. A security protocol which provides data privacy services and optional data authentication, and anti-replay services. ESP encapsulates the data to be protected.

Supported Hardware, Switching Paths, and Encapsulation

IPSec has certain requirements for hardware, switching paths, and encapsulation methods as follows:

Supported Hardware

This section contains the following subsections:

VPN Accelerator Module (VAM) Support

The VAM is a single-width acceleration module. It provides high-performance, hardware-assisted tunneling and encryption services suitable for VPN remote access, site-to-site intranet, and extranet applications. It also provides platform scalability and security while working with all services necessary for successful VPN deployments—security, quality of service (QoS), firewall and intrusion detection, service-level validation, and management. The VAM off-loads IPSec processing from the main processor, thus freeing resources on the processor engines for other tasks.

The VAM provides hardware-accelerated support for the following multiple encryption functions:

- 56-bit DES standard mode: CBC
- 3-Key Triple DES (168-bit)
- SHA-1 and MD5
- Rivest, Shamir, Adelman (RSA) public-key algorithm
- Diffie-Hellman key exchange RC4-40

For more information on VAMs, see the document “VPN Acceleration Module (VAM).”

AIMs and NM Support

The data encryption Advanced Integration Module (AIM) and Network Module (NM) provide hardware-based encryption.

The data encryption AIMs and NM are hardware Layer 3 (IPSec) encryption modules and provide DES and Triple DES IPSec encryption for multiple T1s or E1s of bandwidth. These products also have hardware support for Diffie-Hellman, RSA, and DSA key generation.

Before using either module, note that RSA manual keying is not supported.

See Table 44 to determine which VPN encryption module to use.

IPPCP Software for Use with AIMs and NMs in Cisco 2600 and Cisco 3600 Series Routers

Software IPPCP with AIMs and NMs allow customers to use Lempel-Ziv-Stac (LZS) software compression with IPSec when a VPN module is in Cisco 2600 and Cisco 3600 series routers, allowing users to effectively increase the bandwidth on their interfaces.

Without IPPCP software, compression is not supported with the VPN encryption hardware AIM and NM; that is, a user had to remove the VPN module from the router and run software encryption with software compression. IPPCP enables all VPN modules to support LZS compression in software when the VPN module is in the router, thereby, allowing users to configure data compression and increase their bandwidth, which is useful for a low data link.

Without IPPCP, compression occurs at Layer 2, and encryption occurs at Layer 3. After a data stream is encrypted, it is passed on for compression services. When the compression engine receives the encrypted data streams, the data expands and does not compress. This feature enables both compression and encryption of the data to occur at Layer 3 by selecting LZS with the IPSec transform set; that is, LZS compression occurs before encryption, and it is able to get better compression ratio.

Table 44 AIM/VPN Encryption Module Support by Cisco IOS Release

	Encryption Module Support by Cisco IOS Release				
Platform	**12.2(13)T**	**12.3(4)T**	**12.3(5)**	**12.3(6)**	**12.3(7)T**
Cisco 831	Software-based AES				
Cisco 1710 Cisco 1711 Cisco 1721 Cisco 1751 Cisco 1760	Software-based AES				
Cisco 2600 XM	—			AIM-VPN/BPII-Plus Hardware Encryption Module	
Cisco 2611 XM Cisco 2621 XM Cisco 2651 XM	—	AIM-VPN/BPII Hardware Encryption Module			AIM-VPN/BPII-Plus Hardware Encryption Module
Cisco 2691 XM	AIM-VPN/EPII Hardware Encryption Module				AIM-VPN/EPII-Plus Hardware Encryption Module
Cisco 3735	AIM-VPN/EPII Hardware Encryption Module		AIM-VPN/EPII-Plus Hardware Encryption Module		
Cisco 3660 Cisco 3745	AIM-VPN/HPII Hardware Encryption Module		AIM-VPN/HPII-Plus Hardware Encryption Module		

For more information on AIMs and NM, see *Installing Advanced Integration Modules in Cisco 2600 Series, Cisco 3600 Series, and Cisco 3700 Series Routers*.

Supported Switching Paths

Table 45 lists the supported switching paths that work with IPSec.

Table 45 Supported Switching Paths for IPSec

Switching Paths	Examples
Process switching	`interface ethernet0/0` ` no ip route-cache`
Fast switching	`interface ethernet0/0` ` ip route-cache` `! Ensure that you will not hit flow switching.` ` no ip route-cache flow` `! Disable CEF for the interface, which supercedes global CEF.` ` no ip route-cache cef`
Cisco Express Forwarding (CEF)	`ip cef` `interface ethernet0/0` ` ip route-cache` `! Ensure that you will not hit flow switching.` ` no ip route-cache flow`
Fast-flow switching	`interface ethernet0/0` ` ip route-cache` `! Enable flow switching` `p route-cache flow` `! Disable CEF for the interface.` `  no ip route-cache cef`
CEF-flow switching	`! Enable global CEF.` `ip cef` `interface ethernet0/0` ` ip route-cache` ` ip route-cache flow` `! Enable CEF for the interface` ` ip route-cache cef`

Supported Encapsulation

IPSec works with the following serial encapsulations: High-Level Data-Links Control (HDLC), PPP, and Frame Relay.

IPSec also works with the Generic Routing Encapsulation (GRE) and IPinIP Layer 3, Layer 2 Forwarding (L2F), Layer 2 Tunneling Protocol (L2TP), Data Link Switching+ (DLSw+), and SRB tunneling protocols; however, multipoint tunnels are not supported. Other Layer 3 tunneling protocols may not be supported for use with IPSec.

Because the IPSec Working Group has not yet addressed the issue of group key distribution, IPSec currently cannot be used to protect group traffic (such as broadcast or multicast traffic).

IPSec Functionality Overview

IPSec provides the following network security services. (In general, local security policy dictates the use of one or more of these services.)

- Data Confidentiality—The IPSec sender can encrypt packets before transmitting them across a network.
- Data Integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.

- Data Origin Authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service is dependent upon the data integrity service.
- Anti-Replay—The IPSec receiver can detect and reject replayed packets.

IPSec provides secure *tunnels* between two peers, such as two routers. You define which packets are considered sensitive and should be sent through these secure tunnels, and you define the parameters that should be used to protect these sensitive packets, by specifying characteristics of these tunnels. Then, when the IPSec peer recognizes such a sensitive packet, the peer sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer. (The use of the term *tunnel* in this chapter does not refer to using IPSec in tunnel mode.)

More accurately, these *tunnels* are sets of security associations (SAs) that are established between two IPSec peers. The SAs define which protocols and algorithms should be applied to sensitive packets and specify the keying material to be used by the two peers. SAs are unidirectional and are established per security protocol (AH or ESP).

With IPSec you define what traffic should be protected between two IPSec peers by configuring access lists and applying these access lists to interfaces by way of crypto map sets. Therefore, traffic may be selected on the basis of source and destination address, and optionally Layer 4 protocol, and port. (The access lists used for IPSec are used only to determine which traffic should be protected by IPSec, not which traffic should be blocked or permitted through the interface. Separate access lists define blocking and permitting at the interface.)

A crypto map set can contain multiple entries, each with a different access list. The crypto map entries are searched in order—the router attempts to match the packet to the access list specified in that entry.

When a packet matches a **permit** entry in a particular access list, and the corresponding crypto map entry is tagged as **cisco**, connections are established if necessary. If the crypto map entry is tagged as **ipsec-isakmp**, IPSec is triggered. If no SA exists that IPSec can use to protect this traffic to the peer, IPSec uses IKE to negotiate with the remote peer to set up the necessary IPSec SAs on behalf of the data flow. The negotiation uses information specified in the crypto map entry as well as the data flow information from the specific access list entry. (The behavior is different for dynamic crypto map entries. See the section "Creating Dynamic Crypto Maps" section later in this module.)

If the crypto map entry is tagged as **ipsec-manual**, IPSec is triggered. If no SA exists that IPSec can use to protect this traffic to the peer, the traffic is dropped. In this case, the SAs are installed via the configuration, without the intervention of IKE. If the SAs did not exist, IPSec did not have all of the necessary pieces configured.

Once established, the set of SAs (outbound, to the peer) is then applied to the triggering packet and to subsequent applicable packets as those packets exit the router. "Applicable" packets are packets that match the same access list criteria that the original packet matched. For example, all applicable packets could be encrypted before being forwarded to the remote peer. The corresponding inbound SAs are used when processing the incoming traffic from that peer.

Multiple IPSec tunnels can exist between two peers to secure different data streams, with each tunnel using a separate set of SAs. For example, some data streams might be authenticated only while other data streams must both be encrypted and authenticated.

Access lists associated with IPSec crypto map entries also represent which traffic the router requires to be protected by IPSec. Inbound traffic is processed against the crypto map entries—if an unprotected packet matches a **permit** entry in a particular access list associated with an IPSec crypto map entry, that packet is dropped because it was not sent as an IPSec-protected packet.

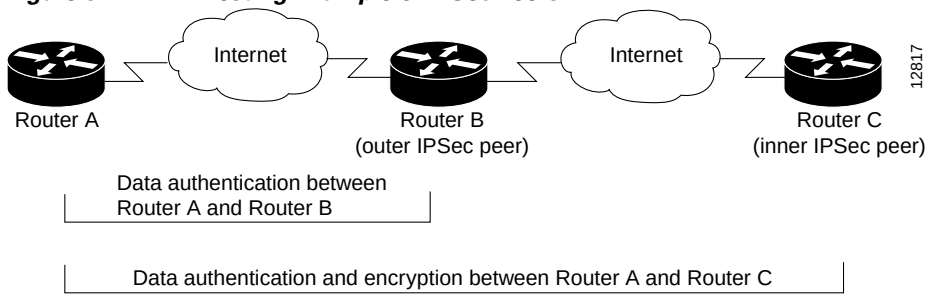
Crypto map entries also include transform sets. A transform set is an acceptable combination of security protocols, algorithms, and other settings to apply to IPSec protected traffic. During the IPSec SA negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

IPSec Traffic Nested to Multiple Peers

You can nest IPSec traffic to a series of IPSec peers. For example, in order for traffic to traverse multiple firewalls (these firewalls have a policy of not letting through traffic that they have not authenticated), the router must establish IPSec tunnels with each firewall in turn. The “nearer” firewall becomes the “outer” IPSec peer.

In the example shown in Figure 54, Router A encapsulates the traffic destined for Router C in IPSec (Router C is the inner IPSec peer). However, before Router A can send this traffic, it must first reencapsulate this traffic in IPSec in order to send it to Router B (Router B is the “outer” IPSec peer).

Figure 54 Nesting Example of IPSec Peers



It is possible for the traffic between the “outer” peers to have one kind of protection (such as data authentication) and for traffic between the “inner” peers to have different protection (such as both data authentication and encryption).

How to Configure IPSec VPNs

Perform the tasks in the following sections to create IPSec VPNs:

- Creating Crypto Access Lists, page 8
- Defining Transform Sets: A Combination of Security Protocols and Algorithms, page 14
- Creating Crypto Map Sets, page 17
- Applying Crypto Map Sets to Interfaces, page 30

Creating Crypto Access Lists

To create crypto access lists that define which traffic is protected via IPSec tunnels, you should understand the following concepts:

- Crypto Access List Overview
- When to Use the permit and deny Keywords in Crypto Access Lists
- Mirror Image Crypto Access Lists at Each IPSec Peer
- When to Use the any Keyword in Crypto Access Lists

Crypto Access List Overview

Crypto access lists are used to define which IP traffic is protected by crypto and which traffic is not protected by crypto. (These access lists are *not* the same as regular access lists, which determine what traffic to forward or block at an interface.) For example, access lists can be created to protect all IP traffic between Subnet A and Subnet Y or Telnet traffic between Host A and Host B.

The access lists themselves are not specific to IPSec. It is the crypto map entry referencing the specific access list that defines whether IPSec processing is applied to the traffic matching a **permit** in the access list.

Crypto access lists associated with IPSec crypto map entries have four primary functions:

- Select outbound traffic to be protected by IPSec (permit = protect).
- Indicate the data flow to be protected by the new SAs (specified by a single **permit** entry) when initiating negotiations for IPSec security associations.
- Process inbound traffic in order to filter out and discard traffic that should have been protected by IPSec.
- Determine whether or not to accept requests for IPSec security associations on behalf of the requested data flows when processing IKE negotiation from the IPSec peer.
- Negotiation is performed only for **ipsec-isakmp** crypto map entries. In order to be accepted, if the peer initiates the IPSec negotiation, it must specify a data flow that is "permitted" by a crypto access list associated with an **ipsec-isakmp** crypto map entry.

If you want certain traffic to receive one combination of IPSec protection (for example, authentication only) and other traffic to receive a different combination of IPSec protection (for example, both authentication and encryption), you need to create two different crypto access lists to define the two different types of traffic. These different access lists are then used in different crypto map entries which specify different IPSec policies.

When to Use the permit and deny Keywords in Crypto Access Lists

Crypto protection can be permitted or denied for certain IP traffic in a crypto access list as follows:

- To protect IP traffic that matches the specified policy conditions in its corresponding crypto map entry, use the **permit** keyword in an access list.
- To refuse protection for IP traffic that matches the specified policy conditions in its corresponding crypto map entry, use the **deny** keyword in an access list.

Note IP traffic is not protected by crypto if it is refused protection in all of the crypto map entries for an interface.

After the corresponding crypto map entry is defined and the crypto map set is applied to the interface, the defined crypto access list is applied to an interfacethe defined crypto access list is applied to an interface. Different access lists must be used in different entries of the same crypto map set. However, both inbound and outbound traffic is evaluated against the same "outbound" IPSec access list. Therefore, the access list's criteria is applied in the forward direction to traffic exiting your router and in the reverse direction to traffic entering your router.

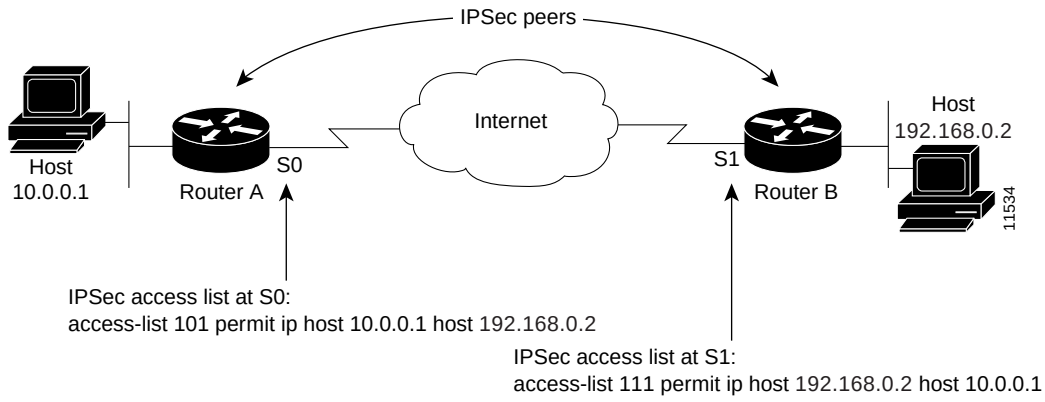
In Figure 55, IPSec protection is applied to traffic between Host 10.0.0.1 and Host 192.168.0.2 as the data exits Router A's S0 interface en route to Host 192.168.0.2. For traffic from Host 10.0.0.1 to Host 192.168.0.2, the access list entry on Router A is evaluated as follows:

```
source = host 10.0.0.1
dest = host 192.168.0.2
```

For traffic from Host 192.168.0.2 to Host 10.0.0.1, that same access list entry on Router A is evaluated as follows:

```
source = host 192.168.0.2
dest = host 10.0.0.1
```

Figure 55 How Crypto Access Lists Are Applied for Processing IPSec



Traffic exchanged between hosts 10.0.0.1 and 192.168.0.2
is protected between Router A S0 and Router B S1

If you configure multiple statements for a given crypto access list that is used for IPSec, in general the first **permit** statement that is matched is the statement used to determine the scope of the IPSec SA. That is, the IPSec SA is set up to protect traffic that meets the criteria of the matched statement only. Later, if traffic matches a different **permit** statement of the crypto access list, a new, separate IPSec SA is negotiated to protect traffic matching the newly matched access list statement.

Any unprotected inbound traffic that matches a **permit** entry in the crypto access list for a crypto map entry flagged as IPSec is dropped, because this traffic was expected to be protected by IPSec.

Note If you view your router's access lists by using a command such as **show ip access-lists**, *all* extended IP access lists are shown in the command output. This display output includes extended IP access lists that are used for traffic filtering purposes and those that are used for crypto. The **show** command output does not differentiate between the different uses of the extended access lists.

The following example shows that if overlapping networks are used, then the most specific networks are defined in crypto sequence numbers before less specific networks are defined. In this example, the more specific network is covered by the crypto map sequence number 10, followed by the less specific network in the crypto map, which is sequence number 20.

```
crypto map mymap 10 ipsec-isakmp
 set peer 192.168.1.1
 set transform-set test
 match address 101
crypto map mymap 20 ipsec-isakmp
 set peer 192.168.1.2
 set transform-set test
 match address 102
```

```
access-list 101 permit ip 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255

access-list 102 permit ip 10.0.0.0 0.255.255.255 172.16.0.0 0.15.255.255
```

The following example shows how having a **deny** keyword in one crypto map sequence number and having a **permit** keyword for the same subnet and IP range in another crypto map sequence number is not supported.

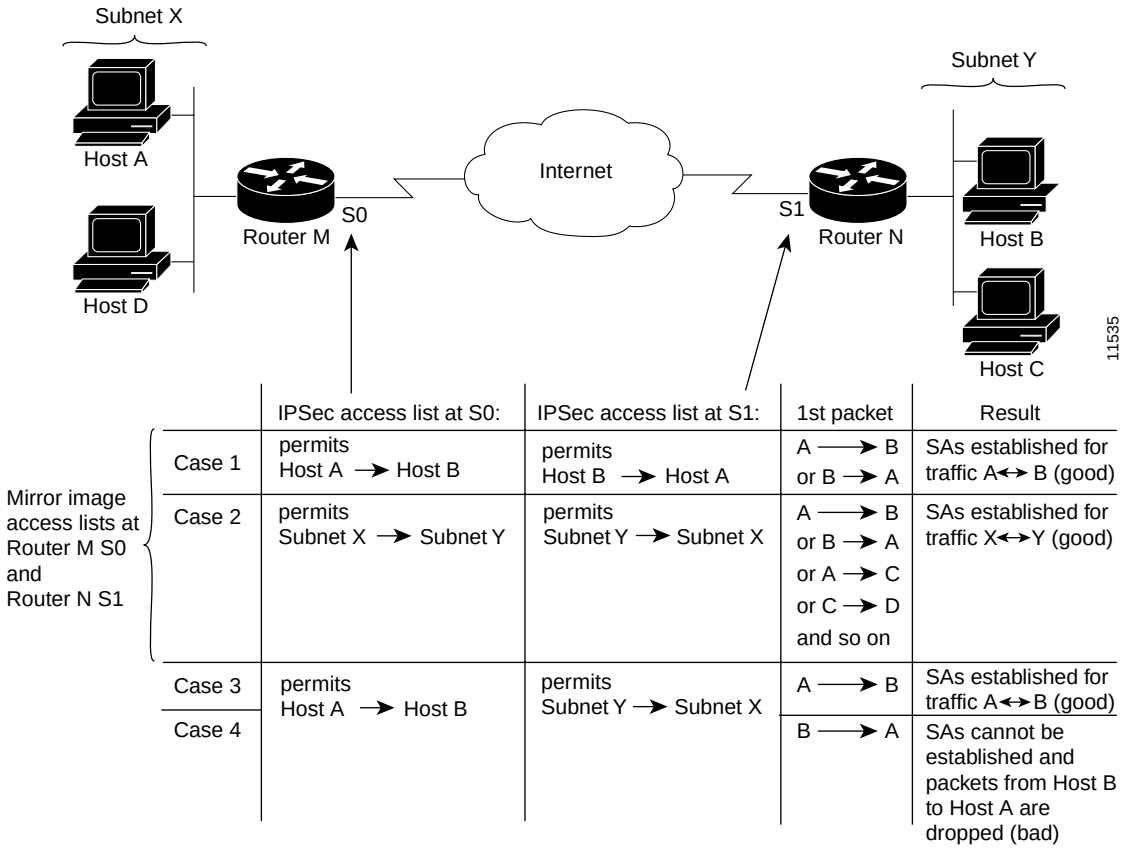
```
crypto map mymap 10 ipsec-isakmp
 set peer 192.168.1.1
 set transform-set test
 match address 101
crypto map mymap 20 ipsec-isakmp
 set peer 192.168.1.2
 set transform-set test
 match address 102

access-list 101 deny   ip 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255
access-list 101 permit ip 10.0.0.0 0.255.255.255 172.16.0.0 0.15.255.255

access-list 102 permit ip 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255
```

Mirror Image Crypto Access Lists at Each IPSec Peer

Cisco recommends that for every crypto access list specified for a static crypto map entry that you define at the local peer, you define a “mirror image” crypto access list at the remote peer. This ensures that traffic that has IPSec protection applied locally can be processed correctly at the remote peer. (The crypto map entries themselves must also support common transforms and must refer to the other system as a peer.)

Figure 56 shows some sample scenarios when you have mirror image access lists and when you do not have mirror image access lists.

Figure 56 Mirror Image vs. Nonmirror Image Crypto Access Lists (for IPSec)



		IPSec access list at S0:	IPSec access list at S1:	1st packet	Result
Mirror image access lists at Router M S0 and Router N S1	Case 1	permits Host A → Host B	permits Host B → Host A	A → B or B → A	SAs established for traffic A↔B (good)
	Case 2	permits Subnet X → Subnet Y	permits Subnet Y → Subnet X	A → B or B → A or A → C or C → D and so on	SAs established for traffic X↔Y (good)
	Case 3	permits Host A → Host B	permits Subnet Y → Subnet X	A → B	SAs established for traffic A↔B (good)
	Case 4			B → A	SAs cannot be established and packets from Host B to Host A are dropped (bad)

As Figure 56 indicates, IPSec SAs can be established as expected whenever the two peers' crypto access lists are mirror images of each other. However, an IPSec SA can be established only some of the time when the access lists are not mirror images of each other. This can happen in the case where an entry in one peer's access list is a subset of an entry in the other peer's access list, such as shown in Cases 3 and 4 of Figure 56. IPSec SA establishment is critical to IPSec—without SAs, IPSec does not work, causing any packets matching the crypto access list criteria to be silently dropped instead of being forwarded with IPSec.

In Figure 56, an SA cannot be established in Case 4. This is because SAs are always requested according to the crypto access lists at the initiating packet's end. In Case 4, Router N requests that all traffic between Subnet X and Subnet Y be protected, but this is a superset of the specific flows permitted by the crypto access list at Router M so the request is therefore not permitted. Case 3 works because Router M's request is a subset of the specific flows permitted by the crypto access list at Router N.

Because of the complexities introduced when crypto access lists are not configured as mirror images at peer IPSec devices, Cisco strongly encourages you to use mirror image crypto access lists.

When to Use the any Keyword in Crypto Access Lists

When you create crypto access lists, using the **any** keyword could cause problems. Cisco discourages the use of the **any** keyword to specify source or destination addresses.

The **any** keyword in a **permit** statement is discouraged when you have multicast traffic flowing through the IPSec interface; the **any** keyword can cause multicast traffic to fail.

The **permit any any** statement is strongly discouraged, because this causes all outbound traffic to be protected (and all protected traffic sent to the peer specified in the corresponding crypto map entry) and requires protection for all inbound traffic. Then, all inbound packets that lack IPSec protection are silently dropped, including packets for routing protocols, Network Time Protocol (NTP), echo, echo response, and so on.

You need to be sure you define which packets to protect. If you *must* use the **any** keyword in a **permit** statement, you must preface that statement with a series of **deny** statements to filter out any traffic (that would otherwise fall within that **permit** statement) that you do not want to be protected.

Also, use of **any** keyword in access control lists (ACLs) with reverse route injection (RRI) is not supported. (For more information on RRI, see the section "Creating Crypto Map Sets.")

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**log**]

   or

   ip access-list extended *name*
4. Repeat Step 3 for each crypto access list you want to create.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**log**] **Example:** `Router(config)# access-list 100 permit ip 10.0.68.0 0.0.0.255 10.1.1.0 0.0.0.255` or **ip access-list extended** *name* **Example:** `Router(config)# ip access-list extended vpn-tunnel`	Specifies conditions to determine which IP packets are protected.[1] Enable or disable crypto for traffic that matches these conditions. **Tip** Cisco recommends that you configure "mirror image" crypto access lists for use by IPSec and that you avoid using the **any** keyword.
Step 4	—	Repeat Step 3 for each crypto access list you want to create.

1. You specify conditions using an IP access list designated by either a number or a name. The **access-list** command designates a numbered extended access list; the **ip access-list extended** command designates a named access list.

What to Do Next

After at least one crypto access list is created, a transform set needs to be defined as described in the section "Defining Transform Sets: A Combination of Security Protocols and Algorithms."

Next the crypto access lists need to be associated to particular interfaces when you configure and apply crypto map sets to the interfaces are configured and applied (following instructions in the sections "Creating Crypto Map Sets" and "Applying Crypto Map Sets to Interfaces").

Defining Transform Sets: A Combination of Security Protocols and Algorithms

Perform this task to define a transform set that is to be used by the IPSec peers during IPSec security association negotiations with IKE.

Restrictions

If you are specifying SEAL encryption, note the following restrictions:

- Your router and the other peer must not have hardware IPSec encryption.
- Your router and the other peer must support IPSec.
- Your router and the other peer must support the k9 subsystem.
- SEAL encryption is available only on Cisco equipment. Therefore, interoperability is not possible.

About Transform Sets

A transform set represents a certain combination of security protocols and algorithms. During the IPSec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

You can specify multiple transform sets, and then specify one or more of these transform sets in a crypto map entry. The transform set defined in the crypto map entry is used in the IPSec SA negotiation to protect the data flows specified by that crypto map entry's access list.

During IPSec security association negotiations with IKE, the peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and is applied to the protected traffic as part of both peers' IPSec SAs. (With manually established SAs, there is no negotiation with the peer, so both sides must specify the same transform set.)

If you change a transform set definition, the change is only applied to crypto map entries that reference the transform set. The change is not applied to existing security associations, but is used in subsequent negotiations to establish new SAs. If you want the new settings to take effect sooner, you can clear all or part of the SA database by using the **clear crypto sa** command

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]]

4. **mode** [**tunnel** | **transport**]
5. **exit**
6. **clear crypto sa** [peer {*ip-address* | **peer-name**} | sa map *map-name* | sa entry *destination-address protocol spi*]
7. **show crypto ipsec transform-set** [**tag** *transform-set-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]] **Example:** `Router(config)# crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac`	Defines a transform set and enters crypto transform configuration mode. There are complex rules defining which entries you can use for the transform arguments. These rules are explained in the command description for the **crypto ipsec transform-set** command, and Table 46 provides a list of allowed transform combinations.
Step 4	**mode** [**tunnel** | **transport**] **Example:** `Router(cfg-crypto-tran)# mode transport`	(Optional) Changes the mode associated with the transform set. The mode setting is applicable only to traffic whose source and destination addresses are the IPSec peer addresses; it is ignored for all other traffic. (All other traffic is in tunnel mode only.)
Step 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

	Command or Action	Purpose
Step 6	**clear crypto sa** [**peer** {*ip-address* | *peer-name*} | **sa map** *map-name* | **sa entry** *destination-address protocol spi*] **Example:** `Router# clear crypto sa`	(Optional) Clears existing IPSec security associations so that any changes to a transform set takes effect on subsequently established security associations. Manually established SAs are reestablished immediately. • Using the **clear crypto sa** command without parameters clear out the full SA database, which clears out active security sessions. • You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database.
Step 7	**show crypto ipsec transform-set** [**tag** *transform-set-name*] **Example:** `Router# show crypto ipsec transform-set`	(Optional) Displays the configured transform sets.

Table 46 shows allowed transform combinations.

Table 46 Allowed Transform Combinations

Transform Type	Transform	Description
AH Transform	**ah-md5-hmac**	AH with the MD5 (a Hash-based Message Authentication Code [HMAC] variant) authentication algorithm
	ah-sha-hmac	AH with the SHA (an HMAC variant) authentication algorithm
ESP Encryption Transform	**esp-aes**	ESP with the 128-bit AES encryption algorithm
	esp-aes 192	ESP with the 192-bit AES encryption algorithm
	esp-aes 256	ESP with the 256-bit AES encryption algorithm
	esp-des	ESP with the 56-bit DES encryption algorithm
	esp-3des	ESP with the 168-bit DES encryption algorithm (3DES or Triple DES)
	esp-null	Null encryption algorithm
	esp-seal	ESP with the 160-bit SEAL encryption algorithm.

Table 46 Allowed Transform Combinations (continued)

Transform Type	Transform	Description
ESP Authentication Transform	**esp-md5-hmac**	ESP with the MD5 (HMAC variant) authentication algorithm
	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm
IP Compression Transform	**comp-lzs**	IP compression with the LZS algorithm

What to Do Next

After you have defined a transform set, you should create a crypto map as specified in the section "Creating Crypto Map Sets."

Creating Crypto Map Sets

See one of the following sections, as appropriate, to help create crypto map sets:

- Creating Static Crypto Maps
- Creating Dynamic Crypto Maps
- Creating Crypto Map Entries to Establish Manual SAs

Prerequisites

Before you create crypto map entries, you should determine which type of crypto map—static, dynamic, or manual—best addresses the needs of your network. You should also understand the following concepts:

- About Crypto Maps
- Load Sharing Among Crypto Maps
- Crypto Map Guidelines

About Crypto Maps

Crypto map entries created for IPSec pull together the various parts used to set up IPSec SAs, including:

- Which traffic should be protected by IPSec (per a crypto access list)
- The granularity of the flow to be protected by a set of SAs
- Where IPSec-protected traffic should be sent (who the remote IPSec peer is)
- The local address to be used for the IPSec traffic (See the section "Applying Crypto Map Sets to Interfaces" for more details.)
- What IPSec SA should be applied to this traffic (selecting from a list of one or more transform sets)
- Whether SAs are manually established or are established via IKE
- Other parameters that might be necessary to define an IPSec SA

How Crypto Maps Work

Crypto map entries with the same crypto map name (but different map sequence numbers) are grouped into a crypto map set. Later, you apply these crypto map sets to interfaces; then, all IP traffic passing through the interface is evaluated against the applied crypto map set. If a crypto map entry sees outbound IP traffic that should be protected and the crypto map specifies the use of IKE, a SA is negotiated with the remote peer according to the parameters included in the crypto map entry; otherwise, if the crypto map entry specifies the use of manual SAs, an SA should have already been established via configuration. (If a dynamic crypto map entry sees outbound traffic that should be protected and no security association exists, the packet is dropped.)

The policy described in the crypto map entries is used during the negotiation of SAs. If the local router initiates the negotiation, it uses the policy specified in the static crypto map entries to create the offer to be sent to the specified IPSec peer. If the IPSec peer initiates the negotiation, the local router checks the policy from the static crypto map entries, as well as any referenced dynamic crypto map entries to decide whether to accept or reject the peer's request (offer).

For IPSec to succeed between two IPSec peers, both peers' crypto map entries must contain compatible configuration statements.

Compatible Crypto Maps: Establishing an SA

When two peers try to establish a SA, they must each have at least one crypto map entry that is compatible with one of the other peer's crypto map entries. For two crypto map entries to be compatible, they must at least meet the following criteria:

- The crypto map entries must contain compatible crypto access lists (for example, mirror image access lists). In the case where the responding peer is using dynamic crypto maps, the entries in the local crypto access list must be "permitted" by the peer's crypto access list.
- The crypto map entries must each identify the other peer (unless the responding peer is using dynamic crypto maps).
- The crypto map entries must have at least one transform set in common.

Load Sharing Among Crypto Maps

You can define multiple remote peers using crypto maps to allow for load sharing. Load sharing is useful because if one peer fails, there continues to be a protected path. The peer that packets are actually sent to is determined by the last peer that the router heard from (received either traffic or a negotiation request from) for a given data flow. If the attempt fails with the first peer, IKE tries the next peer on the crypto map list.

If you are not sure how to configure each crypto map parameter to guarantee compatibility with other peers, you might consider configuring dynamic crypto maps as described in the section "Creating Dynamic Crypto Maps." Dynamic crypto maps are useful when the establishment of the IPSec tunnels is initiated by the remote peer (such as in the case of an IPSec router fronting a server). They are not useful if the establishment of the IPSec tunnels is locally initiated, because the dynamic crypto maps are policy templates, not complete statements of policy. (Although the access lists in any referenced dynamic crypto map entry are used for crypto packet filtering.)

Crypto Map Guidelines

You can apply only one crypto map set to a single interface. The crypto map set can include a combination of IPSec/IKE and IPSec/manual entries. Multiple interfaces can share the same crypto map set if you want to apply the same policy to multiple interfaces.

If you create more than one crypto map entry for a given interface, use the *seq-num* argument of each map entry to rank the map entries: the lower the *seq-num* argument, the higher the priority. At the interface that has the crypto map set, traffic is evaluated against higher priority map entries first.

You must create multiple crypto map entries for a given interface if any of the following conditions exist:

- If different data flows are to be handled by separate IPSec peers.
- If you want to apply different IPSec security to different types of traffic (to the same or separate IPSec peers); for example, if you want traffic between one set of subnets to be authenticated, and traffic between another set of subnets to be both authenticated and encrypted. In this case the different types of traffic should have been defined in two separate access lists, and you must create a separate crypto map entry for each crypto access list.
- If you are not using IKE to establish a particular set of security associations, and want to specify multiple access list entries, you must create separate access lists (one per **permit** entry) and specify a separate crypto map entry for each access list.

Creating Static Crypto Maps

When IKE is used to establish SAs, the IPSec peers can negotiate the settings they use for the new security associations. This means that you can specify lists (such as lists of acceptable transforms) within the crypto map entry.

Perform this task to create crypto map entries that use IKE to establish the SAs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **match address** *access-list-id*
5. **set peer** {*hostname* | *ip-address*}
6. **set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]
7. **set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*}
8. **set security-association level per-host**
9. **set pfs** [**group1** | **group2** | **group5**]
10. **exit**
11. exit
12. show crypto map [**interface** *interface* | tag *map-name*]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router(config)# crypto map static-map 1 ipsec-isakmp`	Names the crypto map entry to create (or modify), and enters crypto map configuration mode.
Step 4	**match address** *access-list-id* **Example:** `Router(config-crypto-m)# match address vpn-tunnel`	Names an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry.
Step 5	**set peer** {*hostname* | *ip-address*} **Example:** `Router(config-crypto-m)# set-peer 192.168.101.1`	Specifies a remote IPSec peer, the peer to which IPSec protected traffic can be forwarded. Repeat for multiple remote peers.
Step 6	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config-crypto-m)# set transform-set aesset`	Specifies which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first).
Step 7	**set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*} **Example:** `Router (config-crypto-m)# set security-association lifetime seconds 2700`	(Optional) Specifies a SA lifetime for the crypto map entry. By default, the SAs of the crypto map are negotiated according to the global lifetimes.
Step 8	**set security-association level per-host** **Example:** `Router(config-crypto-m)# set security-association level per-host`	(Optional) Specifies that separate SAs should be established for each source and destination host pair. By default, a single IPSec "tunnel" can carry traffic for multiple source hosts and multiple destination hosts. ⚠ **Caution** Use this command with care, because multiple streams between given subnets can rapidly consume resources.

	Command	Purpose
Step 9	**set pfs** [**group1** | **group2** | **group 5**] **Example:** `Router(config-crypto-m)# set pfs group2`	(Optional) Specifies that IPSec either should ask for perfect forward secrecy (PFS) when requesting new SAs for this crypto map entry or should demand PFS in requests received from the IPSec peer. By default, PFS is not requested. If no group is specified with this command, group1 is used as the default.
Step 10	**exit** **Example:** `Router(config-crypto-m)# exit`	Exits crypto-map configuration mode.
Step 11	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 12	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Router# show crypto map`	Displays your crypto map configuration.

Troubleshooting Tips

Certain configuration changes take effect only when negotiating subsequent SAs. If you want the new settings to take immediate effect, you must clear the existing SAs so that they are re-established with the changed configuration. If the router is actively processing IPSec traffic, it is desirable to clear only the portion of the SA database that would be affected by the configuration changes (that is, clear only the SAs established by a given crypto map set). Clearing the full SA database should be reserved for large-scale changes, or when the router is processing very little other IPSec traffic.

To clear IPSec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters clears out the full SA database, which clears active security sessions.)

What to Do Next

After you have successfully created a static crypto map, you must apply the crypto map set to each interface through which IPSec traffic flows. To complete this task, see the section "Applying Crypto Map Sets to Interfaces."

Creating Dynamic Crypto Maps

Dynamic crypto maps can ease IPSec configuration and are recommended for use with networks where the peers are not always predetermined. To create dynamic crypto maps, you should understand the following concepts:

- Dynamic Crypto Maps Overview
- Tunnel Endpoint Discovery (TED)

Dynamic Crypto Maps Overview

Dynamic crypto maps are only available for use by IKE.

A dynamic crypto map entry is essentially a static crypto map entry without all the parameters configured. It acts as a policy template where the missing parameters are later dynamically configured (as the result of an IPSec negotiation) to match a remote peer's requirements. This allows remote peers to exchange IPSec traffic with the router even if the router does not have a crypto map entry specifically configured to meet all of the remote peer's requirements.

Dynamic crypto maps are not used by the router to initiate new IPSec security associations with remote peers. Dynamic crypto maps are used when a remote peer tries to initiate an IPSec security association with the router. Dynamic crypto maps are also used in evaluating traffic.

A dynamic crypto map set is included by reference as part of a crypto map set. Any crypto map entries that reference dynamic crypto map sets should be the lowest priority crypto map entries in the crypto map set (that is, have the highest sequence numbers) so that the other crypto map entries are evaluated first; that way, the dynamic crypto map set is examined only when the other (static) map entries are not successfully matched.

If the router accepts the peer's request, at the point that it installs the new IPSec security associations it also installs a temporary crypto map entry. This entry is filled in with the results of the negotiation. At this point, the router performs normal processing, using this temporary crypto map entry as a normal entry, even requesting new security associations if the current ones are expiring (based upon the policy specified in the temporary crypto map entry). Once the flow expires (that is, all of the corresponding security associations expire), the temporary crypto map entry is then removed.

For both static and dynamic crypto maps, if unprotected inbound traffic matches a **permit** statement in an access list, and the corresponding crypto map entry is tagged as "IPSec," then the traffic is dropped because it is not IPSec-protected. (This is because the security policy as specified by the crypto map entry states that this traffic must be IPSec-protected.)

For static crypto map entries, if outbound traffic matches a **permit** statement in an access list and the corresponding SA is not yet established, the router initiates new SAs with the remote peer. In the case of dynamic crypto map entries, if no SA existed, the traffic would simply be dropped (because dynamic crypto maps are not used for initiating new SAs).

Note Use care when using the **any** keyword in **permit** entries in dynamic crypto maps. If it is possible for the traffic covered by such a **permit** entry to include multicast or broadcast traffic, the access list should include **deny** entries for the appropriate address range. Access lists should also include **deny** entries for network and subnet broadcast traffic, and for any other traffic that should not be IPSec protected.

Restrictions for Dynamic Crypto Maps

Dynamic crypto map entries specify crypto access lists that limit traffic for which IPSec SAs can be established. A dynamic crypto map entry that does not specify an access list is ignored during traffic filtering. A dynamic crypto map entry with an empty access list causes traffic to be dropped. If there is only one dynamic crypto map entry in the crypto map set, it must specify acceptable transform sets.

Tunnel Endpoint Discovery (TED)

Defining a dynamic crypto map allows only the receiving router to dynamically determine an IPSec peer. TED allows the initiating router to dynamically determine an IPSec peer for secure IPSec communications.

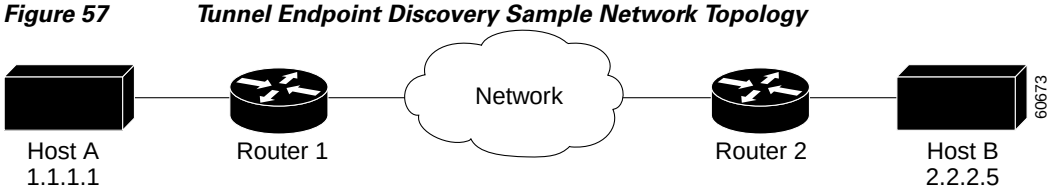
Dynamic TED helps to simplify IPSec configuration on the individual routers within a large network. Each node has a simple configuration that defines the local network that the router is protecting and the required IPSec transforms.

To have a large, fully-meshed network *without* TED, each peer needs to have static crypto maps to every other peer in the network. For example, if there are 100 peers in a large, fully-meshed network, each router needs 99 static crypto maps for each of its peers. With TED, only a single dynamic crypto map with TED enabled is needed because the peer is discovered dynamically. Thus, static crypto maps do not need to be configured for each peer.

Note TED helps only in discovering peers; otherwise, TED does not function any differently than normal IPSec. TED does not improve the scalability of IPSec (in terms of performance or the number of peers or tunnels).

Figure 57 and the corresponding steps explain a sample TED network topology.

Figure 57 Tunnel Endpoint Discovery Sample Network Topology



Step 1 Host A sends a packet that is destined for Host B.

Step 2 Router 1 intercepts and reads the packet. According to the IKE policy, Router 1 contains the following information: the packet must be encrypted, there are no SAs for the packet, and TED is enabled. Thus, Router 1 drops the packet and sends a TED probe into the network. (The TED probe contains the IP address of Host A (as the source IP address) and the IP address of Host B (as the destination IP address) embedded in the payload.

Step 3 Router 2 intercepts the TED probe and checks the probe against the ACLs that it protects; after the probe matches an ACL, it is recognized as a TED probe for proxies that the router protects. It then sends a TED reply with the IP address of Host B (as the source IP address) and the IP address of Host A (as the destination IP address) embedded in the payload.

Step 4 Router 1 intercepts the TED reply and checks the payloads for the IP address and half proxy of Router 2. It then combines the source side of its proxy with the proxy found in the second payload and initiates an IKE session with Router 2; thereafter, Router 1 initiates an IPSec session with Router 2.

Note IKE cannot occur until the peer is identified.

TED Versions

The following table lists the available TED versions:

Version	First Available Release	Description
TEDv1	12.0(5)T	Performs basic TED functionality on nonredundant networks.
TEDv2	12.1M	Enhanced to work with redundant networks with paths through multiple security gateways between the source and the destination.
TEDv3	12.2M	Enhanced to allow non-IP-related entries to be used in the access list.

TED Restrictions

TED has the following restrictions:

- It is Cisco proprietary.
- It is available only on dynamic crypto maps. (The dynamic crypto map template is based on the dynamic crypto map performing peer discovery. Although there are no access-list restrictions on the dynamic crypto map template, the dynamic crypto map template should cover data sourced from the protected traffic and the receiving router using the **any** keyword. When using the **any** keyword, include explicit **deny** statements to exempt routing protocol traffic prior to entering the **permit any** command.)
- TED works only in tunnel mode; that is, it does not work in transport mode.
- It is limited by the performance and scalability of limitation of IPSec on each individual platform.

  Note Enabling TED slightly decreases the general scalability of IPSec because of the set-up overhead of peer discovery, which involves an additional “round-trip” of IKE messages (TED probe and reply). Although minimal, the additional memory used to store data structures during the peer discovery stage adversely affects the general scalability of IPSec.

- The IP addresses must be able to be routed within the network.
- The access list used in the crypto map for TED can only contain IP-related entries—TCP, UDP, or any other protocol cannot be used in the access list.

  Note This restriction is no longer applicable in TEDv3.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic-map** *dynamic-map-name dynamic-seq-num*
4. **set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]
5. **match address** *access-list-id*
6. **set peer** {*hostname* | *ip-address*}
7. **set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*}

8. **set pfs** [**group1** | **group2** | **group5**]
9. **exit**
10. **exit**
11. **show crypto dynamic-map** [**tag** *map-name*]
12. **configure terminal**
13. **crypto map** *map-name seq-num* **ipsec-isakmp dynamic** *dynamic-map-name* [**discover**]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **Example:** `Router(config)# crypto dynamic-map test-map 1`	Creates a dynamic crypto map entry and enters crypto map configuration mode.
Step 4	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config-crypto-m)# set transform-set aesset`	Specifies which transform sets are allowed for the crypto map entry. List multiple transform sets in order of priority (highest priority first). This is the only configuration statement required in dynamic crypto map entries.

	Command	Purpose
Step 5	**match address** *access-list-id* **Example:** `Router(config-crypto-m)# match address 101`	(Optional) Accesses list number or name of an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry. **Note** Although access-lists are optional for dynamic crypto maps, they are highly recommended. If this is configured, the data flow identity proposed by the IPSec peer must fall within a **permit** statement for this crypto access list. If this is not configured, the router accepts any data flow identity proposed by the IPSec peer. However, if this is configured but the specified access list does not exist or is empty, the router drops all packets. This is similar to static crypto maps because they also require that an access list be specified. Care must be taken if the **any** keyword is used in the access list, because the access list is used for packet filtering as well as for negotiation. You must configure a match address; otherwise, the behavior is not secure, and you cannot enable TED because packets are sent in the clear (unencrypted.)
Step 6	**set peer** {*hostname* | *ip-address*} **Example:** `Router(config-crypto-m)# set peer 192.168.101.1`	(Optional) Specifies a remote IPSec peer. Repeat for multiple remote peers. **Note** This is rarely configured in dynamic crypto map entries. Dynamic crypto map entries are often used for unknown remote peers.
Step 7	**set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*} **Example:** `Router (config-crypto-m)# set security-association lifetime seconds 7200`	(Optional) Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IP Security SAs.
Step 8	**set pfs** [**group1** | **group2** | **group5**] **Example:** `Router(config-crypto-m)# set pfs group2`	(Optional) Specifies that IPSec should ask for PFS when requesting new security associations for this crypto map entry or should demand PFS in requests received from the IPSec peer. By default, PFS is not requested. If no group is specified with this command, **group1** is used as the default.
Step 9	**exit** **Example:** `Router(config-crypto-m)# exit`	Exits crypto-map configuration mode and returns to global configuration mode.

	Command	Purpose
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show crypto dynamic-map** [*tag map-name*] **Example:** `Router# show crypto dynamic-map`	(Optional) Displays information about dynamic crypto maps.
Step 12	**configure terminal** **Example:** `Router# configure terminal`	Returns to global configuration mode.
Step 13	**crypto map** *map-name seq-num* **ipsec-isakmp dynamic** *dynamic-map-name* [**discover**] **Example:** `Router(config)# crypto map static-map 1 ipsec-isakmp dynamic test-map discover`	(Optional) Adds a dynamic crypto map to a crypto map set. You should set the crypto map entries referencing dynamic maps to the lowest priority entries in a crypto map set. **Note** You must issue the **discover** keyword to enable TED.

Troubleshooting Tips

Certain configuration changes take effect only when negotiating subsequent SAs. If you want the new settings to take immediate effect, you must clear the existing SAs so that they are reestablished with the changed configuration. If the router is actively processing IPSec traffic, it is desirable to clear only the portion of the SA database that would be affected by the configuration changes (that is, clear only the SAs established by a given crypto map set). Clearing the full SA database should be reserved for large-scale changes, or when the router is processing very little other IPSec traffic.

To clear IPSec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters clears the full SA database, which clears active security sessions.)

What to Do Next

After you have successfully created a crypto map set, you must apply the crypto map set to each interface through which IPSec traffic flows. To complete this task, see the section "Applying Crypto Map Sets to Interfaces."

Creating Crypto Map Entries to Establish Manual SAs

The use of manual security associations is a result of a prior arrangement between the users of the local router and the IPSec peer. The two parties may begin with manual SAs and then move to using SAs established via IKE, or the remote party's system may not support IKE. If IKE is not used for establishing the SAs, there is no negotiation of SAs, so the configuration information in both systems must be the same in order for traffic to be processed successfully by IPSec.

The local router can simultaneously support manual and IKE-established SAs, even within a single crypto map set.

There is very little reason to disable IKE on the local router (unless the router only supports manual SAs, which is unlikely).

Note Access lists for crypto map entries tagged as **ipsec-manual** are restricted to a single **permit** entry and subsequent entries are ignored. In other words, the SAs established by that particular crypto map entry are only for a single data flow. To be able to support multiple manually established SAs for different kinds of traffic, define multiple crypto access lists, and then apply each one to a separate **ipsec-manual** crypto map entry. Each access list should include one **permit** statement defining what traffic to protect.

To create crypto map entries to establish manual SAs (that is, when IKE is not used to establish the SAs), perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-manual**
4. **match address** *access-list-id*
5. **set peer** {*hostname* | *ip-address*}
6. **set transform-set** *transform-set-name*
7. **set session-key inbound ah** *spi hex-key-string*

   or

   set session-key outbound ah *spi hex-key-string*
8. **set session-key inbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*]

   or

   set session-key outbound esp *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*]
9. **exit**
10. **exit**
11. **show crypto map** [**interface** *interface* | **tag** *map-name*]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 3	**crypto map** *map-name seq-num* **ipsec-manual** **Example:** `Router(config)# crypto map mymap 10 ipsec-manual`	Specifies the crypto map entry to create or modify and enters crypto map configuration mode.
Step 4	**match address** *access-list-id* **Example:** `Router(config-crypto-m)# match address 102`	Names an IPSec access list that determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec in the context of this crypto map entry. (The access list can specify only one **permit** entry when IKE is not used.)
Step 5	**set peer** {*hostname* | *ip-address*} **Example:** `Router(config-crypto-m)# set peer 10.0.0.5`	Specifies the remote IPSec peer. This is the peer to which IPSec protected traffic should be forwarded. (Only one peer can be specified when IKE is not used.)
Step 6	**set transform-set** *transform-set-name* **Example:** `Router(config-crypto-m)# set transform-set someset`	Specifies which transform set should be used. This must be the same transform set that is specified in the remote peer's corresponding crypto map entry. **Note** Only one transform set can be specified when IKE is not used.
Step 7	**set session-key inbound ah** *spi hex-key-string* **Example:** `Router(config-crypto-m)# set session-key inbound ah 256 9876543210987654987654321098765 4` and **set session-key outbound ah** *spi hex-key-string* **Example:** `Router(config-crypto-m)# set session-key outbound ah 256 fedcbafedcbafedcfedcbafedcbafedc`	Sets the AH security parameter indexes (SPIs) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the AH protocol. (This manually specifies the AH security association to be used with protected traffic.)
Step 8	**set session-key inbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*] **Example:** `Router(config-crypto-m)# set session-key inbound esp 256 cipher 0123456789012345` and **set session-key outbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*] **Example:** `Router(config-crypto-m)# set session-key outbound esp 256 cipher abcdefabcdefabcd`	Sets the ESP SPIs and keys to apply to inbound and outbound protected traffic if the specified transform set includes the ESP protocol. Specifies the cipher keys if the transform set includes an ESP cipher algorithm. Specifies the authenticator keys if the transform set includes an ESP authenticator algorithm. (This manually specifies the ESP security association to be used with protected traffic.)

	Command	Purpose
Step 9	**exit** **Example:** `Router(config-crypto-m)# exit`	Exits crypto-map configuration mode and returns to global configuration mode.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Router# show crypto map`	Displays your crypto map configuration.

Troubleshooting Tips

For manually established SAs, you must clear and reinitialize the SAs for the changes to take effect. To clear IPSec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters clears the full SA database, which clears active security sessions.)

What to Do Next

After you have successfully created a crypto map set, you must apply the crypto map set to each interface through which IPSec traffic flows. To complete this task, see the section "Applying Crypto Map Sets to Interfaces."

Applying Crypto Map Sets to Interfaces

You need to apply a crypto map set to each interface through which IPSec traffic flows. Applying the crypto map set to an interface instructs the router to evaluate all the interface's traffic against the crypto map set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by crypto.

Perform this task to apply a crypto map to an interface.

Redundant Interfaces Sharing the Same Crypto Map

For redundancy, you could apply the same crypto map set to more than one interface. The default behavior is as follows:

- Each interface has its own piece of the security association database.
- The IP address of the local interface is used as the local address for IPSec traffic originating from or destined to that interface.

If you apply the same crypto map set to multiple interfaces for redundancy purposes, you need to specify an identifying interface. One suggestion is to use a loopback interface as the identifying interface. This has the following effects:

- The per-interface portion of the IPSec security association database is established one time and shared for traffic through all the interfaces that share the same crypto map.

- The IP address of the identifying interface is used as the local address for IPSec traffic originating from or destined to those interfaces sharing the same crypto map set.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **crypto map** *map-name*
5. **exit**
6. **crypto map** *map-name* **local-address** *interface-id*
7. **exit**
8. **show crypto map** [**interface** *interface*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# Interface FastEthernet 0/0`	Configures an interface and enters interface configuration mode.
Step 4	**crypto map** *map-name* **Example:** `Router(config-if)# crypto map mymap`	Applies a crypto map set to an interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**crypto map** *map-name* **local-address** *interface-id* **Example:** `Router(config)# crypto map mymap local-address loopback0`	(Optional) Permits redundant interfaces to share the same crypto map using the same local identity.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config)# exit`	(Optional) Exits global configuration mode.
Step 8	**show crypto map** [**interface** *interface*] **Example:** `Router# show crypto map`	(Optional) Displays your crypto map configuration

Configuration Examples for Configuring an IPSec VPN

This section contains the following configuration example:

- AES-Based Static Crypto Map: Example, page 32

AES-Based Static Crypto Map: Example

The following example is a portion of the **show running-config** command. This example shows how to configure a static crypto map and define AES as the encryption method.

```
crypto isakmp policy 10
 encryption aes 256
 authentication pre-share
 lifetime 180

crypto isakmp key cisco123 address 10.0.110.1
!
!
crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac
 mode transport
!
crypto map aesmap 10 ipsec-isakmp
 set peer 10.0.110.1
 set transform-set aesset
 match address 120
!
!
!
voice call carrier capacity active
!
!
mta receive maximum-recipients 0
!
!
interface FastEthernet0/0
 ip address 10.0.110.2 255.255.255.0
 ip nat outside
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
 crypto map aesmap
!
interface Serial0/0
```

```
 no ip address
 shutdown
!
interface FastEthernet0/1
 ip address 11.0.110.1 255.255.255.0
 ip nat inside
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
!
ip nat inside source list 110 interface FastEthernet0/0 overload
ip classless
ip route 0.0.0.0 0.0.0.0 10.5.1.1
ip route 12.0.110.0 255.255.255.0 FastEthernet0/0
ip route 172.18.124.0 255.255.255.0 10.5.1.1
ip route 172.18.125.3 255.255.255.255 10.5.1.1
ip http server
!
!
access-list 110 deny   ip 11.0.110.0 0.0.0.255 12.0.110.0 0.0.0.255
access-list 110 permit ip 11.0.110.0 0.0.0.255 any
access-list 120 permit ip 11.0.110.0 0.0.0.255 12.0.110.0 0.0.0.255
!
```

Additional References

The following sections provide references related to IPSec VPN configuration.

Related Documents

Related Topic	Document Title
IKE configuration	"Configuring IKE for IPSec VPNs" module
IKE, IPSec, and PKI configuration commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*, Release 12.4

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
• CISCO-IPSEC-FLOW-MONITOR- MIB • CISCO-IPSEC-MIB • CISCO-IPSEC-POLICY-MAP-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2401	Security Architecture for the Internet Protocol
RFC 2402	*IP Authentication Header*
RFC 2403	The Use of HMAC-MD5-96 within ESP and AH
RFC 2404	The Use of HMAC-SHA-1-96 within ESP and AH
RFC 2405	The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC 2406	*IP Encapsulating Security Payload (ESP)*
RFC 2407	The Internet IP Security Domain of Interpretation for ISAKMP
RFC 2408	Internet Security Association and Key Management Protocol (ISAKMP)

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Glossary

anti-replay—Security service where the receiver can reject old or duplicate packets to protect itself against replay attacks. IPSec provides this optional service by use of a sequence number combined with the use of data authentication. Cisco IOS IPSec provides this service whenever it provides the data authentication service, except for manually established SAs (that is, SAs established by configuration and not by IKE).

data authentication—Verification of the integrity and origin of the data.

Data authentication can refer either to integrity alone or to both of these concepts (although data origin authentication is dependent upon data integrity).

data confidentiality—Security service in which the protected data cannot be observed.

data flow—Grouping of traffic, identified by a combination of source address or mask; destination address or mask; IP next protocol field, and source and destination ports, where the protocol and port fields can have the values of **any**. IPSec protection is applied to data flows.

peer—In the context of this module, a "peer" is a router or other device that participates in IPSec.

PFS—perfect forward secrecy. Cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised, because subsequent keys are not derived from previous keys.

SA—security association. Description of how two or more entities use security services in the context of a particular security protocol (AH or ESP) to communicate securely on behalf of a particular data flow. The transform and the shared secret keys are used for protecting the traffic.

SPI—security parameter index. A number which, together with a destination IP address and security protocol, uniquely identifies a particular security association. Without IKE, the SPI is manually specified for each security association.

transform—List of operations performed on a dataflow to provide data authentication, data confidentiality, and data compression. For example, one transform is the ESP protocol with the HMAC-MD5 authentication algorithm; another transform is the AH protocol with the 56-bit DES encryption algorithm and the ESP protocol with the HMAC-SHA authentication algorithm.

tunnel—In the context of this module, "tunnel" is a secure communication path between two peers, such as two routers. It does not refer to using IPSec in tunnel mode.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Feature Information for Security for VPNs with IPSec

Table 47 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Table 47 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 47 Feature Information for Configuring Security for IPSec VPNs

Feature Name	Software Releases	Feature Configuration Information
Advanced Encryption Standard (AES)	12.2(8)T	This feature adds support for the new encryption standard AES, which is a privacy transform for IPSec and IKE and has been developed to replace DES. The following sections provide information about this feature: • Supported Standards • Defining Transform Sets: A Combination of Security Protocols and Algorithms The following commands were modified by this feature: **crypto ipsec transform-set**, **encryption (IKE policy)**, **show crypto ipsec transform-set**, **show crypto isakmp policy**
DES/3DES/AES VPN Encryption Module (AIM-VPN/EPII, AIM-VPN/HPII, AIM-VPN/BPII Family)	12.3(7)T	This feature describes which VPN encryption hardware AI) and NM are supported in certain Cisco IOS software releases. The following section provides information about this feature: • AIMs and NM Support
SEAL Encryption	12.3(7)T	This feature adds support for SEAL encryption in IPSec. The following section provides information about this feature: • Supported Standards • Defining Transform Sets: A Combination of Security Protocols and Algorithms The following command was modified by this feature: **crypto ipsec transform-set**
Software IPPCP (LZS) with Hardware Encryption	12.2(13)T	This feature allows customers to use LZS software compression with IPSec when a VPN module is in Cisco 2600 and Cisco 3600 series routers. The following section provides information about this feature: • AIMs and NM Support
IKE Shared Secret Using AAA Server	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Crypto Conditional Debug Support

The Crypto Conditional Debug Support feature introduces three new command-line interfaces (CLIs) that allow users to debug an IP Security (IPSec) tunnel on the basis of predefined crypto conditions such as the peer IP address, connection-ID of a crypto engine, and security parameter index (SPI). By limiting debug messages to specific IPSec operations and reducing the amount of debug output, users can better troubleshoot a router with a large number of tunnels.

Feature History for Crypto Conditional Debug Support

Feature History	
Release	**Modification**
12.3(2)T	This feature was introduced.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Crypto Conditional Debug Support

To use the new crypto CLIs, you must be using a crypto image, such as the k8 or k9 subsystem.

Restrictions for Crypto Conditional Debug Support

- This feature does not support debug message filtering for hardware crypto engines.
- Although conditional debugging is useful for troubleshooting peer-specific or functionality related Internet Key Exchange (IKE) and IPSec problems, conditional debugging may not be able to define and check large numbers of debug conditions.

  Because extra space is needed to store the debug condition values, additional processing overhead is added to the CPU and memory usage is increased. Thus, enabling crypto conditional debugging on a router with heavy traffic should be used with caution.

Information About Crypto Conditional Debug Support

To enable the conditional crypto debug commands, you should understand the following concept:

- Supported Condition Types, page 2

Supported Condition Types

The new crypto conditional debug CLIs—**debug crypto condition**, **debug crypto condition unmatched**, and **show crypto debug-condition**—allow you to specify conditions (filter values) in which to generate and display debug messages related only to the specified conditions. Table 1 lists the supported condition types.

Table 1 Supported Condition Types for Crypto Debug CLI

Condition Type (Keyword)	Description
connid[1]	An integer between 1–32766. Relevant debug messages will be shown if the current IPSec operation uses this value as the connection ID to interface with the crypto engine.
flowid[1]	An integer between 1–32766. Relevant debug messages will be shown if the current IPSec operation uses this value as the flow-ID to interface with the crypto engine.
FVRF	The name string of a virtual private network (VPN) routing and forwarding (VRF) instance. Relevant debug messages will be shown if the current IPSec operation uses this VRF instance as its front-door VRF (FVRF).

Table 1 Supported Condition Types for Crypto Debug CLI (continued)

Condition Type (Keyword)	Description
IVRF	The name string of a VRF instance. Relevant debug messages will be shown if the current IPSec operation uses this VRF instance as its inside VRF (IVRF).
peer group	A Unity group-name string. Relevant debug messages will be shown if the peer is using this group name as its identity.
peer hostname	A fully qualified domain name (FQDN) string. Relevant debug messages will be shown if the peer is using this string as its identity; for example, if the peer is enabling IKE Xauth with this FQDN string.
peer ipaddress	A single IP address. Relevant debug messages will be shown if the current IPSec operation is related to the IP address of this peer.
peer subnet	A subnet and a subnet mask that specify a range of peer IP addresses. Relevant debug messages will be shown if the IP address of the current IPSec peer falls into the specified subnet range.
peer username	A username string. Relevant debug messages will be shown if the peer is using this username as its identity; for example, if the peer is enabling IKE Extended Authentication (Xauth) with this username.
SPI[1]	A 32-bit unsigned integer. Relevant debug messages will be shown if the current IPSec operation uses this value as the SPI.

1. If an IPSec connid, flowid, or SPI is used as a debug condition, the debug messages for a related IPSec flow are generated. An IPSec flow has two connids, flowids, and SPIs—one inbound and one outbound. Both two connids, flowids, and SPIs can be used as the debug condition that triggers debug messages for the IPSec flow.

How to Enable Crypto Conditional Debug Support

This section contains the following procedures:

- Enabling Crypto Conditional Debug Messages, page 3
- Enabling Crypto Error Debug Messages, page 5

Enabling Crypto Conditional Debug Messages

To enable crypto conditional debug filtering, you must perform the following tasks.

Performance Considerations

- Before enabling crypto conditional debugging, you must decide what debug condition types (also known as debug filters) and values will be used. The volume of debug messages is dependent on the number of conditions you define.

Note Specifying numerous debug conditions may consume CPU cycles and negatively affect router performance.

- Your router will perform conditional debugging only after at least one of the global crypto debug commands—**debug crypto isakmp**, **debug crypto ipsec**, and **debug crypto engine**—has been enabled. This requirement helps to ensure that the performance of the router will not be impacted when conditional debugging is not being used.

Disable Crypto Debug Conditions

If you choose to disable crypto conditional debugging, you must first disable any crypto global debug CLIs you have issued; thereafter, you can disable conditional debugging.

Note The **reset** keyword can be used to disable all configured conditions at one time.

SUMMARY STEPS

1. **enable**
2. **debug crypto condition** [**connid** *integer* **engine-id** *integer*] [**flowid** *integer* **engine-id** *integer*] [**fvrf** *string*] [**ivrf** *string*] [**peer** [**group** *string*] [**hostname** *string*] [**ipv4** *ipaddress*] [**subnet** *subnet mask*] [**username** *string*]] [**spi** *integer*] [**reset**]
3. **show crypto debug-condition** {[**peer**] [**connid**] [**spi**] [**fvrf**] [**ivrf**] [**unmatched**]}
4. **debug crypto isakmp**
5. **debug crypto ipsec**
6. **debug crypto engine**
7. **debug crypto condition unmatched** [**isakmp** | **ipsec** | **engine**] (optional)

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto condition** [**connid** *integer* **engine-id** *integer*] [**flowid** *integer* **engine-id** *integer*] [**fvrf** *string*] [**ivrf** *string*] [**peer** [**group** *string*] [**hostname** *string*] [**ipv4** *ipaddress*] [**subnet** *subnet mask*] [**username** *string*]] [**spi** *integer*] [**reset**] **Example:** `Router# debug crypto condition connid 2000 engine-id 1`	Defines conditional debug filters.
Step 3	**show crypto debug-condition** {[**peer**] [**connid**] [**spi**] [**fvrf**] [**ivrf**] [**unmatched**]} **Example:** `Router# show crypto debug-condition spi`	Displays crypto debug conditions that have already been enabled in the router.
Step 4	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Enables global IKE debugging.
Step 5	**debug crypto ipsec** **Example:** `Router# debug crypto ipsec`	Enables global IPSec debugging.
Step 6	**debug crypto engine** **Example:** `Router# debug crypto engine`	Enables global crypto engine debugging.
Step 7	debug crypto condition **unmatched** [**isakmp** | **ipsec** | **engine**] **Example:** `Router# debug crypto condition unmatched ipsec`	(Optional) Displays debug conditional crypto messages when no context information is available to check against debug conditions. If none of the optional keywords are specified, all crypto-related information will be shown.

Enabling Crypto Error Debug Messages

To enable crypto error debug messages, you must perform the following tasks.

debug crypto error CLI

Enabling the **debug crypto error** command displays only error-related debug messages, thereby, allowing you to easily determine why a crypto operation, such as an IKE negotiation, has failed within your system.

Note When enabling this command, ensure that global crypto debug commands are not enabled; otherwise, the global commands will override any possible error-related debug messages.

SUMMARY STEPS

1. **enable**
2. **debug crypto** {**isakmp** | **ipsec** | **engine**} **error**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`debug crypto {isakmp | ipsec | engine} error` **Example:** `Router# debug crypto ipsec error`	Enables only error debugging messages for a crypto area.

Configuration Examples for the Crypto Conditional Debug CLIs

This section includes the following examples:

- Enabling Crypto Conditional Debugging: Example, page 6
- Disabling Crypto Conditional Debugging: Example, page 7

Enabling Crypto Conditional Debugging: Example

The following example shows how to display debug messages when the peer IP address is 10.1.1.1, 10.1.1.2, or 10.1.1.3, and when the connection-ID 2000 of crypto engine 0 is used. This example also shows how to enable global debug crypto CLIs and enable the **show crypto debug-condition** command to verify conditional settings.

```
Router# debug crypto condition connid 2000 engine-id 1
Router# debug crypto condition peer ipv4 10.1.1.1
Router# debug crypto condition peer ipv4 10.1.1.2
Router# debug crypto condition peer ipv4 10.1.1.3
Router# debug crypto condition unmatched
! Verify crypto conditional settings.
Router# show crypto debug-condition
```

```
Crypto conditional debug currently is turned ON
IKE debug context unmatched flag:ON
IPsec debug context unmatched flag:ON
Crypto Engine debug context unmatched flag:ON

IKE peer IP address filters:
10.1.1.1  10.1.1.2   10.1.1.3

Connection-id filters:[connid:engine_id]2000:1,
! Enable global crypto CLIs to start conditional debugging.
Router# debug crypto isakmp
Router# debug crypto ipsec
Router# debug crypto engine
```

Disabling Crypto Conditional Debugging: Example

The following example shows how to disable all crypto conditional settings and verify that those settings have been disabled:

```
Router# debug crypto condition reset
! Verify that all crypto conditional settings have been disabled.
Router# show crypto debug-condition

Crypto conditional debug currently is turned OFF
IKE debug context unmatched flag:OFF
IPsec debug context unmatched flag:OFF
Crypto Engine debug context unmatched flag:OFF
```

Additional References

The following sections provide references to the Crypto Conditional Debug Support feature.

Related Documents

Related Topic	Document Title
IPSec and IKE configuration tasks	*Cisco IOS Security Configuration Guide*
IPSec and IKE commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features

- **debug crypto condition**
- **debug crypto condition unmatched**
- **debug crypto error**
- **show crypto debug-condition**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Distinguished Name Based Crypto Maps

Feature History

Release	Modification
12.2(4)T	This feature was introduced.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This feature module describes the Distinguished Name Based Crypto Map feature in Cisco IOS Release 12.2(4)T. It includes the following sections:

Feature Overview

The Distinguished Name Based Crypto Maps feature allows you to configure the router to restrict access to selected encrypted interfaces for those peers with specific certificates, especially certificates with particular Distinguished Names (DNs).

Previously, if the router accepted a certificate or a shared secret from the encrypting peer, Cisco IOS did not have a method of preventing the peer from communicating with any encrypted interface other than the restrictions on the IP address of the encrypting peer. This feature allows you to configure which crypto maps are usable to a peer based on the DN that a peer used to authenticate itself, thereby, enabling you to control which encrypted interfaces a peer with a specified DN can access.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Benefits

The Distinguished Name Based Crypto Maps feature allows you to set restrictions in the router configuration that prevent peers with specific certificates—especially certificates with particular DNs—from having access to selected encrypted interfaces.

Restrictions

System Requirements

To configure this feature, your router must support IP Security.

Performance Impact

If you restrict access to a large number of DNs, it is recommended that you specify a few number of crypto maps referring to large identity sections instead of specifying a large number of crypto maps referring to small identity sections.

Related Documents

The following documents provide information related to the Distinguished Name Based Crypto Maps feature:

- *Cisco IOS Security Command Reference*, Release 12.2
- *Cisco IOS Security Configuration Guide*, Release 12.2

Supported Platforms

This feature is supported on the following platforms:

- Cisco 1700 series
- Cisco 2600 series
- Cisco 3620
- Cisco 3640
- Cisco 3660
- Cisco 7100 series
- Cisco 7200 series
- Cisco uBR905 Cable Access Router
- Cisco uBR925 Cable Access Router

Determining Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

None

Prerequisites

Before configuring a DN based crypto map, you must perform the following tasks:

- Create an Internet Key Exchange (IKE) policy at each peer.

  For more information on creating IKE policies, refer to the chapter “Configuring Internet Key Exchange Security Protocol” of the *Cisco IOS Security Configuration Guide*.
- Create crypto map entries for IPSec.

  For more information on creating crypto map entries, refer to the chapter “Configuring IPSec Network Security” of the *Cisco IOS Security Configuration Guide*.

Configuration Tasks

See the following sections for configuration tasks for the Distinguished Name Based Crypto Maps feature. Each task in the list is identified as either required or optional.

- Configuring DN Based Crypto Maps (authenticated by DN) (required)
- Configuring DN Based Crypto Maps (authenticated by hostname) (required)
- Applying Identity to DN Based Crypto Maps (required)
- Verifying DN Based Crypto Maps (optional)

Configuring DN Based Crypto Maps (authenticated by DN)

To configure a DN based crypto map that can be used only by peers that have been authenticated by a DN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto identity name`	Configures the identity of a router with the given list of DNs in the certificate of the router and enters crypto identity configuration mode.
Step 2	`Router(crypto-identity)# dn name=string [,name=string]`	Associates the identity of the router with the DN in the certificate of the router. **Note** The identity of the peer must match the identity in the exchanged certificate.

Configuring DN Based Crypto Maps (authenticated by hostname)

To configure a DN based crypto map that can be used only by peers that have been authenticated by a hostname, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto identity name`	Configures the identity of a router with the given list of DNs in the certificate of the router and enters crypto identity configuration mode.
Step 2	`Router(crypto-identity)# fqdn name`	Associates the identity of the router with the hostname that the peer used to authenticate itself. **Note** The identity of the peer must match the identity in the exchanged certificate.

Applying Identity to DN Based Crypto Maps

To apply the identity (within the crypto map context), use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num ipsec-isakmp`	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
Step 2	`Router(config-crypto-map)# identity name`	Applies the identity to the crypto map. When this command is applied, only the hosts that match a configuration listed within the **identity** *name* can use the specified crypto map. **Note** If the **identity** command does not appear within the crypto map, the encrypted connection does not have any restrictions other than the IP address of the encrypting peer.

Verifying DN Based Crypto Maps

To verify that this functionality is properly configured, use the following command in EXEC mode:

Command	Purpose
`Router# show crypto identity`	Displays the configured identities.

Troubleshooting Tips

If an encrypting peer attempts to establish a connection that is blocked by the DN based crypto map configuration, the following error message will be logged:

```
<time>: %CRYPTO-4-IKE_QUICKMODE_BAD_CERT: encrypted connection attempted with a peer
without the configured certificate attributes.
```

Configuration Examples

This section provides the following configuration example:

- DN Based Crypto Map Configuration Example

DN Based Crypto Map Configuration Example

The following example shows how to configure DN based crypto maps that have been authenticated by DN and hostname. Comments are included inline to explain various commands.

```
! DN based crypto maps require you to configure an IKE policy at each peer.
crypto isakmp policy 15
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
crypto isakmp policy 20
```

```
 authentication pre-share
 lifetime 10000
crypto isakmp key 1234567890 address 171.69.224.33
!
! The following is an IPSec crypto map (part of IPSec configuration). It can be used only
! by peers that have been authenticated by DN and if the certificate belongs to BigBiz.
crypto map map-to-bigbiz 10 ipsec-isakmp
 set peer 172.21.114.196
 set transform-set my-transformset
 match address 124
 identity to-bigbiz
!
crypto identity to-bigbiz
 dn ou=BigBiz
!
!
! This crypto map can be used only by peers that have been authenticated by hostname
! and if the certificate belongs to little.com.
crypto map map-to-little-com 10 ipsec-isakmp
 set peer 172.21.115.119
 set transform-set my-transformset
 match address 125
 identity to-little-com
!
crypto identity to-little-com
 fqdn little.com
!
```

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **crypto identity**
- **dn**
- **fqdn**
- **identity**

For information about these commands, see the Cisco IOS Security Command Reference at

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at

http://tools.cisco.com/Support/CLILookup or the Master Command List.

Dynamic Multipoint VPN (DMVPN)

First Published: November 25, 2002
Last Updated: December 11, 2006

The Dynamic Multipoint VPN (DMVPN) feature allows users to better scale large and small IP Security (IPsec) Virtual Private Networks (VPNs) by combining generic routing encapsulation (GRE) tunnels, IPsec encryption, and Next Hop Resolution Protocol (NHRP).

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Dynamic Multipoint VPN (DMVPN)" section on page 53.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Dynamic Multipoint VPN (DMVPN)

- Before a multipoint GRE (mGRE) and IPsec tunnel can be established, you must define an Internet Key Exchange (IKE) policy by using the **crypto isakmp policy** command.
- For the NAT-Transparency Aware enhancement to work, you must use IPsec transport mode on the transform set. Also, even though NAT-Transparency can support two peers (IKE and IPsec) being translated to the same IP address (using the User Datagram Protocol [UDP] ports to differentiate them [that is, Peer Address Translation (PAT)]), this functionality is not supported for DMVPN. All DMVPN spokes must have a unique IP address after they have been NAT translated. They can have the same IP address before they are NAT translated.
- To enable 2547oDMPVN—Traffic Segmentation Within DMVPN you must configure multiprotocol label switching (MPLS) by using the **mpls ip** command.

Restrictions for Dynamic Multipoint VPN (DMVPN)

- If you use the Dynamic Creation for Spoke-to-Spoke Tunnels benefit of this feature, you must use IKE certificates or wildcard preshared keys for Internet Security Association Key Management Protocol (ISAKMP) authentication.

Note It is highly recommended that you *do not use* wildcard preshared keys because the attacker will have access to the VPN if one spoke router is compromised.

- GRE tunnel keepalives (that is, the **keepalive** command under a GRE interface) are not supported on point-to-point or multipoint GRE tunnels in a DMVPN Network.
- For best DMVPN functionality, it is recommended that you run the latest Cisco IOS software Release 12.4 mainline,12.4T, or 12.2(18)SXF.

DMVPN Support on the Cisco 6500 and Cisco 7600

Blade-to-Blade Switchover on the Cisco 6500 and Cisco 7600

- DMVPN does not support blade-to-blade switchover on the Cisco 6500 and Cisco 7600.

Cisco 6500 or Cisco 7600 As a DMVPN Hub

- A Cisco 6500 or Cisco 7600 that is functioning as a DMVPN hub cannot be located behind a NAT router.
- If a Cisco 6500 or Cisco 7600 is functioning as a DMVPN hub, the spoke behind NAT must be a Cisco 6500 or Cisco 7600, respectively, or the router must be upgraded to Cisco IOS software Release 12.3(11)T02 or a later release.

Cisco 6500 or Cisco 7600 As a DMVPN Spoke

- If a Cisco 6500 or Cisco 7600 is functioning as a spoke, the hub cannot be behind NAT.
- If a Cisco 6500 or Cisco 7600 is functioning as a DMVPN spoke behind NAT, the hub must be a Cisco 6500 or Cisco 7600, respectively, or the router must be upgraded to Cisco IOS Release 12.3(11)T02 or a later release.

DMVPN Hub or Spoke Supervisor Engine

- Only a Supervisor Engine 720 can be used as a DMVPN hub or spoke. A Supervisor Engine 2 cannot be used.

Encrypted Multicast with GRE

- Encrypted Multicast with GRE is not supported on the Cisco 6500 nor on the Cisco 7600.

mGRE Interfaces

- If there are two mGRE interfaces on the same DMVPN node and they both do not have a tunnel key, the two mGRE interfaces must each have a unique tunnel source address (or interface) configured.
- On the Cisco 6500 and Cisco 7600, each GRE interface (multipoint or point-to-point) must have a unique tunnel source address (or interface).
- The following commands are not supported under mGRE with DMVPN: **ip tcp adjust-mss**, **qos pre-classify tunnel vrf**, **tunnel path-mtu-discovery**, and **tunnel vrf**.

Quality of Service (QoS)

- You cannot use QoS for DMVPN packets on a Cisco 6500 or Cisco 7600.

Tunnel Key

- The use of a tunnel key on a GRE (multipoint or point-to-point) interface is not supported in the hardware switching ASICs on the Cisco 6500 and Cisco 7600 platforms. If a tunnel key is configured, throughput performance is greatly reduced.
- In Cisco IOS Release 12.3(11)T3 and Release 12.3(14)T, the requirement that a mGRE interface must have a tunnel key was removed. Therefore, in a DMVPN network that includes a Cisco 6500 or Cisco 7600 as a DMVPN node, you should remove the tunnel key from all DMVPN nodes in the DMVPN network, thus preserving the throughput performance on the Cisco 6500 and Cisco 7600 platforms.
- If the tunnel key is not configured on any DMVPN node within a DMVPN network, it must not be configured on all DMVPN nodes with the DMVPN network.

VRF-Aware DMVPN Scenarios

- The **mls mpls tunnel-recir** command must be configured on the provider equipment (PE) DMVPN hub if customer equipment (CE) DMVPN spokes need to “talk” to other CEs across the MPLS cloud.
- The mGRE interface should be configured with a large enough IP maximum transmission unit (1400 packets to avoid having the route processor doing fragmentation.
- Enhanced Interior Gateway Routing Protocol (EIGRP) should be avoided.

Information About Dynamic Multipoint VPN (DMVPN)

To configure the Dynamic Multipoint VPN (DMVPN) feature, you must understand the following concepts:

- Benefits of Dynamic Multipoint VPN (DMVPN), page 4
- Feature Design of Dynamic Multipoint VPN (DMVPN), page 5
- IPsec Profiles, page 6

Benefits of Dynamic Multipoint VPN (DMVPN)

Hub Router Configuration Reduction

- Currently, for each spoke router, there is a separate block of configuration lines on the hub router that define the crypto map characteristics, the crypto access list, and the GRE tunnel interface. This feature allows users to configure a single mGRE tunnel interface, a single IPsec profile, and no crypto access lists on the hub router to handle all spoke routers. Thus, the size of the configuration on the hub router remains constant even if spoke routers are added to the network.
- DMVPN architecture can group many spokes into a single multipoint GRE interface, removing the need for a distinct physical or logical interface for each spoke in a native IPsec installation.

Automatic IPsec Encryption Initiation

- GRE has the peer source and destination address configured or resolved with NHRP. Thus, this feature allows IPsec to be immediately triggered for the point-to-point GRE tunneling or when the GRE peer address is resolved via NHRP for the multipoint GRE tunnel.

Support for Dynamically Addressed Spoke Routers

- When using point-to-point GRE and IPsec hub-and-spoke VPN networks, the physical interface IP address of the spoke routers must be known when configuring the hub router because IP address must be configured as the GRE tunnel destination address. This feature allows spoke routers to have dynamic physical interface IP addresses (common for cable and DSL connections). When the spoke router comes online, it will send registration packets to the hub router: within these registration packets, is the current physical interface IP address of this spoke.

Dynamic Creation for Spoke-to-Spoke Tunnels

- This feature eliminates the need for spoke-to-spoke configuration for direct tunnels. When a spoke router wants to transmit a packet to another spoke router, it can now use NHRP to dynamically determine the required destination address of the target spoke router. (The hub router acts as the NHRP server, handling the request for the source spoke router.) The two spoke routers dynamically create an IPsec tunnel between them so data can be directly transferred.

VRF Integrated DMVPN

- DMVPNs can be used to extend the Multiprotocol Label Switching (MPLS) networks that are deployed by service providers to take advantage of the ease of configuration of hub and spokes, to provide support for dynamically addressed customer premises equipment (CPEs), and to provide zero-touch provisioning for adding new spokes into a DMVPN.

Feature Design of Dynamic Multipoint VPN (DMVPN)

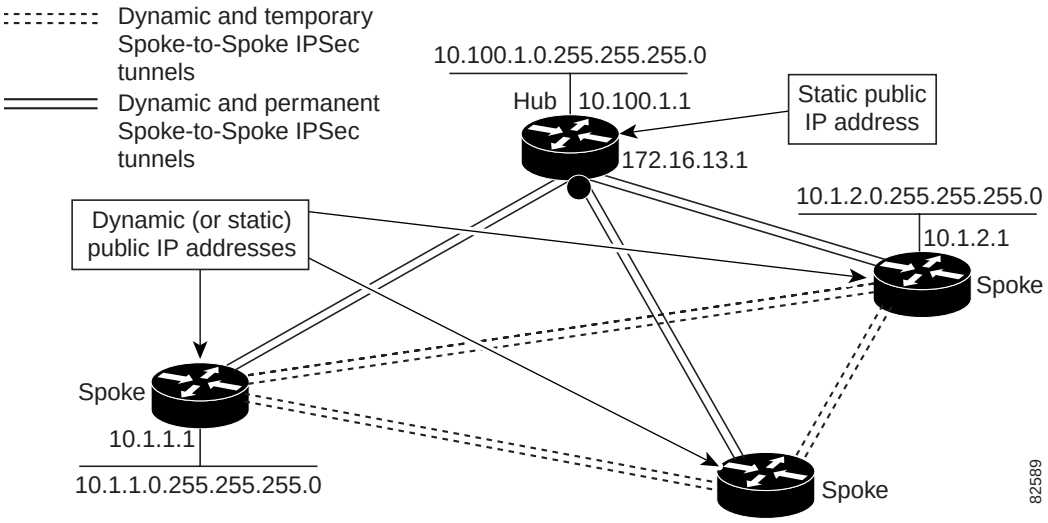
The Dynamic Multipoint VPN (DMVPN) feature combines GRE tunnels, IPsec encryption, and NHRP routing to provide users an ease of configuration via crypto profiles—which override the requirement for defining static crypto maps—and dynamic discovery of tunnel endpoints.

This feature relies on the following two Cisco enhanced standard technologies:

- NHRP—A client and server protocol where the hub is the server and the spokes are the clients. The hub maintains an NHRP database of the public interface addresses of the each spoke. Each spoke registers its real address when it boots and queries the NHRP database for real addresses of the destination spokes to build direct tunnels.
- mGRE Tunnel Interface —Allows a single GRE interface to support multiple IPsec tunnels and simplifies the size and complexity of the configuration.

The topology shown in Figure 1 and the corresponding bullets explain how this feature works.

Figure 1 Sample mGRE and IPsec Integration Topology



- Each spoke has a permanent IPsec tunnel to the hub, not to the other spokes within the network. Each spoke registers as clients of the NHRP server.
- When a spoke needs to send a packet to a destination (private) subnet on another spoke, it queries the NHRP server for the real (outside) address of the destination (target) spoke.
- After the originating spoke “learns” the peer address of the target spoke, it can initiate a dynamic IPsec tunnel to the target spoke.
- The spoke-to-spoke tunnel is built over the multipoint GRE interface.
- The spoke-to-spoke links are established on demand whenever there is traffic between the spokes. Thereafter, packets can bypass the hub and use the spoke-to-spoke tunnel.

Note After a preconfigured amount of inactivity on the spoke-to-spoke tunnels, the router will tear down those tunnels to save resources (IPsec security associations [SAs]).

IPsec Profiles

IPsec profiles abstract IPsec policy information into a single configuration entity, which can be referenced by name from other parts of the configuration. Therefore, users can configure functionality such as GRE tunnel protection with a single line of configuration. By referencing an IPsec profile, the user does not have to configure an entire crypto map configuration. An IPsec profile contains only IPsec information; that is, it does not contain any access list information or peering information.

VRF Integrated DMVPN

VPN Routing and Forwarding (VRF) Integrated DMVPN enables users to map DMVPN multipoint interfaces into MPLS VPNs. This mapping allows Internet service providers (ISPs) to extend their existing MPLS VPN services by mapping off-network sites (typically a branch office) to their respective MPLS VPNs. Customer equipment (CE) routers are terminated on the DMVPN PE router, and traffic is placed in the VRF instance of an MPLS VPN.

DMVPN can interact with MPLS VPNs in two ways:

1. The **ip vrf forwarding** command is used to inject the data IP packets (those packets inside the mGRE+IPsec tunnel) into the MPLS VPN. The **ip vrf forwarding** command is supported for DMVPN in Cisco IOS Release 12.3(6) and Release 12.3(7)T.
2. The **tunnel vrf** command is used to transport (route) the mGRE+IPsec tunnel packet itself within an MPLS VPN. The **tunnel vrf** command is supported in Cisco IOS Release 12.3(11)T but not in Cisco IOS Release 12.2(18)SXE.

Note Clear-text data IP packets are forwarded in a VRF using the **ip vrf forwarding** command, and encrypted tunnel IP packets are forwarded in a VRF using the **tunnel vrf** command.

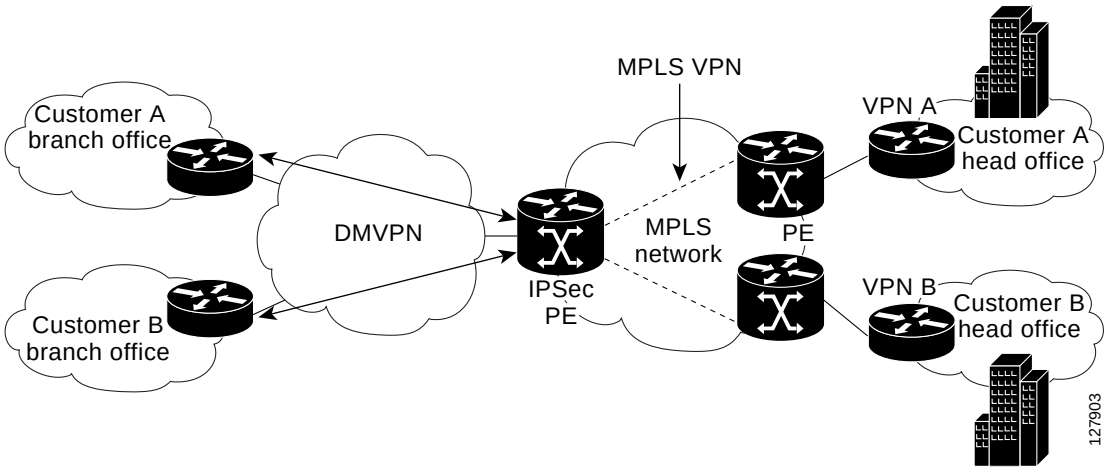
The **ip vrf forwarding** and **tunnel vrf** commands may be used at the same time. If they are used at the same time, the VRF name of each command may be the same or different.

For information about configuring the forwarding of clear-text data IP packets into a VRF, see the section "Configuring the Forwarding of Clear-Text Data IP Packets into a VRF." For information about configuring the forwarding of encrypted tunnel packets into a VRF, see the section "Configuring the Forwarding of Encrypted Tunnel Packets into a VRF."

For more information about configuring VRF, see reference in the "Related Documents" section.

Figure 2 illustrates a typical VRF Integrated DMVPN scenario.

Figure 2 VRF Integrated DMVPN

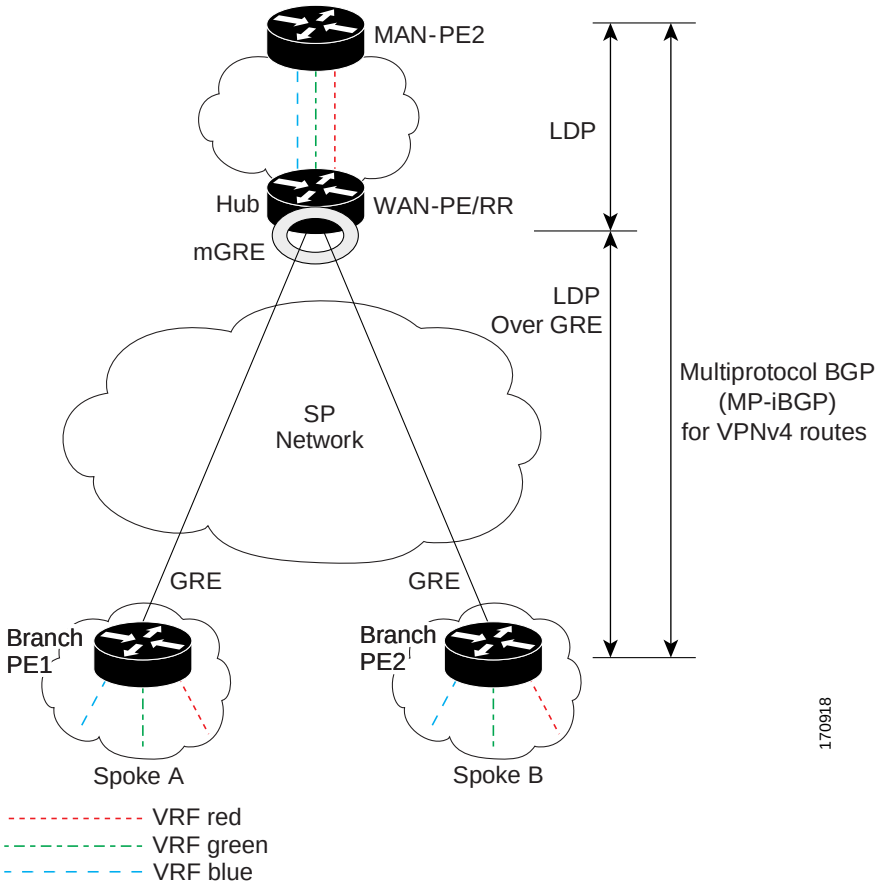


DMVPN—Enabling Traffic Segmentation Within DMVPN

Cisco IOS Release 12.4(11)T provides an enhancement that allows you to segment VPN traffic within a DMVPN tunnel. VRF instances are labeled, using MPLS, to indicate their source and destination.

The diagram in Figure 3 and the corresponding bullets explain how traffic segmentation within DMVPN works.

Figure 3 Traffic Segmentation with DMVPN



- The hub shown in the diagram is a WAN-PE and a route reflector, and the spokes (PE routers) are clients.
- There are three VRFs, designated “red,” “green,” and “blue.”
- Each spoke has both a neighbor relationship with the hub (multiprotocol Border Gateway Protocol [MP-iBGP] peering) and a GRE tunnel to the hub.
- Each spoke advertises its routes and VPNv4 prefixes to the hub.
- The hub sets its own IP address as the next-hop route for all the VPNv4 addresses it learns from the spokes and assigns a local MPLS label for each VPN when it advertises routes back to the spokes. As a result, traffic from Spoke A to Spoke B is routed via the hub.

An example illustrates the process:

1. Spoke A advertises a VPNv4 route to the hub, and applies the label *X* to the VPN.
2. The hub changes the label to *Y* when the hub advertises the route to Spoke B.
3. When Spoke B has traffic to send to Spoke A, it applies the *Y* label, and the traffic goes to the hub.
4. The hub swaps the VPN label, by removing the *Y* label and applying an *X* label, and sends the traffic to Spoke A.

NAT-Transparency Aware DMVPN

DMVPN spokes are often situated behind a NAT router (which is often controlled by the ISP for the spoke site) with the outside interface address of the spoke router being dynamically assigned by the ISP using a private IP address (per Internet Engineering Task Force [IETF] RFC 1918).

Prior to Cisco IOS Release 12.3(6) and 12.3(7)T, these spoke routers had to use IPsec tunnel mode to participate in a DMVPN network. In addition, their assigned outside interface private IP address had to be unique across the DMVPN network. Even though ISAKMP and IPsec would negotiate NAT-T and "learn" the correct NAT public address for the private IP address of this spoke, NHRP could only "see" and use the private IP address of the spoke for its mapping entries. Effective with the NAT-Transparency Aware DMVPN enhancement, NHRP can now learn and use the NAT public address for its mappings as long as IPsec transport mode is used (which is the recommend IPsec mode for DMVPN networks). The restriction that the private interface IP address of the spoke must be unique across the DMVPN network has been removed. It is recommended that all DMVPN routers be upgraded to the new code before you try to use the new functionality even though spoke routers that are not behind NAT do not need to be upgraded. In addition, you cannot convert upgraded spoke routers that are behind NAT to the new configuration (IPsec transport mode) until the hub routers have been upgraded.

Also added in Cisco IOS Releases 12.3(9a) and 12.3(11)T is the capability to have the hub DMVPN router behind static NAT. This was a change in the ISAKMP NAT-T support. For this functionality to be used, all the DMVPN spoke routers and hub routers must be upgraded, and IPsec must use transport mode.

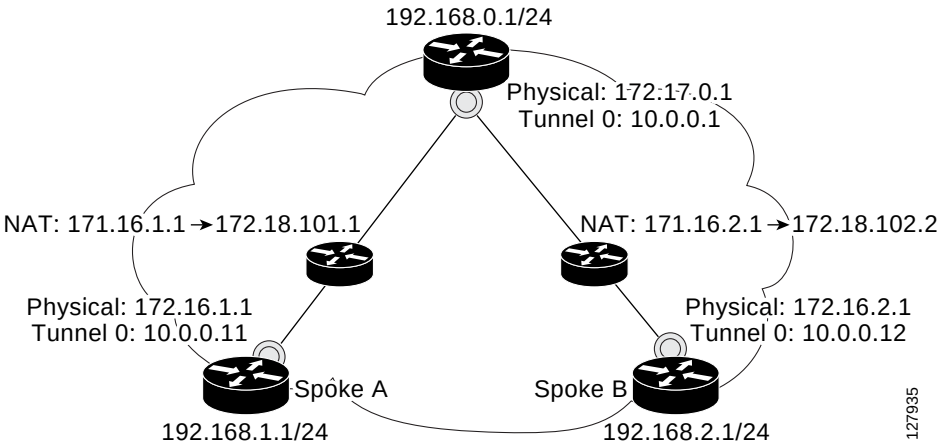
For these NAT-Transparency Aware enhancements to work, you must use IPsec transport mode on the transform set. Also, even though NAT-Transparency (IKE and IPsec) can support two peers (IKE and IPsec) being translated to the same IP address (using the UDP ports to differentiate them), this functionality is not supported for DMVPN. All DMVPN spokes must have a unique IP address after they have been NAT translated. They can have the same IP address before they are NAT translated.

Figure 4 illustrates a NAT-Transparency Aware DMVPN scenario.

Note In Cisco IOS Release 12.4(6)T or earlier, DMVPN spokes behind NAT *will not* participate in dynamic direct spoke-to-spoke tunnels. Any traffic to or from a spoke that is behind NAT will be forwarded using the DMVPN hub routers. DMVPN spokes that are not behind NAT in the same DMVPN network may create dynamic direct spoke-to-spoke tunnels between each other.

In Cisco IOS Release 12.4(6)T or later releases, DMVPN spokes behind NAT *will* participate in dynamic direct spoke-to-spoke tunnels. The spokes must be behind NAT boxes that are preforming NAT, not PAT. The NAT box must translate the spoke to the same outside NAT IP address for the spoke-spoke connections as the NAT box does for the spoke-hub connection. If there is more than one DMVPN spoke behind the same NAT box, then the NAT box *must* translate the DMVPN spokes to different outside NAT IP addresses. It is also likely that you may not be able to build a direct spoke-spoke tunnel between these spokes. If a spoke-spoke tunnel fails to form, then the spoke-spoke packets will continue to be forwarded via the spoke-hub-spoke path.

Figure 4 NAT-Transparency Aware DMVPN



Call Admission Control with DMVPN

In a DMVPN network, it is easy for a DMVPN router to become "overwhelmed" with the number of tunnels it is trying to build. Call Admission Control can be used to limit the number of tunnels that can be built at any one time, thus protecting the memory of the router and CPU resources.

It is most likely that Call Admission Control will be used on a DMVPN spoke to limit the total number of ISAKMP sessions (DMVPN tunnels) that a spoke router will attempt to initiate or accept. This limiting is accomplished by configuring an IKE SA limit under Call Admission Control, which configures the router to drop new ISAKMP session requests (inbound and outbound) if the current number of ISAKMP SAs exceeds the limit.

It is most likely that Call Admission Control will be used on a DMVPN hub to rate limit the number of DMVPN tunnels that are attempting to be built at the same time. The rate limiting is accomplished by configuring a system resource limit under Call Admission Control, which configures the router to drop new ISAKMP session requests (new DMVPN tunnels) when the system utilization is above a specified percentage. The dropped session requests allow the DMVPN hub router to complete the current ISAKMP session requests, and when the system utilization drops, it can process the previously dropped sessions when they are reattempted.

No special configuration is required to use Call Admission Control with DMVPN. For information about configuring Call Admission Control, see the reference in the section "Related Documents."

NHRP Rate-Limiting Mechanism

NHRP has a rate-limiting mechanism that restricts the total number of NHRP packets from any given interface. The default values, which are set using the **ip nhrp max-send** command, are 100 packets every 10 seconds per interface. If the limit is exceeded, you will get the following system message:

```
%NHRP-4-QUOTA: Max-send quota of [int]pkts/[int]Sec. exceeded on [chars]
```

For more information about this system message, see the document *12.4T System Message Guide*.

How to Configure Dynamic Multipoint VPN (DMVPN)

To enable mGRE and IPsec tunneling for hub and spoke routers, you must configure an IPsec profile that uses a global IPsec policy template and configure your mGRE tunnel for IPsec encryption. This section contains the following procedures:

Configuring an IPsec Profile

The IPsec profile shares most of the same commands with the crypto map configuration, but only a subset of the commands are valid in an IPsec profile. Only commands that pertain to an IPsec policy can be issued under an IPsec profile; you cannot specify the IPsec peer address or the access control list (ACL) to match the packets that are to be encrypted.

Prerequisites

Before configuring an IPsec profile, you must define a transform set by using the **crypto ipsec transform-set** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *name*
4. **set transform-set** *transform-set-name*
5. **set identity**
6. **set security association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*}
7. **set pfs** [**group1** | **group2**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec profile** *name* **Example:** `Router(config)# crypto ipsec profile vpnprof`	Defines the IPsec parameters that are to be used for IPsec encryption between "spoke and hub" and "spoke and spoke" routers. This command enters crypto map configuration mode. • The *name* argument specifies the name of the IPsec profile.
Step 4	**set transform-set** *transform-set-name* **Example:** `Router(config-crypto-map)# set transform-set trans2`	Specifies which transform sets can be used with the IPsec profile. • The *transform-set-name* argument specifies the name of the transform set.
Step 5	**set identity** **Example:** Router(config-crypto-map)# set identity	(Optional) Specifies identity restrictions to be used with the IPsec profile.
Step 6	**set security association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*} **Example:** Router(config-crypto-map)# set security association lifetime seconds 1800	(Optional) Overrides the global lifetime value for the IPsec profile. • The **seconds** *seconds* option specifies the number of seconds a security association will live before expiring; the **kilobytes** *kilobytes* option specifies the volume of traffic (in kilobytes) that can pass between IPsec peers using a given security association before that security association expires. • The default for the *seconds* argument is 3600 seconds.
Step 7	**set pfs** [**group1** | **group2**] **Example:** `Router(config-crypto-map)# set pfs group2`	(Optional) Specifies that IPsec should ask for perfect forward secrecy (PFS) when requesting new security associations for this IPsec profile. If this command is not specified, the default (**group1**) will be enabled. • The **group1** keyword specifies that IPsec should use the 768-bit Diffie-Hellman (DH) prime modulus group when performing the new DH exchange; the **group2** keyword specifies the 1024-bit DH prime modulus group.

What to Do Next

Proceed to the following sections "Configuring the Hub for DMVPN" and "Configuring the Spoke for DMVPN."

Configuring the Hub for DMVPN

To configure the hub router for mGRE and IPsec integration (that is, associate the tunnel with the IPsec profile configured in the previous procedure), use the following commands:

Note NHRP network IDs are locally significant and can be different. It makes sense from a deployment and maintenance perspective to use unique network ID numbers (using the **ip nhrp network-id** command) across all routers in a DMVPN network, but it is not necessary that they be the same.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip address** *ip-address mask* [*secondary*]
5. **ip mtu** *bytes*
6. **ip nhrp authentication** *string*
7. **ip nhrp map multicast dynamic**
8. **ip nhrp network-id** *number*
9. **tunnel source** {*ip-address* | *type number*}
10. **tunnel key** *key-number*
11. **tunnel mode gre multipoint**
12. **tunnel protection ipsec profile** *name*
13. **bandwidth** *kbps*
14. **ip tcp adjust-mss** *max-segment-size*
15. **ip nhrp holdtime** *seconds*
16. **delay** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 5`	Configures a tunnel interface and enters interface configuration mode • The *number* argument specifies the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 4	**ip address** *ip-address mask* [*secondary*] **Example:** `Router(config-if)# ip address 10.0.0.1 255.255.255.0`	Sets a primary or secondary IP address for the tunnel interface. **Note** All hubs and spokes that are in the same DMVPN network must be addressed in the same IP subnet.
Step 5	**ip mtu** *bytes* **Example:** `Router(config-if)# ip mtu 1400`	Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface.
Step 6	**ip nhrp authentication** *string* **Example:** `Router(config-if)# ip nhrp authentication donttell`	Configures the authentication string for an interface using NHRP. **Note** The NHRP authentication string must be set to the same value on all hubs and spokes that are in the same DMVPN network.
Step 7	**ip nhrp map multicast dynamic** **Example:** Router(config-if)# ip nhrp map multicast dynamic	Allows NHRP to automatically add spoke routers to the multicast NHRP mappings.
Step 8	**ip nhrp network-id** *number* **Example:** `Router(config-if)# ip nhrp network-id 99`	Enables NHRP on an interface. • The *number* argument specifies a globally unique 32-bit network identifier from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 9	**tunnel source** {*ip-address* | *type number*} **Example:** `Router (config-if)# tunnel source Ethernet0`	Sets source address for a tunnel interface.

	Command or Action	Purpose
Step 10	**tunnel key** *key-number* **Example:** `Router (config-if)# tunnel key 100000`	(Optional) Enables an ID key for a tunnel interface. • The *key-number* argument specifies a number from 0 to 4,294,967,295 that identifies the tunnel key. **Note** The key number must be set to the same value on all hubs and spokes that are in the same DMVPN network. **Note** This command should not be configured if you are using a Cisco 6500 or Cisco 7600 platform.
Step 11	**tunnel mode gre multipoint** **Example:** Router(config-if)# `tunnel mode gre multipoint`	Sets the encapsulation mode to mGRE for the tunnel interface.
Step 12	**tunnel protection ipsec profile** *name* **Example:** `Router(config-if)# tunnel protection ipsec profile vpnprof`	Associates a tunnel interface with an IPsec profile. • The *name* argument specifies the name of the IPsec profile; this value must match the *name* specified in the **crypto ipsec profile** *name* command.
Step 13	**bandwidth** *kbps* **Example:** `Router(config-if)# bandwidth 1000`	Sets the current bandwidth value for an interface to higher-level protocols. • The *kbps* argument specifies the bandwidth in kilobits per second. The default value is 9. The recommend bandwidth value is 1000 or greater. Setting the bandwidth value to at least 1000 is critical if EIGRP is used over the tunnel interface. Higher bandwidth values may be necessary depending on the number of spokes supported by a hub.
Step 14	**ip tcp adjust-mss** *max-segment-size* **Example:** `Router(config-if)# ip tcp adjust-mss 1360`	Adjusts the maximum segment size (MSS) value of TCP packets going through a router. • The *max-segment-size* argument specifies the maximum segment size, in bytes. The range is from 500 to 1460. The recommended value is 1360 when the number of IP MTU bytes is set to 1400. With these recommended settings, TCP sessions quickly scale back to 1400-byte IP packets so the packets will “fit” in the tunnel.

	Command or Action	Purpose
Step 15	**ip nhrp holdtime** *seconds* **Example:** `Router(config-if)# ip nhrp holdtime 450`	Changes the number of seconds that NHRP NBMA addresses are advertised as valid in authoritative NHRP responses. • The *seconds* argument specifies the time in seconds that NBMA addresses are advertised as valid in positive authoritative NHRP responses. The recommended value ranges from 300 seconds to 600 seconds.
Step 16	**delay** *number* **Example:** `Router(config-if)# delay 1000`	(Optional) Used to change the EIGRP routing metric for routes learned over the tunnel interface. • The *number* argument specifies the delay time in seconds. The recommend value is 1000.

Configuring the Spoke for DMVPN

To configure spoke routers for mGRE and IPsec integration, use the following commands.

Note NHRP network IDs are locally significant and can be different. It makes sense from a deployment and maintenance perspective to use unique network ID numbers (using the **ip nhrp network-id** command) across all routers in a DMVPN network, but it is not necessary that they be the same.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip address** *ip-address mask* [*secondary*]
5. **ip mtu** *bytes*
6. **ip nhrp authentication** *string*
7. **ip nhrp map** *hub-tunnel-ip-address hub-physical-ip-address*
8. **ip nhrp map multicast** *hub-physical-ip-address*
9. **ip nhrp nhs** *hub-tunnel-ip-address*
10. **ip nhrp network-id** *number*
11. **tunnel source** {*ip-address* | *type number*}
12. **tunnel key** *key-number*
13. **tunnel mode gre multipoint**

    or

    tunnel destination *hub-physical-ip-address*
14. **tunnel protection ipsec profile** *name*
15. **bandwidth** *kbps*
16. **ip tcp adjust-mss** *max-segment-size*
17. **ip nhrp holdtime** *seconds*
18. **delay** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 5`	Configures a tunnel interface and enters interface configuration mode. • The *number* argument specifies the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 4	**ip address** *ip-address mask* [*secondary*] **Example:** `Router(config-if)# ip address 10.0.0.2 255.255.255.0`	Sets a primary or secondary IP address for the tunnel interface. **Note** All hubs and spokes that are in the same DMVPN network must be addressed in the same IP subnet.
Step 5	**ip mtu** *bytes* **Example:** `Router(config-if)# ip mtu 1400`	Sets the MTU size, in bytes, of IP packets sent on an interface.
Step 6	**ip nhrp authentication** *string* **Example:** `Router(config-if)# ip nhrp authentication donttell`	Configures the authentication string for an interface using NHRP. **Note** The NHRP authentication string be set to the same value on all hubs and spokes that are in the same DMVPN network.
Step 7	`ip nhrp map` *hub-tunnel-ip-address hub-physical-ip-address* **Example:** Router(config-if)# ip nhrp map 10.0.0.1 172.17.0.1	Statically configures the IP-to-NBMA address mapping of IP destinations connected to an MBMA network. • *hub-tunnel-ip-address*—Defines the NHRP server at the hub, which is permanently mapped to the static public IP address of the hub. • *hub-physical-ip-address*—Defines the static public IP address of the hub.
Step 8	**ip nhrp map multicast** *hub-physical-ip-address* **Example:** Router(config-if)# ip nhrp map multicast 172.17.0.1	Enables the use of a dynamic routing protocol between the spoke and hub, and sends multicast packets to the hub router.

	Command or Action	Purpose
Step 9	**ip nhrp nhs** *hub-tunnel-ip-address* **Example:** Router(config-if)# ip nhrp nhs 10.0.0.1	Configures the hub router as the NHRP next-hop server.
Step 10	**ip nhrp network-id** *number* **Example:** `Router(config-if)# ip nhrp network-id 99`	Enables NHRP on an interface. • The *number* argument specifies a globally unique 32-bit network identifier from a NBMA network. The range is from 1 to 4294967295.
Step 11	**tunnel source** {*ip-address* | *type number*} **Example:** `Router (config-if)# tunnel source Ethernet0`	Sets the source address for a tunnel interface.
Step 12	**tunnel key** *key-number* **Example:** `Router (config-if)# tunnel key 100000`	(Optional) Enables an ID key for a tunnel interface. • The *key-number* argument specifies a number from 0 to 4,294,967,295 that identifies the tunnel key. • The key number must be set to the same value on all hubs and spokes that are in the same DMVPN network. **Note** This command should not be configured if you are using a Cisco 6500 or Cisco 7600 platform.
Step 13	**tunnel mode gre multipoint** or **tunnel destination** *hub-physical-ip-address* **Example:** Router(config-if)# `tunnel mode gre multipoint` or Router(config-if)# `tunnel destination 172.17.0.1`	Sets the encapsulation mode to mGRE for the tunnel interface. Use this command if data traffic can use dynamic spoke-to-spoke traffic. Specifies the destination for a tunnel interface. Use this command if data traffic can use hub-and-spoke tunnels.
Step 14	**tunnel protection ipsec profile** *name* **Example:** `Router(config-if)# tunnel protection ipsec profile vpnprof`	Associates a tunnel interface with an IPsec profile. • The *name* argument specifies the name of the IPsec profile; this value must match the *name* specified in the **crypto ipsec profile** *name* command.
Step 15	**bandwidth** *kbps* **Example:** `Router(config-if)# bandwidth 1000`	Sets the current bandwidth value for an interface to higher-level protocols. • The *kbps* argument specifies the bandwidth in kilobits per second. The default value is 9. The recommend bandwidth value is 1000 or greater. The bandwidth setting for the spoke does not need to equal the bandwidth setting for the DMVPN hub. It is usually easier if all of the spokes use the same or similar value.

	Command or Action	Purpose
Step 16	**ip tcp adjust-mss** *max-segment-size* **Example:** `Router(config-if)# ip tcp adjust-mss 1360`	Adjusts the maximum segment size (MSS) value of TCP packets going through a router. • The *max-segment-size* argument specifies the maximum segment size, in bytes. The range is from 500 to 1460. The recommended number value is 1360 when the number of IP MTU bytes is set to 1400. With these recommended settings, TCP sessions quickly scale back to 1400-byte IP packets so the packets will "fit" in the tunnel.
Step 17	**ip nhrp holdtime** *seconds* **Example:** `Router(config-if)# ip nhrp holdtime 450`	Changes the number of seconds that NHRP NBMA addresses are advertised as valid in authoritative NHRP responses. • The *seconds* argument specifies the time in seconds that NBMA addresses are advertised as valid in positive authoritative NHRP responses. The recommended value ranges from 300 seconds to 600 seconds.
Step 18	**delay** *number* **Example:** `Router(config-if)# delay 1000`	(Optional) Used to change the EIGRP routing metric for routes learned over the tunnel interface. • The *number* argument specifies the delay time in seconds. The recommend value is 1000.

Configuring the Forwarding of Clear-Text Data IP Packets into a VRF

To configure the forwarding of clear-text date IP packets into a VRF, perform the following steps. This configuration assumes that the VRF BLUE has already been configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip vrf forwarding** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *type number* **Example:** `Router (config)# interface tunnel0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Router (config-if)# ip vrf forwarding BLUE`	Associates a VPN VRF with an interface or subinterface.

Configuring the Forwarding of Encrypted Tunnel Packets into a VRF

To configure the forwarding of encrypted tunnel packets into a VRF, perform the following steps. This configuration assumes that the VRF RED has already been configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. interface *type number*
4. **tunnel vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	`interface type number` **Example:** `Router (config)# interface tunnel0`	Configures an interface type and enters interface configuration mode.
Step 4	**`tunnel vrf`** *`vrf-name`* **Example:** `Router (config-if)# tunnel vrf RED`	Associates a VPN VRF instance with a specific tunnel destination, interface, or subinterface.

Configuring DMVPN—Traffic Segmentation Within DMVPN

There are no new commands to use for configuring traffic segmentation, but there are tasks you must complete in order to segment traffic within a DMVPN tunnel:

- Enabling MPLS on the VPN Tunnel, page 22
- Configuring Multiprotocol BGP on the Hub Router, page 23
- Configuring Multiprotocol BGP on the Spoke Routers, page 25

Prerequisites

The tasks that follow assume that the DMVPN tunnel and the VRFs “red” and “blue” have already been configured.

For information on configuring a DMVPN tunnel, see the “Configuring the Hub for DMVPN” section on page 13 and the “Configuring the Spoke for DMVPN” section on page 17. For details about VRF configuration, see the “Configuring the Forwarding of Clear-Text Data IP Packets into a VRF” section on page 20 and the “Configuring the Forwarding of Encrypted Tunnel Packets into a VRF” section on page 21.

Enabling MPLS on the VPN Tunnel

Because traffic segmentation within a DMVPN tunnel depends upon MPLS, you must configure MPLS for each VRF instance in which traffic will be segmented. For detailed information about configuring MPLS, see *Cisco IOS Multiprotocol Label Switching Configuration Guide*, Release 12.4.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. interface *type number*
4. **mpls ip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`interface` *`type number`* **Example:** `Router (config)# interface tunnel0`	Configures an interface type and enters interface configuration mode.
Step 4	`mpls ip` **Example:** `Router (config-if)# mpls ip`	Enables MPLS tagging of packets on the specified tunnel interface.

Configuring Multiprotocol BGP on the Hub Router

You must configure multiprotocol iBGP (MP-iBGP) to enable advertisement of VPNv4 prefixes and labels to be applied to the VPN traffic. Use BGP to configure the hub as a route reflector. To force all traffic to be routed via the hub, configure the BGP route reflector to change the next hop to itself when it advertises VPNv4 prefixes to the route reflector clients (spokes).

For more information about the BGP routing protocol, see the "BGP" chapter in the *Cisco IOS IP Routing Protocols Configuration Guide,* Release 12.4.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp**
4. **neighbor** *ipaddress* **remote-as** *as-number*
5. **neighbor** *ipaddress* **update-source** *interface*
6. **address-family vpnv4**
7. **neighbor** *ipaddress* **activate**
8. **neighbor** *ipaddress* **send-community extended**
9. **neighbor** *ipaddress* **route-reflector-client**
10. **neighbor** *ipaddress* **route-map nexthop out**
11. **exit-address family**
12. address-family ipv4 *vrf-name*

13. redistribute connected
14. route-map
15. set ip next-hop *ipaddress*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** **Example:** `Router (config)# router bgp`	Enters BGP configuration mode.
Step 4	**neighbor** *ipaddress* **remote-as** *as-number* **Example:** `Router (config)# neighbor 10.0.0.11 remote-as 1`	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 5	**neighbor** *ipaddress* **update-source** *interface* **Example:** `Router (config)# neighbor 10.10.10.11 update-source Tunnel1`	Configures the Cisco IOS software to allow BGP sessions to use any operational interface for TCP connections.
Step 6	**address-family vpnv4** **Example:** `Router (config)# address-family vpnv4`	Enters address family configuration mode to configure a routing session using Virtual Private Network (VPN) Version 4 address prefixes.
Step 7	**neighbor** *ipaddress* **activate** **Example:** `Router (config)# neighbor 10.0.0.11 activate`	Enables the exchange of information with a BGP neighbor.
Step 8	**neighbor** *ipaddress* **send-community extended** **Example:** `Router (config)# neighbor 10.0.0.11 send-community extended`	Specifies that extended community attributes should be sent to a BGP neighbor.

	Command or Action	Purpose
Step 9	**neighbor** *ipaddress* **route-reflector-client** **Example:** `Router (config)# neighbor 10.0.0.11 route-reflector-client`	Configures the router as a BGP route reflector and configures the specified neighbor as its client.
Step 10	**neighbor** *ipaddress* **route-map nexthop out** **Example:** `Router (config)# neighbor 10.0.0.11 route-map nexthop out`	Forces all traffic to be routed via the hub.
Step 11	**exit-address-family** **Example:** `Router (config)# exit-address-family`	Exits the address family configuration mode for VPNv4.
Step 12	**address-family ipv4** *vrf-name* **Example:** `Router (config)# address-family ipv4 vrf red`	Enters address family configuration mode to configure a routing session using standard IP Version 4 address prefixes.
Step 13	**redistribute connected** **Example:** `Router (config)# redistribute connected`	Redistributes routes that are established automatically by virtue of having enabled IP on an interface from one routing domain into another routing domain.
Step 14	**route-map** **Example:** `Router (config)# route-map nexthop permit 10`	Enters route map configuration mode to configure the next-hop that will be advertised to the spokes.
Step 15	**set ip next-hop** *ipaddress* **Example:** `Router (config)# set ip next-hop 10.0.0.1`	Sets the next hop to be the hub.

Configuring Multiprotocol BGP on the Spoke Routers

Multiprotocol-iBGP (MP-iBGP) must be configured on the spoke routers and the hub. Follow the steps below for each spoke router in the DMVPN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp**
4. **neighbor** *ipaddress* **remote-as** *as-number*
5. **neighbor** *ipaddress* **update-source** *interface*
6. **address-family vpnv4**

7. **neighbor** *ipaddress* **activate**
8. **neighbor** *ipaddress* **send-community extended**
9. **exit-address-family**
10. address-family ipv4 *vrf-name*
11. redistribute connected
12. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** **Example:** `Router (config)# router bgp 1`	Enters BGP configuration mode.
Step 4	**neighbor** *ipaddress* **remote-as** *as-number* **Example:** `Router (config)# neighbor 10.0.0.1 remote-as 1`	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 5	**neighbor** *ipaddress* **update-source** *interface* **Example:** `Router (config)# neighbor 10.10.10.1 update-source Tunnel1`	Configures the Cisco IOS software to allow BGP sessions to use any operational interface for TCP connections.
Step 6	**address-family vpnv4** **Example:** `Router (config)# address-family vpnv4`	Enters address family configuration mode to configure a routing session using Virtual Private Network (VPN) Version 4 address prefixes.
Step 7	**neighbor** *ipaddress* **activate** **Example:** `Router (config)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 8	**neighbor** *ipaddress* **send-community extended** **Example:** `Router (config)# neighbor 10.0.0.1 send-community extended`	Specifies that extended community attributes should be sent to a BGP neighbor.

	Command or Action	Purpose
Step 9	**exit-address-family** **Example:** `Router (config)# exit-address-family`	Exits the address family configuration mode.
Step 10	**address-family ipv4** *vrf-name* **Example:** `Router (config)# address-family ipv4 vrf red`	Enters address family configuration mode to configure a routing session using standard IP Version 4 address prefixes.
Step 11	**redistribute connected** **Example:** `Router (config)# redistribute connected`	Redistributes routes that are established automatically by virtue of having enabled IP on an interface from one routing domain into another routing domain.
Step 12	**exit-address-family** **Example:** `Router (config)# exit-address-family`	Exits the address family configuration mode. **Note** Repeat Steps 10–12 for each VRF.

Troubleshooting Dynamic Multipoint VPN (DMVPN)

After configuring DMVPN, to verify that DMVPN is operating correctly, to clear DMVPN statistics or sessions, or to debug DMVPN, you may perform the following optional steps:

SUMMARY STEPS

1. **clear dmvpn session** [**peer** {**nbma** | **tunnel**} *ip-address*] [**interface** {**tunnel** *number*}] [**vrf** *vrf-name*] [**static**]
2. **clear dmvpn statistics** [**peer** {**nbma** | **tunnel**} *ip-address*] [**interface** {**tunnel** *number*}] [**vrf** *vrf-name*]
3. **debug dmvpn** {[{**condition** [**unmatched**] | [**peer** [**nbma** | **tunnel** {*ip-address*}]] | [**vrf** {*vrf-name*}] | [**interface** {**tunnel** *number*}]}] | [{**error** | **detail** | **packet** | **all**} {**nhrp** | **crypto** | **tunnel** | **socket** | **all**}]}
4. **debug nhrp condition**
5. **debug nhrp error**
6. **logging dmvpn** [**rate-limit** *seconds*]
7. **show crypto ipsec sa** [**active** | **standby**]
8. **show crypto isakmp sa**
9. **show crypto map**
10. **show dmvpn** [**peer** [**nbma** | **tunnel** {*ip-address*}] | [**network** {*ip-address*} {mask}]] [vrf {*vrf-name*}] [**interface** {**tunnel** *number*}] [**detail**] [**static**] [**debug-condition**]
11. **show ip nhrp traffic** [**interface** {**tunnel** *number*}]

DETAILED STEPS

Step 1 The **clear dmvpn session** command is used to clear DMVPN sessions.

The following example clears only dynamic DMVPN sessions:

Router# **clear dmvpn session peer nbma**

The following example clears all DMVPN sessions, both static and dynamic, for the specified tunnel:

Router# **clear dmvpn session interface tunnel 100 static**

Step 2 The **clear dmvpn statistics** command is used to clear DMVPN related counters. The following example shows how to clear DMVPN related session counters for the specified tunnel interface:

```
Router# clear dmvpn statistics peer tunnel 192.0.2.3
```

Step 3 The **debug dmvpn** command is used to debug DMVPN sessions. You can enable or disable DMVPN debugging based on a specific condition. There are three levels of DMVPN debugging, listed in the order of details from lowest to highest:

- Error level
- Detail level
- Packet level

The following example shows how to enable conditional DMVPN debugging that displays all error debugs for next hop routing protocol (NHRP), sockets, tunnel protection and crypto information:

Router# **debug dmvpn error all**

Step 4 The **debug nhrp condition** command enables or disables debugging based on a specific condition. The following example shows how to enable conditional NHRP debugging:

```
Router# debug nhrp condition
```

Step 5 The **debug nhrp error** command displays information about NHRP error activity. The following example shows how to enable debugging for NHRP error messages:

```
Router# debug nhrp error
```

Step 6 The **logging dmvpn** command is used to enable DMVPN system logging. The following command shows how to enable DMVPN system logging at the rate of 1 message every 20 seconds:

```
Router(config)# logging dmvpn rate-limit 20
```

The following example shows a sample system log with DMVPN messages:

```
%DMVPN-7-CRYPTO_SS: Tunnel101-192.0.2.1 socket is UP
%DMVPN-5-NHRP_NHS: Tunnel101 192.0.2.251 is UP
%DMVPN-5-NHRP_CACHE: Client 192.0.2.2 on Tunnel1 Registered.
%DMVPN-5-NHRP_CACHE: Client 192.0.2.2 on Tunnel101 came UP.
%DMVPN-3-NHRP_ERROR: Registration Request failed for 192.0.2.251 on Tunnel101
```

Step 7 The **show crypto ipsec sa** command displays the settings used by the current SAs. The following example output shows the IPsec SA status of only the active device:

```
Router# show crypto ipsec sa active

interface: Ethernet0/0
    Crypto map tag: to-peer-outside, local addr 209.165.201.3
   protected vrf: (none
   local  ident (addr/mask/prot/port): (192.168.0.1/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (172.16.0.1/255.255.255.255/0/0)
   current_peer 209.165.200.225 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 3, #pkts encrypt: 3, #pkts digest: 3
```

```
    #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0
     local crypto endpt.: 209.165.201.3, remote crypto endpt.: 209.165.200.225
     path mtu 1500, media mtu 1500
     current outbound spi: 0xD42904F0(3559458032)
     inbound esp sas:
      spi: 0xD3E9ABD0(3555306448)
        transform: esp-3des ,
        in use settings ={Tunnel, }
        conn id: 2006, flow_id: 6, crypto map: to-peer-outside
        sa timing: remaining key lifetime (k/sec): (4586265/3542)
        HA last key lifetime sent(k): (4586267)
        ike_cookies: 9263635C CA4B4E99 C14E908E 8EE2D79C
        IV size: 8 bytes
        replay detection support: Y
        Status: ACTIVE
```

Step 8 The **show crypto isakmp sa** command displays all current IKE SAs at a peer. For example, the following sample output is displayed after IKE negotiations have successfully completed between two peers.

```
Router# show crypto isakmp sa

dst             src             state          conn-id    slot
172.17.63.19    172.16.175.76   QM_IDLE              2       0
172.17.63.19    172.17.63.20    QM_IDLE              1       0
172.16.175.75   172.17.63.19    QM_IDLE              3       0
```

Step 9 The **show crypto map** command displays the crypto map configuration.

The following sample output is displayed after a crypto map has been configured:

```
Router# show crypto map

Crypto Map "Tunnel5-head-0" 10 ipsec-isakmp
        Profile name: vpnprof
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={trans2, }

Crypto Map "Tunnel5-head-0" 20 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.16.175.75
        Extended IP access list
            access-list permit gre host 172.17.63.19 host 172.16.175.75
        Current peer: 172.16.175.75
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={trans2, }

Crypto Map "Tunnel5-head-0" 30 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.17.63.20
        Extended IP access list
            access-list permit gre host 172.17.63.19 host 172.17.63.20
        Current peer: 172.17.63.20
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={trans2, }

Crypto Map "Tunnel5-head-0" 40 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.16.175.76
```

```
        Extended IP access list
            access-list permit gre host 172.17.63.19 host 172.16.175.76
        Current peer: 172.16.175.76
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={trans2, }
        Interfaces using crypto map Tunnel5-head-0:
    Tunnel5
```

Step 10 The **show dmvpn** command displays DMVPN specific session information. The following example shows example summary output:

```
Router# show dmvpn

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer

! The line below indicates that the sessions are being displayed for Tunnel1.
! Tunnel1 is acting as a spoke and is a peer with three other NBMA peers.

Tunnel1, Type: Spoke, NBMA Peers: 3,
 # Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     2      192.0.2.21     192.0.2.116   IKE     3w0d D
     1     192.0.2.102      192.0.2.11  NHRP 02:40:51 S
     1     192.0.2.225      192.0.2.10    UP     3w0d S

Tunnel2, Type: Spoke, NBMA Peers: 1,
 # Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     1        192.0.2.25      192.0.2.171   IKE    never S
```

Step 11 The **show ip nhrp traffic** command displays NHRP statistics. The following example shows output for a specific tunnel, tunnel7:

```
Router# show ip nhrp traffic interface tunnel7

Tunnel7: Max-send limit:100Pkts/10Sec, Usage:0%
   Sent: Total 79
         18 Resolution Request  10 Resolution Reply  42 Registration Request
         0 Registration Reply    3 Purge Request      6 Purge Reply
         0 Error Indication      0 Traffic Indication
   Rcvd: Total 69
         10 Resolution Request  15 Resolution Reply  0 Registration Request
         36 Registration Reply   6 Purge Request     2 Purge Reply
         0 Error Indication      0 Traffic Indication
```

What to Do Next

If you have troubleshooted your DMVPN configuration and proceed to contact technical support, the **show tech-support** command includes information for DMVPN sessions. For more information, see the **show tech-support** command in the Cisco IOS Configuration Fundamentals Command Reference.

Configuration Examples for Dynamic Multipoint VPN (DMVPN) Feature

This section provides the following comprehensive configuration examples:

Hub Configuration for DMVPN: Example

In the following example, which configures the hub router for multipoint GRE and IPsec integration, no explicit configuration lines are needed for each spoke; that is, the hub is configured with a global IPsec policy template that all spoke routers can talk to. In this example, EIGRP is configured to run over the private physical interface and the tunnel interface.

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
mode transport
!
crypto ipsec profile vpnprof
 set transform-set trans2
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.1 255.255.255.0
! Ensures longer packets are fragmented before they are encrypted; otherwise, the
receiving router would have to do the reassembly.
 ip mtu 1400
! The following line must match on all nodes that “want to use” this mGRE tunnel:
 ip nhrp authentication donttell
! Note that the next line is required only on the hub.
 ip nhrp map multicast dynamic
! The following line must match on all nodes that want to use this mGRE tunnel:
 ip nhrp network-id 99
 ip nhrp holdtime 300
! Turns off split horizon on the mGRE tunnel interface; otherwise, EIGRP will not
advertise routes that are learned via the mGRE interface back out that interface.
 no ip split-horizon eigrp 1
! Enables dynamic, direct spoke-to-spoke tunnels when using EIGRP.
 no ip next-hop-self eigrp 1
 ip tcp adjust-mss 1360
 delay 1000
! Sets IPsec peer address to Ethernet interface’s public address.
 tunnel source Ethernet0
 tunnel mode gre multipoint
! The following line must match on all nodes that want to use this mGRE tunnel.
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address 172.17.0.1 255.255.255.0
!
interface Ethernet1
 ip address 192.168.0.1 255.255.255.0
```

```
!
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 192.168.0.0 0.0.0.255
!
```

For information about defining and configuring ISAKMP profiles, see the references in the "Related Documents" section.

Spoke Configuration for DMVPN: Example

In the following example, all spokes are configured the same except for tunnel and local interface address, thereby, reducing necessary configurations for the user:

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
 mode transport
!
crypto ipsec profile vpnprof
 set transform-set trans2
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.2 255.255.255.0
 ip mtu 1400
! The following line must match on all nodes that want to use this mGRE tunnel:
 ip nhrp authentication donttell
! Definition of NHRP server at the hub (10.0.0.1), which is permanently mapped to the
static public address of the hub (172.17.0.1).
 ip nhrp map 10.0.0.1 172.17.0.1
! Sends multicast packets to the hub router, and enables the use of a dynamic routing
protocol between the spoke and the hub.
 ip nhrp map multicast 172.17.0.1
! The following line must match on all nodes that want to use this mGRE tunnel:
 ip nhrp network-id 99
 ip nhrp holdtime 300
! Configures the hub router as the NHRP next-hop server.
 ip nhrp nhs 10.0.0.1
 ip tcp adjust-mss 1360
 delay 1000
 tunnel source Ethernet0
 tunnel mode gre multipoint
! The following line must match on all nodes that want to use this mGRE tunnel:
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
! This is a spoke, so the public address might be dynamically assigned via DHCP.
interface Ethernet0
 ip address dhcp hostname Spoke1
!
interface Ethernet1
 ip address 192.168.1.1 255.255.255.0
!
! EIGRP is configured to run over the inside physical interface and the tunnel.
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 192.168.1.0 0.0.0.255
```

VRF Aware DMVPN: Example

When configuring VRF Aware DMVPN, you must create a separate DMVPN network for each VRF instance. In the following example, there are two DMVPN networks: BLUE and RED. In addition, a separate source interface has been used on the hub for each DMVPN tunnel—a must for Cisco IOS Release 12.2(18)SXE. For other Cisco IOS releases, you can configure the same tunnel source for both of the tunnel interfaces, but you must configure the **tunnel key** and **tunnel protection (tunnel protection ipsec profile {***name***} shared)** commands.

Note If you use the **shared** keyword, then you should be running Cisco IOS Release 12.4(5) or Release 12.4(6)T, or a later release. Otherwise the IPsec/GRE tunnels under the two mGRE tunnel interfaces may not function correctly.

Hub Configuration

```
interface Tunnel0
! Note the next line.
   ip vrf forwarding BLUE
   bandwidth 1000
   ip address 10.0.0.1 255.255.255.0
   ip mtu 1436
   ! Note the next line.
   ip nhrp authentication BLUE!KEY
   ip nhrp map multicast dynamic
   ! Note the next line
   ip nhrp network-id 100000
   ip nhrp holdtime 600
   no ip split-horizon eigrp 1
   no ip next-hop-self eigrp 1
   ip tcp adjust-mss 1360
   delay 1000
   ! Note the next line.
   tunnel source Ethernet0
   tunnel mode gre multipoint
   tunnel protection ipsec profile vpnprof!
   interface Tunnel1
   ! Note the next line.
   ip vrf forwarding RED
   bandwidth 1000
   ip address 10.0.0.1 255.255.255.0
   ip mtu 1436
   ! Note the next line.
   ip nhrp authentication RED!KEY
   ip nhrp map multicast dynamic
   ! Note the next line.
   ip nhrp network-id 20000
   ip nhrp holdtime 600
   no ip split-horizon eigrp 1
   no ip next-hop-self eigrp 1
   ip tcp adjust-mss 1360
   delay 1000
   ! Note the next line.
   tunnel source Ethernet1
   tunnel mode gre multipoint
   tunnel protection ipsec profile vpnprof!
   interface Ethernet0
   ip address 172.17.0.1 255.255.255.0
!
   interface Ethernet1
```

```
   ip address 192.0.2.171 255.255.255.0
```

Note For the hub configuration shown above, a separate DMVPN network is configured for each VPN. The NHRP network ID and authentication keys must be unique on the two mGRE interfaces.

EIGRP Configuration on the Hub

```
router eigrp 1
auto-summary
!
address-family ipv4 vrf BLUE
network 10.0.0.0 0.0.0.255
no auto-summary
autonomous-system 1
exit-address-family
!
address-family ipv4 vrf RED
network 10.0.0.0 0.0.0.255
no auto-summary
autonomous-system 1
exit-address-family
```

Spoke Configurations

Spoke 1:

```
interface Tunnel0
   bandwidth 1000
   ip address 10.0.0.2 255.255.255.0
   ip mtu 1436
   ! Note the next line.
   ip nhrp authentication BLUE!KEY
   ip nhrp map 10.0.0.1 172.17.0.1
   ip nhrp network-id 100000
   ip nhrp holdtime 300
   ip nhrp nhs 10.0.0.1
   ip tcp adjust-mss 1360
   delay 1000
   tunnel mode gre multipoint
   tunnel source Ethernet0
   tunnel destination 172.17.0.1
   tunnel protection ipsec profile vpnprof
```

Spoke 2:

```
interface Tunnel0
   bandwidth 1000
   ip address 10.0.0.2 255.255.255.0
   ip mtu 1436
   ip nhrp authentication RED!KEY
   ip nhrp map 10.0.0.1 192.0.2.171
   ip nhrp network-id 200000
   ip nhrp holdtime 300
   ip nhrp nhs 10.0.0.1
   ip tcp adjust-mss 1360
   delay 1000
   tunnel source Ethernet0
   tunnel destination 192.0.2.171
   tunnel protection ipsec profile vpnprof!
```

2547oDMVPN with Traffic Segmentation (with BGP only): Example

The following example show a traffic segmentation configuration in which traffic is segmented between two spokes that serve as provider edge (PE) devices.

Hub Configuration

```
hostname hub-pe1

boot-start-marker
boot-end-marker

no aaa new-model

resource policy

clock timezone EST 0
ip cef
no ip domain lookup

!This section refers to the forwarding table for VRF blue:
ip vrf blue
 rd 2:2
 route-target export 2:2
 route-target import 2:2

!This section refers to the forwarding table for VRF red:
ip vrf red
 rd 1:1
 route-target export 1:1
 route-target import 1:1

mpls label protocol ldp

crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0

crypto ipsec transform-set t1 esp-des
 mode transport

crypto ipsec profile prof
 set transform-set t1

interface Tunnel1
 ip address 10.9.9.1 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
 ip nhrp network-id 1

!The command below enables MPLS on the DMVPN network:
mpls ip
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile prof

interface Loopback0
 ip address 10.0.0.1 255.255.255.255

interface Ethernet0/0
 ip address 172.0.0.1 255.255.255.0
```

```
!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes
learned from the spokes and advertises the VPN prefix:
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.0.0.11 remote-as 1
 neighbor 10.0.0.11 update-source Tunnel1
 neighbor 10.0.0.12 remote-as 1
 neighbor 10.0.0.12 update-source Tunnel1
 no auto-summary

 address-family vpnv4
 neighbor 10.0.0.11 activate
 neighbor 10.0.0.11 send-community extended
 neighbor 10.0.0.11 route-reflector-client
 neighbor 10.0.0.11 route-map NEXTHOP out
 neighbor 10.0.0.12 activate
 neighbor 10.0.0.12 send-community extended
 neighbor 10.0.0.12 route-reflector-client
 neighbor 10.0.0.12 route-map NEXTHOP out
 exit-address-family

 address-family ipv4 vrf red
 redistribute connected
 no synchronization
 exit-address-family

 address-family ipv4 vrf blue
 redistribute connected
 no synchronization
 exit-address-family

no ip http server
no ip http secure-server

!In this route map information, the hub sets the next hop to itself, and the VPN prefixes
are advertised:
route-map NEXTHOP permit 10
 set ip next-hop 10.0.0.1

control-plane

line con 0
 logging synchronous
line aux 0
line vty 0 4
 no login

end
```

Spoke Configurations

Spoke 2

```
hostname spoke-pe2

boot-start-marker
boot-end-marker

no aaa new-model
```

```
resource policy

clock timezone EST 0
ip cef
no ip domain lookup

!This section refers to the forwarding table for VRF blue:
ip vrf blue
 rd 2:2
 route-target export 2:2
 route-target import 2:2

!This section refers to the forwarding table for VRF red:
ip vrf red
 rd 1:1
 route-target export 1:1
 route-target import 1:1

mpls label protocol ldp

crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0

crypto ipsec transform-set t1 esp-des
 mode transport

crypto ipsec profile prof
 set transform-set t1

interface Tunnel1
 ip address 10.0.0.11 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
 ip nhrp map 10.0.0.1 172.0.0.1
 ip nhrp map multicast 172.0.0.1
 ip nhrp network-id 1
 ip nhrp nhs 10.0.0.1

!The command below enables MPLS on the DMVPN network:
 mpls ip
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile prof

interface Loopback0
 ip address 10.9.9.11 255.255.255.255

interface Ethernet0/0
 ip address 172.0.0.11 255.255.255.0
!
```

```
!
interface Ethernet1/0
 ip vrf forwarding red
 ip address 192.168.11.2 255.255.255.0

interface Ethernet2/0
 ip vrf forwarding blue
 ip address 192.168.11.2 255.255.255.0

!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes
learned from the spokes and advertises the VPN prefix:
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.0.0.1 remote-as 1
 neighbor 10.0.0.1 update-source Tunnel1
 no auto-summary

 address-family vpnv4
 neighbor 10.0.0.1 activate
 neighbor 10.0.0.1 send-community extended
 exit-address-family

!
 address-family ipv4 vrf red
 redistribute connected
 no synchronization
 exit-address-family

 !
 address-family ipv4 vrf blue
 redistribute connected
 no synchronization
 exit-address-family

no ip http server
no ip http secure-server

control-plane

line con 0
 logging synchronous
line aux 0
line vty 0 4
 no login

end
```

Spoke 3

```
hostname spoke-PE3

boot-start-marker
boot-end-marker

no aaa new-model

resource policy

clock timezone EST 0
ip cef
no ip domain lookup
```

```
!This section refers to the forwarding table for VRF blue:
ip vrf blue
 rd 2:2
 route-target export 2:2
 route-target import 2:2

!This section refers to the forwarding table for VRF red:
ip vrf red
 rd 1:1
 route-target export 1:1
 route-target import 1:1

mpls label protocol ldp

crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0

crypto ipsec transform-set t1 esp-des
 mode transport

crypto ipsec profile prof
 set transform-set t1

interface Tunnel1
 ip address 10.0.0.12 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
 ip nhrp map 10.0.0.1 172.0.0.1
 ip nhrp map multicast 172.0.0.1
 ip nhrp network-id 1
 ip nhrp nhs 10.0.0.1

!The command below enables MPLS on the DMVPN network:
 mpls ip
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile prof
!
interface Loopback0
 ip address 10.9.9.12 255.255.255.255

interface Ethernet0/0
 ip address 172.0.0.12 255.255.255.0

interface Ethernet1/0
 ip vrf forwarding red
 ip address 192.168.12.2 255.255.255.0

interface Ethernet2/0
 ip vrf forwarding blue
 ip address 192.168.12.2 255.255.255.0

!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes
learned from the spokes and advertises the VPN prefix:
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.0.0.1 remote-as 1
 neighbor 10.0.0.1 update-source Tunnel1
 no auto-summary
```

```
 address-family vpnv4
 neighbor 10.0.0.1 activate
 neighbor 10.0.0.1 send-community extended
 exit-address-family

 address-family ipv4 vrf red
 redistribute connected
 no synchronization
 exit-address-family

 address-family ipv4 vrf blue
 redistribute connected
 no synchronization
 exit-address-family

no ip http server
no ip http secure-server

control-plane

line con 0
 logging synchronous
line aux 0
line vty 0 4
 no login

end
```

2547oDMVPN with Traffic Segmentation (Enterprise Branch): Example

The following example shows a configuration for segmenting traffic between two spokes located at branch offices of an enterprise. In this example, EIGRP is configured to learn routes to reach BGP neighbors within the DMVPN.

Hub Configuration

```
hostname HUB

boot-start-marker
boot-end-marker

no aaa new-model

resource policy

clock timezone EST 0
ip cef
no ip domain lookup

!This section refers to the forwarding table for VRF blue:
ip vrf blue
 rd 2:2
 route-target export 2:2
 route-target import 2:2
```

```
!This refers to the forwarding table for VRF red:
ip vrf red
 rd 1:1
 route-target export 1:1
 route-target import 1:1

mpls label protocol ldp

crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0

crypto ipsec transform-set t1 esp-des
 mode transport

crypto ipsec profile prof
 set transform-set t1

interface Tunnel1
 ip address 10.0.0.1 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
 ip nhrp network-id 1

!EIGRP is enabled on the DMVPN network to learn the IGP prefixes:
no ip split-horizon eigrp 1

!The command below enables MPLS on the DMVPN network:
 mpls ip
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile prof

!This address is advertised by EIGRP and used as the BGP endpoint:
interface Loopback0
 ip address 10.9.9.1 255.255.255.255

interface Ethernet0/0
 ip address 172.0.0.1 255.255.255.0

!EIGRP is configured to learn the BGP peer addresses (10.9.9.x networks)
router eigrp 1
 network 10.9.9.1 0.0.0.0
 network 10.0.0.0 0.0.0.255
 no auto-summary

!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes
learned from the spokes and advertises the VPN prefix:
router bgp 1
 no synchronization
 bgp router-id 10.9.9.1
 bgp log-neighbor-changes
 neighbor 10.9.9.11 remote-as 1
 neighbor 10.9.9.11 update-source Loopback0
 neighbor 10.9.9.12 remote-as 1
 neighbor 10.9.9.12 update-source Loopback0
 no auto-summary

 address-family vpnv4
 neighbor 10.9.9.11 activate
 neighbor 10.9.9.11 send-community extended
 neighbor 10.9.9.11 route-reflector-client
```

```
 neighbor 10.9.9.12 activate
 neighbor 10.9.9.12 send-community extended
 neighbor 10.9.9.12 route-reflector-client
 exit-address-family

 address-family ipv4 vrf red
 redistribute connected
 no synchronization
 exit-address-family

 address-family ipv4 vrf blue
 redistribute connected
 no synchronization
 exit-address-family

no ip http server
no ip http secure-server

control-plane

line con 0
 logging synchronous
line aux 0
line vty 0 4
 no login

end
```

Spoke Configurations

Spoke 2

```
hostname Spoke2

boot-start-marker
boot-end-marker

no aaa new-model

resource policy

clock timezone EST 0
ip cef
no ip domain lookup

!This section refers to the forwarding table for VRF blue:
ip vrf blue
 rd 2:2
 route-target export 2:2
 route-target import 2:2

!This section refers to the forwarding table for VRF red:
ip vrf red
 rd 1:1
 route-target export 1:1
 route-target import 1:1

mpls label protocol ldp

crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0
```

```
crypto ipsec transform-set t1 esp-des
 mode transport

crypto ipsec profile prof
 set transform-set t1

interface Tunnel1
 ip address 10.0.0.11 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
 ip nhrp map 10.0.0.1 172.0.0.1
 ip nhrp map multicast 172.0.0.1
 ip nhrp network-id 1
 ip nhrp nhs 10.0.0.1

!The command below enables MPLS on the DMVPN network:
 mpls ip
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile prof

!This address is advertised by EIGRP and used as the BGP endpoint:
interface Loopback0
 ip address 10.9.9.11 255.255.255.255

interface Ethernet0/0
 ip address 172.0.0.11 255.255.255.0

interface Ethernet1/0
 ip vrf forwarding red
 ip address 192.168.11.2 255.255.255.0

interface Ethernet2/0
 ip vrf forwarding blue
 ip address 192.168.11.2 255.255.255.0

!EIGRP is enabled on the DMVPN network to learn the IGP prefixes:
router eigrp 1
 network 10.9.9.11 0.0.0.0
 network 10.0.0.0 0.0.0.255
 no auto-summary

!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes
learned from the spokes and advertises the VPN prefix:
router bgp 1
 no synchronization
 bgp router-id 10.9.9.11
 bgp log-neighbor-changes
 neighbor 10.9.9.1 remote-as 1
 neighbor 10.9.9.1 update-source Loopback0
 no auto-summary

 address-family vpnv4
 neighbor 10.9.9.1 activate
 neighbor 10.9.9.1 send-community extended
 exit-address-family

 address-family ipv4 vrf red
 redistribute connected
 no synchronization
 exit-address-family
```

```
 address-family ipv4 vrf blue
 redistribute connected
 no synchronization
 exit-address-family

no ip http server
no ip http secure-server

control-plane

line con 0
 logging synchronous
line aux 0
line vty 0 4
 no login

end
```

Spoke 3

```
hostname Spoke3

boot-start-marker
boot-end-marker

no aaa new-model

resource policy

clock timezone EST 0
ip cef
no ip domain lookup

!This section refers to the forwarding table for VRF blue:
ip vrf blue
 rd 2:2
 route-target export 2:2
 route-target import 2:2

!This section refers to the forwarding table for VRF red:
ip vrf red
 rd 1:1
 route-target export 1:1
 route-target import 1:1

mpls label protocol ldp

crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0

crypto ipsec transform-set t1 esp-des
 mode transport

crypto ipsec profile prof
 set transform-set t1

interface Tunnel1
 ip address 10.0.0.12 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
```

```
 ip nhrp map 10.0.0.1 172.0.0.1
 ip nhrp map multicast 172.0.0.1
 ip nhrp network-id 1
 ip nhrp nhs 10.0.0.1

!The command below enables MPLS on the DMVPN network:
 mpls ip
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile prof

!This address is advertised by EIGRP and used as the BGP endpoint:
interface Loopback0
 ip address 10.9.9.12 255.255.255.255

interface Ethernet0/0
 ip address 172.0.0.12 255.255.255.0

interface Ethernet1/0
 ip vrf forwarding red
 ip address 192.168.12.2 255.255.255.0

interface Ethernet2/0
 ip vrf forwarding blue
 ip address 192.168.12.2 255.255.255.0

!EIGRP is enabled on the DMVPN network to learn the IGP prefixes:
router eigrp 1
 network 10.9.9.12 0.0.0.0
 network 10.0.0.0 0.0.0.255
 no auto-summary

!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes
learned from the spokes and advertises the VPN prefix:
router bgp 1
 no synchronization
 bgp router-id 10.9.9.12
 bgp log-neighbor-changes
 neighbor 10.9.9.1 remote-as 1
 neighbor 10.9.9.1 update-source Loopback0
 no auto-summary

 address-family vpnv4
 neighbor 10.9.9.1 activate
 neighbor 10.9.9.1 send-community extended
 exit-address-family

 address-family ipv4 vrf red
 redistribute connected
 no synchronization
 exit-address-family

 address-family ipv4 vrf blue
 redistribute connected
 no synchronization
 exit-address-family

no ip http server
no ip http secure-server

control-plane
```

```
line con 0
 logging synchronous
line aux 0
line vty 0 4
 no login

end
```

Sample Command Output: show mpls ldp bindings

```
Spoke2# show mpls ldp bindings

  tib entry: 10.9.9.1/32, rev 8
        local binding:  tag: 16
        remote binding: tsr: 10.9.9.1:0, tag: imp-null
  tib entry: 10.9.9.11/32, rev 4
        local binding:  tag: imp-null
        remote binding: tsr: 10.9.9.1:0, tag: 16
  tib entry: 10.9.9.12/32, rev 10
        local binding:  tag: 17
        remote binding: tsr: 10.9.9.1:0, tag: 17
  tib entry: 10.0.0.0/24, rev 6
        local binding:  tag: imp-null
        remote binding: tsr: 10.9.9.1:0, tag: imp-null
  tib entry: 172.0.0.0/24, rev 3
        local binding:  tag: imp-null
        remote binding: tsr: 10.9.9.1:0, tag: imp-null
Spoke2#
```

Sample Command Output: show mpls forwarding-table

```
Spoke2# show mpls forwarding-table

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     Pop tag     10.9.9.1/32        0          Tu1        10.0.0.1
17     17          10.9.9.12/32       0          Tu1        10.0.0.1
18     Aggregate   192.168.11.0/24[V]   \
                                     0
19     Aggregate   192.168.11.0/24[V]   \
                                     0
Spoke2#
```

Sample Command Output: show ip route vrf red

```
Spoke2# show ip route vrf red

Routing Table: red
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

B    192.168.12.0/24 [200/0] via 10.9.9.12, 00:00:02
C    192.168.11.0/24 is directly connected, Ethernet1/0
Spoke2#
```

Sample Command Output: show ip route vrf blue

```
Spoke2# show ip route vrf blue

Routing Table: blue
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

B    192.168.12.0/24 [200/0] via 10.9.9.12, 00:00:08
C    192.168.11.0/24 is directly connected, Ethernet2/0
Spoke2#
Spoke2# show ip cef vrf red 192.168.12.0

192.168.12.0/24, version 5, epoch 0
0 packets, 0 bytes
  tag information set
    local tag: VPN-route-head
    fast tag rewrite with Tu1, 10.0.0.1, tags imposed: {17 18}
  via 10.9.9.12, 0 dependencies, recursive
    next hop 10.0.0.1, Tunnel1 via 10.9.9.12/32
    valid adjacency
    tag rewrite with Tu1, 10.0.0.1, tags imposed: {17 18}
Spoke2#
```

Sample Command Output: show ip bgp neighbors

```
Spoke2# show ip bgp neighbors

BGP neighbor is 10.9.9.1,  remote AS 1, internal link
  BGP version 4, remote router ID 10.9.9.1
  BGP state = Established, up for 00:02:09
  Last read 00:00:08, last write 00:00:08, hold time is 180, keepalive interval is 60
seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                4          4
    Keepalives:             4          4
    Route Refresh:          0          0
    Total:                  9          9
  Default minimum time between advertisement runs is 0 seconds

 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1/0
 Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
```

```
                                Sent       Rcvd
  Prefix activity:              ----       ----
    Prefixes Current:              0          0
    Prefixes Total:                0          0
    Implicit Withdraw:             0          0
    Explicit Withdraw:             0          0
    Used as bestpath:            n/a          0
    Used as multipath:           n/a          0

                                 Outbound    Inbound
  Local Policy Denied Prefixes:  --------    -------
    Total:                              0          0
  Number of NLRIs in the update sent: max 0, min 0

 For address family: VPNv4 Unicast
  BGP table version 9, neighbor version 9/0
 Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
                                Sent       Rcvd
  Prefix activity:              ----       ----
    Prefixes Current:              2          2 (Consumes 136 bytes)
    Prefixes Total:                4          2
    Implicit Withdraw:             2          0
    Explicit Withdraw:             0          0
    Used as bestpath:            n/a          2
    Used as multipath:           n/a          0

                                 Outbound    Inbound
  Local Policy Denied Prefixes:  --------    -------
    ORIGINATOR loop:                 n/a          2
    Bestpath from this peer:           4        n/a
    Total:                             4          2
  Number of NLRIs in the update sent: max 1, min 1

  Connections established 1; dropped 0
  Last reset never
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled
Local host: 10.9.9.11, Local port: 179
Foreign host: 10.9.9.1, Foreign port: 12365

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x2D0F0):
Timer          Starts    Wakeups            Next
Retrans             6          0             0x0
TimeWait            0          0             0x0
AckHold             7          3             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0

iss: 3328307266  snduna: 3328307756  sndnxt: 3328307756     sndwnd:  15895
irs: 4023050141  rcvnxt: 4023050687  rcvwnd:      16384  delrcvwnd:      0

SRTT: 165 ms, RTTO: 1457 ms, RTV: 1292 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs
IP Precedence value : 6
```

```
Datagrams (max data segment is 536 bytes):
Rcvd: 13 (out of order: 0), with data: 7, total data bytes: 545
Sent: 11 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 0), with
data: 6, total data bytes: 489
Spoke2#
```

Additional References

The following sections provide references related to Dynamic Multipoint VPN (DMVPN):

Related Documents

Related Topic	Document Title
Call Admission Control	*Call Admission Control for IKE*, Cisco IOS Release 12.4
GRE tunnel keepalive information	*Generic Routing Encapsulation (GRE) Tunnel Keepalive*, Cisco IOS Release 12.2(8)T
IKE configuration tasks such as defining an IKE policy	The chapter "*Configuring Internet Key Exchange for IPSec VPNs*" in the *Cisco IOS Security Configuration Guide*, Release 12.4
IPsec configuration tasks	*The chapter "Configuring Security for VPNs with IPSec"* in *the Cisco IOS Security Configuration Guide*, Release 12.4
Tunnel interface configuration tasks	*The section "Implementing Tunnels"* in the cha*pter* "Interface Configuration Overview" in the *Cisco IOS Interface and Hardware Component Configuration Guide*, Release 12.4
Configuring VRF-Aware IPsec	*VRF-Aware IPsec*, in the *Cisco IOS Security Configuration Guide*, Release 12.4
Configuring MPLS	*Cisco IOS Multiprotocol Label Switching Configuration Guide*, Release 12.4.
Configuring BGP	The chapter "BGP" in the *Cisco IOS IP Routing Protocols Configuration Guide*, Release 12.4
System messages	*12.4T System Message Guide*
Defining and configuring ISAKMP profiles	"Certificate to ISAKMP Profile Mapping" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.4

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2547	BGP/MPLS VPNs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features

- **clear dmvpn session**
- **clear dmvpn statistics**
- **debug dmvpn**
- **debug nhrp condition**
- **debug nhrp error**
- **logging dmvpn**
- **show dmvpn**
- **show ip nhrp traffic**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Feature Information for Dynamic Multipoint VPN (DMVPN)

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Dynamic Multipoint VPN (DMVPN)

Feature Name	Releases	Feature Information
2547oDMVPN—Enabling Traffic Segmentation Within DMVPN	12.4(11)T	The 2547oDMVPN feature allows users to segment VPN traffic within a DMVPN tunnel by applying MPLS labels to VRF instances to indicate the source and destination of each VRF.
Mangeability Enhancements for DMVPN	12.4(9)T	DMVPN session manageabilty was expanded with DMVPN specific commands for debugging, show output, session and counter control, and system log information. The following sections provide information about this feature: • Troubleshooting Dynamic Multipoint VPN (DMVPN) The following commands were introduced or modified by this feature: **clear dmvpn session**, **clear dmvpn statistics**, **debug dmvpn**, **debug nhrp condition**, **debug nhrp error**, **logging dmvpn**, **show dmvpn**, **show ip nhrp traffic**
DMVPN Phase 2	12.2(18)SXE 12.3(9)a 12.3(8)T1	DMVPN Spoke-to-Spoke functionality was made more production ready. If you are using this functionality in a production network, the minimum release is Release 12.3(9a) or Release 12.3(8)T1. In Release 12.2(18)SXE, support was added for the Cisco Catalyst 6500 series switch and the Cisco 7600 series router.

Table 1 Feature Information for Dynamic Multipoint VPN (DMVPN)

Feature Name	Releases	Feature Information
—	12.3(6) 12.3(7)T	Virtual Route Forwarding Integrated DMVPN and Network Address Translation-Transparency (NAT-T) Aware DMVPN enhancements were added. In addition, DMVPN Hub-to-Spoke functionality was made more production ready. If you are using this functionality in a production network, the minimum release requirement is Cisco IOS Release12.3(6) or 12.3(7)T. The enhancements added in Cisco IOS Release 12.3(6) were integrated into Cisco IOS Release 12.3(7)T.
Dynamic Multipoint VPN (DMVPN) Phase 1	12.2(13)T	The Dynamic Multipoint VPN (DMVPN) feature allows users to better scale large and small IPsec Virtual Private Networks (VPNs) by combining generic routing encapsulation (GRE) tunnels, IP security (IPsec) encryption, and Next Hop Resolution Protocol (NHRP).
DMVPN - Phase 2	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Dynamic Multipoint VPN (DMVPN) Phase 1	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Glossary

AM—aggressive mode. A mode during IKE negotiation. Compared to MM, AM eliminates several steps, making it faster but less secure than MM. Cisco IOS software will respond in aggressive mode to an IKE peer that initiates aggressive mode.

GRE—generic routing encapsulation. Tunnels that provide a specific pathway across the shared WAN and encapsulate traffic with new packet headers to ensure delivery to specific destinations. The network is private because traffic can enter a tunnel only at an endpoint. Tunnels do not provide true confidentiality (encryption does) but can carry encrypted traffic.

GRE tunneling can also be used to encapsulate non-IP traffic into IP and send it over the Internet or IP network. The Internet Package Exchange (IPX) and AppleTalk protocols are examples of non-IP traffic.

IKE—Internet Key Exchange. A hybrid protocol that implements Oakley key exchange and Skeme key exchange inside the ISAKMP framework. Although IKE can be used with other protocols, its initial implementation is with IPsec. IKE provides authentication of the IPsec peers, negotiates IPsec keys, and negotiates IPsec security associations.

IPsec—IP security. A framework of open standards developed by the Internet Engineering Task Force (IETF). IPsec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (“peers”), such as Cisco routers.

ISAKMP—Internet Security Association Key Management Protocol. A protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.

MM—main mode. Mode that is slower than aggressive mode but more secure and more flexible than aggressive mode because it can offer an IKE peer more security proposals. The default action for IKE authentication (rsa-sig, rsa-encr, or preshared) is to initiate main mode.

NHRP—Next Hop Resolution Protocol. Routers, access servers, and hosts can use NHRP to discover the addresses of other routers and hosts connected to a NBMA network.

The Cisco implementation of NHRP supports the IETF draft version 11 of *NBMA Next Hop Resolution Protocol (NHRP)*.

The Cisco implementation of NHRP supports IP Version 4, Internet Packet Exchange (IPX) network layers, and, at the link layer, ATM, Ethernet, SMDS, and multipoint tunnel networks. Although NHRP is available on Ethernet, NHRP need not be implemented over Ethernet media because Ethernet is capable of broadcasting. Ethernet support is unnecessary (and not provided) for IPX.

PFS—Perfect Forward Secrecy. A cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised, because subsequent keys are not derived from previous keys.

SA—security association. Describes how two or more entities will utilize security services to communicate securely. For example, an IPsec SA defines the encryption algorithm (if used), the authentication algorithm, and the shared session key to be used during the IPsec connection.

Both IPsec and IKE require and use SAs to identify the parameters of their connections. IKE can negotiate and establish its own SA. The IPsec SA is established either by IKE or by manual user configuration.

transform—The list of operations done on a dataflow to provide data authentication, data confidentiality, and data compression. For example, one transform is the ESP protocol with the HMAC-MD5 authentication algorithm; another transform is the AH protocol with the 56-bit DES encryption algorithm and the ESP protocol with the HMAC-SHA authentication algorithm.

VPN—Virtual Private Network. A framework that consists of multiple peers transmitting private data securely to one another over an otherwise public infrastructure. In this framework, inbound and outbound network traffic is protected using protocols that tunnel and encrypt all data. This framework permits networks to extend beyond their local topology, while remote users are provided with the appearance and functionality of a direct network connection.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Cisco Easy VPN Remote

First Published: November 25, 2002
Last Updated: November 4, 2008

This document provides information on configuring and monitoring the Cisco Easy VPN Remote feature to create IPsec Virtual Private Network (VPN) tunnels between a supported router and an Easy VPN server (Cisco IOS router, VPN 3000 concentrator, or Cisco PIX Firewall) that supports this form of IPsec encryption and decryption.

For the benefits of this feature, see the section "Benefits of the Cisco Easy VPN Remote Feature."

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Easy VPN Remote" section on page 108.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Cisco Easy VPN Remote

Cisco Easy VPN Remote Feature

- A Cisco 800 series router running Cisco IOS Release 12.2(15)T, 12.3(2)T, 12.3(4)T, 12.3(7)T, or 12.3(7)XR2 configured as a Cisco Easy VPN remote.
- A Cisco 1700 series router running Cisco IOS Release 12.2(15)T, 12.3(2)T, 12.3(4)T, 12.3(7)T, or 12.3(7)XR, configured as a Cisco Easy VPN remote.
- A Cisco 1800 series fixed configuration router running Cisco IOS Release 12.3(8)YI.
- A Cisco uBR905 or Cisco uBR925 cable access router running Cisco IOS Release 12.2(15)T, configured as a Cisco Easy VPN remote.
- Another Cisco router or VPN concentrator that supports the Cisco Easy VPN Server feature and that is configured as a Cisco IOS Easy VPN server. See the "Required Easy VPN Servers" section for a detailed list.

Reactivate Primary Peer Feature

- An existing Easy VPN remote configuration can be enhanced to accommodate the Reactivate Primary Peer feature using the **peer** command (and **default** keyword) and the **idle-time** command. After the tunnel between the Easy VPN remote and a nondefault peer is working, the Reactivate Primary Peer features takes effect, that is, the Easy VPN remote periodically tries to check the connectivity with the primary peer. Any time the Easy VPN remote detects that the link is working, the Easy VPN remote tears down the existing connection and brings up the tunnel with the primary peer.

Restrictions for Cisco Easy VPN Remote

Required Easy VPN Servers

The Cisco Easy VPN Remote feature requires that the destination peer be a Cisco IOS Easy VPN server or VPN concentrator that supports the Cisco Easy VPN Server feature. At the time of publication, servers or concentrators that support this feature include the following platforms when running the indicated software releases:

- Cisco 806, Cisco 826, Cisco 827, Cisco 828, Cisco 831, Cisco 836, and Cisco 837 routers—Cisco IOS Release 12.2(8)T or later release. Cisco 800 series routers are not supported in Cisco IOS Release 12.3(7)XR, but they are supported in Cisco IOS Release 12.3(7)XR2.
- Cisco 870 series—Cisco IOS Release 12.3(8)YI1.
- Cisco 1700 series—Cisco IOS Release 12.2(8)T or later release.
- Cisco 1800 series fixed configuration router—Cisco IOS Release 12.3(8)YI.
- Cisco 1812 router—Cisco IOS Release 12.3(8)YH.
- Cisco 2600 series—Cisco IOS Release 12.2(8)T or later release.
- Cisco 3620—Cisco IOS Release 12.2(8)T or later release.
- Cisco 3640—Cisco IOS Release 12.2(8)T or later release.
- Cisco 3660—Cisco IOS Release 12.2(8)T or later release.
- Cisco 7100 series VPN routers—Cisco IOS Release 12.2(8)T or later release.
- Cisco 7200 series routers—Cisco IOS Release 12.2(8)T or later release.

- Cisco 7500 series routers—Cisco IOS Release 12.2(8)T or later release.
- Cisco PIX 500 series—Software Release 6.2 or later release.
- Cisco VPN 3000 series—Software Release 3.11 or later release.

Only ISAKMP Policy Group 2 Supported on Easy VPN Servers

The Unity Protocol supports only Internet Security Association Key Management Protocol (ISAKMP) policies that use group 2 (1024-bit Diffie-Hellman) Internet Key Exchange (IKE) negotiation, so the Easy VPN server being used with the Cisco Easy VPN Remote feature must be configured for a group 2 ISAKMP policy. The Easy VPN server cannot be configured for ISAKMP group 1 or group 5 when being used with a Cisco Easy VPN client.

Transform Sets Supported

To ensure a secure tunnel connection, the Cisco Easy VPN Remote feature does not support transform sets that provide encryption without authentication (ESP-DES and ESP-3DES) or transform sets that provide authentication without encryption (ESP-NULL ESP-SHA-HMAC and ESP-NULL ESP-MD5-HMAC).

Note The Cisco Unity Client Protocol does not support Authentication Header (AH) authentication, but Encapsulation Security Protocol (ESP) is supported.

Dial Backup for Easy VPN Remotes

Line-status–based backup is not supported in this feature.

Network Address Translation Interoperability Support

Network Address Translation (NAT) interoperability is not supported in client mode with split tunneling.

Multicast and Static NAT

Multicast and static NAT are supported only for Easy VPN remotes using dynamic virtual tunnel interfaces (DVTIs).

Virtual IPsec Interface Restrictions

- For the Virtual IPsec Interface Support feature to work, virtual templates support is needed.
- If you are using a virtual tunnel interface on the Easy VPN remote device, it is recommended that you configure the server for a virtual tunnel interface.

Dual Tunnel Support

The following restrictions apply if you are using dual tunnels that share common inside and outside interfaces:

- If dual tunnels are configured, one of the tunnels should have a split tunnel configured on the server.
- Web Intercept can be configured for only one of the tunnels. Web Intercept should not be used for the voice tunnel.
- Web Intercept cannot be used for IP phones until authorization proxy becomes aware of how to bypass the IP phone.
- Some features, such as Pushing a Configuration URL Through a Mode-Configuration Exchange, can be used only through a single tunnel.

cTCP Support on Easy VPN Cients

- cTCP listens on only up to 10 ports.
- If there are other applications registered for the port on which cTCP is enabled, those applications will not work.

Information About Cisco Easy VPN Remote

To configure the Cisco Easy VPN Remote features, you should understand the following concepts:

Benefits of the Cisco Easy VPN Remote Feature

- Allows dynamic configuration of end-user policy, requiring less manual configuration by end users and field technicians, thus reducing errors and further service calls.
- Allows the provider to change equipment and network configurations as needed, with little or no reconfiguration of the end-user equipment.
- Provides for centralized security policy management.
- Enables large-scale deployments with rapid user provisioning.
- Eliminates the need for end users to purchase and configure external VPN devices.
- Eliminates the need for end users to install and configure Easy VPN Client software on their PCs.
- Offloads the creation and maintenance of the VPN connections from the PC to the router.
- Reduces interoperability problems between the different PC-based software VPN clients, external hardware-based VPN solutions, and other VPN applications.
- Sets up a single IPsec tunnel regardless of the number of multiple subnets that are supported and the size of the split-include list.

Cisco Easy VPN Remote Overview

Cable modems, xDSL routers, and other forms of broadband access provide high-performance connections to the Internet, but many applications also require the security of VPN connections that perform a high level of authentication and that encrypt the data between two particular endpoints. However, establishing a VPN connection between two routers can be complicated and typically requires tedious coordination between network administrators to configure the VPN parameters of the two routers.

The Cisco Easy VPN Remote feature eliminates much of this tedious work by implementing Cisco Unity Client Protocol, which allows most VPN parameters to be defined at a Cisco IOS Easy VPN server. This server can be a dedicated VPN device, such as a Cisco VPN 3000 concentrator or a Cisco PIX Firewall or a Cisco IOS router that supports the Cisco Unity Client Protocol.

After the Cisco Easy VPN server has been configured, a VPN connection can be created with minimal configuration on an Easy VPN remote, such as a Cisco 800 series router or a Cisco 1700 series router. When the Easy VPN remote initiates the VPN tunnel connection, the Cisco Easy VPN server pushes the IPsec policies to the Easy VPN remote and creates the corresponding VPN tunnel connection.

The Cisco Easy VPN Remote feature provides for automatic management of the following details:

- Negotiating tunnel parameters, such as addresses, algorithms, and lifetime.
- Establishing tunnels according to the parameters that were set.
- Automatically creating the NAT or Port Address Translation (PAT) and associated access lists that are needed, if any.
- Authenticating users, that is, ensuring that users are who they say they are by way of usernames, group names, and passwords.
- Managing security keys for encryption and decryption.
- Authenticating, encrypting, and decrypting data through the tunnel.

Modes of Operation

The Cisco Easy VPN Remote feature supports three modes of operation: client, network extension, and network extension plus:

- Client—Specifies that NAT or PAT be done so that the PCs and other hosts at the remote end of the VPN tunnel form a private network that does not use any IP addresses in the IP address space of the destination server.

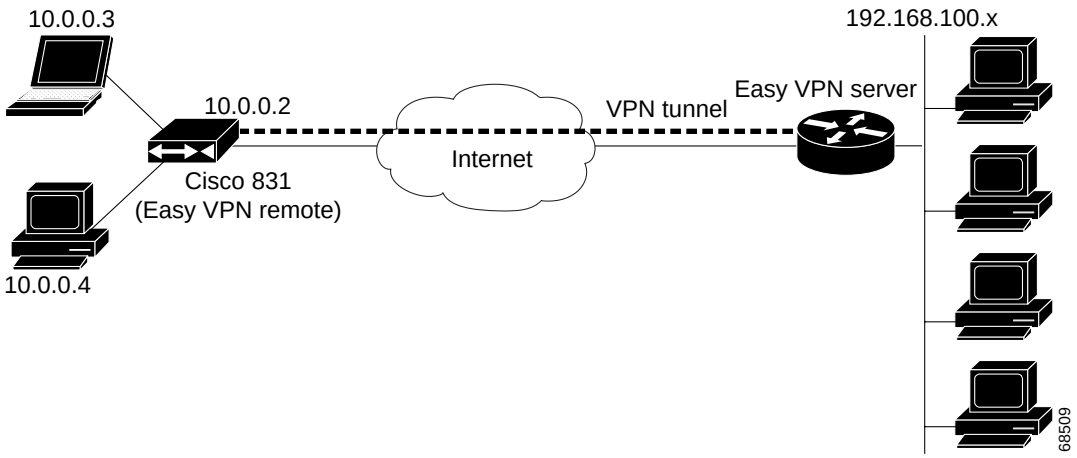
  An enhancement has been made so that the IP address that is received via mode configuration is automatically assigned to an available loopback interface. The IPsec Security Associations (SAs) for this IP address are automatically created by Easy VPN Remote. The IP address is typically used for troubleshooting (using ping, Telnet, and Secure Shell).
- Network extension—Specifies that the PCs and other hosts at the client end of the VPN tunnel should be given IP addresses that are fully routable and reachable by the destination network over the tunneled network so that they form one logical network. PAT is not used, which allows the client PCs and hosts to have direct access to the PCs and hosts at the destination network.
- Network extension plus (mode network-plus)—Identical to network extension mode with the additional capability of being able to request an IP address via mode configuration and automatically assign it to an available loopback interface. The IPsec SAs for this IP address are automatically created by Easy VPN Remote. The IP address is typically used for troubleshooting (using ping, Telnet, and Secure Shell).

All modes of operation also optionally support split tunneling, which allows secure access to corporate resources through the VPN tunnel while also allowing Internet access through a connection to an Internet service provider (ISP) or other service—thereby eliminating the corporate network from the path for web access.

Client Mode and Network Extension Mode Scenarios

Figure 1 illustrates the client mode of operation. In this example, the Cisco 831 router provides access to two PCs, which have IP addresses in the 10.0.0.0 private network space. These PCs connect to the Ethernet interface on the Cisco 831 router, which also has an IP address in the 10.0.0.0 private network space. The Cisco 831 router performs NAT or PAT translation over the VPN tunnel so that the PCs can access the destination network.

Figure 1 Cisco Easy VPN Remote Connection



Note The diagram in Figure 1 could also represent a split tunneling connection, in which the client PCs can access public resources in the global Internet without including the corporate network in the path for the public resources.

Figure 2 also illustrates the client mode of operation, in which a VPN concentrator provides destination endpoints to multiple xDSL clients. In this example, Cisco 800 series routers provide access to multiple small business clients, each of which uses IP addresses in the 10.0.0.0 private network space. The Cisco 800 series routers perform NAT or PAT translation over the VPN tunnel so that the PCs can access the destination network.

Figure 2 Cisco Easy VPN Remote Connection (using a VPN concentrator)

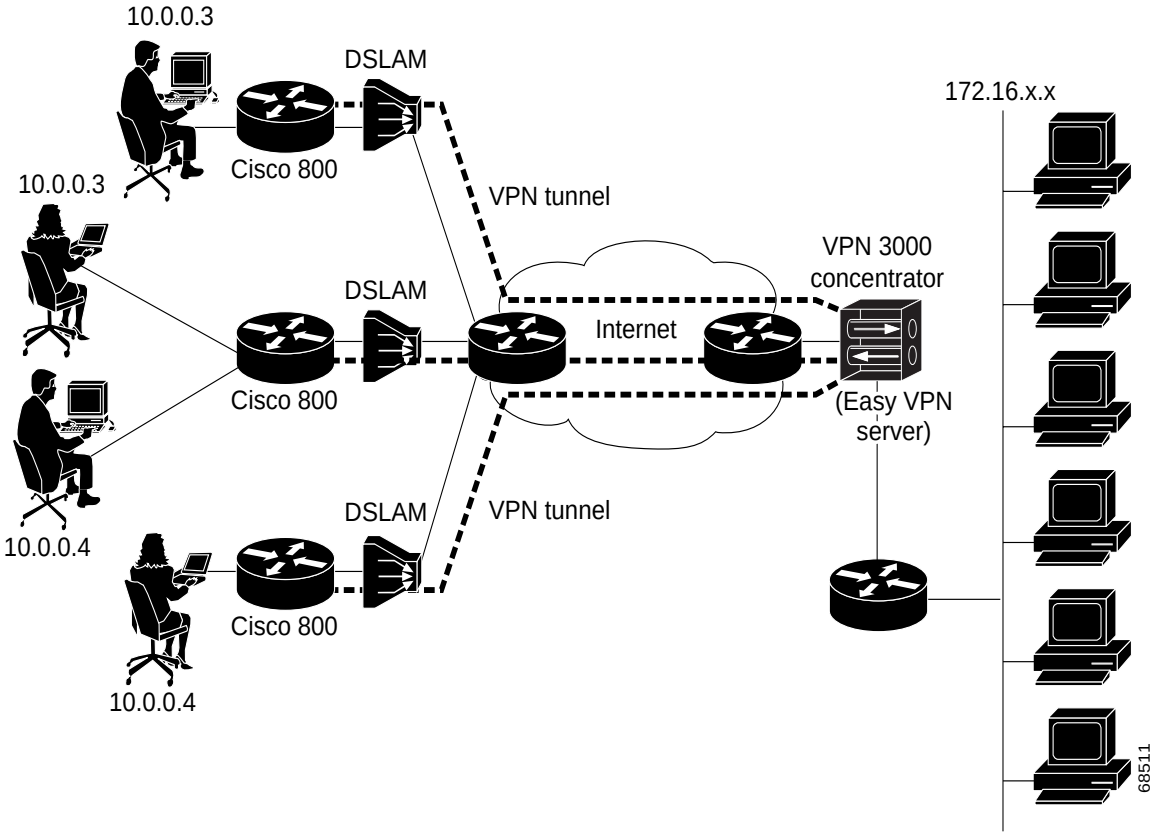
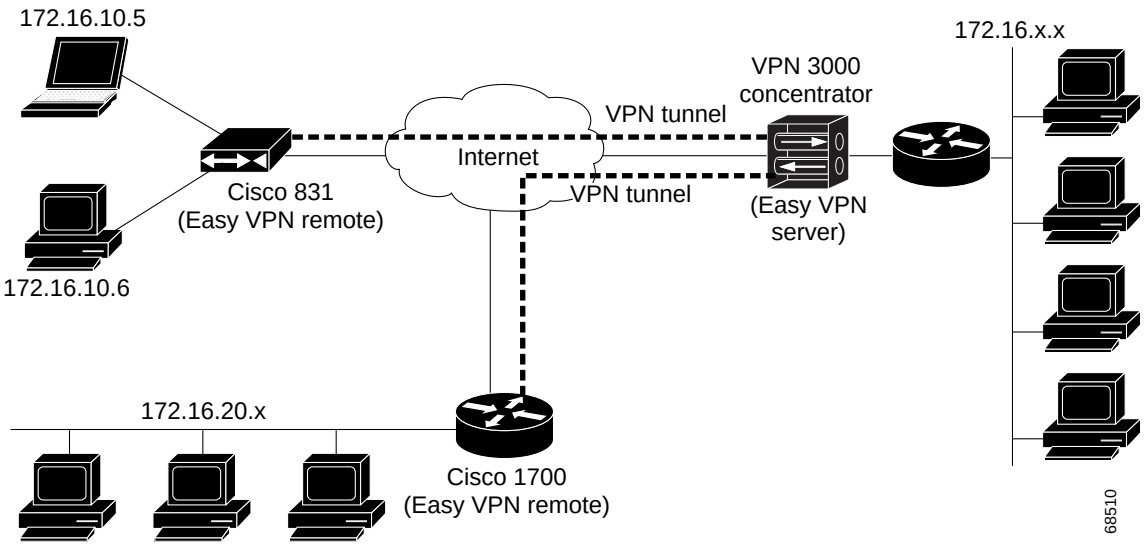


Figure 3 illustrates the network extension mode of operation. In this example, the Cisco 831 router and Cisco 1700 series router both act as Cisco Easy VPN remote devices, connecting to a Cisco VPN 3000 concentrator.

The client hosts are given IP addresses that are fully routable by the destination network over the tunnel. These IP addresses could be either in the same subnet space as the destination network or in separate subnets, assuming that the destination routers are configured to properly route those IP addresses over the tunnel.

In this example, the PCs and hosts attached to the two routers have IP addresses that are in the same address space as the destination enterprise network. The PCs connect to the Ethernet interface of the Cisco 831 router, which also has an IP address in the enterprise address space. This scenario provides a seamless extension of the remote network.

Figure 3 Cisco Easy VPN Network Extension Connection



Authentication

The Cisco Easy VPN Remote feature supports a two-stage process for authenticating the remote router to the central concentrator. The first step is Group Level Authentication and is part of the control channel creation. In this first stage, two types of authentication credentials can be used: either preshared keys or digital certificates. The following paragraphs provide details about these options.

The second authentication step is called Extended Authentication or Xauth. In this step, the remote side (in this case the Easy VPN router) submits a username and password to the central site router. This step is the same process as that which occurs when a user of the Cisco VPN software client on a PC enters his or her username and password to activate his or her VPN tunnel. When using the router, the difference is that the router itself is being authenticated to the network, not a PC with Cisco VPN Client software. Xauth is an optional step (it can be disabled) but is normally enabled to improve security. After Xauth is successful and the tunnel comes up, all PCs behind the Easy VPN remote router have access to the tunnel.

If Xauth is enabled, it is key to decide how to input the username and password. There are two options. The first option is to store the Xauth username and password in the configuration file of the router. This option is typically used if the router is shared between several PCs and the goal is to keep the VPN tunnel up all the time (see the section "Automatic Activation") or to have the router automatically bring up the tunnel whenever there is data to be sent (see the section "Traffic-Triggered Activation"). An example of this application is a branch office situation, in which the users in the branch office want the VPN tunnel to be available whenever they have data to send and do not want to have to do anything special to activate the VPN tunnel. If the PCs in the branch office must be individually authenticated on the basis of the ID of each user, the correct configuration is to put the Easy VPN router in Automatic Activation mode to keep the tunnel "up" all the time and to use Cisco IOS Authentication Proxy or 802.1x to authenticate the individual PCs. Because the tunnel is always up, Authentication Proxy or 802.1x can access a central site user database such as AAA/RADIUS to authenticate the individual user requests as they are submitted by PC users. (See the "Related Documents" sections "General information on IPsec and VPN" for a reference to configuring Authentication Proxy and "802.1x authentication" for a reference to configuring 802.1x authentication.)

The second option for entry of the Xauth username and password is not to store it on the router. Instead, a PC user who is connected to the router is presented with a special web page that allows the user to manually enter the username and password (see the section "Manual Activation"). The router sends the username and password to the central site concentrator, and if the username and password are correct, the tunnel comes up. The typical application for this configuration is a teleworker network. The teleworker wants to control when the tunnel is up and has to enter his or her personal user credentials (which could include one-time passwords) to activate the tunnel. Also, the network administrator may want teleworker tunnels up only when someone is using them to conserve resources on the central concentrators. (See the section "Web-Based Activation" for details about this configuration.)

The Xauth username and password can also be manually entered from the command-line interface (CLI) of the router. This method is not recommended for most situations because the user must first log in to the router (and needs a user ID on the router to do so). However, it can be useful for network administrators during troubleshooting.

Using Preshared Keys

Using preshared keys, each peer is aware of the key of the other peer. Preshared keys are displayed in running configurations, so they can be seen by anyone (referred to as clear format). When a more secure type of authentication is required, Cisco software also supports another type of preshared key: the encrypted preshared key.

Using an encrypted preshared key for authentication allows you to securely store plain-text passwords in type 6 (encrypted) format in NVRAM. A group preshared key can be preconfigured on both VPN-tunnel peers. The encrypted form of the keyword can be seen in the running configuration, but the actual keyword is not visible. (For more information about encrypted preshared keys, see *Encrypted Preshared Key*.)

Using Digital Certificates

Digital certificates provide for the support of Rivest, Shamir, and Adelman (RSA) signatures on Easy VPN remote devices. The support is provided through a RSA certificate that can be stored on or off the remote device.

Note The recommended timeout for Easy VPN using digital certificates is 40 seconds.

For more information about digital certificates, see the *Easy VPN Remote RSA Signature Support* feature guide, Release 12.3(7)T1.

Using Xauth

Xauth is an additional level of authentication that can be used. Xauth is applicable when either group preshared keys or digital certificates are used. Xauth credentials can be entered using a web interface manager, such as Security Device Manager (SDM), or using the CLI. (See the section "Cisco Easy VPN Remote Web Managers.")

The Save Password feature allows the Xauth username and password to be saved in the Easy VPN Remote configuration so that you are not required to enter the username and password manually. One-Time Passwords (OTPs) are not supported by the Save Password feature and must be entered manually when Xauth is requested. The Easy VPN server must be configured to "Allow Saved Passwords." (For more information about how to configure the Save Password feature, see the section "Dead Peer Detection Periodic Message Option.")

Xauth is controlled by the Easy VPN server. When the Cisco IOS Easy VPN server requests Xauth authentication, the following messages are displayed on the console of the router:

```
EZVPN: Pending XAuth Request, Please enter the following command:
crypto ipsec client ezvpn xauth
```

When you see this message, you can provide the necessary user ID, password, and other information by entering the **crypto ipsec client ezvpn connect** command and responding to the prompts that follow.

The recommended Xauth timeout is 50 seconds or fewer.

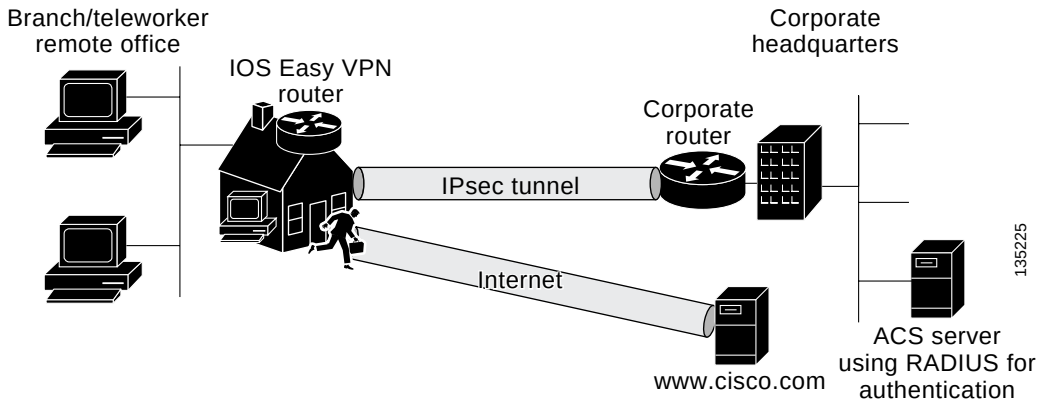
Note The timeout for entering the username and password is determined by the configuration of the Cisco IOS Easy VPN server. For servers running Cisco IOS software, this timeout value is specified by the **crypto isakmp xauth timeout** command.

Web-Based Activation

Web-Based Activation provides a user-friendly method for a remote teleworker to authenticate the VPN tunnel between his or her remote Easy VPN router and the central site router. This feature allows administrators to set up their remote LANs so that the initial HTTP request that is coming from any of the remote PCs is intercepted by the remote Easy VPN router. A login page is returned to the user, whereby the user may enter credentials to authenticate the VPN tunnel. After the VPN tunnel comes up, all users behind this remote site can access the corporate LAN without being reprompted for the username and password. Alternatively, the user may choose to bypass the VPN tunnel and connect only to the Internet, in which case a password is not required.

A typical application for web-based activation is a home teleworker who brings up the Easy VPN tunnel only when he or she needs to connect to the corporate LAN. If the remote teleworker is not present, other members of the household (such as a spouse or children) can use the Internet Only option to browse the Internet without activating the VPN tunnel. Figure 4 shows a typical scenario for web-based activation.

Figure 4 Typical Web-Based Activation Scenario



Note Entering the Xauth credentials brings up the tunnel for all users who are behind this remote site. After the tunnel is up, any additional PCs that are behind the remote site do not get prompted for Xauth credentials. Web-Based Activation is an authentication to bring up the VPN tunnel for all remote PCs and cannot be considered individual user authentication. Individual user authentication for VPN tunnel access is available using the Cisco IOS Authentication Proxy or 802.1x features, which can be

configured on the remote Easy VPN router. (See the "Related Documents" sections "General information on IPsec and VPN" for a reference to configuring Authentication Proxy and "802.1x authentication" for a reference to configuring 802.1x authentication.)

To configure web-based activation, see the section "Configuring Web-Based Activation."

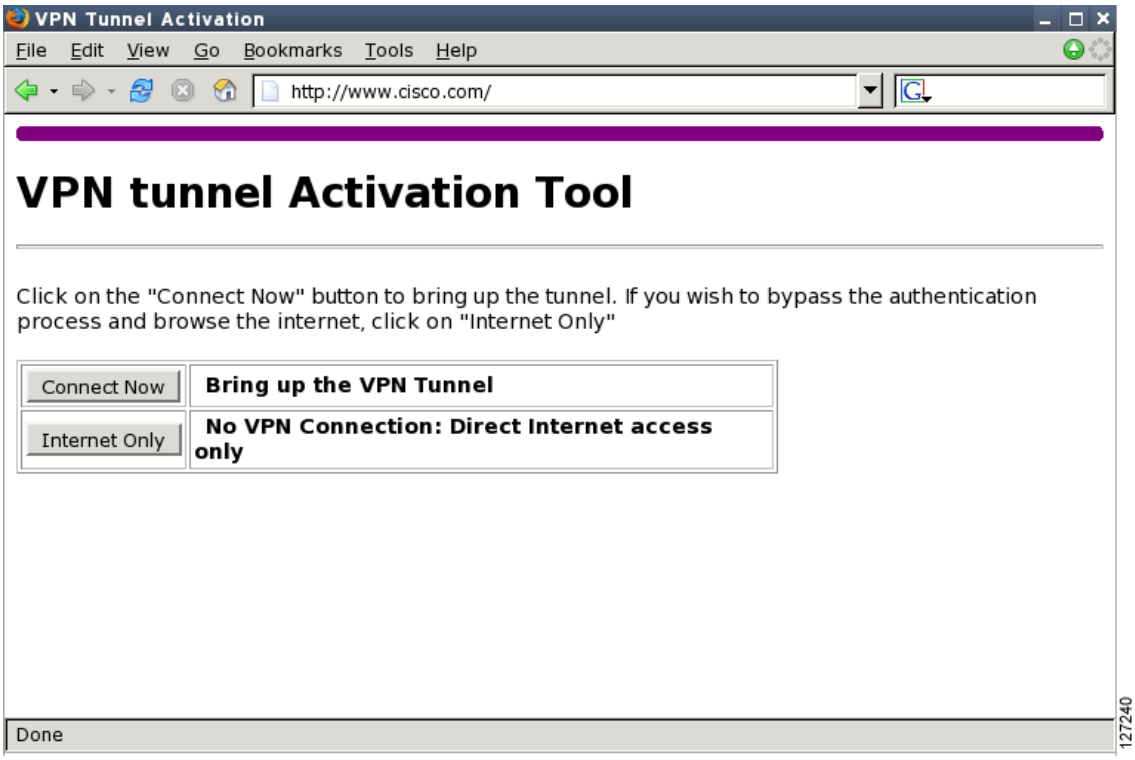
The following sections show the various screen shots that a remote teleworker sees when the Web-Based Activation feature is turned on:

Web-Based Activation Portal Page

Figure 5 is an example of a web-based activation portal page. The user may choose to connect to the corporate LAN by clicking Connect Now or he or she may choose to connect only to the Internet by clicking Internet Only.

Note If the user chooses to connect only to the Internet, a password is not required.

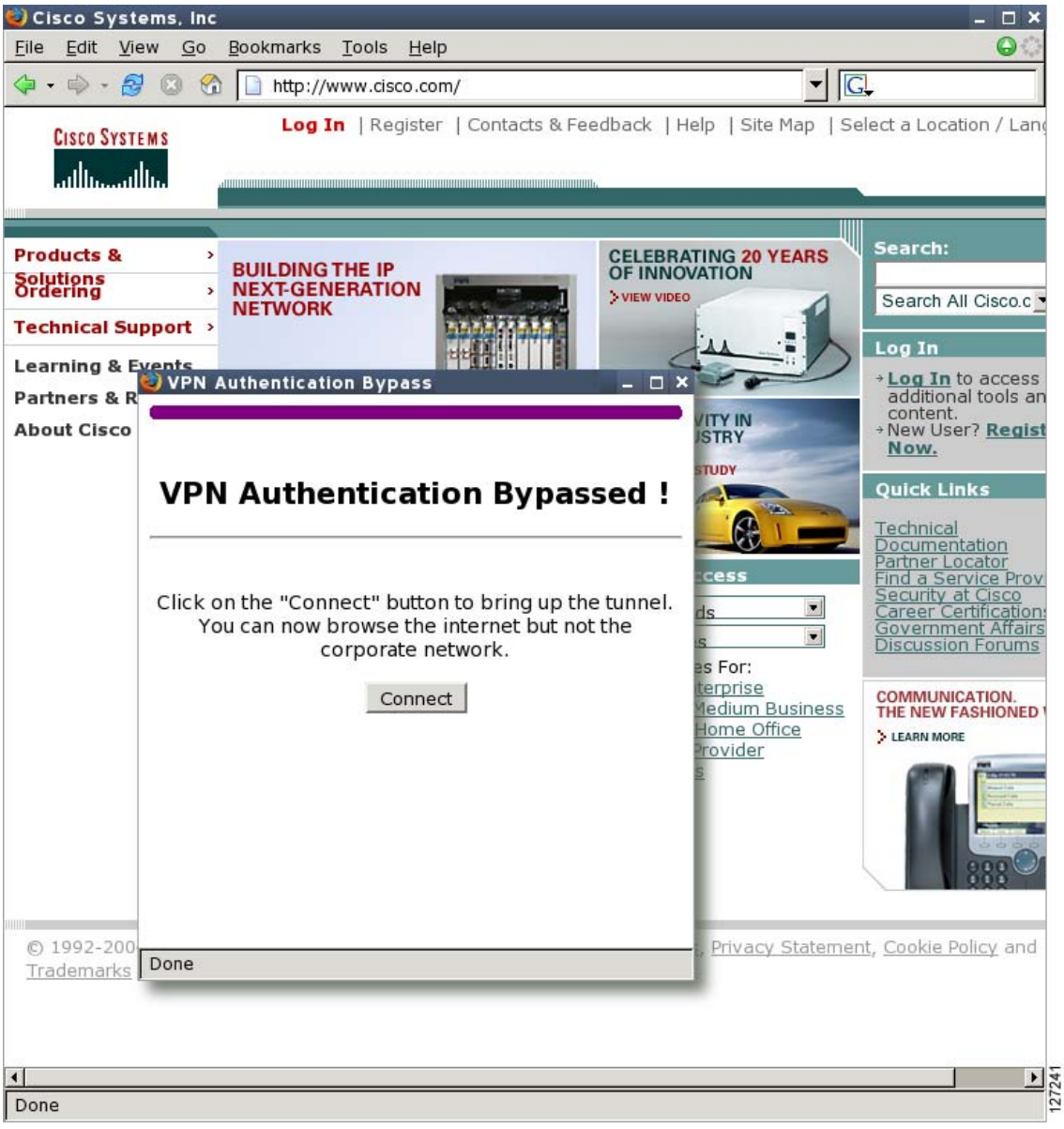
Figure 5 Portal Page



VPN Authentication Bypass

Figure 6 is an example of a web-based activation in which the user chose to connect only to the Internet by clicking the Internet Only option. This option is most useful for household members who need to browse the Internet while the remote teleworker is not available to authenticate the VPN tunnel for corporate use.

Figure 6 VPN Authentication Bypass Page



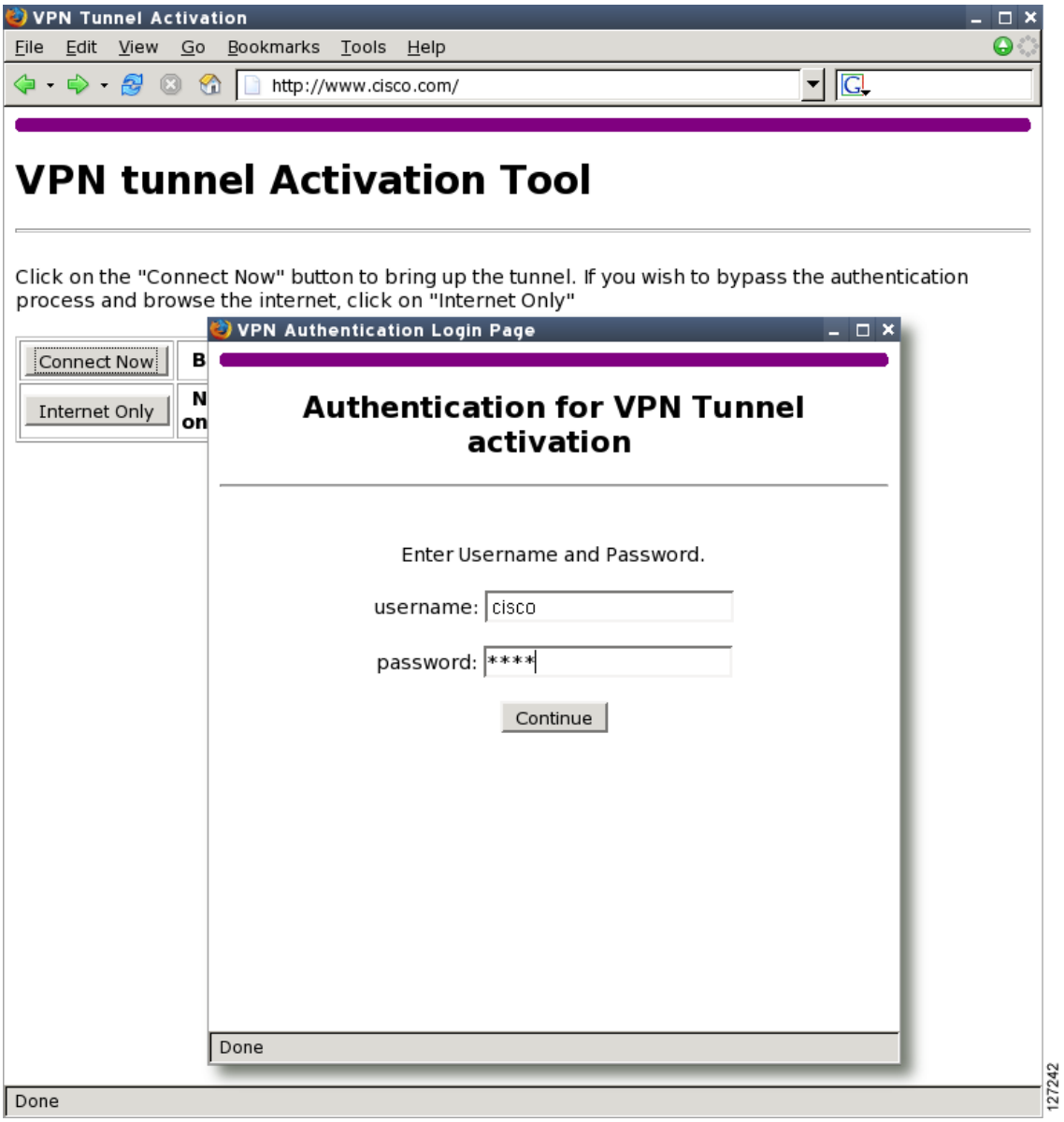
Note If the Web-Based Activation window is mistakenly closed, to connect again, a user should follow this two-step process:

1. In a browser, type “http://routeripaddress/ezvpn/bypass” and try to connect to the URL. Entering this URL clears the bypass state that was created for your IP address (when the “Internet only” button was pressed). If you get a message saying that no such page is found, it does not matter because the only purpose of accessing the URL is to clear the bypass state.
2. After clearing the bypass state, you can browse to any external site. The Connect and Bypass page appears again. You can connect to VPN by pressing the Connect button.

VPN Tunnel Authentication

Figure 7 is an example of a web-based activation in which the user chose to connect to the corporate LAN by entering a username and password. After the user is successfully authenticated, the Easy VPN tunnel is brought up for this remote site. If there are multiple PCs behind this remote site, none of the additional users who are connecting to the corporate LAN will be requested for the Xauth credentials because the tunnel is already up.

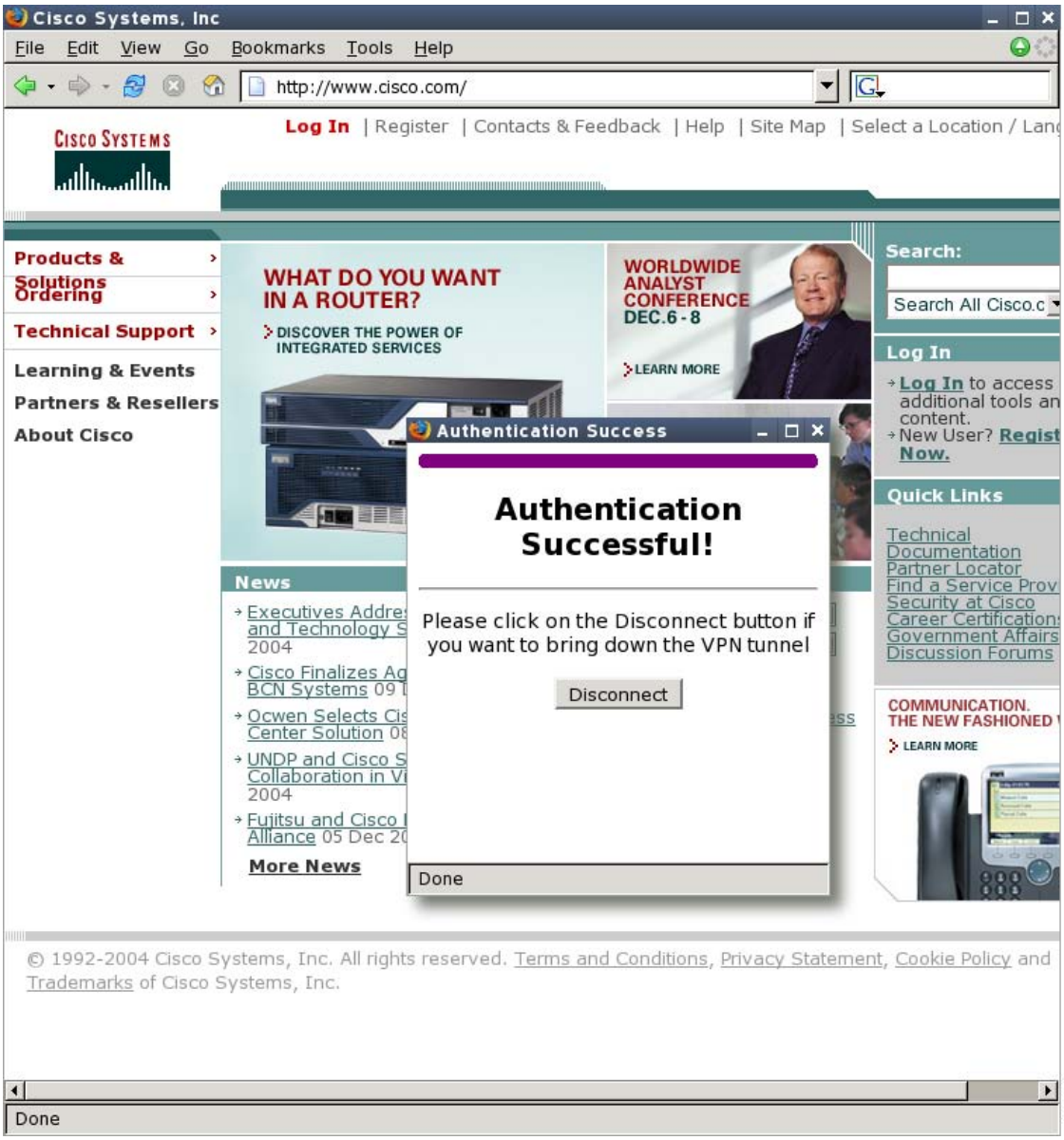
Figure 7 VPN Tunnel Authentication



Successful Authentication

Figure 8 is an example of a successful activation. If the user chooses to deactivate the VPN tunnel, he or she should click the Disconnect button. After the IKE security association (SA) times out (the default value is 24 hours), the remote teleworker has to enter the Xauth credentials to bring up the tunnel.

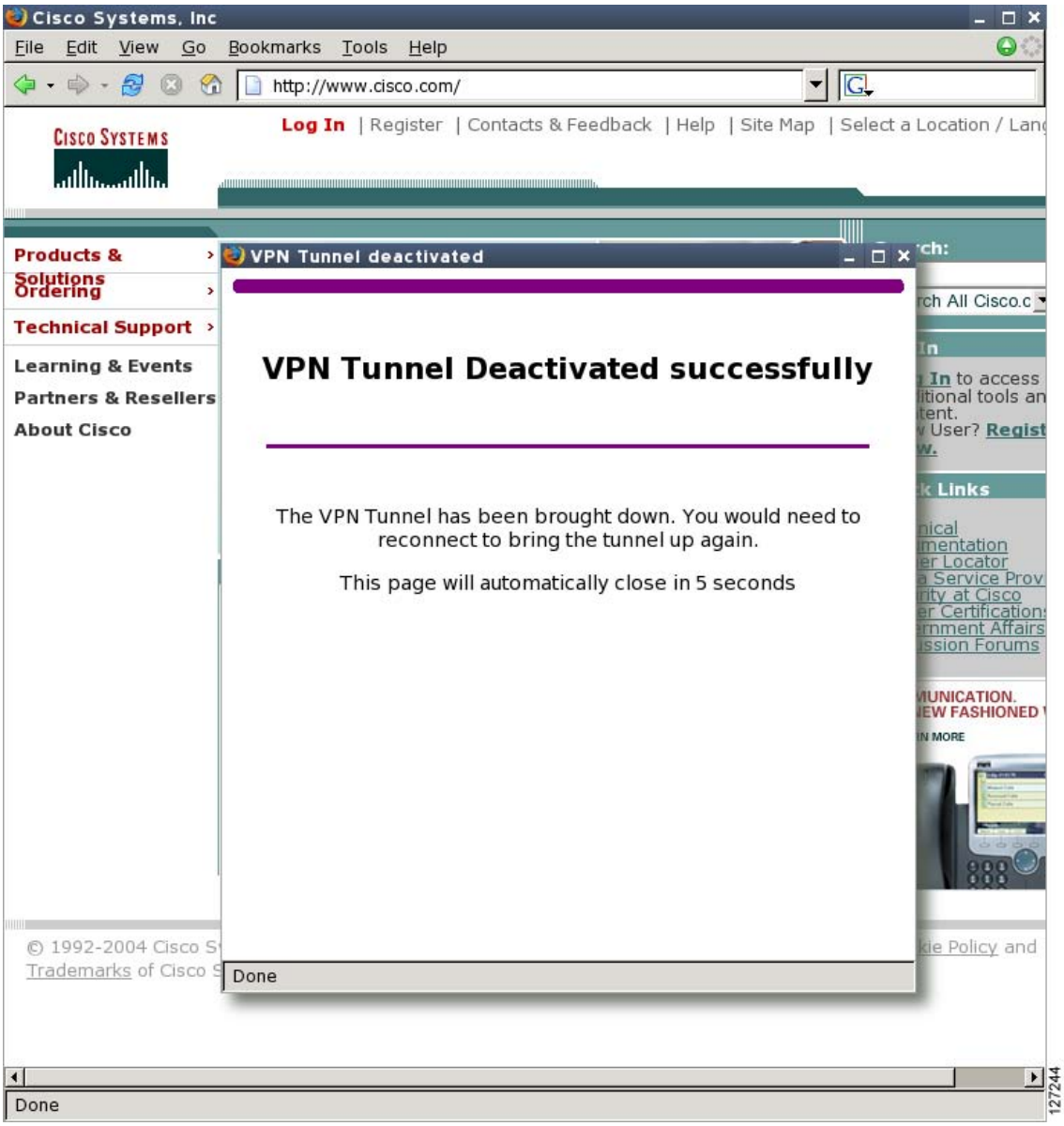
Figure 8 Successful Activation



Deactivation

Figure 9 is an example of a VPN tunnel that has been deactivated successfully. The page automatically closes in 5 seconds.

Figure 9 VPN Tunnel Deactivated Successfully



802.1x Authentication

The 802.1x Authentication feature allows you to combine Easy VPN client mode operation with 802.1x authentication on Cisco IOS routers. For more information about this feature, see "802.1 Authentication" in the section "Additional References."

Tunnel Activation Options

There are three tunnel activation options:

- Automatic activation
- Manual activation
- Traffic-triggered activation (not available in Cisco IOS Release 12.3(11)T)

Tunnel connect and disconnect options are available with SDM.

Automatic Activation

The Cisco Easy VPN tunnel is automatically connected when the Cisco Easy VPN Remote feature is configured on an interface. If the tunnel times out or fails, the tunnel automatically reconnects and retries indefinitely.

To specify automatic tunnel control on a Cisco Easy VPN remote device, you need to configure the **crypto ipsec client ezvpn** command and then the **connect auto** subcommand. However, you do not need to use these two commands when you are creating a new Easy VPN remote configuration because the default is "automatic."

To disconnect or reset a particular tunnel, you should use the **clear crypto ipsec client ezvpn** command, or you can use SDM.

Manual Activation

The Cisco Easy VPN Remote software implements manual control of the Cisco Easy VPN tunnels so that you can establish and terminate the tunnel on demand.

To specify manual tunnel control on a Cisco Easy VPN remote device, you need to input the **crypto ipsec client ezvpn** command and then the **connect manual** command.

The manual setting means that the Cisco Easy VPN remote will wait for a command before attempting to establish the Cisco Easy VPN Remote connection. When the tunnel times out or fails, subsequent connections will also have to wait for the command.

If the configuration is manual, the tunnel is connected only after you issue the command **crypto ipsec client ezvpn connect**.

To disconnect or reset a particular tunnel, you should use the **clear crypto ipsec client ezvpn** command, or you can use SDM.

See the "Configuring Manual Tunnel Control" section for specific information on how to configure manual control of a tunnel.

Traffic-Triggered Activation

Note This feature is not available in Cisco IOS Release 12.3(11)T.

The Traffic-Triggered Activation feature is recommended for transactional-based VPN applications. It is also recommended for use with the Easy VPN dial backup feature for the backup Easy VPN configuration so that backup is activated only when there is traffic to send across the tunnel.

To use Access Control List (ACL) tunnel control, you must first describe the traffic that is considered "interesting." For more information about ACLs, see the chapter "Access Control Lists: Overview and Guidelines" in the "Traffic Filtering and Firewalls" section of the *Cisco IOS Security Configuration Guide, Release 12.3*. To actually configure an ACL-triggered tunnel, use the **crypto ipsec client ezvpn** command with the **connect acl** subcommand.

Dead Peer Detection Stateless Failover Support

Two options are available for configuring Dead Peer Detection Stateless Failover Support:

- Backup Server List Local Configuration
- Backup Server List Auto Configuration

Backup Server List Local Configuration

Backup Server List Local Configuration allows users to enter multiple peer statements. With this feature configured, if the client is connecting to a peer and the negotiation fails, Easy VPN fails over to the next peer. This failover continues through the list of peers. When the last peer is reached, Easy VPN rolls over to the first peer. The IKE and IPsec SAs to the previous peer are deleted. Multiple peer statements work for both IP addresses as well as for hostnames. Setting or unsetting the peer statements will not affect the order of the peer statements.

To use this feature, use the **peer** subcommand of the **crypto ipsec client ezvpn** command.

Backup Server List Auto Configuration

Easy VPN remote that is based on Cisco IOS software can have up to 10 backup servers configured for redundancy. The Backup Server feature allows the Easy VPN server to "push" the backup server list to the Easy VPN remote.

The backup list allows the administrator to control the backup servers to which a specific Easy VPN remote will connect in case of failure, retransmissions, or dead peer detection (DPD) messages.

Note Before the backup server feature can work, the backup server list has to be configured on the server.

How a Backup Server Works

If remote A goes to server A and the connection fails, remote A goes to server B. If server B has a backup list configured, that list will override the backup server list of server A. If the connection to server B fails, remote A will continue through the backup servers that have been configured.

Note If you are in auto mode and you have a failure, you will transition automatically from server A to server B. However, if you are in manual mode, you have to configure the transition manually. To configure the transition manually, use the **crypto ipsec client ezvpn** command with the **connect** keyword.

No new configuration is required at the Easy VPN remote to enable this feature. If you want to display the current server, you can use the **show crypto ipsec client ezvpn** command. If you want to find out which peers were pushed by the Easy VPN server, you can use the same command.

To troubleshoot this feature, use the **debug crypto ipsec client ezvpn** command. If more information is needed for troubleshooting purposes, use the **debug crypto isakmp** command. The **show crypto ipsec client ezvpn** command may also be used for troubleshooting.

Cisco Easy VPN Remote Features

The Cisco Easy VPN Remote feature is a collection of features that improves the capabilities of the Cisco Easy VPN Remote feature introduced in Cisco IOS Release 12.2(4)YA. The Cisco Easy VPN Remote feature includes the following:

- Default Inside Interface, page 20—This feature supports the autoconfiguration of the default Easy VPN inside interface for Cisco 800 series routers.
- Multiple Inside Interfaces, page 21—This feature allows you to configure up to eight inside interfaces on the Cisco Easy VPN remote.
- Multiple Outside Interfaces, page 21—This feature allows you to configure up to four outside tunnels for outside interfaces.
- VLAN Support, page 21—This feature allows VLANs to be configured as valid Easy VPN inside interfaces.
- Multiple Subnet Support, page 22—This feature allows multiple subnets from the Easy VPN inside interface to be included in the Easy VPN tunnel.
- NAT Interoperability Support, page 22—This feature automatically restores the NAT configuration when the IPsec VPN tunnel is disconnected.
- Local Address Support, page 22—The Cisco Easy VPN Remote feature is enhanced to support an additional **local-address** attribute that specifies which interface is used to determine the IP address used to source the Easy VPN tunnel traffic.
- Peer Hostname, page 23—When a peer is defined as a hostname, the hostname is stored and the Domain Name System (DNS) lookup is done at the time of tunnel connection.
- Proxy DNS Server Support, page 23—This feature allows you to configure the router in a Cisco Easy VPN remote configuration to act as a proxy DNS server for LAN-connected users.
- Cisco IOS Firewall Support, page 23—This feature supports Cisco IOS Firewall configurations on all platforms.
- Easy VPN Remote and Server on the Same Interface, page 23—The Easy VPN remote and Easy VPN server are supported on the same interface, which makes it possible to establish a tunnel to another Easy VPN server and terminate the Easy VPN software client on the same interface simultaneously.
- Easy VPN Remote and Site to Site on the Same Interface, page 23—The Easy VPN Remote and site to site (crypto map) are supported on the same interface, which makes it possible to establish a tunnel to another Easy VPN server and have another site to site on the same interface simultaneously.
- Cisco Easy VPN Remote Web Managers, page 24—Users can manage the Cisco Easy VPN Remote feature on the Cisco uBR905 and Cisco uBR925 cable access routers using a built-in web interface.
- Dead Peer Detection Periodic Message Option, page 24—This feature allows you to configure your router to query the liveliness of its IKE peer at regular intervals.
- Load Balancing, page 24—If a remote device is loaded and unable to accept more traffic, the VPN 3000 will send a notify message that contains an IP address that represents the new IKE server to which the remote should connect.

- Management Enhancements, page 25—This feature allows for remote management of the VPN remote.
- PFS Support, page 25—The PFS configuration mode attribute is sent by the server if requested by the VPN remote device.
- Dial Backup, page 25—This feature allows you to configure a dial backup tunnel connection on your remote device.
- Virtual IPsec Interface Support, page 27 —This feature allows you to selectively send traffic to different Easy VPN concentrators as well as to the Internet (includes a reference to the IPsec Virtual Tunnel Interface feature.)
- Dual Tunnel Support, page 29—This feature allows you to configure multiple Easy VPN tunnels that share common inside and outside interfaces to connect two peers to two different VPN servers simultaneously.
- Banner, page 32—The EasyVPN remote device can download a banner that has been pushed by the Easy VPN server. The banner can be used for Xauth and web-based activation. The banner is displayed when the Easy VPN tunnel is “up” on the Easy VPN remote console or as an HTML page in the case of web-based activation.
- Configuration Management Enhancements (Pushing a Configuration URL Through a Mode-Configuration Exchange), page 33—The Easy VPN remote device can download a URL that is pushed by the Easy VPN server, allowing the Easy VPN remote device to download configuration content and apply it to the running configuration.
- Reactivate Primary Peer, page 33—This feature allows you to designate a primary peer. When an Easy VPN device fails over from the primary peer to a backup peer and the primary peer is again available, connections with the backup peer are torn down and a connection is made with the primary peer.
- Identical Addressing Support, page 33—This feature integrates Network Address Translation (NAT) with Easy VPN to allow remotes with overlapping internal IP addressing to connect to the Easy VPN server.
- cTCP Support on Easy VPN Clients, page 34–When cTCP is enabled on a remote device (client) and headend device, IKE and ESP (Protocol 50) traffic is encapsulated in the TCP header so that the firewalls in between the client and the headend device permit this traffic (considering it the same as TCP traffic).

Default Inside Interface

Easy VPN Remote supports the autoconfiguration of the default Easy VPN inside interface for Cisco 800 series routers. The interface Ethernet 0 is the default inside interface.

If you want to disable the default inside interface and configure another inside interface on the Cisco 800 series router, you must configure the other inside interface first and then disable the default inside interface. You can use the following command to disable the default inside interface:

no crypto ipsec client ezvpn *name* **inside**

If you did not configure the other inside interface first before disabling the default inside interface, you will receive a message such as the following (see lines three and four):

```
Router (config)# interface ethernet0
Router (config-if)# no crypto ipsec client ezvpn hw-client inside
Cannot remove the single inside interface unless
one other inside interface is configured
```

Multiple Inside Interfaces

Inside interface support is enhanced in the Cisco Easy VPN Remote feature to support multiple inside interfaces for all platforms. Inside interfaces can be configured manually with the enhanced command and subcommand:

```
interface interface-name
    crypto ipsec client ezvpn name [outside | inside]
```

See the "Configuring Multiple Inside Interfaces" section for information on how to configure more than one inside interface.

Multiple inside interfaces offer the following capabilities:

- Up to eight inside interfaces are supported on the Cisco 800 and Cisco 1700 series routers.
- At least one inside interface must be configured for each outside interface; otherwise, the Cisco Easy VPN Remote feature does not establish a connection.
- Adding a new inside interface or removing an existing inside interface automatically resets the Cisco Easy VPN Remote connection (the currently established tunnel). You must reconnect a manually configured tunnel, and if Xauth is required by the Cisco Easy VPN server, the user is reprompted. If you have set the Cisco Easy VPN Remote configuration to connect automatically and no Xauth is required, no user input is required.
- Inside interfaces that are configured or the default setting can be shown by using the **show crypto ipsec client ezvpn** command.

Multiple Outside Interfaces

The Easy VPN Remote feature supports one Easy VPN tunnel per outside interface. You can configure up to four Easy VPN tunnels per Cisco router. Each Easy VPN tunnel can have multiple inside interfaces configured, but they cannot overlap with another Easy VPN tunnel unless dial backup is configured. For more information about dial backup, see the section "Dial Backup." To configure multiple outside interfaces, use the **crypto ipsec client ezvpn** command and **outside** keyword.

To disconnect or clear a specific tunnel, the **clear crypto ipsec client ezvpn** command specifies the IPsec VPN tunnel name. If there is no tunnel name specified, all existing tunnels are cleared.

See the "Configuring Multiple Outside Interfaces" section for more information on configuring more than one outside interface.

VLAN Support

Inside interface support on VLANs makes it possible to have valid Easy VPN inside interface support on a VLAN, which was not possible before Cisco IOS Release 12.3(7)XR. With this feature, SAs can be established at connection using the VLAN subnet address or mask as a source proxy.

For the inside interface support on VLANs to work, you must define each VLAN as an Easy VPN inside interface. In addition, IPsec SAs should be established for each inside interface in the same manner as for other inside interfaces. For more information about inside and outside interfaces, see the sections "Multiple Inside Interfaces" and "Multiple Outside Interfaces."

Inside interface support on VLANs is supported only on Cisco routers that support VLANs.

Multiple Subnet Support

For situations in which you have multiple subnets connected to an Easy VPN inside interface, you can optionally include these subnets in the Easy VPN tunnel. First, you must specify the subnets that should be included by defining them in an ACL. To configure an ACL, see "Access control lists, configuring" in the "Additional References" section. Next, you have to use the **acl** subcommand of the **crypto ipsec client ezvpn** (global) command to link your ACL to the Easy VPN configuration. Easy VPN Remote will automatically create the IPsec SAs for each subnet that is defined in the ACL as well as for the subnets that are defined on the Easy VPN inside interface.

Note Multiple subnet support is not supported in client mode.

NAT Interoperability Support

Cisco Easy VPN Remote supports interoperability with NAT. You can have a NAT configuration and a Cisco Easy VPN Remote configuration that coexist. When an IPsec VPN tunnel is down, the NAT configuration works.

In the Cisco Easy VPN Remote feature, the router automatically restores the previous NAT configuration when the IPsec VPN tunnel is torn down. The user-defined access lists are not disturbed. Users can continue to access nontunnel areas of the Internet when the tunnel times out or disconnects.

Note NAT interoperability is not supported in client mode with split tunneling.

Local Address Support

The Cisco Easy VPN Remote feature is enhanced to support an additional **local-address** attribute. This attribute specifies which interface is used to determine the IP address that is used to source the Easy VPN Remote tunnel traffic. After specifying the interface with the **local-address** subcommand, you can manually assign a static IP address to the interface or use the **cable-modem dhcp-proxy interface** command to automatically configure the specified interface with a public IP address. See the "Configuring Proxy DNS Server Support" section for configuration information.

Local Address Support is available for all platforms, but it is more applicable to the Cisco uBR905 and Cisco uBR925 cable access routers in conjunction with the **cable-modem dhcp-proxy interface** command. Typically, the loopback interface is the interface used to source tunnel traffic for the Cisco uBR905 and Cisco uBR925 cable access routers.

In a typical DOCSIS network, the Cisco uBR905 and Cisco uBR925 cable access routers are normally configured with a private IP address on the cable modem interface. In the initial Cisco Easy VPN Remote feature, a public IP address was required on the cable modem interface to support the Easy VPN remote.

In the Cisco Easy VPN Remote feature, cable providers can use the Cable DHCP Proxy feature to obtain a public IP address and assign it to the cable modem interface, which is usually the loopback interface.

For more information on the **cable-modem dhcp-proxy interface** command, see the "Cable CPE Commands" chapter in the *Cisco Broadband Cable Command Reference Guide*.

Note The **cable-modem dhcp-proxy interface** command is supported only for the Cisco uBR905 and Cisco uBR925 cable access routers.

Peer Hostname

The peer in a Cisco Easy VPN Remote configuration can be defined as an IP address or a hostname. Typically, when a peer is defined as a hostname, a DNS lookup is done immediately to get an IP address. In the Cisco Easy VPN Remote feature, the peer hostname operation is enhanced to support DNS entry changes. The text string of the hostname is stored so that the DNS lookup is done at the time of the tunnel connection, not when the peer is defined as a hostname.

See the "Configuring and Assigning the Easy VPN Remote Configuration" section for information on enabling the peer hostname functionality.

Proxy DNS Server Support

When the Easy VPN tunnel is down, the DNS addresses of the ISP or cable provider should be used to resolve DNS requests. When the WAN connection is up, the DNS addresses of the enterprise should be used.

As a way of implementing use of the DNS addresses of the cable provider when the WAN connection is down, the router in a Cisco Easy VPN Remote configuration can be configured to act as a proxy DNS server. The router, acting as a proxy DNS server for LAN-connected users, receives DNS queries from local users on behalf of the real DNS server. The DHCP server then can send out the LAN address of the router as the IP address of the DNS server. After the WAN connection comes up, the router forwards the DNS queries to the real DNS server and caches the DNS query records.

See the "Configuring Proxy DNS Server Support" section for information on enabling the proxy DNS server functionality.

Cisco IOS Firewall Support

The Cisco Easy VPN Remote feature works in conjunction with Cisco IOS Firewall configurations on all platforms.

Easy VPN Remote and Server on the Same Interface

This feature allows the Easy VPN remote and Easy VPN server to be supported on the same interface, making it possible to both establish a tunnel to another Easy VPN server and terminate the Easy VPN software client on the same interface simultaneously. A typical application would be a geographically remote location for which Easy VPN Remote is being used to connect to a corporate Easy VPN server and also to terminate local software client users.

For more information about the Easy VPN Remote and Server on the Same Interface feature, see "Easy VPN Remote and Server on the Same Interface" in the section "Additional References."

Easy VPN Remote and Site to Site on the Same Interface

This feature allows the Easy VPN remote and site to site (crypto map) to be supported on the same interface, making it possible to both establish a tunnel to another Easy VPN server and have another site to site on the same interface simultaneously. A typical application would be a third-party VPN service provider that is managing a remote router via the site-to-site tunnel and using Easy VPN Remote to connect the remote site to a corporate Easy VPN server.

For more information about the Easy VPN Remote and Site to Site on the Same Interface feature, see "Easy VPN Remote and Site to Site on the Same Interface" in the section "Additional References."

Cisco Easy VPN Remote Web Managers

Web interface managers may be used to manage the Cisco Easy VPN Remote feature. One such web interface manager is SDM, which is supported on the Cisco 830 series, Cisco 1700 series, Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers. SDM enables you to connect or disconnect the tunnel and provides a web interface for Xauth. For more information about SDM, see *Cisco Security Device Manager*.

A second web interface manager is the Cisco Router Web Setup (CRWS) tool, which is supported on the Cisco 806 router. The CRWS provides a similar web interface as SDM.

A third web interface manager, Cisco Easy VPN Remote Web Manager, is used to manage the Cisco Easy VPN Remote feature for Cisco uBR905 and Cisco uBR925 cable access routers. You do not need access to the CLI to manage the Cisco Easy VPN remote connection.

The web interface managers allow you to do the following:

- See the current status of the Cisco Easy VPN remote tunnel.
- Connect a tunnel that is configured for manual control.
- Disconnect a tunnel that is configured for manual control or reset a tunnel configured for automatic connection.
- Be prompted for Xauth information, if needed.

See the "Troubleshooting the VPN Connection" section for more information about Cisco Easy VPN Remote Web Manager.

Dead Peer Detection Periodic Message Option

The dead peer detection periodic message option allows you to configure your router to query the liveliness of its IKE peer at regular intervals. The benefit of this approach over the default approach (on-demand dead peer detection) is earlier detection of dead peers. For more information about the dead peer detection periodic message option, see *"Dead peer detection" in the section "Additional References."*

Load Balancing

When the Cisco VPN 3000 concentrator is configured for load balancing, the VPN 3000 will accept an incoming IKE request from the VPN remote on its virtual IP address. If the device is loaded and unable to accept more traffic, the VPN 3000 will send a notify message that contains an IP address that represents the new IKE server to which the remote should connect. The old connection will be torn down and a new connection established to the redirected VPN gateway.

There is no configuration required for load balancing to occur. If the VPN gateway is configured for load balancing, and it notifies the VPN remote that it is performing load balancing, the VPN remote has access to the load balancing feature.

To verify whether load balancing is occurring, use the **debug crypto isakmp**, **debug crypto ipsec client ezvpn**, and **show crypto ipsec** commands. To troubleshoot the load balancing process, use the **show crypto ipsec** command.

Management Enhancements

Management enhancements for Easy VPN remotes allow for the remote management of the VPN remote. The feature provides for the IPv4 address to be pushed by configuration mode to the VPN remote. The IPv4 address is assigned to the first available loopback interface on the VPN remote, and any existing statically defined loopbacks are not overridden. On disconnect, the address and loopback interface are removed from the list of active interfaces.

After the VPN remote is connected, the loopback interface should be accessible from the remote end of the tunnel. All PAT activities will be translated through this interface IP address.

If a loopback exists, and an IP address is associated with it and its state is unassigned, the interface is a good candidate for mode configuration address management.

Note After you assign an address to the loopback interface, if you save the configuration to NVRAM and reboot the VPN remote, the configuration address is permanently contained in the configuration. If you saved the configuration to NVRAM and rebooted the VPN remote, you must enter configuration mode and remove the IP address from the loopback interface manually.

You can use the **show ip interface** command with the **brief** keyword to verify that a loopback has been removed. The output of this **show** command also displays the interface.

PFS Support

The PFS configuration mode attribute is sent by the server if requested by the VPN remote device. If any subsequent connection by the remote device shows that PFS is not received by the remote, PFS will not be sent in IPsec proposal suites.

Note The PFS group that will be proposed in the IPsec proposal suites is the same as the group used for IKE.

You can use the **show crypto ipsec client ezvpn** command to display the PFS group and to verify that you are using PFS.

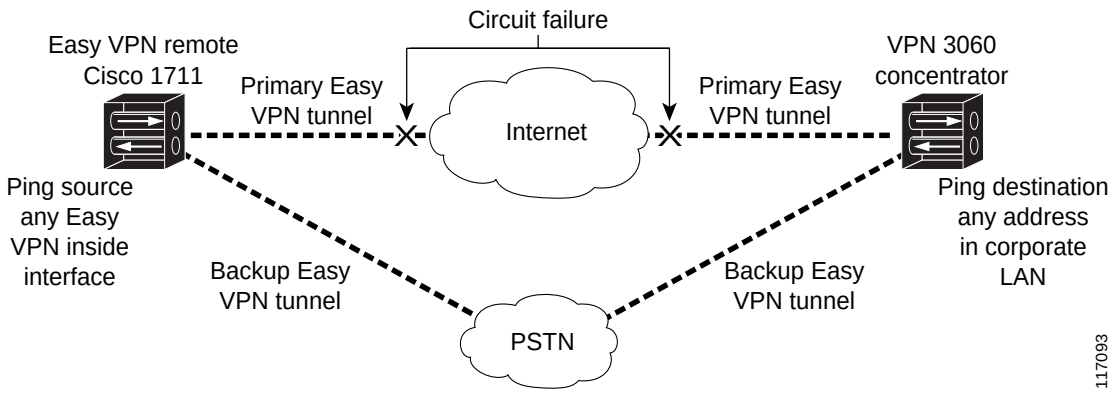
Dial Backup

Note The Dial Backup feature is not available in Cisco IOS Release 12.3(11)T.

Dial backup for Easy VPN remotes allows you to configure a dial backup tunnel connection on your remote device. The backup feature is “brought up” only when real data has to be sent, eliminating the need for expensive dialup or ISDN links that must be created and maintained even when there is no traffic.

Figure 10 illustrates a typical Easy VPN remote-with-dial-backup scenario. In this scenario, a Cisco 1751 remote device is attempting to connect to another Cisco 1751 (acting as a server). There is a failure in the primary Easy VPN tunnel, and the connection is rerouted through the Easy VPN backup tunnel to the Cisco 1751 server.

Figure 10 Dial Backup for Easy VPN Scenario



Dial Backup Using a Dial-on-Demand Solution

IP static route tracking enable Cisco IOS software to identify when a Point-to-Point Protocol over Ethernet (PPPoE) or IPsec VPN tunnel “goes down” and initiates a Dial-on-Demand (DDR) connection to a preconfigured destination from any alternative WAN or LAN port (for example, a T1, ISDN, analog, or auxiliary port). The failure may be caused by several catastrophic events (for example, by Internet circuit failures or peer device failure). The remote route has only a static route to the corporate network. The IP static-route-tracking feature allows an object to be tracked (using an IP address or hostname) using Internet Control Message Protocol (ICMP), TCP, or other protocols, and it installs or removes the static route on the basis of the state of the tracked object. If the tracking feature determines that Internet connectivity is lost, the default route for the primary interface is removed, and the floating static route for the backup interface is enabled.

Dial Backup Using Object Tracking

IP static route tracking must be configured for dial backup on an Easy VPN remote device to work. The object tracking configuration is independent of the Easy VPN remote dial backup configuration. (For more information about object tracking, see the feature guide *Reliable Static Routing Backup Using Object Tracking*.)

Easy VPN Remote Dial Backup Support Configuration

You can configure dial backup for your Easy VPN remote using two Easy VPN remote options that allow a connection to the backup Easy VPN configuration and a connection to the tracking system.

- To specify the Easy VPN configuration that will be activated when backup is triggered, use the **backup** subcommand of the **crypto ipsec client ezvpn** (global) command.
- The Easy VPN remote device registers to the tracking system to get the notifications for change in the state of the object. Use the **track** subcommand to inform the tracking process that the Easy VPN remote device is interested in tracking an object, which is identified by the object number. The tracking process, in turn, informs the Easy VPN remote device when the state of this object changes. This notification prompts the Easy VPN remote device when the state of this object changes. This notification prompts the Easy VPN remote device to bring up the backup connection when the tracked object state is DOWN. When the tracked object is UP again, the backup connection is torn down and the Easy VPN remote device will switch back to using the primary connection.

Note Only one backup configuration is supported for each primary Easy VPN configuration. Each inside interface must specify the primary and backup Easy VPN configuration.

Dynamically Addressed Environments

To allow dial backup to be deployed in dynamically addressed environments, use the IP SLA Pre-Routed ICMP Echo Probe feature. (For more information about this feature, see *Cisco 1700 Series- Cisco IOS Release 12.3(7)XR* release notes. To use the IP SLA Pre-Routed ICMP Echo Probe feature, use the **icmp-echo** command with the **source-interface** keyword.

Dial Backup Examples

For examples of dial backup configurations, see the section "Dial Backup: Examples."

Virtual IPsec Interface Support

The Virtual IPsec Interface Support feature provides a routable interface to selectively send traffic to different Easy VPN concentrators as well as to the Internet.

Before Cisco IOS Release 12.4(4)T, at the tunnel-up/tunnel-down transition, attributes that were pushed during the mode configuration had to be parsed and applied. When such attributes resulted in the configurations being applied on the interface, the existing configuration had to be overridden. With the Virtual IPsec Interface Support feature, the tunnel-up configuration can be applied to separate interfaces, making it easier to support separate features at tunnel-up time. Features that are applied to the traffic going into the tunnel can be separate from the features that are applied to traffic that is not going through the tunnel (for example, split-tunnel traffic and traffic leaving the device when the tunnel is not up). When the Easy VPN negotiation is successful, the line protocol state of the virtual-access interface gets changed to up. When the Easy VPN tunnel goes down because the security association (SA) expires or is deleted, the line protocol state of the virtual-access interfaces changes to down.

Routes act as traffic selectors in an Easy VPN virtual interface, that is, the routes replace the access list on the crypto map. In a virtual-interface configuration, Easy VPN negotiates a single IPsec SA if the Easy VPN server has been configured with a dynamic virtual IPsec interface. This single SA is created irrespective of the Easy VPN mode that is configured.

After the SA is established, routes that point to the virtual-access interface are added to direct traffic to the corporate network. Easy VPN also adds a route to the VPN concentrator so that IPsec-encapsulated packets get routed to the corporate network. A default route that points to the virtual-access interface is added in the case of a nonsplit mode. When the Easy VPN server "pushes" the split tunnel, the split tunnel subnet becomes the destination to which the routes that point to the virtual access are added. In either case, if the peer (VPN concentrator) is not directly connected, Easy VPN adds a route to the peer.

Note

- Most routers that run the Cisco Easy VPN Client software have a default route configured. The default route that is configured should have a metric value greater than 1. The metric value must be greater than 1 because Easy VPN adds a default route that has a metric value of 1. The route points to the virtual-access interface so that all traffic is directed to the corporate network when the concentrator does not "push" the split tunnel attribute.

For more information about the IPsec Virtual Tunnel Interface feature, see the document *IPSec Virtual Tunnel Interface* (URL link provided in the *"Related Documents"* section of this document [*General Information on IPsec and VPN*]).

Table 1 presents the different methods of configuring a remote device and the corresponding headend IPsec aggregator configurations. Each row represents a way to configure a remote device. The third column shows the different headend configurations that can be used with IPsec interfaces. See Table 2 for a description of terms that are used in Table 1 and Table 3.

Table 1 How Different Remote Device Configurations Interact with Various Headends and Configurations

Remote Device Configurations	IOS Headend – Using Crypto Maps	IOS Headend – Using IPsec Interfaces	VPN3000/ASA
Crypto maps	• Supported.	—	—
Easy VPN virtual interface	• Supported. • Will create multiple SAs for a split tunnel. • Because there is no interface on the headend, interface features cannot be supported. • Limited quality of service (QoS) is supported.	• Supported. • Creates only a single SA in split and no-split tunnels. • Route injection is accomplished on the server. • Routes are injected on the remote devices to direct traffic to the interface.	• Supported. • Will create multiple SAs for a split tunnel.
Legacy Easy VPN	• Creates a single IPsec SA on the headend when a default policy is pushed. • Creates multiple SAs when a split-tunnel policy is pushed to the remote device.	• Not supported. • Cannot be used with split tunnels because the headend interface does not support multiple SAs on a single interface.	• Supported. • Creates multiple SAs for split tunnels.
Static virtual interface	• Not supported.	• Supported. • Can be used with a static interface or dynamic interface on the headend. • Routing support is mandatory to reach the network.	• Not supported.

Table 2 provides a description of the terms used in Table 1 and Table 3.

Table 2 Terms Used in Table 1 and Table 3

Terms	Description
ASA	Cisco Adaptive Security Appliance, a threat-management security appliance.
Crypto maps	Commonly used for configuring IPsec tunnels. The crypto map is attached to an interface. For more information on crypto maps, see the section "Creating Crypto Map Sets" of the "Configuring Security for VPNs with IPSec" chapter of the *Cisco IOS Security Configuration Guid*e. (URL link provided in the "Related Documents" section of this document.)
Easy VPN dual tunnel remote device	Two Easy VPN remote device configurations in which both are using a dynamic IPsec virtual tunnel interface.
Easy VPN virtual interface remote device (Easy VPN virtual interface)	Easy VPN remote configuration that configures the usage of a dynamic IPsec virtual tunnel interface.
IPsec interface	Consists of static and dynamic IPsec virtual interfaces.
IPsec Virtual Tunnel Interface	Tunnel interface that is created from a virtual template tunnel interface using mode IPsec. For more information on virtual tunnel interface configurations, see the document *IPSec Virtual Tunnel Interface (*URL link provided in the *"Related Documents"* section of this document [*General Information on IPsec and VPN*]).
Legacy Easy VPN	Easy VPN remote device configuration that uses crypto maps and does not use IPsec interfaces.
Static IPsec virtual tunnel interface (static virtual tunnel interface)	Tunnel interface used with mode IPsec that proposes and accepts only an "ipv4 any any" selector. For more information on static virtual tunnel interface configurations, see the document *IPSec Virtual Tunnel Interface* (URL link provided in the *"Related Documents"* section of this document [*General Information on IPsec and VPN*]).
VPN 3000	Cisco VPN 3000 series routers.

Dual Tunnel Support

Easy VPN now supports the ability to configure two easy VPN tunnels that have the same inside and outside interfaces. The feature is called the Easy VPN Dual Tunnel. Configuring multiple tunnels on a single remote device can be accomplished in a number of ways, which are listed below in Table 3 along with their configuration and usage considerations. Further discussion in this section refers to only one such method of configuring dual tunnels using Easy VPN tunnels that have virtual interfaces. This method will be referred to as Dual Tunnel Support.

In a dual-tunnel Easy VPN setup, each Easy VPN tunnel is configured using virtual IPsec interface support, as shown in the section "Virtual IPsec Interface Support." Each Easy VPN tunnel has its unique virtual interface, which is created when the Easy VPN configuration is complete.

There are two possible combinations in which the dual tunnels can be used.

- Dual Easy VPN tunnels that have one tunnel using a nonsplit tunnel policy and the other tunnel using a split tunnel policy that has been pushed from the respective headend.

- Dual Easy VPN tunnel in which both tunnels are using an independent split tunnel policy that has been pushed from the respective headend.

Note It is not permitted to have dual Easy VPN tunnels in which both tunnels are using a nonsplit tunnel policy.

The Easy VPN dual tunnel makes use of route injections to direct the appropriate traffic through the correct Easy VPN virtual tunnel interface. When the Easy VPN tunnel on the remote device "comes up," it "learns" the split or nonsplit policy from the headend. The Easy VPN remote device injects routes in its routing table that correspond to the nonsplit networks that have been learned. If the headend pushes a nonsplit tunnel policy to the Easy VPN remote device, the Easy VPN remote device installs a default route in its routing table that directs all traffic out of the Easy VPN virtual interface that corresponds to this Easy VPN tunnel. If the headend pushes split-tunnel networks to the remote device, the remote device installs specific routes to the split networks in its routing table, directing the traffic to these networks out of the virtual tunnel interface.

Note Dual Tunnel Easy VPN uses destination-based routing to send traffic to the respective tunnels.

Output features can be applied to this virtual interface. Examples of such output features are Cisco IOS Quality of Service and Cisco IOS Firewall. These features must be configured on the virtual template that is configured in the Easy VPN client configuration.

Table 3 explains how this feature should be used. See Table 2 for a description of terms that are used in Table 1 and Table 3.

Table 3 Dual Tunnel Usage Guidelines

Dual Tunnel Combinations	Headends Supported	Configuration and Usage Considerations on the Easy VPN Remote Device and Headend
Two legacy Easy VPN tunnels	IOS, ASA, and VPN 3000	• Two tunnels cannot share a common outside interface. • Two tunnels cannot share a common inside interface. • The two tunnels should use separate inside and outside interfaces. • Traffic from an inside interface that belongs to one Easy VPN tunnel cannot be pushed into another tunnel.
One legacy Easy VPN tunnel and one crypto map	IOS, ASA, and VPN 3000	The crypto map can share the same outside interface as the legacy Easy VPN client configuration. However, the behavior of the two remote devices depends on the mode of Easy VPN as well as the IPsec selectors of the crypto map and the Easy VPN remote device. This is not a recommended combination.
One legacy Easy VPN tunnel and one static virtual interface	IOS	Both tunnels cannot terminate on the same headend. The static virtual interface remote device tunnel has to be terminated on a static virtual interface on the headend router. The legacy Easy VPN remote device tunnel can terminate on the virtual tunnel interface or crypto map that is configured on the headend.

Table 3 Dual Tunnel Usage Guidelines (continued)

Dual Tunnel Combinations	Headends Supported	Configuration and Usage Considerations on the Easy VPN Remote Device and Headend
One legacy Easy VPN tunnel and one Easy VPN virtual interface	IOS, ASA, and VPN 3000	• Both tunnels cannot terminate on the same headend. • The legacy Easy VPN tunnel and the Easy VPN virtual interface can share a common inside and outside interface. • An Easy VPN virtual interface should be used only with split tunneling. • Legacy Easy VPN can use a split tunnel or no split tunnel. • The Web-Based Activation feature cannot be applied on both Easy VPN tunnels. • Using two Easy VPN virtual interfaces is preferable to using this combination.
One Easy VPN virtual interface and one static virtual interface	IOS	• Both tunnels cannot terminate on the same peer. The static virtual interface and the Easy VPN virtual interface can use the same outside interface. • The Easy VPN virtual interface should use split tunneling.
Two Easy VPN virtual interfaces	IOS, ASA, and VPN 3000	• Both tunnels cannot terminate on the same peer. • At least one of the tunnels should use split tunneling. • Web-Based Activation cannot be applied to both Easy VPN tunnels.

Banner

The Easy VPN server pushes a banner to the Easy VPN remote device. The Easy VPN remote device can use the banner during Xauth and web-based activation. The Easy VPN remote device displays the banner the first time that the Easy VPN tunnel is brought up.

The banner is configured under group configuration on the Easy VPN server.

Configuration Management Enhancements (Pushing a Configuration URL Through a Mode-Configuration Exchange)

After this feature has been configured on the server using the commands **configuration url** and **configuration version** (subcommands under the **crypto isakmp client configuration group** command), the server can "push" the configuration URL and configuration version number to the Easy VPN remote device. With this information, the Easy VPN remote device can download the configuration content and apply it to its running configuration. For more information about this feature, see the section "Configuration Management Enhancements" in the *Easy VPN Server* feature module.

Reactivate Primary Peer

The Reactivate Primary Peer feature allows a default primary peer to be defined. The default primary peer (a server) is one that is considered better than other peers for reasons such as lower cost, shorter distance, or more bandwidth. With this feature configured, if Easy VPN fails over during Phase 1 SA negotiations from the primary peer to the next peer in its backup list, and if the primary peer is again available, the connections with the backup peer are torn down and the connection is again made with the primary peer.

Dead Peer Detection is one of the mechanisms that acts as a trigger for primary peer reactivation. Idle timers that are configured under Easy VPN is another triggering mechanism. When configured, the idle timer detects inactivity on the tunnel and tears it down. A subsequent connect (which is immediate in auto mode) is attempted with the primary preferred peer rather than with the peer last used.

Note Only one primary peer can be defined.

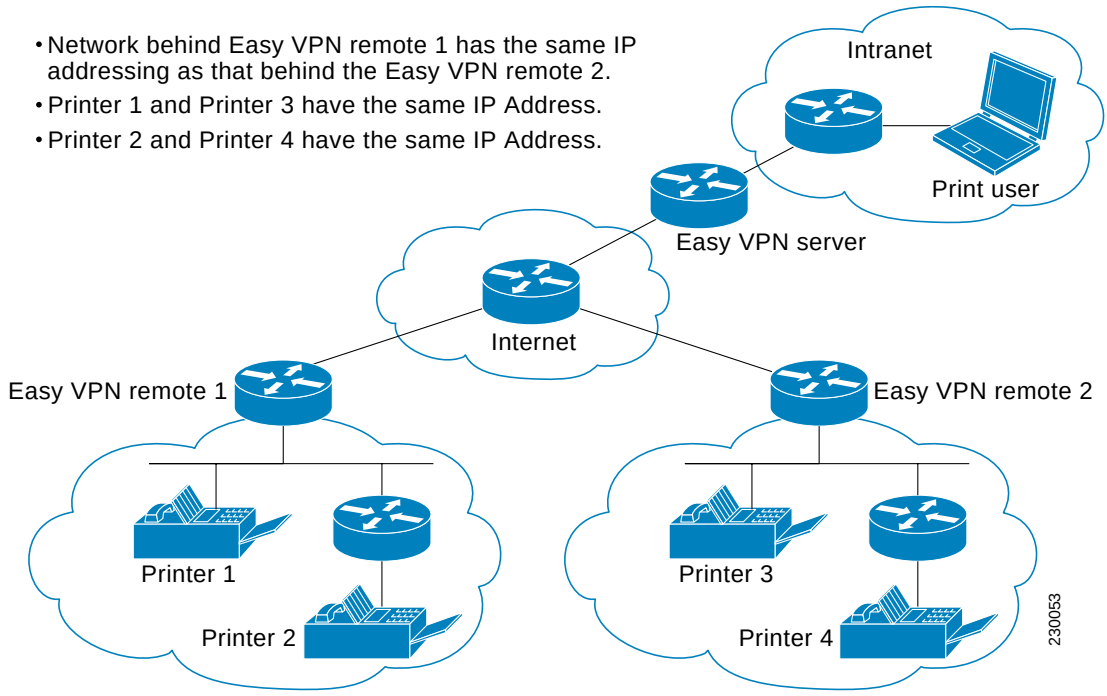
Identical Addressing Support

The Identical Addressing Support feature supports identically addressed LANs on Easy VPN remotes. Network resources, such as printers and web servers on the LAN side of the EasyVPN remotes, that have overlapping addressing with other Easy VPN remotes are now reachable. The Easy VPN Remote feature was enhanced to work with NAT to provide this functionality.

- The Easy VPN server requires no changes to support the Identical Addressing Support feature.
- The Identical Addressing Support feature is supported only in network extension modes (network-extension and network-plus).
- Virtual tunnel interfaces must be configured on the Easy VPN remote before using the Identical Addressing Support feature.

Figure 11 shows an example of the Identical Addressing Support feature configuration.

Figure 11 Identical Addressing Support



The Identical Addressing Support feature can be configured with the following command and enhanced subcommands:

crypto ipsec client ezvpn *<name>*

Enhanced subcommands

- **nat acl** {*acl-name* | *acl-number*}—Enables split tunneling for the traffic specified by the ACL name or the ACL number.
  - The *acl-name* argument is the name of the ACL.
  - The *acl-number* argument is the number of the ACL.
- **nat allow**—Allows NAT to be integrated with Cisco Easy VPN.

For detailed steps on how to configure Identical Addressing Support, see "Configuring Identical Addressing Support."

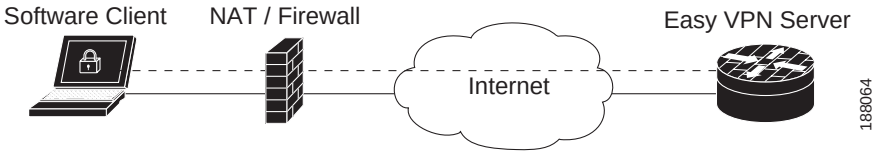
cTCP Support on Easy VPN Clients

The Cisco Tunneling Control Protocol (cTCP) feature can be used for situations in which an Easy VPN client (remote device) is operating in an environment in which standard IPsec does not function or in which it does not function transparently without modification to existing firewall rules. These situations include the following:

- Small office or home office router performing Network Address Translation (NAT) or Port Address Translation (PAT)
- PAT-provided IP address behind a larger router (for example, in a corporation)
- Non-NAT firewall (packet filtering or stateful)
- Proxy server

Figure 12 illustrates how IPsec traffic that is tunneled inside the cTCP traverses Network Address Translation (NAT) and the firewall (see the dashed line).

Figure 12 cTCP on an Easy VPN Remote Device



For detailed steps on how to configure cTCP on Easy VPN remote devices, see the section "Configuring cTCP on an Easy VPN Client."

For more information about cTCP support on Easy VPN remote devices, including configuration and troubleshooting examples, see "cTCP on Cisco Easy VPN remote devices" in the section "Related Documents."

How to Configure Cisco Easy VPN Remote

This section includes the following required and optional tasks.

Remote Tasks

Easy VPN Server Tasks

- Configuring a Cisco IOS Easy VPN Server, page 58 (required)
- Configuring an Easy VPN Server on a VPN 3000 Series Concentrator, page 58 (optional)
- Configuring an Easy VPN Server on a Cisco PIX Firewall, page 60 (optional)

Web Interface Tasks

- Configuring Web-Based Activation, page 61 (optional)
- Monitoring and Maintaining Web-Based Activation, page 61 (optional)
- Using SDM As a Web Manager, page 65 (optional)

Troubleshooting the VPN Connection

- Troubleshooting a VPN Connection Using the Cisco Easy VPN Remote Feature, page 65 (optional)
- Troubleshooting the Client Mode of Operation, page 65 (optional)
- Troubleshooting Remote Management, page 66 (optional)
- Troubleshooting Dead Peer Detection, page 66 (optional)

Remote Tasks

Configuring and Assigning the Easy VPN Remote Configuration

The router acting as the Easy VPN remote must create a Cisco Easy VPN Remote configuration and assign it to the outgoing interface. To configure and assign the remote configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **group** *group-name* **key** *group-key*
5. **peer** [*ip-address* | *hostname*]
6. **mode** {**client** | **network-extension**}
7. **exit**
8. **interface** *interface*
9. **crypto ipsec client ezvpn** *name* [**outside**]
10. **exit**
11. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn easy client remote`	Creates a remote configuration and enters Cisco Easy VPN Remote configuration mode.
Step 4	**group** *group-name* **key** *group-key* **Example:** `Router (config-crypto-ezvpn)# group easy-vpn-remote-groupname key easy-vpn-remote-password`	Specifies the IPsec group and IPsec key value to be associated with this configuration. **Note** The value of the *group-name* argument must match the group defined on the Easy VPN server. On Cisco IOS routers, use the **crypto isakmp client configuration group** and **crypto map dynmap isakmp authorization list** commands. **Note** The value of the *group-key argument* must match the key defined on the Easy VPN server. On Cisco IOS routers, use the **crypto isakmp client configuration group** command.
Step 5	**peer** [*ip-address* | *hostname*] **Example:** `Router (config-crypto-ezvpn)# peer 192.185.0.5`	Specifies the IP address or hostname for the destination peer (typically the IP address on the outside interface of the destination route). • Multiple peers may be configured. **Note** You must have a DNS server configured and available to use the *hostname* option.
Step 6	**mode** {**client** | **network-extension**} **Example:** `Router (config-crypto-ezvpn)# mode client`	Specifies the type of VPN connection that should be made. • **client**—Specifies that the router is configured for VPN client operation, using NAT or PAT address translation. Client operation is the default if the type of VPN connection is not specified • **network-extension**—Specifies that the router is to become a remote extension of the enterprise network at the destination of the VPN connection.
Step 7	**exit** **Example:** `Router (config-crypto-ezvpn)# exit`	Exits Cisco Easy VPN Remote configuration mode.

	Command	Purpose
Step 8	**interface** *interface* **Example:** `Router (config)# interface Ethernet1`	Enters interface configuration mode for the interface. • This interface will become the outside interface for the NAT or PAT translation.
Step 9	**crypto ipsec client ezvpn** *name* [**outside**] **Example:** `Router (config-if)# crypto ipsec client ezvpn easy_vpn_remote1 outside`	Assigns the Cisco Easy VPN Remote configuration to the interface. • This configuration automatically creates the necessary NAT or PAT translation parameters and initiates the VPN connection (if you are in client mode). **Note** The inside interface must be specified on Cisco 1700 and higher platforms.
Step 10	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 11	**exit** **Example:** `Router (config)# exit`	Exits global configuration mode.

Verifying the Cisco Easy VPN Configuration

To verify that the Cisco Easy VPN Remote configuration has been correctly configured, that the configuration has been assigned to an interface, and that the IPsec VPN tunnel has been established, perform the following steps.

SUMMARY STEPS

1. **show crypto ipsec client ezvpn**
2. **show ip nat statistics**

DETAILED STEPS

Step 1 Display the current state of the Cisco Easy VPN Remote connection using the **show crypto ipsec client ezvpn** command. The following is typical output for a Cisco 1700 series router using client mode:

```
Router# show crypto ipsec client ezvpn

Tunnel name : hw1
Inside interface list: FastEthernet0/0, Serial0/0,
Outside interface: Serial1/0
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 10.0.0.5
Mask: 255.255.255.255
Default Domain: cisco.com
Tunnel name : hw2
Inside interface list: Serial0/1,
Outside interface: Serial1/1
Current State: IPSEC_ACTIVE
```

```
Last Event: SOCKET_UP
Default Domain: cisco.com
```

Step 2 Display the NAT or PAT configuration that was automatically created for the VPN connection using the **show ip nat statistics** command. The "Dynamic mappings" field of this display gives the details for the NAT or PAT translation that is occurring on the VPN tunnel.

```
Router# show ip nat statistics

Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Outside interfaces:
  cable-modem0
Inside interfaces:
  Ethernet0
Hits: 1489  Misses: 1
Expired translations: 1
Dynamic mappings:
-- Inside Source
access-list 198 pool enterprise refcount 0
 pool enterprise: netmask 255.255.255.0
        start 192.168.1.90 end 192.168.1.90
        type generic, total addresses 1, allocated 0 (0%), misses 0\
```

If you are seeing IPSEC_ACTIVE in your output at this point, everything is operating as expected.

Configuring Save Password

To configure the Save Password feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **password encryption aes**
4. **crypto ipsec client ezvpn** *name*
5. **username** *name* **password** {**0** | **6**} {*password*}
6. **exit**
7. **show running-config**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**password encryption aes** **Example:** `Router (config)# password encryption aes`	Enables a type 6 encrypted preshared key.
Step 4	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn ezvpn1`	Creates a Cisco Easy VPN remote configuration and enters the Cisco Easy VPN remote configuration mode.
Step 5	**username** *name* **password** {**0** | **6**} {*password*} **Example:** `Router (config-crypto-ezvpn)# username server_1 password 0 blue`	Allows you to save your Xauth password locally on the PC. • The **0** keyword specifies that an unencrypted password will follow. • The **6** keyword specifies that an encrypted password will follow. • The *password* argument *is the unencrypted (cleartext) user password.*
Step 6	**exit** **Example:** `Router (config-crypto-ezvpn)# exit`	Exits the Cisco Easy VPN remote configuration mode.
Step 7	**show running-config** **Example:** `Router (config)# show running-config`	Displays the contents of the configuration file that is currently running.

Configuring Manual Tunnel Control

To configure control of IPsec VPN tunnels manually so that you can establish and terminate the IPsec VPN tunnels on demand, perform the following steps.

Note CLI is one option for connecting the tunnel. The preferred method is via the web interface (using SDM).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **connect** [**auto** | **manual]**
5. **exit**
6. **exit**
7. **crypto ipsec client ezvpn connect** *name*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote1`	Assigns a Cisco Easy VPN remote configuration to an interface and enters Cisco Easy VPN Remote configuration mode. • The *name* argument specifies the configuration name to be assigned to the interface.
Step 4	**connect** [**auto**	**manual]** **Example:** `Router (config-crypto-ezvpn)# connect manual`	Connects the VPN tunnel. Specify **manual** to configure manual tunnel control. • Automatic is the default; you do not need to use the **manual** keyword if your configuration is automatic.
Step 5	**exit** **Example:** `Router (config-crypto-ezvpn)# exit`	Exits Cisco Easy VPN Remote configuration mode.
Step 6	**exit** **Example:** `Router (config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 7	**crypto ipsec client ezvpn connect** *name* **Example:** `Router# crypto ipsec client ezvpn connect easy vpn remote1`	Connects a given Cisco Easy VPN remote configuration. • The *name* argument specifies the IPsec VPN tunnel name. **Note** If the tunnel name is not specified, the active tunnel is connected. If there is more than one active tunnel, the command fails with an error requesting that you specify the tunnel name.

Configuring Automatic Tunnel Control

To configure automatic tunnel control, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **connect** [**auto** | **manual**]
5. **exit**
6. **exit**
7. **crypto ipsec client ezvpn connect** *name*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote1`	Assigns a Cisco Easy VPN remote configuration to an interface and enters Cisco Easy VPN Remote configuration mode. • Specify the configuration name to be assigned to the interface.
Step 4	**connect** [**auto	manual**] **Example:** `Router (config-crypto-ezvpn)# connect auto`	Connects the VPN tunnel. • Specify **auto** to configure automatic tunnel control. Automatic is the default; you do not need to use this subcommand if your configuration is automatic.
Step 5	**exit** **Example:** `Router (config-crypto-ezvpn)# exit`	Exits Cisco Easy VPN Remote configuration mode.
Step 6	**exit** **Example:** `Router (config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

	Command	Purpose
Step 7	**crypto ipsec client ezvpn connect** *name* **Example:** `Router# crypto ipsec client ezvpn connect easy vpn remote1`	Connects a given Cisco Easy VPN remote configuration. • The *name* argument specifies the IPsec VPN tunnel name. **Note** If the tunnel name is not specified, the active tunnel is connected. If there is more than one active tunnel, the command fails with an error requesting that you specify the tunnel name.

Configuring Multiple Inside Interfaces

You can configure up to three inside interfaces for all platforms. You need to manually configure each inside interface using the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-name*
4. **exit**
5. **crypto ipsec client ezvpn** *name* [**outside** | **inside**]
6. **interface** *interface-name*
7. **exit**
8. **crypto ipsec client ezvpn** *name* [**outside** | **inside**]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-name* **Example:** `Router (config)# interface Ethernet0`	Selects the interface you want to configure by specifying the interface name and enters interface configuration mode.
Step 4	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.

	Command	Purpose
Step 5	**crypto ipsec client ezvpn** *name* [**outside** | **inside**] **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote 1 inside`	Specifies the Cisco Easy VPN remote configuration name to be assigned to the first inside interface. • You must specify **inside** for each inside interface.
Step 6	**interface** *interface-name* **Example:** `Router (config)# interface Ethernet1`	Selects the next interface you want to configure by specifying the next interface name and enters interface configuration mode.
Step 7	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 8	**crypto ipsec client ezvpn** *name* [**outside** | **inside**] **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote2 inside`	Specifies the Cisco Easy VPN remote configuration name to be assigned to the next inside interface. • You must specify **inside** for each inside interface. Repeat Step 3 through Step 4 to configure an additional tunnel if desired.

Configuring Multiple Outside Interfaces

You can configure multiple tunnels for outside interfaces, setting up a tunnel for each outside interface. You can configure a maximum of four tunnels using the following procedure for each outside interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-name*
4. **exit**
5. **crypto ipsec client ezvpn** *name* **[outside | inside]**
6. **interface** *interface-name*
7. **exit**
8. **crypto ipsec client ezvpn** *name* [**outside** | **inside**]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-name* **Example:** `Router (config)# interface Ethernet0`	Selects the first outside interface you want to configure by specifying the interface name and enters interface configuration mode.
Step 4	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 5	**crypto ipsec client ezvpn** *name* [**outside** | **inside**] **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote1 outside`	Specifies the Cisco Easy VPN remote configuration name to be assigned to the first outside interface. • Specify **outside** (optional) for each outside interface. If neither **outside** nor **inside** is specified for the interface, the default is **outside**.
Step 6	**interface** *interface-name* **Example:** `Router (config)# interface Ethernet1`	Selects the next outside interface you want to configure by specifying the next interface name.
Step 7	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 8	**crypto ipsec client ezvpn** *name* [**outside** | **inside**] **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote2 outside`	Specifies the Cisco Easy VPN remote configuration name to be assigned to the next outside interface. • Specify **outside** (optional) for each outside interface. If neither **outside** nor **inside** is specified for the interface, the default is **outside**. Repeat Step 3 through Step 4 to configure additional tunnels if desired.

Configuring Multiple Subnet Support

When configuring multiple subnet support, you must first configure an access list to define the actual subnets to be protected. Each source subnet or mask pair indicates that all traffic that is sourced from this network to any destination is protected by IPsec. For information about configuring ACLs, see "Access control lists, configuring" in the section "Additional References."

After you have defined the subnets, you must configure the crypto IPsec client EZVPN profile to use the ACLs.

Note Multiple subnets are not supported in client mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-name*
4. **exit**
5. **crypto ipsec client ezvpn** *name*
6. **acl** {*acl-name* | *acl-number*}

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-name* **Example:** `Router (config)# interface Ethernet1`	Selects the interface you want to configure by specifying the interface name and enters interface configuration mode.
Step 4	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 5	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn ez1`	Creates a Cisco Easy VPN remote configuration and enters crypto Easy VPN configuration mode.
Step 6	**acl** {*acl-name* | *acl-number*} **Example:** `Router (config-crypto-ezvpn)# acl acl-list1`	Specifies multiple subnets in a VPN tunnel.

Configuring Proxy DNS Server Support

As a way of implementing the use of the DNS addresses of the ISP when the WAN connection is down, the router in a Cisco Easy VPN remote configuration can be configured to act as a proxy DNS server. To enable the proxy DNS server functionality with the **ip dns server** command, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns server**

DETAILED STEPS

	Command	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip dns server` **Example:** `Router (config)# ip dns server`	Enables the router to act as a proxy DNS server. **Note** This definition is IOS specific.

What to Do Next

After configuring the router, you configure the Cisco IOS Easy VPN server as follows:

- Under the **crypto isakmp client configuration group** command, configure the *dns* subcommand as in the following example:

  dns *A.B.C.D A1.B1.C1.D1*

  These DNS server addresses should be pushed from the server to the Cisco Easy VPN remote and dynamically added to or deleted from the running configuration of the router.

For information about general DNS server functionality in Cisco IOS software applications, see *Configuring DNS* and *Configuring DNS on Cisco Routers*.

Configuring Dial Backup

Note The Dial Backup feature is not available in Cisco IOS Release 12.3(11)T.

To configure dial backup, perform the following steps.

SUMMARY STEPS

1. Create the Easy VPN backup configuration.
2. Add the backup subcommand details to the primary configuration.
3. Apply the backup Easy VPN configuration to the dial backup outside interface.
4. Apply the Easy VPN profile to the inside interfaces.

DETAILED STEPS

	Command	Purpose
Step 1	*Create the Easy VPN dial backup configuration.*	For details about the backup configuration, see the section "Dial Backup."
Step 2	Add the backup subcommand details to the primary configuration.	Use the **backup** subcommand and **track** keyword of the **crypto ipsec client ezvpn** command.
Step 3	Apply the backup Easy VPN configuration to the dial backup outside interface (for example, serial, async, or dialer).	For details about applying the backup configuration to the dial backup outside interface, see the section "Configuring Multiple Outside Interfaces."
Step 4	Apply the Easy VPN profile to the inside interfaces (there can be more than one).	For details about applying the Easy VPN profile to the inside interfaces, see the section "Configuring Multiple Inside Interfaces."

Configuring the DHCP Server Pool

To configure the Dynamic Host Configuration Protocol (DHCP) server pool, see the chapter "Configuring DHCP" in the *Cisco IOS IP Configuration Guide*, Release 12.3.

Resetting a VPN Connection

To reset the VPN connection, perform the following steps. The **clear** commands can be configured in any order or independent of one another.

SUMMARY STEPS

1. **enable**
2. **clear crypto ipsec client ezvpn**
3. **clear crypto sa**
4. **clear crypto isakmp**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto ipsec client ezvpn** **Example:** `Router# clear crypto ipsec client ezvpn`	Resets the Cisco Easy VPN remote state machine and brings down the Cisco Easy VPN remote connection on all interfaces or on a given interface (tunnel).
Step 3	**clear crypto sa** **Example:** `Router# clear crypto sa`	Deletes IPsec SAs.
Step 4	**clear crypto isakmp** **Example:** `Router# clear crypto isakmp`	Clears active IKE connections.

Monitoring and Maintaining VPN and IKE Events

To monitor and maintain VPN and IKE events, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug crypto ipsec client ezvpn**
3. **debug crypto ipsec**
4. **debug crypto isakmp**

SUMMARY STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto ipsec client ezvpn** **Example:** `Router# debug crypto ipsec client ezvpn`	Displays information showing the configuration and implementation of the Cisco Easy VPN Remote feature.

	Command	Purpose
Step 3	**debug crypto ipsec** **Example:** `Router# debug crypto ipsec`	Displays IPsec events.
Step 4	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays messages about IKE events.

Configuring a Virtual Interface

To configure a virtual interface, perform the following steps.

Note Before the virtual interface is configured, ensure that the Easy VPN profile is not applied on any outside interface. Remove the Easy VPN profile from the outside interface and then configure the virtual interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number* **type** *type-of-virtual-template*
4. **tunnel mode ipsec ipv4**
5. **exit**
6. **crypto ipsec client ezvpn** *name*
7. **virtual-interface** *virtual-template-number*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **type** *type-of-virtual-template* **Example:** `Router (config)# interface virtual-template1 type tunnel`	(Optional) Creates a virtual template of the type tunnel and enters interface configuration mode. • Steps 3, 4, and 5 are optional, but if one is configured, they must all be configured.

	Command	Purpose
Step 4	**tunnel mode ipsec ipv4** **Example:** `Router (if-config)# tunnel mode ipsec ipv4`	(Optional) Configures the tunnel that does the IPsec tunneling.
Step 5	**exit** **Example:** `Router (if-config)# exit`	(Optional) Exits interface (virtual-tunnel) configuration mode.
Step 6	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn EasyVPN1`	Creates a Cisco Easy VPN remote configuration and enters the Cisco Easy VPN remote configuration mode.
Step 7	**virtual-interface** *virtual-template-number* **Example:** `Router (config-crypto-ezvpn)# virtual-interface 3`	Instructs the Easy VPN remote to create a virtual interface to be used as an outside interface. If the virtual template number is specified, the virtual-access interface is derived from the virtual interface that was specified. If a virtual template number is not specified, a generic virtual-access interface is created.

Troubleshooting Dual Tunnel Support

The following **debug** and **show** commands may be used to troubleshoot your dual-tunnel configuration.

SUMMARY STEPS

1. **enable**
2. **debug crypto ipsec client ezvpn**
3. **debug ip policy**
4. **show crypto ipsec client ezvpn**
5. **show ip interface**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto ipsec client ezvpn** **Example:** `Router# debug crypto ipsec client ezvpn`	Displays information about Cisco Easy VPN remote connections.

	Command	Purpose
Step 3	**debug ip policy** **Example:** `Router# debug ip policy`	Displays IP policy routing packet activity.
Step 4	**show crypto ipsec client ezvpn** **Example:** `Router# show crypto ipsec client ezvpn`	Displays the Cisco Easy VPN Remote configuration.
Step 5	**show ip interface** **Example:** `Router# show ip interface`	Displays the usability status of interfaces that are configured for IP.

Configuring Reactivate (a Default) Primary Peer

To configure a default primary peer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **peer** {*ip-address* | *hostname*} [**default**]
5. **idle-time** *idle-time*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn ez1`	Creates a Cisco Easy VPN remote configuration and enters crypto Easy VPN configuration mode.

	Command	Purpose
Step 4	**peer** {*ip-address* | *hostname*} [**default**] **Example:** `Router (config-crypto-ezvpn)# peer 10.2.2.2 default`	Sets the peer IP address or hostname for the VPN connection. • A hostname can be specified only when the router has a DNS server available for hostname resolution. • The peer subcommand may be input multiple times. However, only one default or primary peer entry can exist at a time (for example, 10.2.2.2 default). • The **default** keyword defines the peer as the primary peer.
Step 5	**idle-time** *idle-time* **Example:** `Router (config-crypto-ezvpn)# idle-time 60`	(Optional) Idle time in seconds after which an Easy VPN tunnel is brought down. • Idle time=60 through 86400 seconds. **Note** If idle time is configured, the tunnel for the primary server is not brought down.

Configuring Identical Addressing Support

Configuring Identical Addressing Support comprises the following tasks:

- Defining the Easy VPN remote in network-extension mode and enabling **nat allow**.
- Assigning the Cisco Easy VPN Remote configuration to the Outside interface.
- Creating a loopback interface and assigning the Cisco Easy VPN Remote configuration to the Inside interface of the loopback interface.
- Configuring a one-to-one static NAT translation for each host that needs to be accessible from the EasyVPN server-side network or from other client locations.
- Configuring dynamic overloaded NAT or PAT using an access list for all the desired VPN traffic. The NAT or PAT traffic is mapped to the Easy VPN inside interface IP address.
- And, if split-tunneling is required, using the **nat acl** command to enable split-tunneling for the traffic specified by the *acl-name* or the *acl-number* argument. The ACL is the same as the ACL used by the NAT or PAT mapping in the preceding bullet item.

To configure Identical Addressing Support, perform the following steps on your router.

Prerequisites

Easy VPN Remote must be configured in network extension mode before you can configure the Identical Addressing Support feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **mode network-extension**
5. **nat allow**
6. **exit**
7. **interface** *interface*

8. **crypto ipsec client ezvpn name** *outside*
9. **exit**
10. **interface** *interface*
11. **ip address** *ip mask*
12. **crypto ipsec client ezvpn** *name* **inside**
13. **exit**
14. **ip nat inside source static** *local-ip global-ip*
15. **ip nat inside source list** {*acl-name* | *acl-number*} **interface** *interface* **overload**
16. **crypto ipsec client ezvpn** *name*
17. **nat acl** {*acl-name* | *acl-number}*
18. **exit**
19. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** Router (config)# crypto ipsec client ezvpn easyclient	Creates a remote configuration and enters Cisco Easy VPN Remote configuration mode.
Step 4	**mode network-extension** **Example:** Router (config-crypto-ezvpn)# mode network-extension	Configures Easy VPN client in network-extension mode.
Step 5	**nat allow** **Example:** Router (config-crypto-ezvpn)# nat allow	Allows NAT to be integrated with Easy VPN and enables the Identical Addressing feature.
Step 6	**exit** **Example:** Router (config-crypto-ezvpn)# exit	Exits Cisco Easy VPN Remote configuration mode.

	Command	Purpose
Step 7	**interface** *interface* **Example:** `Router (config)# interface Ethernet1`	Enters interface configuration mode for the interface. • This interface will become the outside interface for the NAT or PAT translation.
Step 8	**crypto ipsec client ezvpn** *name* **outside** **Example:** `Router (config-if)# crypto ipsec client ezvpn easyclient outside`	Assigns the Cisco Easy VPN Remote configuration to the outside interface. • This configuration automatically creates the necessary NAT or PAT translation parameters and initiates the VPN connection (if you are in client mode).
Step 9	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 10	**interface** *interface* **Example:** `Router (config)# interface Loopback0`	Enters interface configuration mode for the loopback interface. • This interface will become the inside interface for the NAT or PAT translation.
Step 11	**ip address** *ip mask* **Example:** `Router (config-if)# ip address 10.1.1.1 255.255.255.252`	Assigns the IP address and mask to the loopback interface.
Step 12	**crypto ipsec client ezvpn** *name* **inside** **Example:** `Router (config-if)# crypto ipsec client ezvpn easyclient inside`	Assigns the Cisco Easy VPN Remote configuration to the inside interface.
Step 13	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 14	**ip nat inside source static** *local-ip global-ip* **Example:** `Router (config)# ip nat inside source static 10.10.10.10 5.5.5.5`	Configure a one-to-one static NAT translation for each host that needs to be accessible from the Easy VPN server side network, or from other client locations.
Step 15	**ip nat inside source list** {*acl-name* | *acl-number*} **interface** *interface* **overload** **Example:** `Router (config)# ip nat inside source list 100 interface Loopback0 overload`	Configure dynamic overloaded NAT or PAT, which uses an ACL for all the desired VPN traffic. The NAT and PAT traffic is mapped to the Easy VPN inside interface IP address. • The *acl-name* argument is the name of the ACL. • The *acl-number* argument is the number of the ACL.

	Command	Purpose
Step 16	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn easyclient`	(Optional, if using split tunneling) Enters Cisco Easy VPN Remote configuration mode.
Step 17	**nat acl** {*acl-name* | *acl-number*} **Example:** `Router (config-crypto-ezvpn)# nat acl 100`	(Optional, if using split tunneling) Enables split-tunneling for the traffic specified by the *acl-name* or the *acl-number* argument. The ACL is the same as the ACL used by the NAT or PAT mapping in the Step 15. • The *acl-name* argument is the name of the ACL. • The *acl-number* argument is the number of the ACL.
Step 18	**exit** **Example:** `Router (config-crypto-ezvpn)# exit`	Exits Cisco Easy VPN Remote configuration mode.
Step 19	**exit** **Example:** `Router (config)# exit`	Exits global configuration mode.

Configuring cTCP on an Easy VPN Client

To configure cTCP on an Easy VPN client (remote device), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ctcp** [**keepalive** *number-of-seconds* | **port** *port-number*]
4. **crypto ipsec client ezvpn** *name*
5. **ctcp port** *port-number*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 3	**crypto ctcp** [**keepalive** *number-of-seconds* | **port** *port-number*] **Example:** `Router (config)# crypto ctcp keepalive 15`	Sets cTCP keepalive interval for the remote device. • *number-of-seconds*—Number of seconds between keepalives. Value = 5 through 3600. • **port** *port-number*—Port number that cTCP listens to. Up to 10 numbers can be configured. **Note** The cTCP client has to send periodic keepalives to the server to keep NAT or firewall sessions alive.
Step 4	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn ezvpn1`	Creates a Cisco Easy VPN remote configuration and enters Cisco Easy VPN remote configuration mode.
Step 5	**ctcp port** *port-number* **Example:** `Router (config-crypto-ezvpn)# ctcp port 200`	Sets the port number for cTCP encapsulation for Easy VPN. • *port-number*—Port number on the hub. Value = 1 through 65535.

Restricting Traffic When a Tunnel Is Down

To restrict the client from sending traffic in clear text when a tunnel is down, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **flow allow acl** [*name* | *number*]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn ezvpn1`	Creates a Cisco Easy VPN remote configuration and enters Cisco Easy VPN remote configuration mode.
Step 4	**flow allow acl** [**name** | **number**] **Example:** `Router (config-crypto-ezvpn)#flow allow acl 102`	Restricts the client from sending traffic in clear text when the tunnel is down. • *name*— Access list name. • *number*— Access list number. Value = 100 through 199.

Easy VPN Server Tasks

Configuring a Cisco IOS Easy VPN Server

For information about configuring the Easy VPN Server, see the following document:

- *Easy VPN Server*

Configuring an Easy VPN Server on a VPN 3000 Series Concentrator

This section describes the guidelines required to configure the Cisco VPN 3000 series concentrator for use with the Cisco Easy VPN Remote feature. As a general rule, you can use the default configuration except for IP addresses, server addresses, routing configurations, and for the following parameters and options:

- Peer Configuration on a Cisco Easy VPN Remote Using the Hostname, page 59
- Interactive Hardware Client Authentication Version 3.5, page 59
- IPsec Tunnel Protocol, page 59
- IPsec Group, page 59
- Group Lock, page 59
- Xauth, page 59
- Split Tunneling, page 60
- IKE Proposals, page 60
- New IPsec SA, page 60

Note You must be using Cisco VPN 3000 series concentrator software Release 3.11 or later to support Cisco Easy VPN software clients and remotes.

Peer Configuration on a Cisco Easy VPN Remote Using the Hostname

After you have configured the Cisco Easy VPN server on the VPN 3000 concentrator to use hostname as its identity, you must configure the peer on the Cisco Easy VPN remote using the hostname. You can either configure DNS on the client to resolve the peer hostname or configure the peer hostname locally on the client using the **ip host** command. As an example, you can configure the peer hostname locally on an Easy VPN remote as follows:

```
ip host crypto-gw.cisco.com 10.0.0.1
```

Or you can configure the Easy VPN remote to use the hostname with the **peer** command and *hostname* argument, as follows:

```
peer crypto-gw.cisco.com.
```

Interactive Hardware Client Authentication Version 3.5

The Cisco Easy VPN Remote feature does not support the Interactive Hardware Client Authentication Version 3.5 feature. This feature must be disabled. You can disable the feature on the VPN 3000 series concentrator by clicking the **HW Client** tab on the **Configuration | User Management | Base Group** screen.

IPsec Tunnel Protocol

IPsec Tunnel Protocol enables the IPsec tunnel protocol so that it is available for users. The IPsec Tunnel Protocol is configured on the Cisco VPN 3000 series concentrator by clicking the **General** tab on the **Configuration | User Management | Base Group** screen.

IPsec Group

IPsec group configures the Cisco VPN 3000 series concentrator with a group name and password that match the values configured for the Cisco Easy VPN remote configuration on the router. These values are configured on the router with the **group** *group-name* **key** *group-key* subcommand and arguments. The values are configured on the Cisco VPN 3000 series concentrator using the **Configuration | User Management | Groups** screen.

Group Lock

If you are defining multiple users in multiple groups on the VPN 3000 series concentrator, you must check the **Group Lock** box in the IPsec tab to prevent users in one group from logging in with the parameters of another group. For example, if you have configured one group for split tunneling access and another group without split tunneling access, clicking the **Group Lock** box prevents users in the second group from gaining access to the split tunneling features. The Group Lock checkbox appears in the **IPsec** tab in the **Configuration | User Management | Base Group** screen and in the **IPsec** tab in the **Configuration | User Management | Groups | Add/Modify** screens.

Xauth

To use Xauth, set the **Authentication** parameter to **None**. The Authentication parameter appears in the **IPsec** tab in the **Configuration | User Management | Base Group** screen and in the **IPsec** tab in the **Configuration | User Management | Groups | Add/Modify** screens.

Split Tunneling

The **Configuration | User Management | Base Group, Mode Configuration Parameters Tab** screen includes a **Split Tunnel** option with a checkbox that says "Allow the networks in the list to bypass the tunnel."

IKE Proposals

The Cisco VPN 3000 series concentrator is preconfigured with a default IKE proposal, CiscoVPNClient-3DES-MD5, that can be used with Cisco Easy VPN remotes. This IKE proposal supports preshared keys with Xauth using the MD5/HMAC-128 algorithm and Diffie-Hellman Group 2.

This IKE proposal is active by default, but you should verify that it is still an active proposal using the **Configuration | System | Tunneling Protocols | IPsec | IKE Proposals** screen.

In addition, as part of configuring the Cisco VPN 3000 series concentrator—for the Cisco Easy VPN Remote image, you do not need to create a new IPsec SA. Use the default IKE and Easy VPN remote lifetime configured on the Cisco VPN 3000 series concentrator.

Note You can also use the default IKE proposals IKE-DES-MD5 and IKE-3DES-MD5, but they do not enable Xauth support by default.

New IPsec SA

You can create a new IPsec SA. Cisco Easy VPN clients use a SA having the following parameters:

- Authentication Algorithm=ESP/MD5/HMAC-128
- Encryption Algorithm=DES-56 or 3DES-168 (recommended)
- Encapsulation Mode=Tunnel
- IKE Proposal=CiscoVPNClient-3DES-MD5 (preferred)

The Cisco VPN 3000 series concentrator is preconfigured with several default security associations (SAs), but they do not meet the IKE proposal requirements. To use an IKE proposal of CiscoVPNClient-3DES-MD5, copy the ESP/IKE-3DES-MD5 SA and modify it to use CiscoVPNClient-3DES-MD5 as its IKE proposal. An IKE proposal is configured on the VPN 3000 series concentrator using the **Configuration | Policy Management | Traffic Management | Security Associations** screen.

Configuring an Easy VPN Server on a Cisco PIX Firewall

For information about configuring an Easy VPN Server on a Cisco PIX Firewall, see the following document:

- *Easy VPN Server*

Web Interface Tasks

Configuring Web-Based Activation

To configure a LAN so that any HTTP requests coming from any of the PCs on the private LAN are intercepted, providing corporate users with access to the corporate Web page, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **xauth userid mode** {**http-intercept** | **interactive** | **local**}

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote1`	Assigns a Cisco Easy VPN remote configuration to an interface and enters Cisco Easy VPN Remote configuration mode. • The *name* argument specifies the configuration name to be assigned to the interface.
Step 4	**xauth userid mode** {**http-intercept** | **interactive** | **local**} **Example:** `Router (config-crypto-ezvpn)# xauth userid mode http-intercept`	Specifies how the VPN device handles Xauth requests or prompts from the server.

Monitoring and Maintaining Web-Based Activation

To monitor and maintain web-based activation, perform the following steps. (The **debug** and **show** commands may be used independently, or they may all be configured.)

SUMMARY STEPS

1. **enable**
2. **debug crypto ipsec client ezvpn**

3. **debug ip auth-proxy ezvpn**
4. **show crypto ipsec client ezvpn**
5. **show ip auth-proxy config**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto ipsec client ezvpn** **Example:** `Router# debug crypto ipsec client ezvpn`	Displays information about the Cisco Easy VPN connection.
Step 3	**debug ip auth-proxy ezvpn** **Example:** `Router# debug ip auth-proxy ezvpn`	Displays information related to proxy authentication behavior for web-based activation.
Step 4	**show crypto ipsec client ezvpn** **Example:** `Router# show crypto ipsec client ezvpn`	Shows that the username and password used for user credentials during Xauth negotiations will be obtained by intercepting HTTP connections from the user.
Step 5	**show ip auth-proxy config** **Example:** `Router# show ip auth-proxy config`	Displays the auth-proxy rule that has been created and applied by Easy VPN.

Examples

Debug Output

The following is sample **debug** output for a typical situation in which a user has opened a browser and connected to the corporate website:

```
Router# debug ip auth-proxy ezvpn

Dec 10 12:41:13.335: AUTH-PROXY: New request received by EzVPN WebIntercept
! The following line shows the ip address of the user.
from 10.4.205.205
Dec 10 12:41:13.335: AUTH-PROXY:GET request received
Dec 10 12:41:13.335: AUTH-PROXY:Normal auth scheme in operation
Dec 10 12:41:13.335: AUTH-PROXY:Ezvpn is NOT active. Sending connect-bypass page to user
```

At this point, the user chooses “connect” on his or her browser:

```
Dec 10 12:42:43.427: AUTH-PROXY: New request received by EzVPN WebIntercept
from 10.4.205.205
Dec 10 12:42:43.427: AUTH-PROXY:POST request received
Dec 10 12:42:43.639: AUTH-PROXY:Found attribute <connect> in form
Dec 10 12:42:43.639: AUTH-PROXY:Sending POST data to EzVPN
Dec 10 12:42:43.639: EZVPN(tunnel22): Communication from Interceptor
  application.
```

```
Request/Response from 10.4.205.205, via Ethernet0
Dec 10 12:42:43.639:         connect: Connect Now
Dec 10 12:42:43.639: EZVPN(tunnel22): Received CONNECT from 10.4.205.205!
Dec 10 12:42:43.643: EZVPN(tunnel22): Current State: CONNECT_REQUIRED
Dec 10 12:42:43.643: EZVPN(tunnel22): Event: CONNECT
Dec 10 12:42:43.643: EZVPN(tunnel22): ezvpn_connect_request
```

Easy VPN contacts the server:

```
Dec 10 12:42:43.643: EZVPN(tunnel22): Found valid peer 192.168.0.1
Dec 10 12:42:43.643: EZVPN(tunnel22): Added PSK for address 192.168.0.1

Dec 10 12:42:43.643: EZVPN(tunnel22): New State: READY
Dec 10 12:42:44.815: EZVPN(tunnel22): Current State: READY
Dec 10 12:42:44.815: EZVPN(tunnel22): Event: IKE_PFS
Dec 10 12:42:44.815: EZVPN(tunnel22): No state change
Dec 10 12:42:44.819: EZVPN(tunnel22): Current State: READY
Dec 10 12:42:44.819: EZVPN(tunnel22): Event: CONN_UP
Dec 10 12:42:44.819: EZVPN(tunnel22): ezvpn_conn_up B8E86EC7 E88A8A18 D0D51422
  8AFF32B7
```

The server requests Xauth information:

```
Dec 10 12:42:44.823: EZVPN(tunnel22): No state change
Dec 10 12:42:44.827: EZVPN(tunnel22): Current State: READY
Dec 10 12:42:44.831: EZVPN(tunnel22): Event: XAUTH_REQUEST
Dec 10 12:42:44.831: EZVPN(tunnel22): ezvpn_xauth_request
Dec 10 12:42:44.831: EZVPN(tunnel22): ezvpn_parse_xauth_msg
Dec 10 12:42:44.831: EZVPN: Attributes sent in xauth request message:
Dec 10 12:42:44.831:         XAUTH_TYPE_V2(tunnel22): 0
Dec 10 12:42:44.831:         XAUTH_USER_NAME_V2(tunnel22):
Dec 10 12:42:44.831:         XAUTH_USER_PASSWORD_V2(tunnel22):
Dec 10 12:42:44.831:         XAUTH_MESSAGE_V2(tunnel22) <Enter Username and
Password.>
Dec 10 12:42:44.831: EZVPN(tunnel22): Requesting following info for xauth
Dec 10 12:42:44.831:         username:(Null)
Dec 10 12:42:44.835:         password:(Null)
Dec 10 12:42:44.835:         message:Enter Username and Password.
Dec 10 12:42:44.835: EZVPN(tunnel22): New State: XAUTH_REQ
```

The username and password prompt are displayed in the browser of the user:

```
Dec 10 12:42:44.835: AUTH-PROXY: Response to POST  is CONTINUE
Dec 10 12:42:44.839: AUTH-PROXY: Displayed POST response successfully
Dec 10 12:42:44.843: AUTH-PROXY:Served POST response to the user
```

When the user enters his or her username and password, the following is sent to the server:

```
Dec 10 12:42:55.343: AUTH-PROXY: New request received by EzVPN WebIntercept
  from 10.4.205.205
Dec 10 12:42:55.347: AUTH-PROXY:POST request received
Dec 10 12:42:55.559: AUTH-PROXY:No of POST parameters is 3
Dec 10 12:42:55.559: AUTH-PROXY:Found attribute <username> in form
Dec 10 12:42:55.559: AUTH-PROXY:Found attribute <password> in form
Dec 10 12:42:55.559: AUTH-PROXY:Found attribute <ok> in form
Dec 10 12:42:55.563: AUTH-PROXY:Sending POST data to EzVPN
Dec 10 12:42:55.563: EZVPN(tunnel22): Communication from Interceptor application.
Request/Response from 10.4.205.205, via Ethernet0
Dec 10 12:42:55.563:         username:http
Dec 10 12:42:55.563:         password:<omitted>
Dec 10 12:42:55.563:         ok:Continue
Dec 10 12:42:55.563: EZVPN(tunnel22): Received usename|password from 10.4.205.205!
Dec 10 12:42:55.567: EZVPN(tunnel22): Current State: XAUTH_PROMPT
Dec 10 12:42:55.567: EZVPN(tunnel22): Event: XAUTH_REQ_INFO_READY
Dec 10 12:42:55.567: EZVPN(tunnel22): ezvpn_xauth_reply
```

```
Dec 10 12:42:55.567:         XAUTH_TYPE_V2(tunnel22): 0
Dec 10 12:42:55.567:         XAUTH_USER_NAME_V2(tunnel22): http
Dec 10 12:42:55.567:         XAUTH_USER_PASSWORD_V2(tunnel22): <omitted>
Dec 10 12:42:55.567: EZVPN(tunnel22): New State: XAUTH_REPLIED
Dec 10 12:42:55.891: EZVPN(tunnel22): Current State: XAUTH_REPLIED
Dec 10 12:42:55.891: EZVPN(tunnel22): Event: XAUTH_STATUS
Dec 10 12:42:55.891: EZVPN(tunnel22): xauth status received: Success
```

After using the tunnel, the user chooses “Disconnect”:

```
Dec 10 12:48:17.267: EZVPN(tunnel22): Received authentic disconnect credential
Dec 10 12:48:17.275: EZVPN(): Received an HTTP request: disconnect
Dec 10 12:48:17.275: %CRYPTO-6-EZVPN_CONNECTION_DOWN: (Client)  User=
  Group=tunnel22  Client_public_addr=192.168.0.13  Server_public_addr=192.168.0.1
  Assigned_client_addr=10.3.4.5
```

Show Output Before the User Is Connected to the Tunnel

The following output from the two **show** commands (**show crypto ipsec client ezvpn** and **show ip auth-proxy config**) displays what you might see before a user is connected to a VPN tunnel:

```
Router# show crypto ipsec client ezvpn tunnel22

Tunnel name : tunnel22
Inside interface list: Ethernet0
Outside interface: Ethernet1
Current State: CONNECT_REQUIRED
Last Event: RESET
Save Password: Disallowed
! Note the next line.
        XAuth credentials: HTTP intercepted
        HTTP return code : 200
        IP addr being prompted: 0.0.0.0
Current EzVPN Peer: 192.168.0.1

Router# show ip auth-proxy config

Authentication global cache time is 60 minutes
Authentication global absolute time is 0 minutes
Authentication Proxy Watch-list is disabled

Authentication Proxy Rule Configuration
! Note that the next line is the Easy VPN-defined internal rule.
    Auth-proxy name ezvpn401***
      Applied on Ethernet0
      http list not specified inactivity-timer 60 minutes
```

Show Output After the User Is Connected to the Tunnel

The following output from the two **show** commands (**show crypto ipsec client ezvpn** and **show ip auth-proxy config**) displays what you might see after the user has been connected to the tunnel:

```
Router# show crypto ipsec client ezvpn tunnel22

Tunnel name : tunnel22
Inside interface list: Ethernet0
Outside interface: Ethernet1
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 10.3.4.5
Mask: 255.255.255.255
Save Password: Disallowed
        XAuth credentials: HTTP intercepted
        HTTP return code : 200
        IP addr being prompted: 192.168.0.0
```

```
Current EzVPN Peer: 192.168.0.1

Router# show ip auth-proxy config

Authentication global cache time is 60 minutes
Authentication global absolute time is 0 minutes
Authentication Proxy Watch-list is disabled

Auth-proxy name ezvpnWeb*** (EzVPN-defined internal rule)
http list not specified inactivity-timer 60 minutes
```

Using SDM As a Web Manager

For information about the SDM web manager, see the following document:

- *Cisco Security Device Manager*

Troubleshooting the VPN Connection

Troubleshooting a VPN Connection Using the Cisco Easy VPN Remote Feature

To troubleshoot a VPN connection created using the Cisco Easy VPN Remote feature, use the following suggested techniques.

- Be aware that any changes to an active Cisco Easy VPN remote configuration or IP address changes to the involved interfaces, such as adding or removing an inside interface, result in a reset of the Cisco Easy VPN Remote connection.
- Enable debugging of the Cisco Easy VPN Remote feature using the **debug crypto ipsec client ezvpn** command.
- Enable debugging of IKE events using the **debug crypto ipsec** and **debug crypto isakmp** commands.
- Display the active IPsec VPN connections using the **show crypto engine connections active** command.
- To reset the VPN connection, use the **clear crypto ipsec client ezvpn** command. If you have debugging enabled, you might prefer to use the **clear crypto sa** and **clear crypto isakmp** commands.

Troubleshooting the Client Mode of Operation

The following information may be used to troubleshoot the Easy VPN Remote configuration for the client mode of operation.

In client mode, the Cisco Easy VPN Remote feature automatically configures the NAT or PAT translation and access lists that are needed to implement the VPN tunnel. These configurations are automatically created when the IPsec VPN connection is initiated. When the tunnel is torn down, the NAT or PAT and access list configurations are automatically deleted.

The NAT or PAT configuration is created with the following assumptions:

- The **ip nat inside** command is applied to all inside interfaces, including default inside interfaces. The default inside interface is the Ethernet 0 interface (for the Cisco 806, Cisco 826, Cisco 827, Cisco 828, Cisco 831, Cisco 836, and Cisco 837 routers).

- The **ip nat outside** command is applied to the interface that is configured with the Cisco Easy VPN Remote configuration. On the Cisco 800 series and Cisco 1700 series routers, the outside interface is configured with the Cisco Easy VPN Remote configuration. On the Cisco 1700 series routers, Cisco 2600 series routers, Cisco 3600 series routers, and Cisco 3700 series routers, multiple outside interfaces can be configured.

Tip The NAT or PAT translation and access list configurations that are created by the Cisco Easy VPN Remote feature are not written to either the startup configuration or running configuration files. These configurations, however, can be displayed using the **show ip nat statistics** and **show access-list** commands.

Troubleshooting Remote Management

To troubleshoot remote management of the VPN remote, use the **show ip interface** command. Using the **brief** keyword, you can verify that the loopback has been removed and that the interface is shown correctly.

Examples

Following is a typical example of output from the **show ip interface** command.

```
Router# show ip interface brief

Interface                  IP-Address      OK? Method Status                Protocol
Ethernet0                  unassigned      YES NVRAM  administratively down down

Ethernet1                  10.0.0.11       YES NVRAM  up                    up

Loopback0                  192.168.6.1     YES manual up                    up

Loopback1                  10.12.12.12     YES NVRAM  up                    up

Router# show ip interface brief

Interface                  IP-Address      OK? Method Status                Protocol
Ethernet0                  unassigned      YES NVRAM  administratively down down

Ethernet1                  10.0.0.11       YES NVRAM  up                    up

Loopback1                  10.12.12.12     YES NVRAM  up                    up
```

Troubleshooting Dead Peer Detection

To troubleshoot dead peer detection, use the **show crypto ipsec client ezvpn** command.

Examples

The following typical output displays the current server and the peers that have been pushed by the Easy VPN server:

```
Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 4
Tunnel name : ez1
Inside interface list: Loopback1,
Outside interface: Ethernet1
```

```
Current State: IPSEC_ACTIVE
Last Event: CONNECT
Address: 192.168.6.5
Mask: 255.255.255.255
DNS Primary: 10.2.2.2
DNS Secondary: 10.2.2.3
NBMS/WINS Primary: 10.6.6.6
Default Domain: cisco.com
Save Password: Allowed
Current EzVPN Peer:10.0.0.110
Backup Gateways
 (0): green.cisco.com
 (1): blue
```

Configuration Examples for Cisco Easy VPN Remote

This section provides the following configuration examples.

Easy VPN Remote Configuration Examples

Easy VPN Server Configuration Examples

Easy VPN Remote Configuration Examples

Client Mode Configuration: Examples

The examples in this section show configurations for the Cisco Easy VPN Remote feature in client mode. Also shown are the Cisco IOS Easy VPN server configurations that correspond to these client configurations.

- Cisco Easy VPN Client in Client Mode (Cisco 831): Example, page 68
- Cisco Easy VPN Client in Client Mode (Cisco 837): Example, page 69
- Cisco Easy VPN Client in Client Mode (Cisco 1700 Series): Example, page 71

For more client-mode configuration examples, see *IPSec VPN* (under the "Technical Documents" and "Cisco IOS IPSec Configuration Documents" sections) and to *Cisco Easy VPN Solutions*.

Note Typically, users configure the Cisco 800 series routers with the SDM or CRWS web interface, not by entering CLI commands. However, the configurations shown here for the Cisco 800 series routers display typical configurations that can be used if manual configuration is desired.

Cisco Easy VPN Client in Client Mode (Cisco 831): Example

In the following example, a Cisco 831 router is configured as an Easy VPN remote using the Cisco Easy VPN Remote feature in client mode. This example shows the following components of the Cisco Easy VPN Remote configuration:

- DHCP server pool—The **ip dhcp pool** command creates a pool of IP addresses to be assigned to the PCs connected to the Ethernet 0 interface of the router. The pool assigns addresses in the class C private address space (192.168.100.0) and configures each PC so that its default route is 192.168.100.1, which is the IP address assigned to the Ethernet interface of the router. The DHCP lease period is one day.
- Cisco Easy VPN remote configuration—The first **crypto ipsec client ezvpn easy vpn remote** command (global configuration mode) creates a Cisco Easy VPN remote configuration named "easy vpn remote." This configuration specifies the group name "easy vpn remote-groupname" and the shared key value "easy vpn remote-password," and it sets the peer destination to the IP address **192.185.0.5** (which is the address assigned to the interface connected to the Internet on the destination peer router). The Cisco Easy VPN Remote configuration is configured for the default **client** mode.

  Note If DNS is also configured on the router, the **peer** keyword option also supports a hostname instead of an IP address.

- The second **crypto ipsec client ezvpn easy vpn remote** command (interface configuration mode) assigns the Cisco Easy VPN Remote configuration to the Ethernet 1 interface so that all traffic that is received and transmitted on that interface is sent through the VPN tunnel.

```
! Cisco Router Web Setup Template
!
no service pad
no service tcp-small-servers
no service udp-small-servers
service timestamps debug uptime
service timestamps log uptime
```

```
service password-encryption
!
hostname 806Router
!
!
ip subnet-zero
ip domain-lookup
ip dhcp excluded-address 10.10.10.1
!
ip dhcp pool CLIENT
   import all
   network 10.10.10.0 255.255.255.255
   default-router 10.10.10.1
   lease 1 0 0
!
!
crypto ipsec client ezvpn easy_vpn_remote
 peer 192.168.0.5
 group easy_vpn_remote_groupname key easy_vpn_remote_password
 mode client
!
!
interface Ethernet0
 ip address 10.10.10.1 255.255.255.255
 no cdp enable
 hold-queue 32 in
!
interface Ethernet1
 ip address dhcp
 no cdp enable
 crypto ipsec client ezvpn easy_vpn_remote
!
ip classless
ip http server
!
!
ip route 10.0.0.0 10.0.0.0 Ethernet1
!
line con 0
 exec-timeout 120 0
 stopbits 1
line vty 0 4
 exec-timeout 0 0
 login local
```

Cisco Easy VPN Client in Client Mode (Cisco 837): Example

In the following example, a Cisco 837 router is configured as an Easy VPN remote using the Cisco Easy VPN Remote feature in the client mode of operation. This example shows the following components of the Cisco Easy VPN remote configuration:

- PPPoE configuration—The ATM 0 interface is configured to support PPPoE connections over the Dialer 1 virtual interface. Because the interfaces use PPPoE, a DHCP IP address pool is not required to provide IP addresses to the connected PCs.
- Cisco Easy VPN Remote configuration—The first **crypto ipsec client ezvpn** command (global configuration mode) creates a Cisco Easy VPN remote configuration named “easy vpn remote.” This configuration specifies the group name “easy vpn remote-groupname” and the shared key value of “easy vpn remote-password,” and it sets the peer destination to the IP address 10.0.0.5 (which is the address assigned to the interface connected to the Internet on the destination peer router). The Cisco Easy VPN remote configuration is configured for the default client mode.

Note If DNS is also configured on the router, the **peer** keyword option also supports a hostname instead of an IP address.

- The second **crypto ipsec client ezvpn** command (interface configuration mode) assigns the Cisco Easy VPN remote configuration to the Dialer 1 interface so that all traffic received and transmitted on that interface is sent through the VPN tunnel.

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname c827
!
!
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
vpdn enable
!
vpdn-group pppoe
 request-dialin
  protocol pppoe
 ip mtu adjust
!!
!
crypto ipsec client ezvpn easy_vpn_remote
 group easy_vpn_remote_groupname key easy_vpn_remote_password
 mode client
 peer 10.0.0.5
!!
!
interface Ethernet0
 ip address 10.0.0.117 255.0.0.0
 hold-queue 100 out
!
interface ATM0
 no ip address
 no atm ilmi-keepalive
 pvc 1/40
  pppoe-client dial-pool-number 1
 !
 dsl operating-mode auto
!
interface Dialer1
 ip address 10.0.0.3 255.0.0.0
 ip mtu 1492
 encapsulation ppp
 dialer pool 1
 crypto ipsec client ezvpn easy_vpn_remote
!
ip classless
ip route 0.0.0.0 0.0.0.0 ATM0
ip route 0.0.0.0 0.0.0.0 Dialer1 permanent
ip route 10.0.0.0 255.0.0.0 10.0.0.13
```

```
ip http server
ip pim bidir-enable
!
line con 0
 stopbits 1
line vty 0 4
 login
!
scheduler max-task-time 5000
end
```

Cisco Easy VPN Client in Client Mode (Cisco 1700 Series): Example

In the following example, a Cisco 1753 router is configured as an Easy VPN remote using the Cisco Easy VPN Remote feature in the client mode of operation. This example shows a running configuration of a Cisco 1753 that has two inside interfaces and one outside interface on one tunnel. The **connect auto** subcommand manually establishes the IPsec VPN tunnel.

```
Router# show running-config

Building configuration...
Current configuration : 881 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname mma-1753
!
!
memory-size iomem 15
ip subnet-zero
!!
!
ip ssh time-out 120
ip ssh authentication-retries 3
! !
!
crypto ipsec client ezvpn easy_vpn_remote
connect auto
group ezvpn key ezvpn
mode client
peer 10.6.6.1
! !
!
interface FastEthernet0/0
ip address 10.4.4.2 255.255.255.0
speed auto
crypto ipsec client ezvpn easy_vpn_remote inside
!
interface Serial0/0
ip address 10.6.6.2 255.255.255.0
no fair-queue
crypto ipsec client ezvpn easy_vpn_remote
!
interface Serial1/0
ip address 10.5.5.2 255.255.255.0
clock rate 4000000
crypto ipsec client ezvpn easy_vpn_remote inside
!
ip classless
no ip http server
```

```
ip pim bidir-enable
! !
!
line con 0
line aux 0
line vty 0 4
login
!
end
```

The following example shows a running configuration of a Cisco 1760 router that has two active, automatically connected tunnels, easy vpn remote1 and easy vpn remote2. Tunnel easy vpn remote1 has two configured inside interfaces and one configured outside interface. Tunnel easy vpn remote2 has one configured inside interface and one configured outside interface. The example also shows the output for the **show crypto ipsec client ezvpn** command that lists the tunnel names and the outside and inside interfaces.

```
Router# show running-config

Building configuration...
Current configuration : 1246 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 1760
!
aaa new-model
!
!
aaa session-id common
!
ip subnet-zero
!!
!
crypto ipsec client ezvpn easy_vpn_remote2
connect auto
group ez key ez
mode network-extension
peer 10.7.7.1
crypto ipsec client ezvpn easy_vpn_remote1
connect auto
group ezvpn key ezvpn
mode client
peer 10.6.6.1
! !
!
interface FastEthernet0/0
ip address 10.5.5.2 255.255.255.0
speed auto
no cdp enable
crypto ipsec client ezvpn easy_vpn_remote1 inside
!
interface Serial0/0
ip address 10.4.4.2 255.255.255.0
no ip route-cache
no ip mroute-cache
no fair-queue
no cdp enable
crypto ipsec client ezvpn easy_vpn_remote1 inside
!
```

```
interface Serial0/1
ip address 10.3.3.2 255.255.255.0
no cdp enable
crypto ipsec client ezvpn easy_vpn_remote2 inside
!
interface Serial1/0
ip address 10.6.6.2 255.255.255.0
clockrate 4000000
no cdp enable
crypto ipsec client ezvpn easy_vpn_remote1
!
interface Serial1/1
ip address 10.7.7.2 255.255.255.0
no keepalive
no cdp enable
crypto ipsec client ezvpn easy_vpn_remote2
!
ip classless
no ip http server
ip pim bidir-enable
!
!
radius-server retransmit 3
radius-server authorization permit missing Service-Type
!
line con 0
line aux 0
line vty 0 4
!
no scheduler allocate
end


Router# show crypto ipsec client ezvpn

Tunnel name : easy_vpn_remote1
Inside interface list: FastEthernet0/0, Serial0/0,
Outside interface: Serial1/0
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 10.0.0.5
Mask: 255.255.255.255
Default Domain: cisco.com
Tunnel name : easy_vpn_remote2
Inside interface list: Serial0/1,
Outside interface: Serial1/1
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Default Domain: cisco.com
```

Local Address Support for Easy VPN Remote: Example

The following example shows that the **local-address** command is used to specify the loopback 0 interface for sourcing tunnel traffic:

```
Router# configure terminal
Router(config)# crypto ipsec client ezvpn telecommuter-client
Router(config-crypto-ezvpn)# local-address loopback0
```

Network Extension Mode Configuration: Examples

In this section, the following examples demonstrate how to configure the Cisco Easy VPN Remote feature in the network extension mode of operation. Also shown are the Cisco IOS Easy VPN server configurations that correspond to these client configurations.

- Cisco Easy VPN Client in Network Extension Mode (Cisco 831): Example, page 74
- Cisco Easy VPN Client in Network Extension Mode (Cisco 837): Example, page 75
- Cisco Easy VPN Client in Network Extension Mode (Cisco 1700 Series): Example, page 77

For more network extension mode configuration examples, see *IPSec VPN* (under the "Technical Documents" and "Cisco IOS IPSec Configuration Documents" sections) and to *Cisco Easy VPN Solutions*.

Cisco Easy VPN Client in Network Extension Mode (Cisco 831): Example

In the following example, a Cisco 831 router is configured as an Easy VPN remote using the Cisco Easy VPN Remote feature. This example shows the following components of the Cisco Easy VPN remote configuration:

- The Ethernet 0 interface is assigned an address in the network address space of the Cisco IOS Easy VPN server. The **ip route** command directs all traffic for this network space from the Ethernet 1 interface to the destination server.
- Cisco Easy VPN Remote configuration—The first **crypto ipsec client ezvpn** command (global configuration mode) creates a Cisco Easy VPN remote configuration named "easy vpn remote." This configuration specifies the group name "easy vpn remote-groupname" and the shared key value "easy vpn remote-password," and it sets the peer destination to the IP address 192.185.0.5 (which is the address assigned to the interface connected to the Internet on the destination peer router). The Cisco Easy VPN remote configuration is configured for network extension mode.

  Note If DNS is also configured on the router, the **peer** keyword option also supports a hostname instead of an IP address.

- The second **crypto ipsec client ezvpn** command (interface configuration mode) assigns the Cisco Easy VPN Remote configuration to the Ethernet 1 interface so that all traffic that is received and transmitted on that interface is sent through the VPN tunnel.

```
! Cisco Router Web Setup Template
!
no service pad
no service tcp-small-servers
no service udp-small-servers
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname Router
!
!
ip subnet-zero
ip domain-lookup
!
!
ip dhcp excluded-address 172.31.1.1
!
ip dhcp pool localpool
```

```
   import all
   network 172.31.1.0 255.255.255.255
   default-router 172.31.1.1
   lease 1 0 0
!
!
crypto ipsec client ezvpn easy_vpn_remote
 peer 192.168.0.5
 group easy_vpn_remote_groupname key easy_vpn_remote_password
 mode network-extension
!
!
interface Ethernet0
 ip address 172.31.1.1 255.255.255.255
 no cdp enable
 hold-queue 32 in
!
interface Ethernet1
 ip address dhcp
 no cdp enable
 crypto ipsec client ezvpn easy_vpn_remote
!
ip classless
ip route 172.31.0.0 255.255.255.255 Ethernet1
ip http server
!
!
line con 0
 exec-timeout 120 0
 stopbits 1
line vty 0 4
 exec-timeout 0 0
 login local
```

Cisco Easy VPN Client in Network Extension Mode (Cisco 837): Example

In the following example, a Cisco 837 router is configured as an Easy VPN remote using the Cisco Easy VPN Remote feature in client mode. This example shows the following components of the Cisco Easy VPN remote configuration:

- PPPoE configuration—The ATM 0 interface is configured to support PPPoE connections over the Dialer 1 virtual interface. Because the interfaces use PPPoE, a DHCP IP address pool is not required to provide IP addresses to the connected PCs.
- The Ethernet 0 interface is assigned an address in the network address space of the Cisco IOS Easy VPN server. The **ip route** command directs all traffic for this network space from the Dialer 1 interface to the destination server.
- Cisco Easy VPN Remote configuration—The first **crypto ipsec client ezvpn** command (global configuration mode) creates a Cisco Easy VPN remote configuration named “easy vpn remote.” This configuration specifies the group name “easy vpn remote-groupname” and the shared key value “easy vpn remote-password,” and it sets the peer destination to the IP address 10.0.0.5 (which is the address assigned to the interface connected to the Internet on the destination peer router). The Cisco Easy VPN remote configuration is configured for the default network extension mode.

Note If DNS is also configured on the router, the **peer** keyword option also supports a hostname instead of an IP address.

- The second **crypto ipsec client ezvpn** command (interface configuration mode) assigns the Cisco Easy VPN remote configuration to the Dialer1 interface so that all traffic that is received and transmitted on that interface is sent through the VPN tunnel.

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname c827
!
!
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
vpdn enable
!
vpdn-group pppoe
 request-dialin
  protocol pppoe
 ip mtu adjust
!
!
crypto ipsec client ezvpn easy_vpn_remote
 group easy_vpn_remote_groupname key easy_vpn_remote_password
 mode network-extension
 peer 10.0.0.5
!
!
interface Ethernet0
 ip address 172.16.0.30 255.255.255.192
 hold-queue 100 out
!
interface ATM0
 no ip address
 no atm ilmi-keepalive
 pvc 1/40
  pppoe-client dial-pool-number 1
 !
 dsl operating-mode auto
!
interface Dialer1
 ip address 10.0.0.3 255.0.0.0
 ip mtu 1492
 encapsulation ppp
 dialer pool 1
 crypto ipsec client ezvpn easy_vpn_remote
!
ip classless
ip route 172.16.0.0 255.255.255.128 Dialer1
ip route 0.0.0.0 0.0.0.0 ATM0
ip route 0.0.0.0 0.0.0.0 Dialer1 permanent
ip route 10.0.0.0 255.0.0.0 10.0.0.13
ip http server
ip pim bidir-enable
!
line con 0
 stopbits 1
```

```
line vty 0 4
 login
!
scheduler max-task-time 5000
```

Cisco Easy VPN Client in Network Extension Mode (Cisco 1700 Series): Example

In the following example, a Cisco 1700 series router is configured as an Easy VPN remote using the Cisco Easy VPN Remote feature in the network extension mode of operation. This example shows the following components of the Cisco Easy VPN remote configuration:

- Cisco Easy VPN Remote configuration—The first **crypto ipsec client ezvpn** command (global configuration mode) creates a Cisco Easy VPN remote configuration that is named “easy vpn remote.” This configuration specifies the group name “easy vpn remote-groupname” and the shared key value “easy vpn remote-password,” and it sets the peer destination to the IP address 10.0.0.2 (which is the address assigned to the interface connected to the Internet on the destination peer router). The Cisco Easy VPN remote configuration is configured for network extension mode.

  Note If DNS is also configured on the router, the **peer** keyword option also supports a hostname instead of an IP address.

- The second **crypto ipsec client ezvpn easy vpn remote** command (interface configuration mode) assigns the Cisco Easy VPN remote configuration to the Ethernet 0 interface so that all traffic that is received and transmitted on that interface is sent through the VPN tunnel.

```
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 1710
!
!
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
!
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
!
ip dhcp excluded-address 10.0.0.10
!
ip dhcp pool localpool
   import all
   network 10.70.0.0 255.255.255.248
   default-router 10.70.0.10
   lease 1 0 0
!
!
crypto ipsec client ezvpn easy_vpn_remote
 group easy_vpn_remote_groupname key easy_vpn_remote_password
 mode network-extension
 peer 10.0.0.2
!
!
```

```
interface Ethernet0
 ip address 10.50.0.10 255.0.0.0
 half-duplex
 crypto ipsec client ezvpn easy_vpn_remote
!
interface FastEthernet0
 ip address 10.10.0.10 255.0.0.0
 speed auto
!
ip classless
ip route 10.20.0.0 255.0.0.0 Ethernet0
ip route 10.20.0.0 255.0.0.0 Ethernet0
no ip http server
ip pim bidir-enable
!!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
```

Save Password Configuration: Example

The following sample **show running-config** output shows that the Save Password feature has been configured (note the **password encryption aes** command and **username** keywords in the output):

```
Router# show running-config

133.CABLEMODEM.CISCO: Oct 28 18:42:07.115: %SYS-5-CONFIG_I: Configured from console by
consolen
Building configuration...

Current configuration : 1269 bytes
!
! Last configuration change at 14:42:07 UTC Tue Oct 28 2003
!
version 12.3
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
!
clock timezone UTC -4
no aaa new-model
ip subnet-zero
no ip routing
!
!
ip audit notify log
ip audit po max-events 100
ip ssh break-string
no ftp-server write-enable
password encryption aes
!
!
```

```
no crypto isakmp enable
!
!
crypto ipsec client ezvpn remote_vpn_client
 connect auto
 mode client
 username greentree password  6 ARiFgh`SOJfMHLK[MHMQJZagR\M
!
!
interface Ethernet0
 ip address 10.3.66.4 255.255.255.0
 no ip route-cache
 bridge-group 59
```

PFS Support: Examples

The following **show crypto ipsec client ezvpn** command output shows the group name ("2") and that PFS is being used:

```
Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 4

Tunnel name : ez1
Inside interface list: Loopback1,
Outside interface: Ethernet1
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 192.168.6.6
Mask: 255.255.255.255
Using PFS Group: 2
Save Password: Allowed
Current EzVPN Peer:10.0.0.110
```

Note that on a Cisco IOS EasyVPN server, PFS must be included in IPsec proposals by adding to the crypto map, as in the following example:

```
crypto dynamic-map mode 1
 set security-association lifetime seconds 180
 set transform-set client
 set pfs group2
 set isakmp-profile fred
 reverse-route
```

Dial Backup: Examples

Static IP Addressing

The following example shows that static IP addressing has been configured for a Cisco 1711 router:

```
Router# show running-config

Building configuration...

Current configuration : 3427 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ph4_R5
```

```
!
boot-start-marker
boot-end-marker
!
no logging buffered
!
username ph4_R8 password 0 cisco
username ph4_R7 password 0 lab
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
aaa new-model
!
!
aaa session-id common
ip subnet-zero
!
!
no ip domain lookup
ip cef
ip ids po max-events 100
ip dhcp-client default-router distance 1
no ftp-server write-enable
!
!
track 123 rtr 3 reachability
!
crypto isakmp keepalive 10 periodic
!
!
crypto ipsec client ezvpn backup_profile_vpn3k
 connect auto
 group hw_client_groupname key password123
 mode client
 peer 10.0.0.5
 username rchu password  password123
crypto ipsec client ezvpn hw_client_vpn3k
 connect auto
 group hw_client_groupname key password123
 backup backup_profile_vpn3k track 123
 mode client
 peer 10.0.0.5
 username rchu password password123
!
!
interface Loopback0
 ip address 10.40.40.50 255.255.255.255
!
interface Loopback1
 ip address 10.40.40.51 255.255.255.255
!
interface Loopback2
 no ip address
!
interface FastEthernet0
 description Primary Link to 10.0.0.2
 ip address 10.0.0.10 255.255.255.0
 duplex auto
 speed auto
 no cdp enable
 crypto ipsec client ezvpn hw_client_vpn3k
!
interface FastEthernet1
```

```
 no ip address
 duplex full
 speed 100
 no cdp enable
!
interface FastEthernet2
 no ip address
 no cdp enable
!
interface FastEthernet3
 no ip address
 no cdp enable
!
interface FastEthernet4
 no ip address
 no cdp enable
!
interface Vlan1
 ip address 10.0.0.1 255.255.255.0
 crypto ipsec client ezvpn backup_profile_vpn3k inside
 crypto ipsec client ezvpn hw_client_vpn3k inside
!
interface Async1
 description Backup Link
 no ip address
 ip nat outside
 ip virtual-reassembly
 encapsulation ppp
 no ip route-cache cef
 dialer in-band
 dialer pool-member 1
 dialer-group 1
 async default routing
 async mode dedicated
!
interface Dialer1
 ip address 10.30.0.1 255.255.255.0
 encapsulation ppp
 no ip route-cache cef
 dialer pool 1
 dialer idle-timeout 60
 dialer string 102
 dialer hold-queue 100
 dialer-group 1
 crypto ipsec client ezvpn backup_profile_vpn3k
!
ip local policy route-map policy_for_rtr
ip classless

ip route 0.0.0.0 0.0.0.0 faste0 track 123


ip route 0.0.0.0 0.0.0.0 Dialer1 240
no ip http server
no ip http secure-server
!
!
ip access-list extended dummy1
 permit ip host 10.0.0.2 host 10.3.0.1
ip access-list extended important_traffic
 permit ip 10.0.0.0 0.0.0.255 10.0.0.2 0.0.0.255
 permit ip 10.0.0.0 0.0.0.255 10.0.0.3 0.0.0.255
ip access-list extended important_traffic_2
 permit ip 10.0.0.0 0.0.0.255 10.0.0.3 0.0.0.255
```

```
access-list 112 permit icmp any host 10.0.10.2 echo
dialer-list 1 protocol ip permit
no cdp run
!
route-map policy_for_rtr permit 10
 match ip address 112
 set interface Null0
 set ip next-hop 10.0.10.2
!
!
control-plane
!
rtr 2
 type echo protocol ipIcmpEcho 10.0.0.2 source-ipaddr 10.0.0.3
 timeout 10000
 threshold 1000
 frequency 11
rtr schedule 2 life forever start-time now
rtr 3
 type echo protocol ipIcmpEcho 10.0.0.2 source-interface FastEthernet0
 timeout 10000
 threshold 1000
 frequency 11
rtr schedule 3 life forever start-time now
!
line con 0
 exec-timeout 0 0
line 1
 modem InOut
 modem autoconfigure discovery
 transport input all
 autoselect ppp
 stopbits 1
 speed 115200
 flowcontrol hardware
line aux 0
line vty 0 4
 password lab
!
```

DHCP Configured on Primary Interface and PPP Async As Backup

The following example shows that a Cisco 1711 router has been configured so that DHCP is configured on the primary interface and PPP asynchronous mode is configured as the backup:

```
Router# show running-config

Building configuration...

Current configuration : 3427 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ph4_R5
!
boot-start-marker
boot-end-marker
!
no logging buffered
!
username ph4_R8 password 0 cisco
```

```
username ph4_R7 password 0 lab
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
aaa new-model
!
!
aaa session-id common
ip subnet-zero
!
!
no ip domain lookup
ip cef
ip ids po max-events 100
ip dhcp-client default-router distance 1
no ftp-server write-enable
!
!
track 123 rtr 3 reachability
!
crypto isakmp keepalive 10 periodic
!
!
crypto ipsec client ezvpn backup_profile_vpn3k
 connect auto
 group hw_client_groupname key password123
 mode client
 peer 10.0.0.5
 username rchu password  password123
crypto ipsec client ezvpn hw_client_vpn3k
 connect auto
 group hw_client_groupname key password123
 backup backup_profile_vpn3k track 123
 mode client
 peer 10.0.0.5
 username rchu password  password123
!
!
interface Loopback0
 ip address 10.40.40.50 255.255.255.255
!
interface Loopback1
 ip address 10.40.40.51 255.255.255.255
!
interface Loopback2
 no ip address
!
interface FastEthernet0
 description Primary Link to 10.0.0.2
 ip dhcp client route track 123
 ip address dhcp
 duplex auto
 speed auto
 no cdp enable
 crypto ipsec client ezvpn hw_client_vpn3k
!
interface FastEthernet1
 no ip address
 duplex full
 speed 100
 no cdp enable
!
interface FastEthernet2
```

```
 no ip address
 no cdp enable
!
interface FastEthernet3
 no ip address
 no cdp enable
!
interface FastEthernet4
 no ip address
 no cdp enable
!
interface Vlan1
 ip address 10.0.0.1 255.255.255.0
 crypto ipsec client ezvpn backup_profile_vpn3k inside
 crypto ipsec client ezvpn hw_client_vpn3k inside
!
interface Async1
 description Backup Link
 no ip address
 ip nat outside
 ip virtual-reassembly
 encapsulation ppp
 no ip route-cache cef
 dialer in-band
 dialer pool-member 1
 dialer-group 1
 async default routing
 async mode dedicated
!
interface Dialer1
 ip address 10.0.0.3 255.255.255.0
 encapsulation ppp
 no ip route-cache cef
 dialer pool 1
 dialer idle-timeout 60
 dialer string 102
 dialer hold-queue 100
 dialer-group 1
 crypto ipsec client ezvpn backup_profile_vpn3k
!
ip local policy route-map policy_for_rtr
ip classless
ip route 0.0.0.0 0.0.0.0 Dialer1 240
no ip http server
no ip http secure-server
!
!
ip access-list extended dummy1
 permit ip host 10.10.0.2 host 10.0.0.1
ip access-list extended important_traffic
 permit ip 10.0.0.0 0.0.0.255 10.0.0.2 0.0.0.255
 permit ip 10.0.0.0 0.0.0.255 10.0.0.3 0.0.0.255
ip access-list extended important_traffic_2
 permit ip 10.0.0.0 0.0.0.255 10.0.0.3 0.0.0.255
access-list 112 permit icmp any host 10.0.0.2 echo
dialer-list 1 protocol ip permit
no cdp run
!
route-map policy_for_rtr permit 10
 match ip address 112
 set interface Null0
 set ip next-hop 10.0.0.2
!
!
```

```
control-plane
!
rtr 2
 type echo protocol ipIcmpEcho 10.0.0.2 source-ipaddr 10.0.0.3
 timeout 10000
 threshold 1000
 frequency 11
rtr schedule 2 life forever start-time now
rtr 3
 type echo protocol ipIcmpEcho 10.0.0.2 source-interface FastEthernet0
 timeout 10000
 threshold 1000
 frequency 11
rtr schedule 3 life forever start-time now
!
line con 0
 exec-timeout 0 0
line 1
 modem InOut
 modem autoconfigure discovery
 transport input all
 autoselect ppp
 stopbits 1
 speed 115200
 flowcontrol hardware
line aux 0
line vty 0 4
 password lab
!
```

Web-Based Activation: Example

The following example shows that HTTP connections from the user are to be intercepted and that the user can do web-based authentication (192.0.0.13 is the VPN client device and 192.0.0.1 is the server device):

```
crypto ipsec client ezvpn tunnel22
  connect manual
  group tunnel22 key 22tunnel
  mode client
  peer 192.168.0.1
  xauth userid mode http-intercept
!
!
interface Ethernet0
  ip address 10.4.23.15 255.0.0.0
  crypto ipsec client ezvpn tunnel22 inside!
interface Ethernet1
  ip address 192.168.0.13 255.255.255.128
  duplex auto
  crypto ipsec client ezvpn tunnel22
!
```

Easy VPN Remote with Virtual IPsec Interface Support Configuration: Examples

The following examples indicate that Virtual IPsec Interface Support has been configured on the Easy VPN remote devices.

Virtual IPsec Interface: Generic Virtual Access

The following example shows an Easy VPN remote device with virtual-interface support using a generic virtual-access IPsec interface.

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
clock timezone IST 0
ip subnet-zero
no ip dhcp use vrf connected
!
!
crypto ipsec client ezvpn ez
 connect manual
 group easy key cisco
 mode client
 peer 10.3.0.2
 virtual-interface
 xauth userid mode interactive
!
!
interface Ethernet0/0
 ip address 10.1.0.2 255.255.255.0
 no keepalive
 no cdp enable
 crypto ipsec client ezvpn ez inside
!
interface Ethernet1/0
 ip address 10.2.0.1 255.255.255.0
 no keepalive
 no cdp enable
 crypto ipsec client ezvpn ez
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.2.0.2 2
no ip http server
no ip http secure-server
!
!
line con 0
line aux 0
line vty 0 4
 login
!
end
```

Virtual IPsec Interface: Virtual Access Derived from Virtual Template

The following example shows an Easy VPN remote device with virtual-interface support using a virtual-template–derived virtual-access IPsec interface:

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
clock timezone IST 0
ip subnet-zero
no ip dhcp use vrf connected
!
!
crypto ipsec client ezvpn ez
 connect manual
 group easy key cisco
 mode client
 peer 10.3.0.2
 virtual-interface 1
 xauth userid mode interactive
!
!
interface Ethernet0/0
 ip address 10.1.0.2 255.255.255.0
 no keepalive
 no cdp enable
 crypto ipsec client ezvpn ez inside
!
interface Ethernet1/0
 ip address 10.2.0.1 255.255.255.0
 no keepalive
 no cdp enable
 crypto ipsec client ezvpn ez
!
interface Virtual-Template1 type tunnel
 no ip address
 tunnel mode ipsec ipv4
!
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.2.0.2 2
no ip http server
no ip http secure-server
!
!
line con 0
line aux 0
line vty 0 4
 login
!
end
```

When the Tunnel Is Down

The result of a virtual-interface configuration on an Easy VPN profile is the creation of a virtual-access interface. This interface provides IPsec encapsulation. The output below shows the configuration of a virtual-access interface when Easy VPN is "down."

```
Router# show running-config interface virtual-access 2

Building configuration...

Current configuration : 99 bytes
!
interface Virtual-Access2
 no ip address
 tunnel source Ethernet1/0
 tunnel mode ipsec ipv4
end
```

A virtual-interface configuration results in the creation of a virtual-access interface. This virtual-access interface is made automatically outside the interface of the Easy VPN profile. The routes that are added later when the Easy VPN tunnels come up point to this virtual interface for sending the packets to the corporate network. If **crypto ipsec client ezvpn name outside** (**crypto ipsec client ezvpn name** command and **outside** keyword) is applied on a real interface, that interface is used as the IKE (IPsec) endpoint (that is, IKE and IPsec packets use the address on the interface as the source address).

```
Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 5

Tunnel name : ez
Inside interface list: Ethernet0/0
Outside interface: Virtual-Access2 (bound to Ethernet1/0)
Current State: CONNECT_REQUIRED
Last Event: TRACKED OBJECT UP
Save Password: Disallowed
Current EzVPN Peer: 10.3.0.2
```

Because a virtual interface, or for that matter any interface, is routable, routes act like traffic selectors. When the Easy VPN tunnel is "down," there are no routes pointing to the virtual interface, as shown in the following example:

```
Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.2.0.2 to network 0.0.0.0

     10.0.0.0/24 is subnetted, 2 subnets
C       10.2.0.0 is directly connected, Ethernet1/0
C       10.1.0.0 is directly connected, Ethernet0/0
S*   0.0.0.0/0 [2/0] via 10.2.0.2
```

When the Tunnel Is Up

In the case of client or network plus mode, Easy VPN creates a loopback interface and assigns the address that is pushed in mode configuration. To assign the address of the loopback to the interface, use the **ip unnumbered** command (**ip unnumbered loopback**). In the case of network extension mode, the virtual access will be configured as **ip unnumbered ethernet0** (the bound interface).

```
Router# show running-config interface virtual-access 2

Building configuration...

Current configuration : 138 bytes
!
interface Virtual-Access2
 ip unnumbered Loopback0
 tunnel source Ethernet1/0
 tunnel destination 10.3.0.2
 tunnel mode ipsec ipv4
end

Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 5

Tunnel name : ez
Inside interface list: Ethernet0/0
Outside interface: Virtual-Access2 (bound to Ethernet1/0)
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 10.5.0.2
Mask: 255.255.255.255
DNS Primary: 10.6.0.2
NBMS/WINS Primary: 10.7.0.1
Default Domain: cisco.com
Using PFS Group: 2
Save Password: Disallowed
Split Tunnel List: 1
       Address    : 10.4.0.0
       Mask       : 255.255.255.0
       Protocol   : 0x0
       Source Port: 0
       Dest Port  : 0
Current EzVPN Peer: 10.3.0.2
```

When the tunnels come up, Easy VPN adds either a default route that points to the virtual-access interface or adds routes for all the split attributes of the subnets that point to the virtual-access interface. Easy VPN also adds a route to the peer (destination or concentrator) if the peer is not directly connected to the Easy VPN device.

The following **show ip route** command output examples are for virtual IPsec interface situations in which a split tunnel attribute was sent by the server and a split tunnel attribute was not sent, respectively.

Split Tunnel Attribute Has Been Sent by the Server

```
Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
```

```
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.2.0.2 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C       10.2.0.0/24 is directly connected, Ethernet1/0
S       10.3.0.2/32 [1/0] via 10.2.0.2, Ethernet1/0  <<< Route to
 peer (EzVPN server)
C       10.1.0.0/24 is directly connected, Ethernet0/0
C       10.5.0.2/32 is directly connected, Loopback0
S       10.4.0.0/24 [1/0] via 0.0.0.0, Virtual-Access2  <<< Split
tunnel attr sent by the server
S*   10.0.0.0/0 [2/0] via 10.2.0.2
```

Split Tunnel Attribute Has Not Been Sent by the Server

All networks in the split attribute should be shown, as in the following example:

```
Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.2.0.0/24 is directly connected, Ethernet1/0
! The following line is the route to the peer (the Easy VPN server).
S       10.3.0.2/32 [1/0] via 10.2.0.2, Ethernet1/0
C       10.1.0.0/24 is directly connected, Ethernet0/0
C       10.5.0.3/32 is directly connected, Loopback0
! The following line is the default route.
S*   10.0.0.0/0 [1/0] via 10.0.0.0, Virtual-Access2
```

Dual Tunnel Configuration: Example

The following is an example of a typical dual-tunnel configuration:

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
enable password lab
!
no aaa new-model
!
resource policy
!
clock timezone IST 0
ip subnet-zero
!
!
```

```
username lab password 0 lab
!
!
crypto ipsec client ezvpn ezvpn1
 connect manual
 group easy key cisco
 mode network-extension
 peer 10.75.1.2
 virtual-interface 1
 xauth userid mode interactive
crypto ipsec client ezvpn ezvpn2
 connect manual
 group easy key cisco
 mode network-extension
 peer 10.75.2.2
 virtual-interface 1
 xauth userid mode interactive
!
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.255
 no keepalive
 crypto ipsec client ezvpn ezvpn1 inside
 crypto ipsec client ezvpn ezvpn2 inside
!
interface Ethernet0/1
 no ip address
 shutdown
!
interface Ethernet0/2
 no ip address
 shutdown
!
interface Ethernet0/3
 no ip address
 shutdown
!
interface Ethernet1/0
 ip address 10.76.1.2 255.255.255.0
 no keepalive
 crypto ipsec client ezvpn ezvpn1
 crypto ipsec client ezvpn ezvpn2
!
interface Serial2/0
 ip address 10.76.2.2 255.255.255.0
 no keepalive
 serial restart-delay 0
!
interface Virtual-Template1 type tunnel
 no ip address
 tunnel mode ipsec ipv4
!
!
ip classless
ip route 10.0.0.0 10.0.0.0 10.76.1.1 2
no ip http server
no ip http secure-server
!
!
no cdp run
!
!
line con 0
 exec-timeout 0 0
```

```
line aux 0
line vty 0 4
 login local
!
end
```

Dual Tunnel Show Output: Examples

The following **show** command examples display information about three phases of a dual tunnel that is coming up:

- First Easy VPN tunnel is up
- Second Easy VPN tunnel is initiated
- Both of the Easy VPN tunnels are up

Before the EzVPN Tunnels Are Up

```
Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 6

Tunnel name : ezvpn1
Inside interface list: Ethernet0/0
Outside interface: Virtual-Access2 (bound to Ethernet1/0)
Current State: CONNECT_REQUIRED
Last Event: TRACKED OBJECT UP
Save Password: Disallowed
Current EzVPN Peer: 10.75.1.2


Tunnel name : ezvpn2
Inside interface list: Ethernet0/0
Outside interface: Virtual-Access3 (bound to Serial2/0)
Current State: CONNECT_REQUIRED
Last Event: TRACKED OBJECT UP
Save Password: Disallowed
Current EzVPN Peer: 10.75.2.2

Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
```

The gateway of last resort is 10.76.1.1 to network 0.0.0.0.

```
10.0.0.0/24 is subnetted, 2 subnets
C       10.76.2.0 is directly connected, Serial2/0
C       10.76.1.0 is directly connected, Ethernet1/0
C    192.168.1.0/24 is directly connected, Ethernet0/0
S*   0.0.0.0/0 [2/0] via 10.76.1.1
```

Note The metric of the default route should be greater than 1 so that the default route that is added later by Easy VPN takes precedence and the traffic goes through the Easy VPN virtual-access interface.

Easy VPN "ezvpn2" Tunnel Is Up

```
Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 6

Tunnel name : ezvpn1
Inside interface list: Ethernet0/0
Outside interface: Virtual-Access2 (bound to Ethernet1/0)
Current State: CONNECT_REQUIRED
Last Event: TRACKED OBJECT UP
Save Password: Disallowed
Current EzVPN Peer: 10.75.1.2


Tunnel name : ezvpn2
Inside interface list: Ethernet0/0
Outside interface: Virtual-Access3 (bound to Serial2/0)
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
DNS Ezvpn1: 10.6.0.2
NBMS/WINS Ezvpn1: 10.7.0.1
Default Domain: cisco.com
Save Password: Disallowed
Current EzVPN Peer: 10.75.2.2

Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
```

The gateway of last resort is 0.0.0.0 to network 0.0.0.0.

```
10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
! The next line is the Easy VPN route.
S       10.75.2.2/32 [1/0] via 10.76.1.1
C       10.76.2.0/24 is directly connected, Serial2/0
C       10.76.1.0/24 is directly connected, Ethernet1/0
C    192.168.1.0/24 is directly connected, Ethernet0/0
! The next line is the Easy VPN route.
S*   0.0.0.0/0 [1/0] via 0.0.0.0, Virtual-Access3
```

One default route and one route to the peer is added as shown above.

Easy VPN "ezvpn2" Is Up and Easy VPN "ezvpn1" Is Initiated

```
Router# crypto ipsec client ezvpn connect ezvpn1

Router# show crypto ipsec cli ent ezvpn

Easy VPN Remote Phase: 6

Tunnel name : ezvpn1
Inside interface list: Ethernet0/0
Outside interface: Virtual-Access2 (bound to Ethernet1/0)
Current State: READY
```

```
Last Event: CONNECT
Save Password: Disallowed
Current EzVPN Peer: 10.75.1.2


Tunnel name : ezvpn2
Inside interface list: Ethernet0/0
Outside interface: Virtual-Access3 (bound to Serial2/0)
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
DNS Ezvpn1: 10.6.0.2
NBMS/WINS Ezvpn1: 10.7.0.1
Default Domain: cisco.com
Save Password: Disallowed
Current EzVPN Peer: 10.75.2.2

Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
```

The gateway of last resort is 10.0.0.0 to network 10.0.0.0.

```
     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
S       10.75.2.2/32 [1/0] via 10.76.1.1
! The next line is the Easy VPN router.
S       10.75.1.2/32 [1/0] via 10.76.1.1
C       10.76.2.0/24 is directly connected, Serial2/0
C       10.76.1.0/24 is directly connected, Ethernet1/0
C    192.168.1.0/24 is directly connected, Ethernet0/0
S*   10.0.0.0/0 [1/0] via 10.0.0.0, Virtual-Access3
```

The route to 10.75.1.2 is added before the Easy VPN “ezvpn1” tunnel has come up. This route is for reaching the Easy VPN “ezvpn1” peer 10.75.1.2.

Both Tunnels Are Up

```
Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 6

Tunnel name : ezvpn1
Inside interface list: Ethernet0/0
Outside interface: Virtual-Access2 (bound to Ethernet1/0)
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
DNS Ezvpn1: 10.6.0.2
NBMS/WINS Ezvpn1: 10.7.0.1
Default Domain: cisco.com
Save Password: Disallowed
Split Tunnel List: 1
       Address    : 192.168.3.0
       Mask       : 255.255.255.255
       Protocol   : 0x0
       Source Port: 0
       Dest Port  : 0
Current EzVPN Peer: 10.75.1.2
```

```
Tunnel name : ezvpn2
Inside interface list: Ethernet0/0
Outside interface: Virtual-Access3 (bound to Serial2/0)
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
DNS Ezvpn1: 10.6.0.2
NBMS/WINS Ezvpn1: 10.7.0.1
Default Domain: cisco.com
Save Password: Disallowed
Current EzVPN Peer: 10.75.2.2

Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
```

The gateway of last resort is 10.0.0.0 to network 10.0.0.0.

```
     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
! The next line is the Easy VPN router (ezvpn2).
S       10.75.2.2/32 [1/0] via 10.76.1.1
! The next line is the Easy VPN router (ezvpn1).
S       10.75.1.2/32 [1/0] via 10.76.1.1
C       10.76.2.0/24 is directly connected, Serial2/0
C       10.76.1.0/24 is directly connected, Ethernet1/0
C    192.168.1.0/24 is directly connected, Ethernet0/0
! The next line is the Easy VPN route (ezvpn1).
S    192.168.3.0/24 [1/0] via 0.0.0.0, Virtual-Access2
! The next line is the Easy VPN (ezvpn2).
S*   10.0.0.0/0 [1/0] via 10.0.0.0, Virtual-Access3
```

The route to split tunnel “192.168.3.0/24” that points to Virtual-Access2 is added for the Easy VPN “ezvpn”" tunnel as shown in the above **show** output.

Reactivate Primary Peer: Example

The following show output illustrates that the default primary peer feature has been activated. The primary default peer is 10.3.3.2.

```
Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 6

Tunnel name : ezc
Inside interface list: Loopback0
Outside interface: Ethernet0/0
Current State: IPSEC_ACTIVE
Primary EzVPN Peer: 10.3.3.2, Last Tried: Dec 30 07:21:23.071
Last Event: CONN_UP
Address: 10.7.7.1
Mask: 255.255.255.255
DNS Primary: 10.1.1.1
NBMS/WINS Primary: 10.5.254.22
```

```
Save Password: Disallowed
Current EzVPN Peer: 10.4.4.2

23:52:44: %CRYPTO-6-EZVPN_CONNECTION_UP(Primary peer):
        User: lab, Group: hw-client-g
        Client_public_addr=10.4.22.103, Server_public_addr=10.4.23.112
        Assigned_client_addr=10.7.7.1
```

Identical Addressing Support Configuration: Example

In the following example, a Cisco router is configured for the Identical Addressing Support feature:

```
interface Virtual-Template1 type tunnel
     no ip address
     ip nat outside
!
crypto ipsec client ezvpn easy
 connect manual
 group easy key work4cisco
 mode network-extension
 peer 10.2.2.2
 virtual-interface 1
 nat allow
 nat acl 100
!
interface Ethernet1
 ip address 10.0.0.1 255.255.255.0
 ip nat outside
 crypto ipsec client ezvpn easy
!
interface Ethernet0
 ip address 10.0.0.2 255.255.255.0
 ip nat inside
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.252
 ip nat enable
crypto ipsec client ezvpn easy inside
!
ip access-list 100 permit ip 10.0.0.0 0.0.0.255 any
!
ip nat inside source list 100 interface Loopback0 overload
!
ip nat inside source static 10.5.5.5 1.1.1.101
```

cTCP on an Easy VPN Client (Remote Device): Examples

For configuration and troubleshooting examples, see the topic “cTCP on Cisco Easy VPN remote devices” in the “Related Documents” section on page 102.

Easy VPN Server Configuration Examples

This section describes basic Cisco Easy VPN server configurations that support the Cisco Easy VPN remote configurations given in the previous sections. For complete information on configuring these servers, see *Easy VPN Server* for Cisco IOS Release 12.3(7)T, available on Cisco.com.

Cisco Easy VPN Server Without Split Tunneling: Example

The following example shows the Cisco Easy VPN server that is the destination peer router for the Cisco Easy VPN remote network extension mode configurations shown earlier in this section. In addition to the other IPsec configuration commands, the **crypto isakmp client configuration group** command defines the attributes for the VPN group that was assigned to the Easy VPN remote router. This includes a matching key value (easy vpn remote password), and the appropriate routing parameters, such as DNS server, for the Easy VPN remotes.

To support the network extension mode of operation, the **ip route** command instructs that incoming packets for the 172.168.0.0 network be directed from the cable modem interface to the Cisco Easy VPN remote. Other **ip route** commands might be needed, depending on the topology of your network.

Note This example shows a Cisco uBR925 cable access router, but typically the destination Easy VPN remote is a router, such as a Cisco VPN 3000 concentrator or a Cisco IOS router, that supports the Easy VPN Server feature.

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname uBR925Server
!
aaa new-model
!
!
aaa authorization network easy vpn remote-groupname local
aaa session-id common
!
!
clock timezone - 0 6
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
crypto isakmp policy 1
 authentication pre-share
 group 2
crypto isakmp client configuration address-pool local dynpool
!
crypto isakmp client configuration group easy vpn remote-groupname
 key easy vpn remote-password
 dns 172.16.0.250 172.16.0.251
 wins 172.16.0.252 172.16.0.253
 domain cisco.com
 pool dynpool
!
```

```
!
crypto ipsec transform-set transform-1 esp-des esp-sha-hmac
!
crypto dynamic-map dynmap 1
 set transform-set transform-1
 reverse-route
!
!
crypto map dynmap isakmp authorization list easy vpn remote-groupname
crypto map dynmap client configuration address respond
crypto map dynmap 1 ipsec-isakmp dynamic dynmap
!
!
interface Ethernet0
 ip address 172.16.0.129 255.255.255.128
!
interface cable-modem0
 no cable-modem compliant bridge
 crypto map dynmap
!
interface usb0
 no ip address
 arp timeout 0
!
ip local pool dynpool 172.16.0.65 172.16.0.127
ip classless
! Add the appropriate ip route commands for network-extension mode
ip route 172.16.1.0 255.255.255.248 cable-modem0
no ip http server
no ip http cable-monitor
!
snmp-server manager
!
line con 0
 exec-timeout 0 0
line vty 0 4
!
scheduler max-task-time 5000
```

Cisco Easy VPN Server Configuration with Split Tunneling: Example

The following example shows a Cisco Easy VPN server that is configured for a split tunneling configuration with a Cisco Easy VPN remote. This example is identical to that shown in the "Cisco Easy VPN Server Without Split Tunneling: Example" except for access list 150, which is assigned as part of the **crypto isakmp client configuration group** command. This access list allows the Cisco Easy VPN remote to use the server to access one additional subnet that is not part of the VPN tunnel without compromising the security of the IPsec connection.

To support network extension mode, the **ip route** command instructs that incoming packets for the 172.168.0.0 network be directed from the cable modem interface to the Cisco Easy VPN remote. Other **ip route** commands might be necessary, depending on the topology of your network.

Note This example shows a Cisco uBR925 cable access router, but typically the destination Easy VPN remote will be a router, such as a VPN 3000 concentrator or a Cisco IOS router, that supports the Easy VPN Server feature.

```
version 12.2
no service pad
service timestamps debug uptime
```

```
service timestamps log uptime
no service password-encryption
service internal
!
hostname uBR925Server
!
aaa new-model
!
!
aaa authorization network easy vpn remote-groupname local
aaa session-id common
!
!
clock timezone - 0 6
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
crypto isakmp policy 1
 authentication pre-share
 group 2
crypto isakmp client configuration address-pool local dynpool
!
crypto isakmp client configuration group easy vpn remote-groupname
 key easy vpn remote-password
 dns 172.16.0.250 172.16.0.251
 wins 172.16.0.252 172.16.0.253
 domain cisco.com
 pool dynpool
 acl 150
!
!
crypto ipsec transform-set transform-1 esp-des esp-sha-hmac
!
crypto dynamic-map dynmap 1
 set transform-set transform-1
 reverse-route
!
!
crypto map dynmap isakmp authorization list easy vpn remote-groupname
crypto map dynmap client configuration address respond
crypto map dynmap 1 ipsec-isakmp dynamic dynmap
!
!
interface Ethernet0
 ip address 172.16.0.129 255.255.255.255
!
interface cable-modem0
 no cable-modem compliant bridge
 crypto map dynmap
!
interface usb0
 no ip address
 arp timeout 0
!
ip local pool dynpool 172.16.0.65 172.16.0.127
ip classless
! Add the appropriate ip route commands for network-extension mode
ip route 172.16.1.0 255.255.255.255 cable-modem0
no ip http server
no ip http cable-monitor
!
access-list 150 permit ip 172.16.0.128 10.0.0.127 any
```

```
snmp-server manager
!
line con 0
 exec-timeout 0 0
line vty 0 4
!
scheduler max-task-time 5000
end
```

Cisco Easy VPN Server Configuration with Xauth: Example

The following example shows a Cisco Easy VPN server configured to support Xauth with the Cisco Easy VPN Remote feature. This example is identical to that shown in the "Cisco Easy VPN Server Configuration with Split Tunneling: Example" except for the following commands that enable and configure Xauth:

- **aaa authentication login userlist local**—Specifies the local username database for authentication at login time. You could also specify the use of RADIUS servers by first using the **aaa authentication login userlist group radius** command and then by specifying the RADIUS servers using the **aaa group server radius** command.
- **crypto isakmp xauth timeout**—Specifies the amount of time, in seconds, that the user has to enter the appropriate username and password to authenticate the session.
- **crypto map dynmap client authentication list userlist**—Creates a crypto map named "**dynmap"** that enables Xauth.
- **username cisco password 7 cisco**—Creates an entry in the local username database for a user with the username of "**cisco"** and an encrypted password of "**cisco."** This command should be repeated for each separate user that accesses the server.

The following commands, which are also present in the non-Xauth configurations, are also required for Xauth use:

- **aaa authorization network easy vpn remote-groupname local**—Requires authorization for all network-related service requests for users in the group named "**easy vpn remote-groupname"** using the local username database.
- **aaa new-model**—Specifies that the router should use the new AAA authentication commands.
- **aaa session-id common**—Specifies that a unique and common session ID should be used for AAA sessions.
- **crypto map dynmap 1 ipsec-isakmp dynamic dynmap**—Specifies that IKE should be used to establish the IPsec SAs, using the crypt map named "dynmap" as the policy template.
- **crypto map dynmap client configuration address respond**—Enables IKE negotiation, accepting requests from any requesting peers.
- **crypto map dynmap isakmp authorization list easy vpn remote-groupname**—Configures the crypto map named "**dynmap"** to use IKE Shared Secret using the group named "**easy vpn remote-groupname**."

Tip This configuration shows the server configured for split tunneling, but Xauth can also be used with nonsplit tunnel configurations as well.

Note This example shows a Cisco uBR925 cable access router, but typically the destination Easy VPN server is a router such as a VPN 3000 concentrator or a Cisco IOS router that supports the Easy VPN Server feature.

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname uBR925Server
!
aaa new-model
!
!
aaa authentication login userlist local
aaa authorization network easy vpn remote-groupname local
aaa session-id common
!
username cisco password 7 cisco
!
!
clock timezone - 0 6
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
crypto isakmp policy 1
 authentication pre-share
 group 2
crypto isakmp client configuration address-pool local dynpool
crypto isakmp xauth timeout 60
!
crypto isakmp client configuration group easy vpn remote-groupname
 key easy vpn remote-password
 dns 172.16.0.250 172.16.0.251
 wins 172.16.0.252 172.16.0.253
 domain cisco.com
 pool dynpool
 acl 150
!
!
crypto ipsec transform-set transform-1 esp-des esp-sha-hmac
!
crypto dynamic-map dynmap 1
 set transform-set transform-1
 reverse-route
!
!
crypto map dynmap client authentication list userlist
crypto map dynmap isakmp authorization list easy vpn remote-groupname
crypto map dynmap client configuration address respond
crypto map dynmap 1 ipsec-isakmp dynamic dynmap
!!
!
interface Ethernet0
 ip address 172.16.0.129 255.255.255.128
!
interface cable-modem0
```

```
 no cable-modem compliant bridge
 crypto map dynmap
!
interface usb0
 no ip address
 arp timeout 0
!
ip local pool dynpool 172.16.0.65 172.16.0.127
ip classless
ip route 172.16.1.0 255.255.255.248 cable-modem0
no ip http server
no ip http cable-monitor
!
access-list 150 permit ip 172.16.0.128 0.0.0.127 any
snmp-server manager
!
line con 0
 exec-timeout 0 0
line vty 0 4
!
scheduler max-task-time 5000
end
```

Easy VPN Server Interoperability Support: Example

For information about this feature, see “General information on IPSec and VPN” in the section “Additional References” (*Managing VPN Remote Access*).

Additional References

The following sections provide references related to Cisco Easy VPN Remote.

Related Documents

Related Topic	Document Title
Platform-specific documentation	
Cisco 800 series routers	• *Cisco 800 Series Routers* • *Cisco 806 Router and SOHO 71 Router Hardware Installation Guide* • *Cisco 806 Router Software Configuration Guide* • *Cisco 826, 827, 828, 831, 836, and 837 and SOHO 76, 77, 78, 91, 96, and 97 Routers Software Configuration Guide* • *Cisco 826 and SOHO 76 Router Hardware Installation Guide* • *Cisco 827 and SOHO 77 Routers Hardware Installation Guide* • *Cisco 828 and SOHO 78 Routers Hardware Installation Guide* • *Cisco 837 ADSL Broadband Router*

Related Topic	Document Title
Cisco uBR905 and Cisco uBR925 cable access routers	• *Cisco uBR925 Cable Access Router Hardware Installation Guide* • *Cisco uBR905 Hardware Installation Guide* • *Cisco uBR905/uBR925 Cable Access Router Software Configuration Guide* • *Cisco uBR905 Cable Access Router Subscriber Setup Quick Start Card* • *Cisco uBR925 Cable Access Router Subscriber Setup Quick Start Card* • *Cisco uBR925 Cable Access Router Quick Start User Guide*
Cisco 1700 series routers	• *Cisco 1700 Series Router Software Configuration Guide* • *Cisco 1710 Security Router Hardware Installation Guide* • *Cisco 1710 Security Router Software Configuration Guide* • *Cisco 1711 Security Access Router* • *Cisco 1720 Series Router Hardware Installation Guide* • *Cisco 1721 Access Router Hardware Installation Guide* • *Cisco 1750 Series Router Hardware Installation Guide* • *Cisco 1751 Router Hardware Installation Guide* • *Cisco 1751 Router Software Configuration Guide* • *Cisco 1760 Modular Access Router Hardware Installation Guide* Also see the Cisco IOS release notes for Cisco IOS Release 12.2(4)YA: • *SOHO 70 and Cisco 800 Series—Release Notes for Release 12.2(4)YA* • *Release Notes for Cisco uBR905 and Cisco uBR925 Cable Access Routers for Cisco IOS Release 12.2 YA* • *Cisco 1700 Series—Release Notes for Release 12.2(4)YA*
Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers	• *Cisco 2600 Series Multiservice Platforms* • *Cisco 2600 Series Routers Hardware Installation Guide* • *Cisco 3600 Series Multiservice Platforms* • *Cisco 3600 Series Hardware Installation Guide* • *Cisco 3700 Series Multiservice Access Routers* • *Cisco 3700 Series Routers Hardware Installation Guide* • *Cisco 2600 Series, 3600 Series, and 3700 Series Regulatory Compliance and Safety Information on Cisco.com*
IPsec and VPN documentation	

Related Topic	Document Title
802.1x authentication	• *Configuring Cisco IOS Easy VPN Remote with 802.1X Authentication* (white paper) • *VPN Access Control Using 802.1X Local Authentication*
Access control lists, configuring	• *Access Control Lists: Overview and Guidelines*
Configuration information (additional in-depth)	• *Cisco Easy VPN Solutions*—Provides white papers and examples for configuring Cisco IOS Easy VPN in network extension mode. • *Cisco IOS Security Command Reference*—Provides a reference for each of the Cisco IOS commands used to configure IPsec encryption and related security features. • *SSL VPN*—Provides information about SSL VPN.
cTCP on Cisco Easy VPN remote devices	• *EFT Deployment Guide for Cisco Tunnel Control Protocol on Cisco EasyVPN*
Dead peer detection	• *IPSec Dead Peer Detection Periodic Message Option*
DHCP, configuring	• *Configuring DHCP* • "Configuring the Cisco IOS DHCP Client" in the *Cisco IOS IP Configuration Guide*
Digital certificates (RSA signature support)	• *Easy VPN Remote RSA Signature Support*
DNS, configuring	• *Configuring DNS* and *Configuring DNS on Cisco Routers*
Easy VPN Server feature, which provides Cisco Unity client support for the Cisco Easy VPN Remote feature	• *Easy VPN Server* • *Cisco Easy VPN* • *Configuring NAC with IPsec Dynamic Virtual Tunnel Interface*
Encrypted Preshared Key feature	• *Encrypted Preshared Key*

Related Topic	Document Title
IPsec and VPN, general information	• *Deploying IPsec*—Provides an overview of IPsec encryption and its key concepts, along with sample configurations. Also provides a link to many other documents on related topics. • *Configuring Authorization and Revocation of Certificates in a PKI*—Describes the concept of digital certificates and how they are used to authenticate IPsec users. • *Configuring Authentication Proxy* • *An Introduction to IP Security (IPsec) Encryption*—Provides a step-by-step description of how to configure IPsec encryption. • *Managing VPN Remote Access*—Describes how to configure the Cisco PIX firewall as an Easy VPN server and how to configure Easy VPN remote software clients. • *Configuring VPN Settings*—Provides information about configuring a PIX firewall to operate as a Cisco Secure VPN client. • *Configuring Security for VPNs with IPSec*—Provides information about configuring crypto maps. • *IPSec Virtual Tunnel Interface*—Provides information about IPsec virtual tunnel interfaces. • IP technical tips sections on Cisco.com.
Object tracking	• *Reliable Static Routing Backup Using Object Tracking*

Note Additional documentation on IPsec becomes available on Cisco.com as new features and platforms are added. Cisco Press also publishes several books on IPsec—go to http://www.ciscopress.com for more information on Cisco Press books.

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
• CISCO-IPSEC-FLOW-MONITOR-MIB—Contains attributes describing IPsec-based VPNs (Internet Engineering Task Force (IETF) IPSec Working Group Draft). • CISCO-IPSEC-MIB—Describes Cisco implementation-specific attributes for Cisco routers implementing IPsec VPNs. • CISCO-IPSEC-POLICY-MAP-MIB—Extends the CISCO-IPSEC-FLOW-MONITOR-MIB to map dynamically created structures to the policies, transforms, cryptomaps, and other structures that created or are using them.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html
For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **ctcp port**
- **clear crypto ipsec client ezvpn**
- **crypto ctcp**
- **crypto ipsec client ezvpn (global)**
- **crypto ipsec client ezvpn (interface)**
- **crypto ipsec client ezvpn connect**
- **crypto ipsec client ezvpn xauth**
- **debug crypto ipsec client ezvpn**
- **debug ip auth-proxy ezvpn**
- **icmp-echo**
- **ip http ezvpn**
- **show crypto ipsec client ezvpn**
- **show tech-support**
- **type echo protocol ipIcmpEcho**
- **xauth userid mode**

Feature Information for Easy VPN Remote

Table 4 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 4 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 4 Feature Information for Easy VPN Remote

Feature Name	Releases	Feature Information
Easy VPN Remote	12.2(4)YA Cisco IOS XE Release 2.1	Support for Cisco Easy VPN Remote (Phase I) of this feature was introduced for Cisco 806, Cisco 826, Cisco 827, and Cisco 828 routers; Cisco 1700 series routers; and Cisco uBR905 and Cisco uBR925 cable access routers. In Cisco IOS XE Release 2.1, support for this feature was introduced on Cisco ASR 1000 Series Routers.
	12.2(13)T	Cisco Easy VPN Remote was integrated into Cisco IOS Release 12.2(13)T.
	12.2(8)YJ	Support for Cisco Easy VPN Remote (Phase II) of this feature was introduced for Cisco 806, Cisco 826, Cisco 827, and Cisco 828 routers; Cisco 1700 series routers; and Cisco uBR905 and Cisco uBR925 cable access routers.
	12.2(15)T	The Cisco Easy VPN Remote (Phase II) feature was integrated into Cisco IOS Release 12.2(15)T. Support for the Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers was added.
	12.3(2)T	The Type 6 Password in the IOS Configuration feature was added.
	12.3(4)T	The Save Password and Multiple Peer Backup features were added. The following sections provide information about the Save Password feature: • Using Xauth, page 9 • Configuring Save Password, page 39 • Save Password Configuration: Example, page 78

Table 4 Feature Information for Easy VPN Remote (continued)

Feature Name	Releases	Feature Information
	12.3(7)T	The following feature was introduced in this release: • Dead Peer Detection Periodic Message Option, page 24
	12.3(7)XR	The following features were introduced: Dead Peer Detection with Stateless Failover (Object Tracking with Easy VPN)—Backup Server List Local Configuration and Backup Server List Auto Configuration, Management Enhancements, Load Balancing, VLAN Support, Multiple Subnet Support, Traffic-Triggered Activation, Perfect Forward Secrecy (PFS) Via Policy Push, 802.1x Authentication, Certificate (PKI) Support, Easy VPN Remote and Server on the Same Interface, and Easy VPN Remote and Site to Site on the Same Interface. The following sections provide information about these features: • 802.1x Authentication, page 16 • Traffic-Triggered Activation, page 17 • Backup Server List Local Configuration, page 18 • Backup Server List Auto Configuration, page 18 • VLAN Support, page 21 • Easy VPN Remote and Server on the Same Interface, page 23 • Easy VPN Remote and Site to Site on the Same Interface, page 23 • Load Balancing, page 24 • Management Enhancements, page 25 • PFS Support, page 25 **Note** Cisco 800 series routers are not supported in Cisco IOS Release 12.3(7)XR. **Note** These features are available only in Cisco Release 12.3(7)XR2.
	12.3(7)XR2	The features in Cisco IOS Release 12.3(7)XR were introduced on Cisco 800 series routers.
	12.3(8)YH	The Dial Backup, Traffic-Triggered Activation, and Web-Based Activation features were introduced on the Cisco 1812 router. The following sections provide information about these features: • Dial Backup, page 25 • Dial Backup: Examples, page 79

Table 4 Feature Information for Easy VPN Remote (continued)

Feature Name	Releases	Feature Information
	12.3(11)T	Except for the Dial Backup and Traffic-Triggered Activation features, all features introduced in Cisco IOS Releases 12.3(7)XR and 12.3(7)XR2 were integrated into Cisco IOS Release 12.3(11)T.
	12.3(14)T	Dial Backup and Traffic-Triggered Activation features were integrated into Cisco IOS Release 12.3(14)T. In addition, the Web-Based Activation feature was integrated into this release.
	12.3(8)YI	The Dial Backup, Traffic-Triggered Activation, and Web-Based Activation features were introduced on the Cisco 1800 series fixed configuration routers.
	12.3(8)YI1	The Dial Backup, Traffic-Triggered Activation, and Web-Based Activation features were introduced on the Cisco 870 series routers.
	12.4(2)T 12.2(33)SXH	The following features were added in this release: Banner, Auto-Update, and Browser-Proxy Enhancements. The following section provides information about these features: • Banner, page 32
	12.4(4)T 12.2(33)SXH	The following features were added in this release: Dual Tunnel Support, Configuration Management Enhancements (Pushing a Configuration URL Through a Mode-Configuration Exchange), Reactivate Primary Peer, and Virtual IPsec Interface Support. In addition, the **flow allow acl** subcommand was added so that traffic can be blocked when a tunnel is down. The following sections provide information about these features: • Virtual IPsec Interface Support, page 27 • Dual Tunnel Support, page 29 • Configuration Management Enhancements (Pushing a Configuration URL Through a Mode-Configuration Exchange), page 33 • Reactivate Primary Peer, page 33 • Restricting Traffic When a Tunnel Is Down, page 57
	12.2(33)SRA	Cisco Easy VPN Remote was integrated into Cisco IOS Release 12.2(33)SRA.
	12.4(11)T	The following feature was added in this release: • Identical Addressing Support The following section provides information about this feature: – Identical Addressing Support, page 33

Table 4 Feature Information for Easy VPN Remote (continued)

Feature Name	Releases	Feature Information
	12.4(20)T	The following features were added in this release: • cTCP Support on Easy VPN Clients The following sections provide information about this feature: – cTCP Support on Easy VPN Clients, page 34 – Configuring cTCP on an Easy VPN Client, page 56 – cTCP on an Easy VPN Client (Remote Device): Examples, page 96 The following commands were introduced or modified for this feature: **crypto ctcp**, **ctcp port**

Glossary

AAA—authentication, authorization, and accounting. Framework of security services that provide the method for identifying users (authentication); for remote access control (authorization); and for collecting and sending security server information used for billing, auditing, and reporting (accounting).

aggressive mode—Mode that eliminates several steps during Internet Key Exchange (IKE) authentication negotiation between two or more IPsec peers. Aggressive mode is faster than main mode but is not as secure.

authorization—Method for remote access control, including one-time authorization or authorization for each service; per-user account list and profile; user group support; and support of IP, IPX, ARA, and Telnet. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared to the information contained in a database for a given user and the result is returned to AAA to determine the actual capabilities and restrictions of the user. The database can be located locally on the access server or router, or it can be hosted remotely on a RADIUS or TACACS+ security server. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. All authorization methods must be defined through AAA.

CA—certificate authority. An entity in a network that issues and manages security credentials and public keys (in the form of X509v3 certificates) for message encryption. As part of a public key infrastructure (PKI), a CA checks with a registration authority (RA) to verify information provided by the requestor of a digital certificate. If the RA verifies the information of the requestor, the CA can then issue a certificate. Certificates generally include the public key of the owner, the expiration date of the certificate, the name of the owner, and other information about the public key owner.

CRWS—Cisco Router Web Setup Tool. Tool that provides web interface capabilities.

cTCP—Cisco Tunneling Control Protocol. When cTCP is enabled on a remote device (client) and headend device, IKE and ESP (Protocol 50) traffic is encapsulated in the TCP header so that the firewalls in between the client and the headend device permits this traffic (considering it the same as TCP traffic).

DPD—dead peer detection. Queries the liveliness of the Internet Key Exchange (IKE) peer of a router at regular intervals.

DSLAM—digital subscriber line access multiplexer. A device that connects many digital subscriber lines to a network by multiplexing the DSL traffic onto one or more network trunk lines.

IKE—Internet Key Exchange. Key management protocol standard that is used in conjunction with the IP Security (IPsec) standard. IPsec is an IP security feature that provides robust authentication and encryption of IP packets. IPsec can be configured without IKE, but IKE enhances IPsec by providing additional features, flexibility, and ease of configuration for the IPsec standard. IKE is a hybrid protocol that implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.

IPsec—IP Security Protocol. Framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPsec provides these security services at the IP layer. IPsec uses IKE to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPsec. IPsec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

main mode—Mode that ensures the highest level of security when two or more IPsec peers are negotiating IKE authentication. It requires more processing time than aggressive mode.

MIB—Management Information Base. Database of network management information that is used and maintained by a network management protocol, such as Simple Network Management Protocol (SNMP) or Common Management Information Protocol (CMIP). The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a graphical user interface (GUI) network management system (NMS). MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

peer—Router or device that participates as an endpoint in IPsec and IKE.

preshared key—Shared, secret key that uses IKE for authentication.

QoS—quality of service. Capability of a network to provide better service to selected network traffic over various technologies, including Frame Relay; Asynchronous Transfer Mode (ATM); Ethernet; and 802.1 networks, SONET, and IP-routed networks that may use any or all of these underlying technologies.

RADIUS—Remote Authentication Dial-In User Service. Distributed client or server system that secures networks against unauthorized access. RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

SA—security association. Instance of security policy and keying material applied to a data flow. Both IKE and IPsec use SAs, although SAs are independent of one another. IPsec SAs are unidirectional, and they are unique in each security protocol. An IKE SA is used by IKE only, and unlike the IPsec SA, it is bi-directional. IKE negotiates and establishes SAs on behalf of IPsec. A user can also establish IPsec SAs manually.

A set of SAs are needed for a protected data pipe, one per direction per protocol. For example, if you have a pipe that supports encapsulating security payload (ESP) between peers, one ESP SA is required for each direction. SAs are uniquely identified by destination (IPsec endpoint) address, security protocol (AH or ESP), and security parameter index (SPI).

SDM—Security Device Manager. Web interface manager that enables you to connect or disconnect a VPN tunnel and that provides a web interface for extended authentication (Xauth).

SNMP—Simple Network Management Protocol. Application-layer protocol that provides a message format for communication between SNMP managers and agents.

trap—Message sent by an SNMP agent to a network management system, console, or terminal to indicate the occurrence of a significant event, such as a specifically defined condition or a threshold that was reached.

VPN—virtual private network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses tunnels to encrypt all information at the IP level.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Easy VPN Remote RSA Signature Support

First Published: March 1, 2004
Last Updated: August 21, 2007

The Easy VPN Remote RSA Signature Support feature provides for the support of Rivest, Shamir, and Adelman (RSA) signatures on Easy VPN remote devices. The support is provided through RSA certificates that can be stored on or off the remote device.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Easy VPN Remote RSA Signature Support" section on page 6.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Prerequisites for Easy VPN Remote RSA Signature Support

- You must have a Cisco Virtual Private Network (VPN) remote device and be familiar with configuring the device.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- You must have a certification authority (CA) available to your network before you configure this interoperability feature. The CA must support the public key infrastructure (PKI) protocol of Cisco Systems, which is the Simple Certificate Enrollment Protocol (SCEP) (formerly called certificate enrollment protocol [CEP]).
- You should be familiar with IP Security (IPSec) and PKI.
- You should be familiar with configuring RSA key pairs.
- You should be familiar with configuring CAs.

Restrictions for Easy VPN Remote RSA Signature Support

- This feature should be configured only when you also configure both IPSec and Internet Key Exchange (IKE) in your network.
- The Cisco IOS software does not support CA server public keys greater than 2048 bits.

Information About Easy VPN Remote RSA Signature Support

To configure the Easy VPN Remote RSA Signature Support feature, you should understand the following concept:

- Easy VPN Remote RSA Signature Support Overview, page 2

Easy VPN Remote RSA Signature Support Overview

The Easy VPN Remote RSA Signature Support feature allows you to configure RSA signatures on your Easy VPN remote device. The signatures can be stored on or off your remote device.

How to Configure Easy VPN Remote RSA Signature Support

This section contains the following procedure:

- Configuring Easy VPN Remote RSA Signature Support, page 2

Configuring Easy VPN Remote RSA Signature Support

The RSA signatures for an Easy VPN remote device are configured the same way that you would configure RSA signatures for any other Cisco device. (For information about configuring RSA signatures, refer to the “Configuring Certification Authority Interoperability” chapter of the “IP Security and Encryption” section of the *Cisco IOS Security Configuration Guide*, Release 12.4.)

To enable the RSA signatures, when you are configuring the Easy VPN remote and assigning the configuration to the outgoing interface, you must omit the **group** command. The content of the first Organizational Unit (OU) field will be used as the group. (For information about configuring Cisco Easy VPN remote devices, refer to the feature document “*Cisco Easy VPN Remote*,” Release 12.4(11)T.)

To troubleshoot your Easy VPN remote RSA signature configuration, you can use the following **debug** commands. The **debug** commands can be used in any order or individually.

SUMMARY STEPS

1. **enable**
2. **debug crypto ipsec client ezvpn**
3. **debug crypto isakmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`debug crypto ipsec client ezvpn` **Example:** `Router# debug crypto ipsec client ezvpn`	Displays information about the VPN tunnel as it relates to the Easy VPN remote configuration.
Step 3	`debug crypto isakmp` **Example:** `Router# debug crypto isakmp`	Displays messages about IKE events.

Additional References

The following sections provide references related to Easy VPN Remote RSA Signature Support.

Related Documents

Related Topic	Document Title
Configuring IPSec	"IP Security and Encryption Overview" chapter of the *Cisco IOS Security Configuration Guide*, Release 12.4
Configuring IKE	"Configuring Internet Key Exchange Security Protocol" chapter of the "IP Security and Encryption" section of the *Cisco IOS Security Configuration Guide*, Release 12.4
Configuring RSA key pairs	Feature document "*Exporting and Importing RSA Keys*," Release 12.2(15)T
Declaring a CA	"Configuring Certification Authority Interoperability" chapter of the "IP Security and Encryption" section of the *Cisco IOS Security Configuration Guide*, Release 12.4
Configuring a Cisco Easy VPN remote device	Feature document "*Cisco Easy VPN Remote*," Release 12.4(11)T
Security commands	*Cisco IOS Security Command Reference*, Release 12.4 T

Standards

Standards	Title
There are no new or modified standards associated with this feature.	—

MIBs

MIBs	MIBs Link
There are no new or modified MIBs associated with this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
There are no new or modified RFCs associated with this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Easy VPN Remote RSA Signature Support

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Easy VPN Remote RSA Signature Support

Feature Name	Releases	Feature Information
Easy VPN Remote RSA Signature Support	12.3(7)T1 12.2(33)SRA 12.2(33)SXH	The Easy VPN Remote RSA Signature Support feature provides for the support of Rivest, Shamir, and Adelman (RSA) signatures on Easy VPN remote devices. The support is provided through RSA certificates that can be stored on or off the remote device. The following sections provide information about this feature: • "Easy VPN Remote RSA Signature Support Overview" section on page 2 • "Configuring Easy VPN Remote RSA Signature Support" section on page 2
Easy VPN Client RSA - Signature Support	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Easy VPN Server

First Published: February 25, 2002
Last Updated: November 4, 2008

The Easy VPN Server feature introduces server support for the Cisco VPN Client Release 3.x and later software clients and Cisco VPN hardware clients (such as the Cisco 800, Cisco 900, Cisco 1700, VPN 3002, and PIX 501 devices). This feature allows a remote end user to communicate using IP Security (IPsec) with any Cisco IOS Virtual Private Network (VPN) gateway. Centrally managed IPsec policies are "pushed" to the client device by the server, minimizing configuration by the end user.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Easy VPN Server" section on page 75.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Easy VPN Server

Nonsupported Protocols

Table 1 outlines IPsec protocol options and attributes that currently are *not* supported by Cisco VPN clients, so these options and attributes should not be configured on the router for these clients.

Table 1 Nonsupported IPsec Protocol Options and Attributes

Options	Attributes
Authentication Types	Authentication with public key encryption Digital Signature Standard (DSS)
Diffie-Hellman (D-H) groups	1
IPsec Protocol Identifier	IPSEC_AH
IPsec Protocol Mode	Transport mode
Miscellaneous	Manual keys Perfect Forward Secrecy (PFS)

Cisco Secure VPN Client 1.x Restrictions

When used with this feature, the Cisco Secure VPN Client 1.x has the following restrictions:

- It does not support dead peer detection (DPD) or any other keepalive scheme.
- It does not support initial contact.

This feature cannot use per-group attribute policy profiles such as IP addresses, and Domain Name Service (DNS). Thus, customers must continue to use existing, globally defined parameters for IP address assignment, Windows Internet Naming Service (WINS) and DNS, and preshared keys.

Multicast and Static NAT

Multicast and static NAT are supported only for Easy VPN servers using dynamic virtual tunnel interfaces (DVTIs).

Virtual IPsec Interface Restrictions

The Virtual IPsec Interface Support feature works only with a Cisco software VPN Client that is version 4.x or later, and an Easy VPN remote device that is configured to use a virtual interface.

cTCP Restrictions

- If a port is being used for Cisco Tunnel Control Protocol (cTCP), it cannot be used for other applications.
- cTCP can be used on only ten ports at a time.
- cTCP is supported on only Cisco IOS Easy VPN servers.
- If a cTCP connection is set up on a port, cTCP cannot be disabled on that port because doing so would cause the existing connection to stop receiving traffic.
- High Availability of cTCP is not currently supported on the Easy VPN server.

Information About Easy VPN Server

Before using the Easy VPN Server Enhancements feature, you should understand the following concepts:

How It Works

When the client initiates a connection with a Cisco IOS VPN device, the “conversation” that occurs between the peers consists of device authentication via Internet Key Exchange (IKE), followed by user authentication using IKE Extended Authentication (Xauth), VPN policy push (using Mode Configuration), and IPsec security association (SA) creation. An overview of this process is as follows:

- The client initiates IKE Phase 1 via aggressive mode (AM) if a preshared key is to be used for authentication; the client initiates main mode (MM) if digital certificates are used. If the client identifies itself with a preshared key, the accompanying group name entered in the configuration GUI (ID_KEY_ID) is used to identify the group profile associated with this client. If digital certificates are used, the organizational unit (OU) field of a distinguished name (DN) is used to identify the group profile.

Note Because the client may be configured for preshared key authentication, which initiates IKE AM, it is recommended that the administrator change the identity of the Cisco IOS VPN device via the **crypto isakmp identity hostname** command. This will not affect certificate authentication via IKE MM.

- The client attempts to establish an IKE SA between its public IP address and the public IP address of the Cisco IOS VPN device. To reduce the amount of manual configuration on the client, every combination of encryption and hash algorithms, in addition to authentication methods and D-H group sizes, is proposed.
- Depending on its IKE policy configuration, the Cisco IOS VPN device will determine which proposal is acceptable to continue negotiating Phase 1.

Tip IKE policy is global for the Cisco IOS VPN device and can consist of several proposals. In the case of multiple proposals, the Cisco IOS VPN device will use the first match, so you should always list your most secure policies first.

Note Device authentication ends and user authentication begins at this point.

- After the IKE SA is successfully established, and if the Cisco IOS VPN device is configured for Xauth, the client waits for a “username/password” challenge and then responds to the challenge of the peer. The information that is entered is checked against authentication entities using

authentication, authorization, and accounting (AAA) protocols such as RADIUS and TACACS+. Token cards may also be used via AAA proxy. During Xauth, it is also possible for a user-specific attribute to be retrieved if the credentials of that user are validated via RADIUS.

Note VPN devices that are configured to handle remote clients should always be configured to enforce user authentication.

- If the Cisco IOS VPN device indicates that authentication was successful, the client requests further configuration parameters from the peer. The remaining system parameters (for example, IP address, DNS, and split tunnel attributes) are pushed to the client at this time using Mode Configuration.

Note The IP address pool and group preshared key (if Rivest, Shamir, and Adelman [RSA] signatures are not being used) are the only required parameter in a group profile, all other parameters are optional.

- After each client is assigned an internal IP address via Mode Configuration, it is important that the Cisco IOS VPN device knows how to route packets through the appropriate VPN tunnel. Reverse route injection (RRI) will ensure that a static route is created on the Cisco IOS VPN device for each client internal IP address.

Note It is recommended that you enable RRI on the crypto map (static or dynamic) for the support of VPN clients unless the crypto map is being applied to a Generic Routing Encapsulation (GRE) tunnel that is already being used to distribute routing information.

- After the configuration parameters have been successfully received by the client, IKE quick mode is initiated to negotiate IPsec SA establishment.
- After IPsec SAs are created, the connection is complete.

RADIUS Support for Group Profiles

Group policy information is stored in a profile that can be defined locally in the router configuration or on a RADIUS server that is accessible by the Cisco IOS VPN device. If RADIUS is used, you must configure access to the server and allow the Cisco IOS VPN device to send requests to the server.

To define group policy attributes for RADIUS, you must do the following task on your RADIUS server:

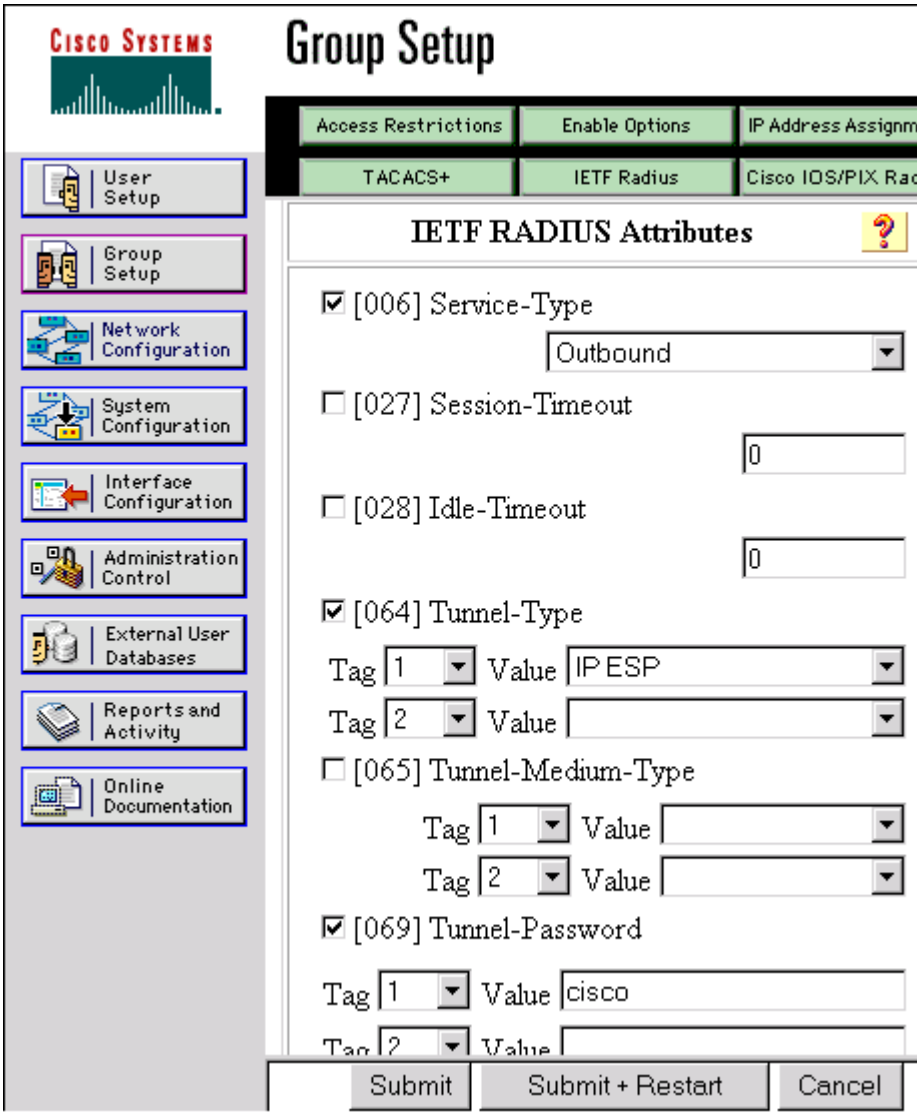
- Define a user that has a name equal to the group name as defined in the client graphical user interface (GUI). For example, if users will be connecting to the Cisco IOS VPN device using the group name “sales,” you will need a user whose name is “sales.” The password for this user is “cisco,” which is a special identifier that is used by the router for RADIUS purposes. The username must then be made a member of a group in which the correct policy is defined. For simplicity, it is recommended that the group name be the same as the username.

For a Cisco Secure Access Control Server

If you are using a Cisco Secure access control server (ACS), you may configure your remote access VPN group profiles on this server. To perform this task, you must ensure that Internet Engineering Task Force (IETF) RADIUS attributes are selected for group configuration as shown in Figure 1. (This figure also

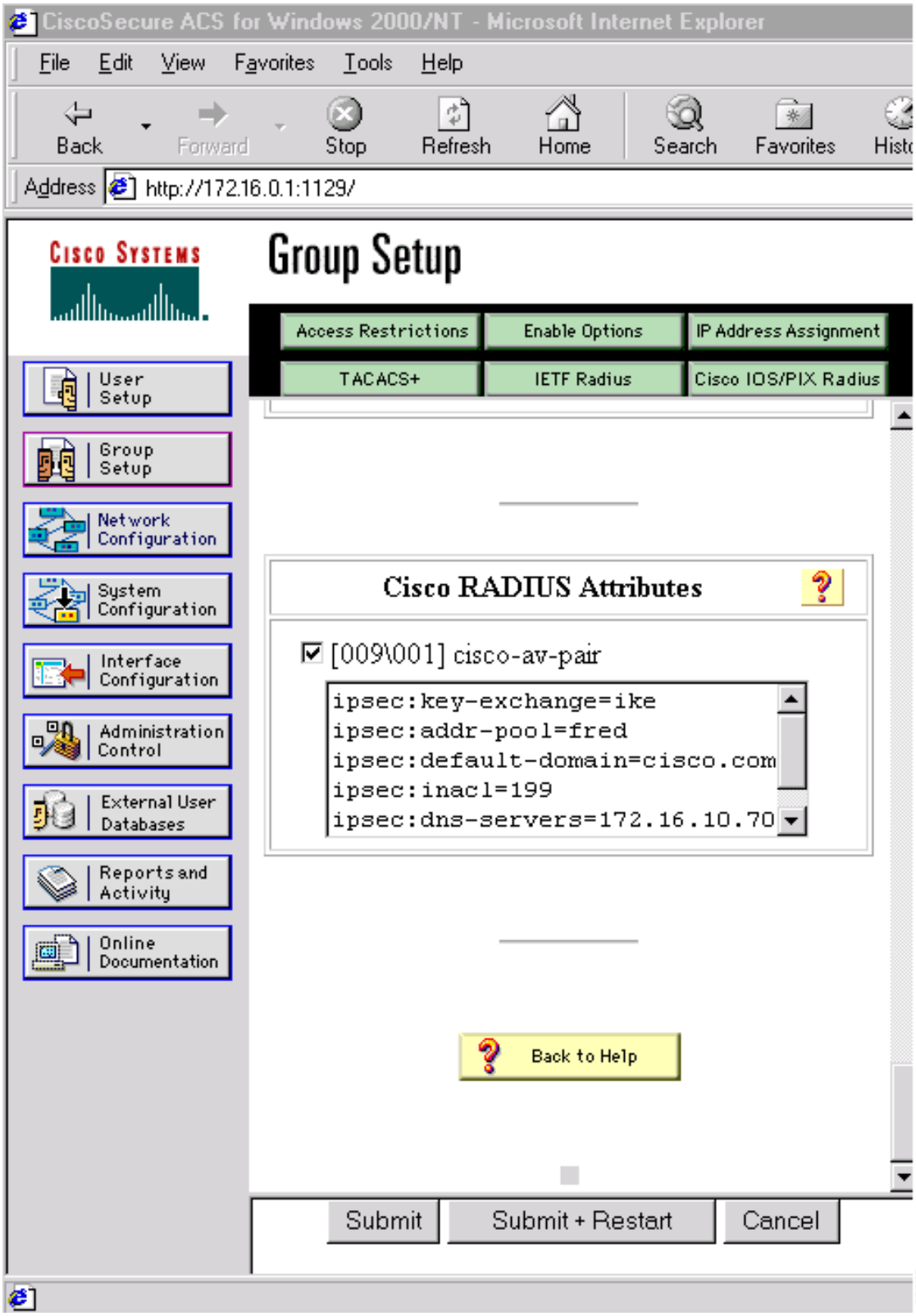
shows the compulsory attributes required for a remote access VPN group.) All values must be entered except the Tunnel-Password attribute, which is actually the preshared key for IKE purposes; if digital certificates are preferred, this attribute may be omitted.

Figure 1 IETF RADIUS Attributes Selection for Group Configuration



In addition to the compulsory attributes shown in Figure 1, other values can be entered that represent the group policy that is pushed to the remote client via Mode Configuration. Figure 2 shows an example of a group policy. All attributes are optional except the addr-pool, key-exchange=preshared-key, and key-exchange=ike attributes. The values of the attributes are the same as the setting that is used if the policy is defined locally on the router rather than in a RADIUS server. (These values are explained in the section "Defining Group Policy Information for Mode Configuration Push" later in this document.)

Figure 2 CiscoSecure ACS Group Policy Setup



After the group profile is created, a user who is a member of the group should be added. (Remember that the username that is defined maps to the group name as defined on the remote client, and the password defined for the username in the RADIUS database must be “cisco.”) If digital certificates are the preferred method of IKE authentication, the username should reflect the OU field in the certificate presented by the remote client.

For All Other RADIUS Servers

Ensure that your RADIUS server allows you to define attribute-value (AV) pairs. (For an example, see the section “Configuring Cisco IOS for Easy VPN Server: Example” later in this document).

Note If digital certificates are used, the username defined in RADIUS must be equal to the OU field of the DN of the certificate of the client.

RADIUS Support for User Profiles

Attributes may also be applied on a per-user basis. If you apply attributes on a per-user basis, you can override a group attribute value with an individual user attribute. The attributes are retrieved at the time that user authentication via Xauth occurs. The attributes are then combined with group attributes and applied during Mode Configuration.

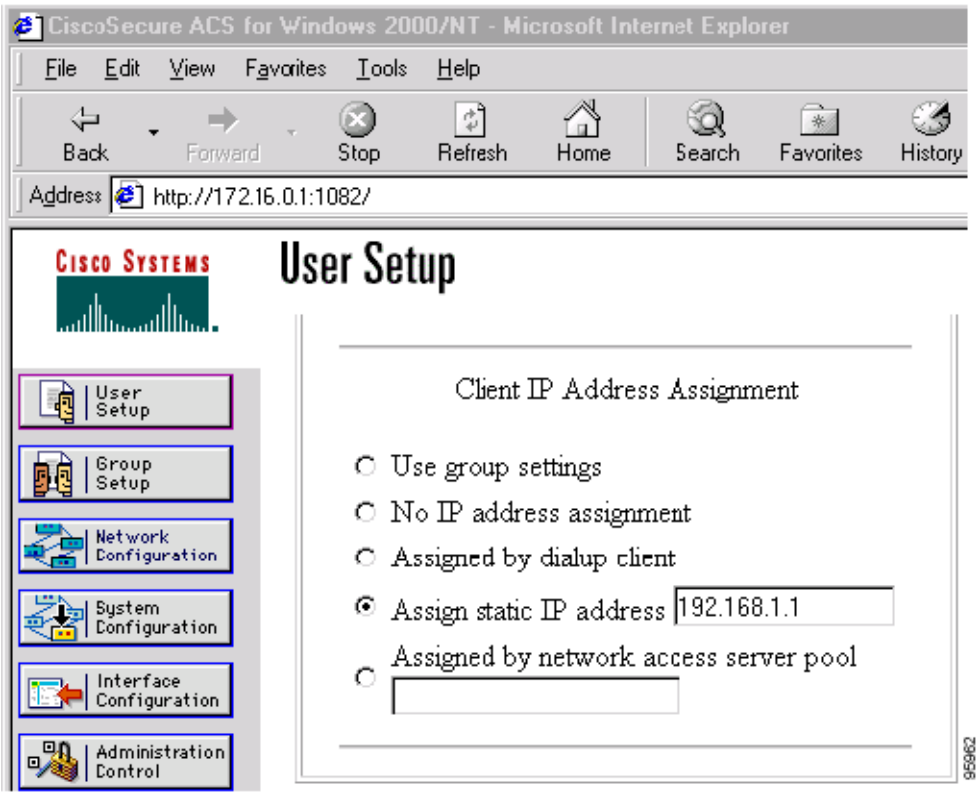
User-based attributes are available only if RADIUS is being used for user authentication.

To define user policy attributes for RADIUS, you must do the following task on your RADIUS server:

- Define a user or add attributes to the existing profile of a user in your RADIUS database. The password for the user will be used during Xauth user authentication, or you may proxy to a third-party server, such as a token card server.

Figure 3 shows how CiscoSecure ACS may be used for user authentication and for the assignment of a Framed-IP-Address attribute that may be pushed to the client. The presence of this attribute means that the local address pool defined for the group to which that user belongs will be overridden.

Figure 3 CiscoSecure ACS User Profile Setup



For All Other RADIUS Servers

Ensure that your RADIUS server allows you to define AV pairs. (For an example, see the "Configuring Cisco IOS for Easy VPN Server: Example" section later in this document.)

Supported Protocols

Table 2 outlines supported IPsec protocol options and attributes that can be configured for this feature. (See Table 1 for nonsupported options and attributes.)

Table 2 Supported IPsec Protocol Options and Attributes

Options	Attributes
Authentication Algorithms	• Hashed Message Authentication Codes with Message Digest 5 (HMAC-MD5) • HMAC-Secure Hash Algorithm 1 (HMAC-SHA1)
Authentication Types	• Preshared keys • RSA digital signatures

Table 2 Supported IPsec Protocol Options and Attributes (continued)

Options	Attributes
D-H groups	• 2 • 5
Encryption Algorithms (IKE)	• Data Encryption Standard (DES) • Triple Data Encryption Standard (3DES)
Encryption Algorithms (IPsec)	• DES • 3DES • NULL
IPsec Protocol Identifiers	• Encapsulating Security Payload (ESP) • IP LZS compression (IPCOMP-LZS)
IPsec Protocol Mode	Tunnel mode

Functions Supported by Easy VPN Server

Mode Configuration Version 6 Support

Mode Configuration version 6 is now supported for more attributes (as described in an IETF draft submission).

Xauth Version 6 Support

Cisco IOS has been enhanced to support version 6 of Xauth. Xauth for user authentication is based on an IETF draft submission.

IKE DPD

The client implements a new keepalives scheme—IKE DPD.

DPD allows two IPsec peers to determine whether the other is still "alive" during the lifetime of a VPN connection. DPD is useful because a host may reboot, or the dialup link of a remote user may disconnect without notifying the peer that the VPN connection has gone away. When an IPsec host determines that a VPN connection no longer exists, the host can notify a user, attempt to switch to another IPsec host, or clean up valuable resources that were allocated for the peer that no longer exists.

A Cisco IOS VPN device can be configured to send and reply to DPD messages. DPD messages are sent if no other traffic is being passed through the VPN tunnel. If a configured amount of time has lapsed since the last inbound data was received, DPD will send a message ("DPD R-U-THERE") the next time it sends outbound IPsec data to the peer. DPD messages are unidirectional and are automatically sent by Cisco VPN clients. DPD *must* be configured on the router *only* if the router wishes to send DPD messages to the VPN client to determine the health of the client.

Split Tunneling Control

Remote clients can support split tunneling, which enables a client to have intranet and Internet access at the same time. If split tunneling is not configured, the client will direct all traffic through the tunnel, even traffic destined for the Internet.

Initial Contact

If a client is suddenly disconnected, the gateway may not be notified. Consequently, removal of connection information (IKE and IPsec SAs) for that client will not immediately occur. Thus, if the client attempts to reconnect to the gateway again, the gateway will refuse the connection because the previous connection information is still valid.

To avoid such a scenario, a new capability called initial contact has been introduced; it is supported by all Cisco VPN products. If a client or router is connecting to another Cisco gateway for the first time, an initial contact message is sent that tells the receiver to ignore and delete any old connection information that has been maintained for that newly connecting peer. Initial contact ensures that connection attempts are not refused because of SA synchronization problems, which are often identified via invalid security parameter index (SPI) messages and which require devices to have their connections cleared.

Group-Based Policy Control

Policy attributes such as IP addresses, DNS, and split tunnel access can be provided on a per-group or per-user basis.

User-Based Policy Control

Attributes may also be applied on a per-user basis. You can override a group attribute value with an individual user attribute. The attributes are retrieved at the time that user authentication via Xauth occurs. They are then combined with group attributes and applied during Mode Configuration.

From Cisco IOS Release 12.3(4)T forward, attributes can be applied on a per-user basis after the user has been authenticated. These attributes can override any similar group attributes. User-based attributes are available only if RADIUS is used as the database.

Framed-IP-Address

To select the Framed-IP-Address attribute for CiscoSecure for NT, do the following: Under the user profile, choose the “use this IP address” option under addressing and manually enter the address. (You should check the method of configuring a framed IP address with your own RADIUS server because this procedure will vary.)

Note If a framed IP address is present, and there is also a local pool address configured for the group that the user belongs to, the framed IP address will override the local pool setting.

DHCP Client Proxy

Easy VPN servers currently assign an IP address to a remote device using either a local pool that is configured on the router or the framed IP address attribute that is defined in RADIUS. Effective with Cisco IOS Release 12.4(9)T, the DHCP Client Proxy feature provides the option of configuring an Easy VPN server to obtain an IP address from a DHCP server. The IP address is pushed to the remote device using mode configuration.

Note This feature does not include functionality for the DHCP server to push the DNS, WINS server, or domain name to the remote client.

To configure DHCP Client Proxy, see the section “Configuring an Easy VPN Server to Obtain an IP Address from a DHCP Server.”

Benefits of DHCP Client Proxy

- The functionality provided with this feature helps in the creation of DDNS (dynamic Domain Name System) entries when a DNS server exists in conjunction with the DHCP server.
- The user is not restricted to IP address pools.

User-Save-Password

As per the group description, the User-Save-Password attribute can be received in addition to the group variant (Save-Password), but if it is received, it will override the value asserted by the group.

The following is an output example of a RADIUS AV pair for the User-Save-Password attribute:

```
ipsec:user-save-password=1
```

User-Include-Local-LAN

As per the group description, the User-Include-Local-LAN attribute can be received in addition to the group variant (Include-Local-LAN), but if it is received, it will override the value asserted by the group.

The following is an output example of a RADIUS AV pair for the User-Include-Local LAN attribute:

```
ipsec:user-include-local-lan=1
```

User-VPN-Group

The User-VPN-Group attribute is a replacement for the Group-Lock attribute. It allows support for both preshared key and RSA signature authentication mechanisms such as certificates.

If you need to check that the group a user is attempting to connect to is indeed the group the user belongs to, use the User-VPN-Group attribute. The administrator sets this attribute to a string, which is the group that the user belongs to. The group the user belongs to is matched against the VPN group as defined by group name (ID_KEY_ID) for preshared keys or by the OU field of a certificate. If the groups do not match, the client connection is terminated.

This feature works only with AAA RADIUS. Local Xauth authentication must still use the Group-Lock attribute.

The following is an output example of a RADIUS AV pair for the Use-VPN-Group attribute:

```
ipsec:user-vpn-group=cisco
```

Group-Lock

If you are only using pre-shared keys (no certificates or other RSA signature authentication mechanisms) with RADIUS or local AAA, you can continue to use the Group-Lock attribute. If you are only using pre-shared keys (no certificates or other RSA signature authentication mechanisms) with RADIUS, you can either continue to use the Group-Lock attribute or you can use the new User-VPN-Group attribute.

Caution Do not use the Group-Lock attribute if you are using RSA signature authentication mechanisms such as certificates. Use the User-VPN-Group attribute instead.

Session Monitoring for VPN Group Access

It is possible to mimic the functionality provided by some RADIUS servers for limiting the maximum number of connections to a specific server group and also for limiting the number of simultaneous logins for users in that group. After user-defined thresholds are defined in each VPN group, connections will be denied until counts drop below these thresholds.

If you use a RADIUS server, such as CiscoSecure ACS, it is recommended that you enable this session control on the RADIUS server if the functionality is provided. In this way, usage can be controlled across a number of servers by one central repository. When enabling this feature on the router itself, only connections to groups on that specific device are monitored. Load-sharing scenarios are not accurately accounted for.

To configure session monitoring using command-line interface (CLI), use the **crypto isakmp client configuration group** command and the **max-users** and **max-logins** subcommands.

The following is an output example of RADIUS AV pairs that have been added to the relevant group:

```
ipsec:max-users=1000
ipsec:max-logins=1
```

Virtual IPsec Interface Support on a Server

Virtual IPsec Interface Support on a Server allows you to selectively send traffic to different Easy VPN concentrators (servers) as well as to the Internet.

Before Cisco IOS Release 12.4(4)T, at the tunnel-up/tunnel-down transition, attributes that were pushed during the mode configuration had to be parsed and applied. When such attributes resulted in the configurations being applied on the interface, the existing configuration had to be overridden.

With the Virtual Ipsec Interface Support feature, the tunnel-up configuration can be applied to separate interfaces, making it easier to support separate features at tunnel-up. Features that are applied to the traffic going into the tunnel can be separate from the features that are applied to traffic that is not going through the tunnel (for example, split-tunnel traffic and traffic leaving the device when the tunnel is not up). When the Easy VPN negotiation is successful, the line protocol state of the virtual-access interface gets changed to up. When the Easy VPN tunnel goes down because the SA expires or is deleted, the line protocol state of the virtual-access interfaces changes to down.

Note This feature does not support multicast.

For more information about this feature, see the document *Cisco Easy VPN Remote*. (This feature is configured on the Easy VPN remote device.)

For information about the IPsec Virtual Tunnel Interface feature, see the document "IPSec Virtual Tunnel Interface" (link in the "Related Documents" section of this document).

Virtual Tunnel Interface Per-User Attribute Support

Effective with Cisco IOS Release 12.4(9)T, Virtual Tunnel Interface provides per-user attribute support for Easy VPN servers.

For more information about this feature, see the document *IPsec Virtual Tunnel Interface*.

Banner, Auto-Update, and Browser Proxy

The following features provide support for attributes that aid in the management of the Cisco Easy VPN remote device.

Banner

An Easy VPN server can be configured to push the banner to the Easy VPN remote device. A banner is needed for the web-based activation feature. The banner is displayed when the Easy VPN tunnel is up on the Easy VPN remote console or as a HTML page in the case of web-based activation.

Auto-Update

An Easy VPN server can be configured to provide an automated mechanism for software and firmware upgrades on an Easy VPN remote device.

Browser Proxy

An Easy VPN server can be configured so that an Easy VPN remote device can access resources on the corporate network. Using this feature, the user does not have to manually modify the proxy settings of his or her web browser when connecting to the corporate network using Cisco IOS VPN Client or manually revert the proxy settings upon disconnecting.

Configuration Management Enhancements

Pushing a Configuration URL Through a Mode-Configuration Exchange

When remote devices connect to a corporate gateway for creating an IPsec VPN tunnel, some policy and configuration information has to be applied to the remote device when the VPN tunnel is active to allow the remote device to become a part of the corporate VPN.

The Pushing a Configuration URL Through a Mode-Configuration Exchange feature provides for a mode-configuration attribute that "pushes" a URL from the concentrator (server) to the Cisco IOS Easy VPN remote device. The URL contains the configuration information that the remote device has to download and apply to the running configuration, and it contains the Cisco IOS CLI listing. (For more information about a Cisco IOS CLI listing, see Cisco IOS documentation for the **configuration url** command.) The CLI for this feature is configured on the concentrator.

The configuration that is pushed to the remote device is persistent by default. That is, the configuration is applied when the IPsec tunnel is "up," but it is not withdrawn when the IPsec tunnel goes "down." However, it is possible to write a section of configuration that is transient in nature, in which case the configuration of the section is reverted when the tunnel is disconnected.

There are no restrictions on where the configuration distribution server is physically located. However, it is recommended that a secure protocol such as HTTPS (Secure HTTP) be used to retrieve the configuration. The configuration server can be located in the corporate network, so because the transfer happens through the IPsec tunnel, insecure access protocols (HTTP) can be used.

Regarding backward compatibility: the remote device asks for the CONFIGURATION-URL and CONFIGURATION-VERSION attributes. Because the CONFIGURATION-URL and CONFIGURATION-VERSION attributes are not mandatory attributes, the server sends them only if it has them configured for the group. There is no built-in restriction to push the configuration, but bootstrap configurations (such as for the IP address) cannot be sent because those configurations are required to set up the Easy VPN tunnel, and the CONFIGURATION-URL comes into effect only after the Easy VPN tunnel comes up.

After the Configuration Has Been Acquired by the Easy VPN Remote Device

After the configuration has been acquired by the Easy VPN remote device, the remote device sends a new ISAKMP notification to the Easy VPN server. The notification contains several manageability information messages about the client (remote device). The Easy VPN server takes two actions when this information is received:

- The Easy VPN server caches the information in its peer database. The information can be displayed by using the **show crypto isakmp peer config** command. This command output displays all manageability information that is sent by the client (remote device).
- If accounting is enabled, the Easy VPN server sends an accounting update record that contains the manageability information messages about the remote device to the accounting RADIUS server. This accounting update is later available in the accounting log of the RADIUS server.

How to Configure This Feature

The commands that are used to configure this feature and the attributes CONFIGURATION-URL and CONFIGURATION-VERSION are described in the **crypto isakmp client configuration group** command documentation.

Per User AAA Policy Download with PKI

With the Support of Per User AAA Policy Download with PKI feature, user attributes are obtained from the AAA server and pushed to the remote device through mode configuration. The username that is used to get the attributes is retrieved from the remote device certificate.

Per-User Attribute Support for Easy VPN Servers

The Per-User Attribute Support for Easy VPN Servers feature provides users with the ability to support per-user atttributes on Easy VPN servers. These attributes are applied on the virtual access interface.

Local Easy VPN AAA Server

For a local Easy VPN AAA server, the per-user attributes can be applied at the group level or at the user level using the command-line interface (CLI).

To configure per-user attributes for a local Easy VPN server, see "Configuring Per-User Attributes on a Local Easy VPN AAA Server."

Remote Easy VPN AAA Server

Attribute value (AV) pairs can be defined on a remote Easy VPN AAA server as shown in this example:

```
cisco-avpair = "ip:outacl#101=permit tcp any any established
```

Per-User Attributes

The following per-user attributes are currently defined in the AAA server and are applicable to IPsec:

- inacl
- interface-config
- outacl
- route
- rte-fltr-in
- rte-fltr-out
- sub-policy-In
- sub-policy-Out
- policy-route
- prefix

Syslog Message Enhancements

Some new syslog messages have been added for Easy VPN in Cisco IOS Release 12.4(4)T. The syslog messages can be enabled on your server by using the command-line interface (CLI). The format of the syslog messages is as follows:

```
timestamp: %CRYPTO-6-VPN_TUNNEL_STATUS: (Server)  <event message>  User=<username>
Group=<groupname>  Client_public_addr=<ip_addr> Server_public_addr=<ip addr>
```

For an authentication-passed event, the syslog message looks like the following:

```
Jul 25 23:33:06.847: %CRYPTO-6-VPN_TUNNEL_STATUS: (Server) Authentication PASS
ED User=blue Group=Cisco1760group Client_public_addr=10.20.20.1
Server_public_addr=10.20.20.2
```

Three of the messages (Max users, Max logins, and Group does not exist) are authorization issues and are printed only with the group name in the format. The reason for only the group name being printed is that authorization check happens much before mode configuration happens. Therefore, the peer information is not yet present and cannot be printed. The following is an example of a "Group does not exit" message.

```
*Jun 30 18:02:58.107: %CRYPTO-6-VPN_TUNNEL_STATUS: Group: group_1 does not exist
```

Easy VPN Syslog Messages That Are Supported

Both ezvpn_connection_up and ezvpn_connection_down were already supported in a previous release of syslog messages. The enhancements in Cisco IOS Release 12.4(4)T follow the same format, but new syslogs are introduced. The added syslogs are as follows:

- Authentication Passed
- Authentication Rejected
  - Group Lock Enabled
  - Incorrect Username or Password
  - Max Users exceeded/Max Logins exceeded
  - No. of Retries exceeded
- Authentication Failed (AAA Not Contactable)
- IP Pool Not present/No Free IP Address available in the pool
- ACL associated with Ezvpn policy but NOT defined (hence, no split tunneling possible)
- Save password Turned ON
- Incorrect firewall record being sent by Client (incorrect vendor | product | capability)
- Authentication Rejected
  - Access restricted via incoming interface
  - Group does not exist

Network Admission Control Support for Easy VPN

Network Admission Control was introduced in Cisco IOS Release 12.3(8)T as a way to determine whether a PC client should be allowed to connect to the LAN. Network Admission Control uses Extensible Authentication Protocol over UDP (EAPoUDP) to query the Cisco trust agent on the PC and allows a PC to access the network if the client status is healthy. Different policies can be applied on the server to deny or limit access of PCs that are infected.

Effective with Cisco IOS Release 12.4(4)T, Network Admission Control can now be used to monitor the status of remote PC clients as well. After the Easy VPN tunnel comes up and the PC starts to send traffic, the traffic is intercepted at the Easy VPN server, and the posture validation process starts. The posture validation process consists of sending an EAPoUDP request over the Easy VPN tunnel and querying the Cisco trust agent. The authentication server is configured inside the trusted network, behind the IPsec aggregator.

The configuration of an Easy VPN server that has Network Admission Control enabled is shown in the output in Network Admission Control: Example, page 64.

Central Policy Push Firewall Policy Push

The Easy VPN server supports Central Policy Push (CPP) Firewall Policy Push. This feature allows administrators to push policies that enforce security to the Cisco Easy VPN (software) Client and related firewall software.

A split tunnel enables access to corporate networks, but it also allows a remote device to be exposed to attacks from the Internet. This feature enables the server to determine whether to allow or deny a tunnel if the remote device does not have a required firewall, thereby reducing exposure to attacks.

The following firewall types are supported:

- Cisco-Integrated-firewall (central-policy-push)
- Cisco-Security-Agent (check-presence)
- Zonelabs-Zonealarm (both)
- Zonelabs-ZonealarmPro (both)

The server can be used either to check the presence of a firewall on the client (remote device) using the check-presence option or to specify the specifics of the firewall policies that must be applied by the client using the central-policy-push.

Note The **policy check-presence command and keyword, which are used with this feature, replace the firewall are-u-there command functionality that was supported before Cisco IOS Release 12.4(6)T. The firewall are-u-there command will continue to be supported for backward compatibility.**

To enable this feature, see the sections "Defining a CPP Firewall Policy Push Using a Local AAA Server" and "Applying a CPP Firewall Policy Push to the Configuration Group."

Syslog Support for CPP Firewall Policy Push

Syslog support can be enabled using the **crypto logging ezvpn** command on your router. CPP syslog messages will be printed for the following error conditions:

- If policy is configured on a group configuration (using the **firewall policy** command), but a global policy with the same name is not defined (using the **crypto isakmp client firewall** command). The syslog message is as follows:

  ```
  Policy enabled on group configuration but not defined
  ```

  Tunnel setup proceeds as normal (with the firewall).

- If an incorrect firewall request (vendor/product/cap incorrect order) is received, the syslog message is as follows:

  ```
  Incorrect firewall record received from client
  ```

- If a policy mismatch occurs between the Cisco VPN Client and the server, the syslog is as follows:

  ```
  CPP policy mismatch between client and headend
  ```

Password Aging

Prior to Cisco IOS Release 12.4(6)T, EasyVPN remote devices (clients) sent username and password values to the Easy VPN server, which in turn sent them to the AAA subsystem. The AAA subsystem generated an authentication request to the RADIUS server. If the password had expired, the RADIUS

server replied with an authentication failure. The reason for the failure was not passed back to the AAA subsystem. The user was denied access due to authentication failure, but he or she did not know that the failure was due to password expiration.

Effective with Cisco IOS Release 12.4(6)T, if you have configured the Password Aging feature, the EasyVPN client is notified when a password has expired, and you are prompted to enter a new password. To configure the Password Aging feature, see the section "Configuring Password Aging."

For more information about Password Aging, see the reference for "Password Aging" in the section Additional References (subsection "Related Documents).

Split DNS

Effective with Cisco IOS Release 12.4(9)T, split DNS functionality is available on Easy VPN servers. This feature enables the Easy VPN hardware client to use primary and secondary DNS values to resolve DNS queries. These values are pushed by the Easy VPN server to the Easy VPN remote device. To configure this feature on your server, use the **split-dns** command (see the section "Defining Group Policy Information for Mode Configuration Push"). Configuring this command adds the split-dns attribute to the policy group. The attribute will include the list of domain names that you configured. All other names will be resolved using the public DNS server.

For more information about configuring split DNS, see "Configuring Split and Dynamic DNS on the Cisco VPN 3000" at the following URL:

http://www.cisco.com/warp/public/471/dns_split_dynam.pdf

cTCP

The Cisco Tunneling Control Protocol (cTCP) feature can be used for situations in which an Easy VPN remote device is operating in an environment in which standard IPsec does not function or in which it does not function transparently without modification to existing firewall rules. These situations include the following:

- Small or home office router performing Network Address Translation (NAT) or Port Address Translation (PAT)
- PAT-provided IP address behind a larger router (for example, in a corporation)
- Non-NAT firewall (packet filtering or stateful)
- Proxy server

The firewall should be configured to allow the headend to accept cTCP connections on the configured cTCP port. This configuration is enabled on the Easy VPN server. If the firewall is not configured, it will not allow the cTCP traffic.

Note cTCP traffic is actually Transmission Control Protocol (TCP) traffic. cTCP packets are IKE or Encapsulating Security Payload (ESP) packets that are being transmitted over TCP.

VRF Assignment by a AAA Server

To assign VRF to Easy VPN users, the following attributes should be enabled on a AAA server:

```
Cisco-avpair "ip:interface-config=ip vrf forwarding example1"
Cisco-avpair "ip:interface-config=ip unnumbered loopback10"
```

How to Configure Easy VPN Server

This section includes the following procedures:

Enabling Policy Lookup via AAA

To enable policy lookup via AAA, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication password-prompt** *text-string*
5. **aaa authentication username prompt** *text-string*
6. **aaa authentication login** [*list-name method1*] [*method2...*]
7. **aaa authorization network** *list-name* **local group radius**
8. **username** *name* **password** *encryption-type encrypted-password*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables AAA.
Step 4	**aaa authentication password-prompt** *text-string* **Example:** `Router (config)# aaa authentication password-prompt "Enter your password now:"`	(Optional) Changes the text displayed when users are prompted for a password.
Step 5	**aaa authentication username-prompt** *text-string* **Example:** `Router (config)# aaa authentication username-prompt "Enter your name here:"`	(Optional) Changes the text displayed when users are prompted to enter a username.

	Command	Purpose
Step 6	**aaa authentication login** [*list-name method1*] [*method2...*] **Example:** `Router (config)# aaa authentication login userlist local group radius`	Sets AAA authentication at login. • A local and RADIUS server may be used together and will be tried in order. **Note** This command must be enabled to enforce Xauth.
Step 7	**aaa authorization network** *list-name* **local group radius** **Example:** `Router (config)# aaa authorization network grouplist local group radius`	Enables group policy lookup. • A local and RADIUS server may be used together and will be tried in order.
Step 8	**username** *name* **password** *encryption-type encrypted-password* **Example:** `Router (config)# username server_r password 7 121F0A18`	(Optional) Defines local users for Xauth if RADIUS or TACACS+ is not used. **Note** Use this command only if no external validation repository will be used.

Defining Group Policy Information for Mode Configuration Push

Although users can belong to only one group per connection, they may belong to specific groups with different policy requirements. Thus, users may decide to connect to the client using a different group ID by changing their client profile on the VPN device. To define the policy attributes that are pushed to the client via Mode Configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** {*group-name* | **default**}
4. **key** *name*
5. **dns** *primary-server secondary-server*
6. **wins** *primary-server secondary-server*
7. **domain** *name*
8. **pool** *name*
9. **netmask** *name*
10. **acl** *number*
11. **access-restrict** {*interface-name*}
12. **policy check-presence**

    or

    firewall are-u-there
13. **group-lock**

14. **include-local-lan**
15. **save-password**
16. **backup-gateway**
17. **pfs**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp client configuration group** {*group-name* | **default**} **Example:** `Router (config)# crypto isakmp client configuration group group1`	Specifies the policy profile of the group that will be defined and enters Internet Security Association Key Management Protocol (ISAKMP) group configuration mode. • If no specific group matches and a default group is defined, users will automatically be given the policy of a default group.
Step 4	**key** *name* **Example:** `Router (config-isakmp-group)# key group1`	Specifies the IKE preshared key for group policy attribute definition. **Note** This command *must* be enabled if the client identifies itself with a preshared key.
Step 5	**dns** *primary-server secondary-server* **Example:** `Router (config-isakmp-group)# dns 10.2.2.2 10.3.3.3`	(Optional) Specifies the primary and secondary DNS servers for the group.
Step 6	**wins** *primary-server secondary-server* **Example:** `Router (config-isakmp-group)# wins 10.10.10.10 10.12.12.12`	(Optional) Specifies the primary and secondary WINS servers for the group.
Step 7	**domain** *name* **Example:** `Router (config-isakmp-group)# domain domain.com`	(Optional) Specifies the DNS domain to which a group belongs.

	Command	Purpose
Step 8	**pool** *name* **Example:** `Router (config-isakmp-group)# pool green`	Defines a local pool address. • Although a user must define at least one pool name, a separate pool may be defined for each group policy. **Note** This command *must* be defined and refer to a valid IP local pool address or the client connection will fail.
Step 9	**netmask** *name* **Example:** `Router (config-isakmp-group)# netmask 255.255.255.255`	(Optional) Specifies that a subnet mask be downloaded to the client for local connectivity. **Note** Some VPN clients use the default mask for their particular classes of address. However, for a router, the host-based mask is typically used (/32). If you want to override the default mask, use the **netmask** command.
Step 10	**acl** *number* **Example:** `Router (config-isakmp-group)# acl 199`	(Optional) Configures split tunneling. • The *number* argument specifies a group of access control list (ACL) rules that represent protected subnets for split tunneling purposes.
Step 11	**access-restrict** {*interface-name*} **Example:** `Router (config-isakmp-group)# access-restrict fastethernet0/0`	Restricts clients in a group to an interface.
Step 12	**policy check-presence** or **firewall are-u-there** **Example:** `Router (config-isakmp-group)# policy check-presence` or `Router (config-isakmp-group)# firewall are-u-there`	(Optional) Denotes that the server should check for the presence of the specified firewall (as shown as the firewall type on the client). or Adds the firewall are-u-there attribute to the server group if your PC is running the Black Ice or Zone Alarm personal firewalls. **Note** The **policy** command and **check-presence** keyword were added to Cisco IOS documentation in Cisco IOS 12.4(6)T. It is recommended that the **policy** command be used instead of the **firewall are-u-there** command because the **policy** command is supported in local AAA and remote AAA configurations. The **firewall are-u-there** command can be figured only locally, but it is still supported for backward compatibility.
Step 13	**group-lock** **Example:** `Router (config-isakmp-group)# group-lock`	Enforces the group lock feature.
Step 14	**include-local-lan** **Example:** `Router (config-isakmp-group)# include-local-lan`	(Optional) Configures the Include-Local-LAN attribute to allow a nonsplit-tunneling connection to access the local subnetwork at the same time as the client.

	Command	Purpose
Step 15	`save-password` **Example:** `Router (config-isakmp-group)# save-password`	(Optional) Saves your Xauth password locally on your PC.
Step 16	`backup-gateway` **Example:** `Router (config-isakmp-group)# backup gateway`	(Optional) Rather than have backup gateways added to client configurations manually, it is possible to have the server "push down" a list of backup gateways to the client device. • These gateways are tried in order in the case of a failure of the previous gateway. The gateways may be specified using IP addresses or host names.
Step 17	`pfs` **Example:** `Router (config-isakmp-group)# pfs`	(Optional) Notifies the client of the central-site policy regarding whether PFS is required for any IPsec SA. • Because the client device does not have a user interface option to enable or disable PFS negotiation, the server will notify the client device of the central site policy using this parameter. The Diffie-Hellman (D-H) group that is proposed for PFS will be the same that was negotiated in Phase 1 of the IKE negotiation.

Enabling VPN Session Monitoring

If you wish to set restrictions on the maximum number of connections to the router per VPN group and the maximum number of simultaneous logins per user, add the following attributes to the VPN group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** *group-name*
4. **exit**
5. **max-logins** *number-of-logins*
6. **max-users** *number-of-users*

DETAILED STEPS

	Command	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp client configuration group** *group-name* **Example:** `Router (config)# crypto isakmp client configuration group group1`	Specifies the policy profile of the group that will be defined and enters ISAKMP group configuration mode. • *group-name*—Group definition that identifies which policy is enforced for users.
Step 4	**exit** **Example:** `Router (config-isakmp-group)# exit`	Exits ISAKMP group configuration mode.
Step 5	**max-logins** *number-of-logins* **Example:** `Router (config-isakmp-group)# max-logins 10`	(Optional) Limits the number of simultaneous logins for users in a specific server group.
Step 6	**max-users** *number-of-users* **Example:** `Router (config)# max-users 1000`	(Optional) Limits the number of connections to a specific server group.

Verifying a VPN Session

To verify a VPN session, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto session group**
3. **show crypto session summary**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command	Purpose
Step 2	**show crypto session group** **Example:** `Router# show crypto session group`	Displays groups that are currently active on the VPN device.
Step 3	**show crypto session summary** **Example:** `Router# show crypto session summary`	Displays groups that are currently active on the VPN device and the users that are connected for each of those groups.

Applying Mode Configuration and Xauth

Mode Configuration and Xauth must be applied to a crypto map to be enforced. To apply Mode Configuration and Xauth to a crypto map, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *tag* **client configuration address** [**initiate** | **respond**]
4. **crypto map** *map-name* **isakmp authorization list** *list-name*
5. **crypto map** *map-name* **client authentication list** *list-name*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *tag* **client configuration address** [**initiate** | **respond**] **Example:** `Router (config)# crypto map dyn client configuration address initiate`	Configures the router to initiate or reply to Mode Configuration requests. **Note** Cisco clients require the **respond** keyword to be used; however, if the Cisco Secure VPN Client 1.x is used, the **initiate** keyword must be used; **initiate** and **respond** keywords may be used simultaneously.

	Command	Purpose
Step 4	**crypto map** *map-name* **isakmp authorization list** *list-name* **Example:** `Router (config)# crypto map ikessaaamap isakmp authorization list ikessaaalist`	Enables IKE querying for group policy when requested by the client. • The *list-name* argument is used by AAA to determine which storage source is used to find the policy (local or RADIUS) as defined in the **aaa authorization network** command.
Step 5	**crypto map** *map-name* **client authentication list** *list-name* **Example:** `Router (config)# crypto map xauthmap client authentication list xauthlist`	Enforces Xauth. • The *list-name* argument is used to determine the appropriate username and password storage location (local or RADIUS) as defined in the **aaa authentication login** command.

Enabling Reverse Route Injection for the Client

To enable RRI on the crypto map (static or dynamic) for VPN client support, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic** *map-name seq-num*

   or

   crypto map *map-name seq-num* **ipsec-isakmp**
4. **set peer** *ip-address*
5. **set transform-set** *transform-set-name*
6. **reverse-route**
7. **match-address**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 3	**crypto dynamic** *map-name seq-num* or **crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router (config)# crypto dynamic mymap 10` or `Router (config)# crypto map yourmap 15 ipsec-isakmp`	Creates a dynamic crypto map entry and enters crypto map configuration mode. or Adds a dynamic crypto map set to a static crypto map set and enters crypto map configuration mode.
Step 4	**set peer** *ip-address* **Example:** `Router (config-crypto-map)# set peer 10.20.20.20`	Specifies an IPsec peer IP address in a crypto map entry. • This step is optional when configuring dynamic crypto map entries.
Step 5	**set transform-set** *transform-set-name* **Example:** `Router (config-crypto-map)# set transform-set dessha`	Specifies which transform sets are allowed for the crypto map entry. • Lists multiple transform sets in order of priority (highest priority first). **Note** This list is the only configuration statement required in dynamic crypto map entries.
Step 6	**reverse-route** **Example:** `Router (config-crypto-map)# reverse-route`	Creates source proxy information.
Step 7	**match address** **Example:** `Router (config-crypto-map)# match address`	Specifies an extended access list for a crypto map entry. • This step is optional when configuring dynamic crypto map entries.

Enabling IKE Dead Peer Detection

To enable a Cisco IOS VPN gateway (instead of the client) to send IKE DPD messages, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp keepalive** *secs retries*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp keepalive** *secs retries* **Example:** `Router (config)# crypto isakmp keepalive 20 10`	Allows the gateway to send DPD messages to the router. • The *secs* argument specifies the number of seconds between DPD messages (the range is from 1 to 3600 seconds); the *retries* argument specifies the number of seconds between retries if DPD messages fail (the range is from 2 to 60 seconds).

Configuring RADIUS Server Support

To configure access to the server and allow the Cisco IOS VPN device to send requests to the server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius server host** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*] [**key** *string*]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**radius server host** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*] [**key** *string*] **Example:** `Router (config)# radius server host 192.168.1.1. auth-port 1645 acct-port 1646 key XXXX`	Specifies a RADIUS server host. **Note** This step is required if you choose to store group policy information in a RADIUS server.

Verifying Easy VPN Server

To verify your configurations for this feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto map** [**interface** *interface* | **tag** *map-name*]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto map** [**interface** *interface*	**tag** *map-name*] **Example:** `Router# show crypto map interface ethernet 0`	Displays the crypto map configuration.

Configuring a Banner

To configure an Easy VPN server to push a banner to an Easy VPN remote device, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** {*group-name*}

4. **banner c** {*banner-text*} **c**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp client configuration group** {*group-name*} **Example:** `Router (config)# crypto isakmp client configuration group Group1`	Specifies to which group a policy profile will bedefined and enters crypto ISAKMP group configuration mode.
Step 4	**banner c** {*banner-text*} **c** **Example:** `Router (config-isakmp-group)# banner c The quick brown fox jumped over the lazy dog c`	Specifies the text of the banner.

Configuring Auto Upgrade

To configure an Easy VPN server to provide an automated mechanism to make software and firmware upgrades automatically available to an Easy VPN remote device, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** {*group-name*}
4. **auto-update client** {*type-of-system*} {**url** *url*} {**rev** *review-version*}

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp client configuration group** {*group-name*} **Example:** `Router (config)# crypto isakmp client configuration group Group2`	Specifies to which group a policy profile will bedefined and enters crypto ISAKMP group configuration mode.
Step 4	**auto-update client** {*type-of-system*} {**url** *url*} {**rev** *review-version*} **Example:** `Router (config-isakmp-group)# auto-update client Win2000 url http:www.ourcompanysite.com/newclient rev 3.0.1(Rel), 3.1(Rel)`	Configures auto-update parameters for an Easy VPN remote device.

Configuring Browser Proxy

To configure an EasyVPN server so that the Easy VPN remote device can access resources on the corporate network when using Cisco IOS VPN Client software, perform the following steps. With this configuration, the user does not have to manually modify the proxy settings of his or her web browser when connecting and does not have to manually revert the proxy settings when disconnecting.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration browser-proxy** {*browser-proxy-name*}
4. **proxy** {*proxy-parameter*}

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp client configuration browser-proxy** {*browser-proxy-name*} **Example:** `Router (config)# crypto isakmp client configuration browser-proxy bproxy`	Configures browser-proxy parameters for an Easy VPN remote device and enters ISAKMP Browser Proxy configuration mode.
Step 4	**proxy** {*proxy-parameter*} **Example:** `Router (config-ikmp-browser-proxy)# proxy auto-detect`	Configures proxy parameters for an Easy VPN remote device.

Configuring the Pushing of a Configuration URL Through a Mode-Configuration Exchange

To configure an Easy VPN server to push a configuration URL through a Mode-Configuration Exchange, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** {*group-name*}
4. **configuration url** {*url*}
5. **configuration version** {*version-number*}

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp client configuration group** {*group-name*} **Example:** `Router (config)# crypto isakmp client configuration group Group1`	Specifies to which group a policy profile will be defined and enters crypto ISAKMP group configuration mode.
Step 4	**configuration url** {*url*} **Example:** `Router (config-isakmp-group)# configuration url http://10.10.88.8/easy.cfg`	Specifies the URL the remote device must use to get the configuration from the server. • The URL must be a non-NULL terminated ASCII string that specifies the complete path of the configuration file.
Step 5	**configuration version** {*version-number*} **Example:** `Router (config-isakmp-group)# configuration version 10`	Specifies the version of the configuration. • The version number will be an unsigned integer in the range 1 through 32767.

Configuring Per User AAA Download with PKI—Configuring the Crypto PKI Trustpoint

To configure a AAA server to push user attributes to a remote device, perform the following steps.

Prerequisites

Before configuring a AAA server to push user attributes to a remote device, you must have configured AAA. The crypto PKI trustpoint must also be configured (see the first configuration task below). It is preferable that the trustpoint configuration contain the **authorization username** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*

5. **revocation-check none**
6. **rsakeypair** *key-label*
7. **authorization username** {**subjectname** *subjectname*}
8. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router (config)# crypto pki trustpoint ca-server`	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:** `Router (config-ca-trustpoint)# enrollment url http://10.7.7.2:80`	Specifies the URL of the certification authority (CA) server to which to send enrollment requests.
Step 5	**revocation-check none** **Example:** `Router (config-ca-trustpoint)# revocation-check none`	Checks the revocation status of a certificate.
Step 6	**rsakeypair** *key-label* **Example:** `Router (config-ca-trustpoint)# rsakeypair rsa-pair`	Specifies which key pair to associate with the certificate.
Step 7	**authorization username** {**subjectname** *subjectname*} **Example:** `Router (config-ca-trustpoint)# authorization username subjectname commonname`	Specifies the parameters for the different certificate fields that are used to build the AAA username.
Step 8	**exit** **Example:** `Router (config-ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.

Configuring the Actual Per User AAA Download with PKI

To configure the actual per-user download with PKI, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp policy** *priority*
4. **group** {**1** | **2**}
5. **exit**
6. **crypto isakmp profile** *profile-name*
7. **match certificate** *certificate-map*
8. **client pki authorization list** *listname*
9. **client configuration address** {*initiate* | *respond*}
10. **virtual-template** *template-number*
11. **exit**
12. **crypto ipsec transform-set** [*transform-set-name transform1*] [*transform2*] [*transform3*] [*transform4*]
13. **crypto ipsec profile** *name*
14. **set transform-set** *transform-set-name*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp policy** *priority* **Example:** `Router (config)# crypto isakmp policy 10`	Defines an IKE policy and enters ISAKMP policy configuration mode.
Step 4	**group** {**1**	**2**} **Example:** `Router (config-isakmp-policy)# group 2`	Specifies the Diffie-Hellman group identifier within an IKE policy.

	Command	Purpose
Step 5	**exit** **Example:** `Router (config-isakmp-policy)# exit`	Exits ISAKMP policy configuration mode.
Step 6	**crypto isakmp profile** *profile-name* **Example:** `Router (config)# crypto isakmp profile ISA-PROF`	Defines an ISAKMP profile and audits IPsec user sessions and enters crypto ISAKMP profile configuration mode.
Step 7	**match certificate** *certificate-map* **Example:** `Router (config-isakmp-profile)# match certificate cert_map`	Assigns an ISAKMP profile to a peer on the basis of the contents of arbitrary fields in the certificate.
Step 8	**client pki authorization list** *listname* **Example:** `Router (config-isakmp-profile)# client pki authorization list usrgrp`	Specifies the authorization list of AAA servers that will be used for obtaining per-user AAA attributes on the basis of the username constructed from the certificate.
Step 9	**client configuration address** {**initiate** | **respond**} **Example:** `Router (config-isakmp-profile)# client configuration address respond`	Configures IKE configuration mode in the ISAKMP profile.
Step 10	**virtual-template** *template-number* **Example:** `Router(config-isakmp-profile)# virtual-template 2`	Specifies which virtual template will be used to clone virtual access interfaces.
Step 11	**exit** **Example:** `Router(config-isakmp-profile)# exit`	Exits crypto ISAKMP profile configuration mode.
Step 12	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*] **Example:** `Router (config)# crypto ipsec transform-set trans2 esp-3des esp-sha-hmac1`	Defines a transform set—an acceptable combination of security protocols and algorithms.

	Command	Purpose
Step 13	**crypto ipsec profile** *name* **Example:** `Router (config)# crypto ipsec profile IPSEC_PROF`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers.
Step 14	**set transform-set** *transform-set-name* **Example:** `Router (config)# set transform-set trans2`	Specifies which transform sets can be used with the crypto map entry.

Configuring Per-User Attributes on a Local Easy VPN AAA Server

To configure per-user attributes on a local Easy VPN AAA server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa attribute list** *list-name*
4. **attribute type** *name value* [**service** *service*] [**protocol** *protocol*]
5. **exit**
6. **crypto isakmp client configuration group** *group-name*
7. **crypto aaa attribute list** *list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa attribute list** *list-name* **Example:** `Router(config)# aaa attribute list list1`	Defines a AAA attribute list locally on a router and enters attribute list configuration mode.

	Command or Action	Purpose
Step 4	**attribute type** *name value* [**service** *service*] [**protocol** *protocol*] **Example:** `Router(config-attr-list)# attribute type attribute xxxx service ike protocol ip`	Defines an attribute type that is to be added to an attribute list locally on a router.
Step 5	**exit** **Example:** `Router(config-attr-list)# exit`	Exits attribute list configuration mode.
Step 6	**crypto isakmp client configuration group** *group-name* **Example:** `Router (config)# crypto isakmp client configuration group group1`	Specifies to which group a policy profile will be defined and enters ISAKMP group configuration mode.
Step 7	**crypto aaa attribute list** *list-name* **Example:** `Router (config-isakmp-group)# crypto aaa attribute list listname1`	Defines a AAA attribute list locally on a router.

Enabling Easy VPN Syslog Messages

To enable Easy VPN syslog messages on a server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto logging ezvpn group** *group-name*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto logging ezvpn** [*group group-name*] **Example:** `Router (config)# crypto logging ezvpn group group1`	Enables Easy VPN syslog messages on a server. • The **group** keyword and *group-name* argument are optional. If a group name is not provided, syslog messages are enabled for all Easy VPN connections to the server. If a group name is provided, syslog messages are enabled for that particular group only.

Defining a CPP Firewall Policy Push Using a Local AAA Server

To define a CPP firewall policy push on a server to allow or deny a tunnel on the basis of whether a remote device has a required firewall for a local AAA server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client firewall** {*policy-name*} {**required** | **optional**} {*firewall-type*}
4. **policy** {**check-presence** | **central-policy-push** {**access-list** {**in** | **out**} *access-list-name* | *access-list-number*}}

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 3	**crypto isakmp client firewall** {*policy-name*} {**required** | **optional**} {*firewall-type*} **Example:** `Router (config)# crypto isakmp client firewall hw-client-g-cpp required Cisco-Security-Agent`	Defines the CPP firewall push policy on a server and enters ISAKMP client firewall configuration mode. The arguments and keywords are as follows: • *policy-name*—Uniquely identifies a policy. A policy name can be associated with the Easy VPN client group configuration of the server (local group configuration) or on the AAA server. • **required**—Policy is mandatory. If the CPP policy is defined as mandatory and is included in the Easy VPN server configuration, the tunnel setup is allowed only if the client confirms this policy. Otherwise, the tunnel is terminated. • **optional**—Policy is optional. If the CPP policy is defined as optional, and is included in the Easy VPN server configuration, the tunnel setup is continued even if the client does not confirm the defined policy. • *firewall-type*—Type of firewall (see the **crypto isakmp client firewall** command for a list of firewall types).
Step 4	**policy** {**check-presence** | **central-policy-push** {**access-list** {**in** | **out**} *access-list-name* | *access-list-number*}} **Example:** `Router (config-ikmp-client-fw)# policy central-policy-push access-list out acl1` `or` `Router (config-ikmp-client-fw)# policy check-presence`	Defines the CPP firewall policy push. The arguments and keywords are as follows: • **check-presence**—Denotes that the server should check for the presence of the specified firewall as shown by the value of the *firewall-type* argument on the client. • **central-policy-push**—The configuration following this keyword specifies the actual policy, such as the input and output access lists that have to be applied by the client firewall, which is of the type specified by the value of the *firewall-type* argument. • **access-list** {**in** | **out**}—Defines the inbound and outbound access lists. • *access-list-name* | *access-list-number*—Name or number of the access list.

What to Do Next

Apply the CPP firewall policy push to the configured group.

Applying a CPP Firewall Policy Push to the Configuration Group

Now that the CPP firewall policy push has been defined, it must be applied to the configuration group by performing the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** {*group-name*}
4. **firewall policy** {*policy-name*}

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp client configuration group** {*group-name*} **Example:** `Router (config)# crypto isakmp client configuration group hw-client-g`	Specifies to which group a policy profile will be defined and enters ISAKMP group configuration mode.
Step 4	**firewall policy** {*policy-name*} **Example:** `Router (crypto-isakmp-group)# firewall policy hw-client-g-cpp`	Specifies the CPP firewall push policy name for the crypto ISAKMP client configuration group on a local authentication, AAA server.

Defining a CPP Firewall Policy Push Using a Remote AAA Server

To define a CPP firewall policy push using a remote AAA server, see the section "Defining a CPP Firewall Policy Push Using a Local AAA Server." The steps are the same for this configuration.

What to Do Next

After defining the CPP firewall policy push, you should add the VSA cpp-policy under the group definition.

Adding the VSA CPP-Policy Under the Group Definition

To add the the VSA cpp-policy under the group definition that is defined in RADIUS, perform the following step.

SUMMARY STEPS

1. Add the VSA cpp-policy under the group definition that is defined in RADIUS.

DETAILED STEPS

	Command	Purpose
Step 1	Add the VSA “cpp-policy” under the group definition that is defined in RADIUS. **Example:** ipsec:cpp-policy=”Enterprise Firewall”	Defines the CPP firewall push policy for a remote server.

Verifying CPP Firewall Policy Push

To verify the CPP firewall push policy on a local or remote AAA server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug crypto isakmp**

DETAILED STEPS

	Command	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`debug crypto isakmp` **Example:** `Router# debug crypto isakmp`	Displays messages about IKE events.

Configuring Password Aging

To configure Password Aging so that the Easy VPN client is notified if the password has expired, perform the following steps.

Restrictions

The following restrictions apply to the Password Aging feature:

- It works only with VPN software clients. It does not work with VPN client hardware.
- It works only with RADIUS servers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** {*list-name*} **password-expiry** *method1* [*method2...*]
5. **radius-server host** {*ip-address*} **auth-port** *port-number* **acct-port** *port-number* **key** *string*}
6. Configure the ISAKMP profile
7. **client authentication list** {*list-name*}

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables AAA.
Step 4	**aaa authentication login** {*list-name*} **password-expiry** *method1* [*method2...*] **Example:** `Router (config)# aaa authentication login userauth paswd-expiry group radius`	Configures the authentication list so that the Password Aging feature is enabled.
Step 5	**radius-server host** {*ip-address*} **auth-port** *port-number* **acct-port** *port-number* **key** *string* **Example:** `Router (config)# radius-server host 172.19.217.96 255.255.255.0 auth-port 1645 acct-port 1646 key cisco radius-server vsa send authentication`	Configures the RADIUS server.

	Command	Purpose
Step 6	Configure the ISAKMP profile. **Example:** see the section "Configuring Password Aging: Example"	Configures the ISAKMP profile and enters ISAKMP profile configuration mode (see the section "Configuring Password Aging: Example").
Step 7	`client authentication list {list-name}` **Example:** `Router (config-isakmp-profile)# client authentication list userauth`	Configures IKE extended authentication (Xauth) in an ISAKMP profile and includes the authentication list that was defined above.

Configuring Split DNS

To configure Split DNS, perform the following steps.

Prerequisites

Before the Split DNS feature can work, the following commands should have been configured on the Easy VPN remote:

- **ip dns server**
- **ip domain-lookup**

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** *group-name*
4. **dns** *primary-server secondary-server*
5. **split-dns** *domain-name*

DETAILED STEPS

	Command	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 3	**crypto isakmp client configuration group** {*group-name* | **default**} **Example:** `Router (config)# crypto isakmp client configuration group group1`	Specifies the policy profile of the group that will be defined and enters ISAKMP group configuration mode. • If no specific group matches and a default group is defined, users will automatically be given the policy of a default group.
Step 4	**dns** *primary-server secondary-server* **Example:** `Router (config-isakmp-group)# dns 10.2.2.2 10.3.3.3`	Specifies the primary and secondary DNS servers for the group.
Step 5	**split-dns** *domain-name* **Example:** `Router (config-isakmp-group)# split-dns green.com`	Specifies a domain name that must be tunneled or resolved to the private network.

Verifying Split DNS

To verify a split DNS configuration, perform the following steps (the **show** commands can be used one at a time or together).

SUMMARY STEPS

1. **enable**
2. **show ip dns name-list** [*name-list-number*]
3. **show ip dns view** [**vrf** *vrf-name*] [**default** | *view-name*]
4. **show ip dns view-list** [*view-list-name*]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip dns name-list** [*name-list-number*] **Example:** `Router# show ip dns name-list 1`	Displays information about DNS name lists.

	Command	Purpose
Step 3	**show ip dns view** [**vrf** *vrf-name*] [**default** | *view-name*] **Example:** `Router# show ip dns view default`	Displays information about DNS views.
Step 4	**show ip dns view-list** [*view-list-name*] **Example:** `Router# show ip dns view-list ezvpn-internal-viewlist`	Displays information about DNS view lists.

Monitoring and Maintaining Split DNS

To monitor and maintain the split DNS configuration on Easy VPN remote devices, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug ip dns name-list**
3. **debug ip dns view**
4. **debug ip dns view-list**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip dns name-list** **Example:** `Router# debug ip dns name-list`	Enables debugging output for Domain Name System (DNS) name-list events.
Step 3	**debug ip dns view** **Example:** `Router# debug ip dns view`	Enables debugging output for DNS view events.
Step 4	**debug ip dns view-list** **Example:** `Router# debug ip dns view-list`	Enables debugging output for DNS view-list events.

Configuring an Easy VPN Server to Obtain an IP Address from a DHCP Server

When the Easy VPN server selects the method for address assignment, it does so in the following order of precedence:

1. Selects the Framed IP address
2. Uses the IP address from the authentication server (group/user)
3. Uses the global IKE address pools
4. Uses DHCP

Note To enable the Easy VPN server to obtain an IP address from a DHCP server, remove other address assignments.

To configure an Easy VPN server to obtain an IP address from a DHCP server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** *group-name*
4. **dhcp server** {*ip-address* | *hostname*}
5. **dhcp timeout** *time*
6. **dhcp giaddr** *scope*

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp client configuration group** *group-name* **Example:** `Router (config)# crypto isakmp client configuration group group1`	Specifies to which group a policy profile will be defined. **Note** Entering this command places the CLI in ISAKMP group configuration mode. From this mode, you can use subcommands to specify characteristics for the group policy.
Step 4	**dhcp server** {*ip-address* | *hostname*} **Example:** `Router (config-isakmp-group)# dhcp server 10.10.1.2`	Specifies a primary (and backup) DHCP server to allocate IP addresses to MS users entering a particular public data network (PDN) access point.

Step 5	**dhcp timeout** *time* **Example:** `Router (config-isakmp-group)# dhcp timeout 6`	Sets the wait time in seconds before the next DHCP server on the list is tried.
Step 6	**dhcp giaddr** *scope* **Example:** `Router (config-isakmp-group)# dhcp giaddr 10.1.1.4`	Specifies the giaddr for the DHCP scope.

Verifying DHCP Client Proxy

To verify your DHCP client proxy configuration, perform the following steps (use the **show** commands one at a time or together).

SUMMARY STEPS

1. **enable**
2. **show dhcp lease**
3. **show ip dhcp pool**
4. **show ip dhcp binding**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 1	**show dhcp lease** **Example:** `Router# show dhcp lease`	Displays information about the DHCP address pools. **Note** Use this command when an external DHCP is used.
Step 2	**show ip dhcp pool** **Example:** `Router# show ip dhcp pool`	Displays information about the DHCP address pools. **Note** This command is applicable only when the Easy VPN server is also the DHCP server (generally not the case because in most cases, the DHCP server is an external server).
Step 3	**show ip dhcp binding** **Example:** `Router# show ip dhcp binding`	Displays address bindings on the DHCP server. **Note** This command is applicable only when the Easy VPN server is also the DHCP server (generally not the case because in most cases, the DHCP server is an external server).

Monitoring and Maintaining DHCP Client Proxy

To monitor and maintain your DHCP client proxy configuration, perform the following steps (use the **debug** commands one at a time or together).

SUMMARY STEPS

1. **enable**
2. **debug crypto isakmp**
3. **debug dhcp**
4. **debug dhcp detail**
5. **debug ip dhcp server events**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays messages about Internet Key Exchange (IKE) event.
Step 3	**debug dhcp** **Example:** `Router# debug dhcp`	Reports server events, like address assignments and database updates.
Step 4	**debug dhcp detail** **Example:** `Router# debug dhcp detail`	Displays detailed DHCP debugging information.
Step 5	**debug ip dhcp server events** **Example:** `Router# debug ip dhcp server events`	Reports server events, like address assignments and database updates. **Note** This command is applicable only when the Easy VPN server is also the DHCP server (generally not the case because in most cases, the DHCP server is an external server).

Configuring cTCP

To enable cTCP, perform the following steps on your Easy VPN server.

Prerequisites

Before configuring cTCP, you should have configured crypto IPsec.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ctcp port [*port-number*]**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ctcp port** [*port-number*] **Example:** `Router (config)# crypto ctcp port 120`	Configures cTCP encapsulation for Easy VPN. • Up to 10 port numbers can be configured. • If the *port-number* argument is not configured, cTCP is enabled on port 80 by default.

Verifying cTCP

To verify your cTCP configuration, perform the following steps (the **show** commands can be used one at a time or together).

SUMMARY STEPS

1. **enable**
2. **show crypto ctcp** [**peer** *ip-address*]

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto ctcp** [**peer** *ip-address*] **Example:** `Router# show crypto ctcp peer 10.76.235.21`	Displays information about a specific cTCP peer.

Monitoring and Maintaining a cTCP Configuration

To monitor and maintain your cTCP configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug crypto ctcp**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto ctcp** **Example:** `Router# debug crypto ctcp`	Displays information about a cTCP session.

Clearing a cTCP Configuration

To clear a cTCP configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **clear crypto ctcp** [**peer** *ip-address*]

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto ctcp** [**peer** *ip-address*] **Example:** `Router# clear crypto ctcp peer 10.76.23.21`	Displays information about a cTCP session.

Troubleshooting a cTCP Configuration

To troubleshoot a cTCP configuration, perform the following steps.

SUMMARY STEPS

1. Ensure that the cTCP session is in the CTCP_ACK_RECEIVED state.
2. If the cTCP session is not in the CTCP_ACK_RECEIVED state, enable the **debug crypto ctcp** command.
3. If no cTCP bugs are seen, ensure that the firewall is allowing the cTCP packets to get to the server.
4. If the firewall configuration is correct, debugging is enabled, and you do not see any cTCP debugs on your console, you must find out why the cTCP port on the router is not receiving packets.

DETAILED STEPS

Step 1 To ensure that the cTCP session is in the CTCP_ACK_RECEIVED state, use the **show crypto ctcp** command.

Step 2 If the cTCP session is not in the CTCP_ACK_RECEIVED state, enable the **debug crypto ctcp** command and then try using the **show crypto ctcp** command again.

Step 3 If no cTCP bugs are seen, ensure that the firewall is allowing the cTCP packets to get to the server (check the firewall configuration).

Step 4 If the firewall configuration is correct, debugging is enabled, and you do not see any cTCP debugs on your console, you must find out why the cTCP port on the router is not receiving packets. If you do not see any cTCP debugs and a cTCP session has not been set up, there is a possibility that cTCP packets that are actually TCP packets could have been delivered to a TCP stack instead of to the cTCP port. By enabling the **debug ip packet** and **debug ip tcp packet** commands, you may be able to determine whether the packet is being given to the TCP stack.

Configuration Examples for Easy VPN Server

This section provides the following configuration examples:

Configuring Cisco IOS for Easy VPN Server: Example

The following example shows how to define group policy information locally for mode configuration. In this example, a group name is named “cisco” and another group name is named “default.” The policy is enforced for all users who do not offer a group name that matches “cisco.”

```
! Enable policy look-up via AAA. For authentication and authorization, send requests to
! RADIUS first, then try local policy.
aaa new-model
aaa authentication login userlist group radius local
aaa authorization network grouplist group radius local
enable password XXXX
!
username cisco password 0 cisco
clock timezone PST -8
ip subnet-zero
! Configure IKE policies, which are assessed in order so that the first policy that
matches the proposal of the client will be used.
crypto isakmp policy 1
 group 2
!
crypto isakmp policy 3
 hash md5
 authentication pre-share
 group 2
crypto isakmp identity hostname
!
! Define “cisco” group policy information for mode config push.
crypto isakmp client configuration group cisco
 key cisco
 dns 10.2.2.2 10.2.2.3
 wins 10.6.6.6
 domain cisco.com
 pool green
 acl 199
! Define default group policy for mode config push.
crypto isakmp client configuration group default
 key cisco
 dns 10.2.2.2 10.3.2.3
 pool green
 acl 199
!
!
crypto ipsec transform-set dessha esp-des esp-sha-hmac
!
crypto dynamic-map mode 1
 set transform-set dessha
!
! Apply mode config and xauth to crypto map “mode.” The list names that are defined here
! must match the list names that are defined in the AAA section of the config.
crypto map mode client authentication list userlist
crypto map mode isakmp authorization list grouplist
crypto map mode client configuration address respond
crypto map mode 1 ipsec-isakmp dynamic mode
```

```
!
!
controller ISA 1/1
!
!
interface FastEthernet0/0
 ip address 10.6.1.8 255.255.0.0
 ip route-cache
 ip mroute-cache
 duplex auto
 speed auto
 crypto map mode
!
interface FastEthernet0/1
 ip address 192.168.1.28 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
! Specify IP address pools for internal IP address allocation to clients.
ip local pool green 192.168.2.1 192.168.2.10
ip classless
ip route 0.0.0.0 0.0.0.0 10.6.0.1
!
! Define access lists for each subnet that should be protected.
access-list 199 permit ip 192.168.1.0 0.0.0.255 any
access-list 199 permit ip 192.168.3.0 0.0.0.255 any
!
! Specify a RADIUS server host and configure access to the server.
radius-server host 192.168.1.1 auth-port 1645 acct-port 1646 key XXXXX
radius-server retransmit 3
!
!
line con 0
 exec-timeout 0 0
 length 25
 transport input none
line aux 0
line vty 5 15
!
```

RADIUS Group Profile with IPsec AV Pairs: Example

The following is an example of a standard RADIUS group profile that includes RADIUS IPsec AV pairs. To get the group authorization attributes, “cisco” must be used as the password.

```
client_r Password = "cisco"
 Service-Type = Outbound

 cisco-avpair = "ipsec:tunnel-type*ESP"
 cisco-avpair = "ipsec:key-exchange=ike"
 cisco-avpair = "ipsec:tunnel-password=lab"
 cisco-avpair = "ipsec:addr-pool=pool1"
 cisco-avpair = "ipsec:default-domain=cisco"
 cisco-avpair = "ipsec:inacl=101"
 cisco-avpair = "ipsec:access-restrict=fastethernet 0/0"
 cisco-avpair = "ipsec:group-lock=1"
 cisco-avpair = "ipsec:dns-servers=10.1.1.1 10.2.2.2"
 cisco-avpair = "ipsec:firewall=1"
 cisco-avpair = "ipsec:include-local-lan=1"
 cisco-avpair = "ipsec:save-password=1"
 cisco-avpair = "ipsec:wins-servers=10.3.3.3 10.4.4.4"
```

```
 cisco-avpair = "ipsec:split-dns=green.com"
 cisoc-avpair = "ipsec:ipsec-backup-gateway=10.1.1.1"
 cisoc-avpair = "ipsec:ipsec-backup-gateway=10.1.1.2"
 cisoc-avpair = "ipsec:pfs=1"
 cisco-avpair = "ipsec:cpp-policy="Enterprise Firewall"
 cisco-avpair = “ipsec:auto-update=”Win http://abc.com 4.0.1”
 cisco-avpair = “ipsec:browser-proxy=bproxy_profile_A”
 cisco-avpair = “ipsec:xauth-banner=”Xauth banner text here”
```

RADIUS User Profile with IPsec AV Pairs: Example

The following is an example of a standard RADIUS user profile that includes RADIUS IPsec AV pairs. These user attributes will be obtained during Xauth.

```
ualluall Password = "uall1234"
        cisco-avpair = "ipsec:user-vpn-group=unity"
        cisco-avpair = "ipsec:user-include-local-lan=1"
        cisco-avpair = "ipsec:user-save-password=1"
        Framed-IP-Address = 10.10.10.10
```

Backup Gateway with Maximum Logins and Maximum Users: Example

The following example shows that five backup gateways have been configured, that the maximum users have been set to 250, and that maximum logins have been set to 2:

```
crypto isakmp client configuration group sdm
 key 6 RMZPPMRQMSdiZNJg`EBbCWTKSTi\d[
 pool POOL1
 acl 150
 backup-gateway 172.16.12.12
 backup-gateway 172.16.12.13
 backup-gateway 172.16.12.14
 backup-gateway 172.16.12.130
 backup-gateway 172.16.12.131
 max-users 250
 max-logins 2
```

Easy VPN with an IPsec Virtual Tunnel Interface: Example

The following output shows that Easy VPN has been configured with an IPsec virtual tunnel interface.

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
!
aaa new-model
!
!
aaa authentication login default local
```

```
aaa authorization network default local
!
aaa session-id common
!
resource policy
!
clock timezone IST 0
ip subnet-zero
ip cef
no ip domain lookup
no ip dhcp use vrf connected
!
username lab password 0 lab
!
crypto isakmp policy 3
 authentication pre-share
 group 2
crypto isakmp xauth timeout 90

!
crypto isakmp client configuration group easy
 key cisco
 domain foo.com
 pool dpool
 acl 101
crypto isakmp profile vi
   match identity group easy
   isakmp authorization list default
   client configuration address respond
   client configuration group easy
   virtual-template 1
!
!
crypto ipsec transform-set set esp-3des esp-sha-hmac
!
crypto ipsec profile vi
 set transform-set set
 set isakmp-profile vi
!
!
interface Loopback0
 ip address 10.4.0.1 255.255.255.0
!
interface Ethernet0/0
 ip address 10.3.0.2 255.255.255.0
 no keepalive
 no cdp enable
interface Ethernet1/0
 no ip address
 no keepalive
 no cdp enable
!
interface Virtual-Template1 type tunnel
 ip unnumbered Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile vi
!
ip local pool dpool 10.5.0.1 10.5.0.10
!
ip classless
ip route 10.2.0.0 255.255.255.0 10.3.0.1
no ip http server
no ip http secure-server
!
```

```
!
access-list 101 permit ip 10.4.0.0 0.0.0.255 any
no cdp run
!
!
line con 0
line aux 0
line vty 0 4
!
end
```

Pushing a Configuration URL Through a Mode-Configuration Exchange: Examples

The following **show crypto ipsec client ezvpn** command output displays the mode configuration URL location and version:

```
Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 5

Tunnel name : branch
Inside interface list: Vlan1
Outside interface: FastEthernet0
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 172.16.1.209
Mask: 255.255.255.255
Default Domain: cisco.com
Save Password: Allowed
Configuration URL [version]: tftp://172.16.30.2/branch.cfg [11]
Config status: applied, Last successfully applied version: 11
Current EzVPN Peer: 192.168.10.1
```

The following **show crypto isakmp peers config** command output displays all manageability information that is sent by the remote device.

```
Router# show crypto isakmp peers config

Client-Public-Addr=192.168.10.2:500; Client-Assigned-Addr=172.16.1.209;
Client-Group=branch; Client-User=branch; Client-Hostname=branch.; Client-Platform=Cisco
1711; Client-Serial=FOC080210E2 (412454448); Client-Config-Version=11;
Client-Flash=33292284; Client-Available-Flash=10202680; Client-Memory=95969280;
Client-Free-Memory=14992140; Client-Image=flash:c1700-advipservicesk9-mz.ef90241;
Client-Public-Addr=192.168.10.3:500; Client-Assigned-Addr=172.16.1.121;
Client-Group=store; Client-User=store; Client-Hostname=831-storerouter.;
Client-Platform=Cisco C831; Client-Serial=FOC08472UXR (1908379618);
Client-Config-Version=2; Client-Flash=24903676; Client-Available-Flash=5875028;
Client-Memory=45298688; Client-Free-Memory=6295596;
Client-Image=flash:c831-k9o3y6-mz.ef90241
```

Per User AAA Policy Download with PKI: Example

The following output shows that the Per User AAA Policy Download with PKI feature has been configured on the Easy VPN server.

```
Router# show running-config

Building configuration...
```

```
Current configuration : 7040 bytes
!
! Last configuration change at 21:06:51 UTC Tue Jun 28 2005
!
version 12.4
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname GEN
!
boot-start-marker
boot-end-marker
!
!
aaa new-model
!
!
aaa group server radius usrgrppki
 server 10.76.248.201 auth-port 1645 acct-port 1646
!
aaa authentication login xauth group usrgrppki
aaa authentication login usrgrp group usrgrppki
aaa authorization network usrgrp group usrgrppki
!
aaa session-id common
!
resource policy
!
ip subnet-zero
!
!
ip cef
!
!
ip address-pool local
!
!
crypto pki trustpoint ca-server
 enrollment url http://10.7.7.2:80
 revocation-check none
 rsakeypair rsa-pair
 ! Specify the field within the certificate that will be used as a username to do a
per-user AAA lookup into the RADIUS database. In this example, the contents of the
 commonname will be used to do a AAA lookup. In the absence of this statement, by default
 the contents of the “unstructured name” field in the certificate is used for AAA lookup.
 authorization username subjectname commonname
!
!
crypto pki certificate map CERT-MAP 1
 subject-name co yourname
 name co yourname
!
crypto pki certificate chain ca-server
 certificate 02
  308201EE 30820157 A0030201 02020102 300D0609 2A864886 F70D0101 04050030
  14311230 10060355 04031309 63612D73 65727665 72301E17 0D303530 36323832
  30303731 345A170D 30363036 32383230 30373134 5A301531 13301106 092A8648
  86F70D01 09021604 47454E2E 30819F30 0D06092A 864886F7 0D010101 05000381
  8D003081 89028181 00ABF8F0 FDFFDF8D F22098D6 A48EE0C3 F505DD96 C0022EA4
  EAB95EE8 1F97F450 990BB0E6 F2B7151F C5C79391 93822FE4 DEE5B00C A03412BB
  9B715AAD D6C31F93 D8802658 AF9A8866 63811942 913D0C02 C3E328CC 1C046E94
```

```
  F73B7C1A 4497F86E 74A627BC B809A3ED 293C15F2 8DCFA217 5160F9A4 09D52044
  350F85AF 08B357F5 D7020301 0001A34F 304D300B 0603551D 0F040403 0205A030
  1F060355 1D230418 30168014 F9BC4498 3DA4D51D 451EFEFD 5B1F5F73 8D7B1C9B
  301D0603 551D0E04 1604146B F6B2DFD1 1FE237FF 23294129 E55D9C48 CCB04630
  0D06092A 864886F7 0D010104 05000381 81004AFF 2BE300C1 15D0B191 C20D06E0
  260305A6 9DF610BB 24211516 5AE73B62 78E01FE4 0785776D 3ADFA3E2 CE064432
  1C93E82D 93B5F2AB 9661EDD3 499C49A8 F87CA553 9132F239 1D50187D 21CC3148
  681F5043 2F2685BC F544F4FF 8DF535CB E55B5F36 31FFF025 8969D9F8 418C8AB7
  C569B022 46C3C63A 22DD6516 C503D6C8 3D81
  quit
 certificate ca 01
  30820201 3082016A A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  14311230 10060355 04031309 63612D73 65727665 72301E17 0D303530 36323832
  30303535 375A170D 30383036 32373230 30353537 5A301431 12301006 03550403
  13096361 2D736572 76657230 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 BA1A4413 96339C6B D36BD720 D25C9A44 E0627A29 97E06F2A
  69B268ED 08C7144E 7058948D BEA512D4 40588B87 322C5D79 689427CA 5C54B3BA
  82FAEC53 F6AC0B5C 615D032C 910CA203 AC6AB681 290D9EED D31EB185 8D98E1E7
  FF73613C 32290FD6 A0CBDC40 6E4D6B39 DE1D86BA DE77A55E F15299FF 97D7C185
  919F81C1 30027E0F 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301F 0603551D 23041830 168014F9
  BC44983D A4D51D45 1EFEFD5B 1F5F738D 7B1C9B30 1D060355 1D0E0416 0414F9BC
  44983DA4 D51D451E FEFD5B1F 5F738D7B 1C9B300D 06092A86 4886F70D 01010405
  00038181 003EF397 F4D98BDE A4322FAF 4737800F 1671F77E BD6C45AE FB91B28C
  F04C98F0 135A40C6 635FDC29 63C73373 5D5BBC9A F1BBD235 F66CE1AD 6B4BFC7A
  AB18C8CC 1AB93AF3 7AC67436 930E9C81 F43F7570 A8FE09AE 3DEA01D1 DA6BD0CB
  83F9A77F 1DFAFE5E 2F1F206B F1FDD8BE 6BB57A3C 8D03115D B1F64A3F 7A7557C1
  09B0A34A DB
  quit
!
!
crypto isakmp policy 10
 group 2
crypto isakmp keepalive 10
crypto isakmp profile ISA-PROF
   match certificate CERT-MAP
   isakmp authorization list usrgrp
   client pki authorization list usrgrp
   client configuration address respond
   client configuration group pkiuser
   virtual-template 2
!
!
crypto ipsec transform-set trans2 esp-3des esp-sha-hmac
!
crypto ipsec profile IPSEC_PROF
 set transform-set trans2
!
crypto ipsec profile ISC_IPSEC_PROFILE_1
 set transform-set trans2
!
!
crypto call admission limit ike sa 40
!
!
interface Loopback0
 ip address 10.3.0.1 255.255.255.255
 no ip route-cache cef
 no ip route-cache
!
interface Loopback1
 ip address 10.76.0.1 255.255.255.255
 no ip route-cache cef
 no ip route-cache
```

```
!
interface Ethernet3/0
 ip address 10.76.248.209 255.255.255.255
 no ip route-cache cef
 no ip route-cache
 duplex half
!
!
interface Ethernet3/2
 ip address 10.2.0.1 255.255.255.0
 no ip route-cache cef
 no ip route-cache
 duplex half
!
!
interface Serial4/0
 no ip address
 no ip route-cache cef
 no ip route-cache
 shutdown
 serial restart-delay 0
!
interface Serial4/1
 no ip address
 no ip route-cache cef
 no ip route-cache
 shutdown
 serial restart-delay 0
!
interface Serial4/2
 no ip address
 no ip route-cache cef
 no ip route-cache
 shutdown
 serial restart-delay 0
!
interface Serial4/3
 no ip address
 no ip route-cache cef
 no ip route-cache
 shutdown
 serial restart-delay 0
!
interface FastEthernet5/0
 ip address 10.9.4.77 255.255.255.255
 no ip route-cache cef
 no ip route-cache
 duplex half
!
interface FastEthernet6/0
 ip address 10.7.7.1 255.255.255.0
 no ip route-cache cef
 no ip route-cache
 duplex full
!
interface Virtual-Template1
 no ip address
!
interface Virtual-Template2 type tunnel
 ip unnumbered Loopback0
 tunnel source Ethernet3/2
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile IPSEC_PROF
!
```

```
router eigrp 20
 network 172.16.0.0
 auto-summary
!
ip local pool ourpool 10.6.6.6
ip default-gateway 10.9.4.1
ip classless
ip route 10.1.0.1 255.255.255.255 10.0.0.2
ip route 10.2.3.0 255.255.0.0 10.2.4.4
ip route 10.9.1.0 255.255.0.0 10.4.0.1
ip route 10.76.0.0 255.255.0.0 10.76.248.129
ip route 10.11.1.1 255.255.255.0 10.7.7.2
!
no ip http server
no ip http secure-server
!
!
logging alarm informational
arp 10.9.4.1 0011.bcb4.d40a ARPA
!
!
radius-server host 10.76.248.201 auth-port 1645 acct-port 1646 key cisco
!
control-plane
!
!
gatekeeper
 shutdown
!
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
!
!
end
```

Per-User Attributes on an Easy VPN Server: Example

The following example shows that per-user attributes have been configured on an Easy VPN server.

```
!

aaa new-model
!
!
aaa authentication login default local
aaa authentication login noAAA none
aaa authorization network default local
!
aaa attribute list per-group
 attribute type inacl "per-group-acl" service ike protocol ip mandatory
!
aaa session-id common
!
resource policy
!
ip subnet-zero
!
!
```

```
ip cef
!
!
username example password 0 example
!
!
crypto isakmp policy 3
 authentication pre-share
 group 2
crypto isakmp xauth timeout 90
!
crypto isakmp client configuration group PerUserAAA
 key cisco
 pool dpool
 crypto aaa attribute list per-group
!
crypto isakmp profile vi
 match identity group PerUserAAA
 isakmp authorization list default
 client configuration address respond
 client configuration group PerUserAAA
 virtual-template 1
!
!
crypto ipsec transform-set set esp-3des esp-sha-hmac
!
crypto ipsec profile vi
 set transform-set set
 set isakmp-profile vi
!
!
interface GigabitEthernet0/0
 description 'EzVPN Peer'
 ip address 192.168.1.1 255.255.255.128
 duplex full
 speed 100
 media-type rj45
 no negotiation auto
!
interface GigabitEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
 no negotiation auto

interface Virtual-Template1 type tunnel
 ip unnumbered GigabitEthernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile vi
!
ip local pool dpool 10.5.0.1 10.5.0.10
ip classless
!
no ip http server
no ip http secure-server
!
!
ip access-list extended per-group-acl
 permit tcp any any
 deny   icmp any any
logging alarm informational
logging trap debugging
```

```
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
line aux 0
 stopbits 1
line vty 0 4
!
!
end
```

Network Admission Control: Example

The following is output for an Easy VPN server that has been enabled with Network Admission Control.

Note Network Admission Control is supported on an Easy VPN server only when the server uses IPsec virtual interfaces. Network Admission Control is enabled on the virtual template interface and applies to all PC clients that use this virtual template interface.

```
Router# show running-config

Building configuration...

Current configuration : 5091 bytes
!
version 12.4
!
hostname Router
!

aaa new-model
!
!
aaa authentication login userlist local
!
aaa authentication eou default group radius
aaa authorization network hw-client-groupname local
aaa accounting update newinfo
aaa accounting network acclist start-stop broadcast group radius
aaa session-id common
!
!
! Note 1: EAPoUDP packets will use the IP address of the loopback interface when sending
the EAPoUDP hello to the Easy VPN client. Using the IP address ensures that the returning
EAPoUDP packets come back encrypted and are associated with the correct virtual access
interface. The ip admission (ip admission source-interface Loopback10) command is
optional. Instead of using this command, you can specify the IP address of the virtual
template to be an address in the inside network space as shown in the configuration of the
virtual template below in Note 2.
ip admission source-interface Loopback10
ip admission name test eapoudp inactivity-time 60
!
!
eou clientless username cisco
eou clientless password cisco
eou allow ip-station-id
```

```
eou logging
!
username lab password 0 lab
username lab@easy password 0 lab
!
!
crypto isakmp policy 3
  encr 3des
  authentication pre-share
  group 2
!
!
crypto isakmp key 0 cisco address 10.53.0.1
crypto isakmp client configuration group easy
  key cisco
  domain cisco.com
  pool dynpool
  acl split-acl
  group-lock
  configuration url tftp://10.13.0.9/Config-URL_TFTP.cfg
  configuration version 111
!
crypto isakmp profile vi
    match identity group easy
    client authentication list userlist
    isakmp authorization list hw-client-groupname
    client configuration address respond
    client configuration group easy
    accounting acclist
    virtual-template 2
!
crypto ipsec security-association lifetime seconds 120
crypto ipsec transform-set set esp-3des esp-sha-hmac
crypto ipsec transform-set aes-trans esp-aes esp-sha-hmac
crypto ipsec transform-set transform-1 esp-des esp-sha-hmac
crypto ipsec profile vi
  set security-association lifetime seconds 3600
  set transform-set set aes-trans transform-1
  set isakmp-profile vi
!
!
crypto dynamic-map dynmap 1
  set transform-set aes-trans transform-1
  reverse-route
!

interface Loopback10
  ip address 10.61.0.1 255.255.255.255
!
interface FastEthernet0/0
  ip address 10.13.11.173 255.255.255.255
  duplex auto
  speed auto
!
interface FastEthernet0/1
  ip address 10.55.0.1 255.255.255.255
  duplex auto
  speed auto
!
!
interface Virtual-Template2 type tunnel
```

```
! Note2: Use the IP address of the loopback10. This ensures that the EAPoUDP packets that
are attached to virtual-access interfaces that are cloned from this virtual template carry
the source address of the loopback address and that response packets from the VPN client
come back encrypted.
!
  ip unnumbered Loopback10
! Enable Network Admission Control for remote VPN clients.
  ip admission test
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile vi
!
!
ip local pool dynpool 172.16.2.65 172.16.2.70
ip classless
ip access-list extended ClientException
  permit ip any host 10.61.0.1
ip access-list extended split-acl
  permit ip host 10.13.11.185 any
  permit ip 10.61.0.0 255.255.255.255 any
  permit ip 10.71.0.0 255.255.255.255 any
  permit ip 10.71.0.0 255.255.255.255 10.52.0.0 0.255.255.255
  permit ip 10.55.0.0 255.255.255.255 any
!
ip radius source-interface FastEthernet0/0
access-list 102 permit esp any any
access-list 102 permit ahp any any
access-list 102 permit udp any any eq 21862
access-list 102 permit ospf any any
access-list 102 deny ip any any
access-list 195 deny ospf any any
access-list 195 permit ip 10.61.0.0 255.255.255.255 10.51.0.0 255.255.255.255
!
!
radius-server attribute 6 on-for-login-auth
radius-server attribute 8 include-in-access-req
radius-server host 10.13.11.185 auth-port 1645 acct-port 1646 key cisco
radius-server vsa send accounting
radius-server vsa send authentication
!
end
```

Configuring Password Aging: Example

The following example shows that password aging has been configured so that if the password expires, the Easy VPN client is notified.

```
Current configuration : 4455 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname xinl-gateway
!
boot-start-marker
boot system flash c2800nm-advsecurityk9-mz.124-7.9.T
boot-end-marker
!
!
aaa new-model
!
```

```
!
aaa authentication login USERAUTH passwd-expiry group radius aaa authorization network
branch local !
aaa session-id common
!

ip cef


username cisco privilege 15 secret 5 $1$A3HU$bCWjlkrEztDJx6JJzSnMV1 !
!
crypto isakmp policy 1
  encr 3des
  authentication pre-share
  group 2
crypto isakmp client configuration address-pool local dynpool !
crypto isakmp client configuration group branch
  key cisco
  domain cisco.com
  pool dynpool
!
!
crypto ipsec transform-set transform-1 esp-3des esp-sha-hmac !
crypto isakmp profile profile2
   client authentication list USERAUTH
   match identity group branch
   isakmp authorization list branch
   client configuration address respond
   virtual-template 1

crypto ipsec profile vi
  set transform-set transform-1


interface GigabitEthernet0/0
  description $ETH-LAN$$ETH-SW-LAUNCH$$INTF-INFO-GE 0/0$
  ip address 192.168.1.100 255.255.255.0
  duplex auto
  speed auto
  crypto map dynmap
!
interface GigabitEthernet0/1
  description $ES_LAN$
  ip address 172.19.217.96 255.255.255.0
  duplex auto
  speed auto

!
!interface Virtual-Template1 type tunnel
  ip unnumbered Ethernet0/0
  no clns route-cache
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile vi
!
ip local pool dpool 10.0.0.1 10.0.0.3

!
radius-server host 172.19.220.149 auth-port 1645 acct-port 1646 key cisco radius-server
vsa send authentication !
control-plane
!
!
end
```

Split DNS: Examples

In the following example, the split tunnel list named “101” contains the 10.168.0.0/16 network. It is necessary to include this network information so that the DNS requests to the internal DNS server of 10.168.1.1 are encrypted.

```
crypto isakmp client configuration group home
  key abcd
  acl 101
  dns 10.168.1.1. 10.168.1.2
```

show Output

The following **show** command output example shows that www.ciscoexample1.com and www.ciscoexample2.com have been added to the policy group:

```
Router# show running-config | security group

 crypto isakmp client configuration group 831server
 key abcd
 dns 10.104.128.248
 split-dns www.ciscoexample1.com
 split-dns www.ciscoexample2.com
 group home2 key abcd
```

The following **show** command output example displays currently configured DNS views:

```
Router# show ip dns view

DNS View default parameters:
Logging is off
DNS Resolver settings:
  Domain lookup is enabled
  Default domain name: cisco.com
  Domain search list:
  Lookup timeout: 3 seconds
  Lookup retries: 2
  Domain name-servers:
    172.16.168.183
DNS Server settings:
  Forwarding of queries is enabled
  Forwarder addresses:

DNS View ezvpn-internal-view parameters:
Logging is off
DNS Resolver settings:
  Domain lookup is enabled
  Default domain name:
  Domain search list:
  Lookup timeout: 3 seconds
  Lookup retries: 2
  Domain name-servers:
    10.104.128.248
DNS Server settings:
  Forwarding of queries is enabled
  Forwarder addresses:
```

The following **show** command output example displays currently configured DNS view lists.

```
Router# show ip dns view-list

View-list ezvpn-internal-viewlist:
  View ezvpn-internal-view:
    Evaluation order: 10
```

```
    Restrict to ip dns name-list: 1
  View default:
    Evaluation order: 20
```

The following **show** command output displays DNS name lists.

```
Router# show ip dns name-list

ip dns name-list 1
    permit www.ciscoexample1.com
    permit www.ciscoexample2.com
```

DHCP Client Proxy: Examples

The following examples display DHCP client proxy output information using **show** and **debug** commands.

show Output

Note To use the **show ip dhcp** command, the DHCP server must be a Cisco IOS server.

The following **show ip dhcp pool** command output provides information about the DHCP parameters:

```
Router# show ip dhcp pool

Pool dynpool :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 0 / 0
 Total addresses                : 254
 Leased addresses               : 1
 Pending event                  : none
 1 subnet is currently in the pool:
 Current index    IP address range          Leased addresses
                  10.3.3.1 - 10.3.3.254     1
 No relay targets associated with class aclass
```

The following **show ip dhcp** command output provides information about the DHCP bindings:

```
Router# show ip dhcp binding

Bindings from all pools not associated with VRF:
IP address          Client-ID/                      Lease expiration      Type
                    Hardware address/User name
            10.3.3.5  0065.7a76.706e.2d63.   Apr 04 2006 06:01 AM  Automatic
            6c69.656e.74
```

debug Output

The following example shows how the **debug crypto isakmp** and **debug ip dhcp server events** commands can be used to troubleshoot your DHCP client proxy support configuration:

```
*Apr  3 06:01:32.047: ISAKMP: Config payload REQUEST *Apr  3 06:01:32.047:
ISAKMP:(1002):checking request:
*Apr  3 06:01:32.047: ISAKMP:    IP4_ADDRESS
*Apr  3 06:01:32.047: ISAKMP:    IP4_NETMASK
*Apr  3 06:01:32.047: ISAKMP:    MODECFG_CONFIG_URL
*Apr  3 06:01:32.047: ISAKMP:    MODECFG_CONFIG_VERSION
*Apr  3 06:01:32.047: ISAKMP:    IP4_DNS
*Apr  3 06:01:32.047: ISAKMP:    IP4_DNS
*Apr  3 06:01:32.047: ISAKMP:    IP4_NBNS
```

```
*Apr  3 06:01:32.047: ISAKMP:    IP4_NBNS
*Apr  3 06:01:32.047: ISAKMP:    SPLIT_INCLUDE
*Apr  3 06:01:32.047: ISAKMP:    SPLIT_DNS
*Apr  3 06:01:32.047: ISAKMP:    DEFAULT_DOMAIN
*Apr  3 06:01:32.047: ISAKMP:    MODECFG_SAVEPWD
*Apr  3 06:01:32.047: ISAKMP:    INCLUDE_LOCAL_LAN
*Apr  3 06:01:32.047: ISAKMP:    PFS
*Apr  3 06:01:32.047: ISAKMP:    BACKUP_SERVER
*Apr  3 06:01:32.047: ISAKMP:    APPLICATION_VERSION
*Apr  3 06:01:32.047: ISAKMP:    MODECFG_BANNER
*Apr  3 06:01:32.047: ISAKMP:    MODECFG_IPSEC_INT_CONF
*Apr  3 06:01:32.047: ISAKMP:    MODECFG_HOSTNAME
*Apr  3 06:01:32.047: ISAKMP/author: Author request for group homesuccessfully sent to AAA
*Apr  3 06:01:32.047: ISAKMP:(1002):Input = IKE_MESG_FROM_PEER, IKE_CFG_REQUEST


*Apr  3 06:01:32.047: ISAKMP:(1002):Old State = IKE_P1_COMPLETE  New State =
IKE_CONFIG_AUTHOR_AAA_AWAIT

*Apr  3 06:01:32.047: ISAKMP:(1002):attributes sent in message:
*Apr  3 06:01:32.047:         Address: 10.2.0.0
*Apr  3 06:01:32.047: Requesting DHCP Server0 address 10.3.3.3 *Apr  3 06:01:32.047:
DHCPD: Sending notification of DISCOVER:
*Apr  3 06:01:32.047:   DHCPD: htype 1 chaddr aabb.cc00.6600
*Apr  3 06:01:32.047:   DHCPD: circuit id 00000000
*Apr  3 06:01:32.047: DHCPD: Seeing if there is an internally specified pool class:
*Apr  3 06:01:32.047:   DHCPD: htype 1 chaddr aabb.cc00.6600
*Apr  3 06:01:32.047:   DHCPD: circuit id 00000000


*Apr  3 06:01:34.063: DHCPD: Adding binding to radix tree (10.3.3.5) *Apr  3 06:01:34.063:
DHCPD: Adding binding to hash tree *Apr  3 06:01:34.063: DHCPD: assigned IP address
10.3.3.5 to client 0065.7a76.706e.2d63.6c69.656e.74.
*Apr  3 06:01:34.071: DHCPD: Sending notification of ASSIGNMENT:
*Apr  3 06:01:34.071:  DHCPD: address 10.3.3.5 mask 255.255.255.0
*Apr  3 06:01:34.071:   DHCPD: htype 1 chaddr aabb.cc00.6600
*Apr  3 06:01:34.071:   DHCPD: lease time remaining (secs) = 86400
*Apr  3 06:01:34.183: Obtained DHCP address 10.3.3.5 *Apr  3 06:01:34.183:
ISAKMP:(1002):allocating address 10.3.3.5 *Apr  3 06:01:34.183: ISAKMP: Sending private
address: 10.3.3.5 *Apr  3 06:01:34.183: ISAKMP: Sending subnet mask: 255.255.255.0
```

cTCP Session: Example

The following **debug crypto ctcp** command output displays information about a cTCP session, and it includes comments about the output:

```
Router# debug crypto ctcp

! In the following two lines, a cTCP SYN packet is received from the client, and the cTCP
connection is created.
*Sep 26 11:14:37.135: cTCP: Connection[648B50C0] 10.76.235.21:3519 10.76.248.239:10000:
created
*Sep 26 11:14:37.135: cTCP: SYN from 10.76.235.21:3519
! In the following line, the SYN acknowledgement is sent to the client.
*Sep 26 11:14:37.135: cTCP: Sending SYN(680723B2)ACK(100C637) to 10.76.235.21:3519
! In the following two lines, an acknowlegement is received, and connection setup is
complete. IKE packets should now be received on this newly created cTCP session.
*Sep 26 11:14:37.135: cTCP: Connection[648B50C0] 10.76.235.21:3519 10.76.248.239:10000:
found
*Sep 26 11:14:37.135: cTCP: ACK from 10.76.235.21:3519
*Sep 26 11:14:37.727: cTCP: Connection[648B50C0] 10.76.235.21:3519 10.76.248.239:10000:
found
```

```
*Sep 26 11:14:37.731: cTCP: updating PEER Seq number to 16828803l
*Sep 26 11:14:37.731: cTCP: Pak with contiguous buffer
*Sep 26 11:14:37.731: cTCP: mangling IKE packet from peer: 10.76.235.21:500->3519
  10.76.248.239:500->500
*Sep 26 11:14:37.731: cTCP: Connection[648B50C0] 10.76.235.21:3519 10.76.248.239:10000:
found
*Sep 26 11:14:37.799: cTCP: demangling outbound IKE packet: 10.76.248.239:500->500
  10.76.235.21:3519->500
*Sep 26 11:14:37.799: cTCP: encapsulating IKE packet
*Sep 26 11:14:37.799: cTCP: updating LOCAL Seq number to 1745298727l
! The above lines show that after the required number of IKE packets are exchanged, IKE
and IPsec SAs are created.
*Sep 26 11:14:40.335: cTCP: updating PEER Seq number to 16830431l
*Sep 26 11:14:40.335: cTCP: Pak with particles
*Sep 26 11:14:40.335: cTCP: encapsulating pak
*Sep 26 11:14:40.339: cTCP: datagramstart 0xF2036D8, network_start 0xF2036D8, size 112
*Sep 26 11:14:40.339: cTCP: Pak with contiguous buffer
*Sep 26 11:14:40.339:  cTCP: allocated new buffer
*Sep 26 11:14:40.339: cTCP: updating LOCAL Seq number to 1745299535l
*Sep 26 11:14:40.339: IP: s=10.76.248.239 (local), d=10.76.235.21 (FastEthernet1/1), len
148, cTCP
! The above lines show that Encapsulating Security Payload (ESP) packets are now being
sent and received.
```

VRF Assignment by a AAA Server: Example

The following output example shows that neither a VRF nor an IP address has been defined:

```
aaa new-model
aaa authentication login VPN group radius
aaa authorization network VPN group radius
!
ip vrf example1
 rd 1:1
!
crypto isakmp profile example1
 match identity group example1group
 client authentication list VPN
 isakmp authorization list VPN
 client configuration address respond
 virtual-template 10
!
crypto ipsec transform-set TS esp-3des esp-sha-hmac
!
crypto ipsec profile example1
 set transform-set TS
 set isakmp-profile example1
!
interface Virtual-Template10 type tunnel
! The next line shows that neither VRF nor an IP address has been defined.
 no ip address
tunnel mode ipsec ipv4
tunnel protection ipsec profile example1
```

Additional References

The following sections provide references related to Easy VPN Server.aaa new-model

Related Documents

Related Topic	Document Title
Configuring a router as a VPN client	*Easy VPN Remote Enhancements*, Cisco IOS Release 12.4(4)T feature module
General information on IPsec and VPN	Refer to the following information in the product literature and in IP technical tips sections on Cisco.com: • *Cisco IOS Security Configuration Guide* • *Cisco IOS Security Command Reference*, Release 12.4 • *An Introduction to IP Security (IPSec) Encryption* • *Deploying IPSec* • *Certificate Authority Support for IPSec Overview* • *Cisco Secure VPN Client* • *IPSec VPN High Availability Enhancements*, Cisco IOS Release 12.2(8)T feature module • *Cisco Easy VPN* • *Configuring NAC with IPsec Dynamic Virtual Tunnel Interface*
IPsec Protocol options and attributes	"Configuring Internet Key Exchange Security Protocol" chapter in the *Cisco IOS Security Configuration Guide*
IPsec virtual tunnels	*IPSec Virtual Tunnel Interface*, Cisco IOS Release 12.3(14)T feature module
Network Admission Control	*Network Admission Control*, Cisco IOS Release 12.3(8)T
RRI	*IPSec VPN High Availability Enhancements*, Cisco IOS Release 12.2(8)T feature module
Split DNS	*Configuring Split and Dynamic DNS on the Cisco VPN 3000*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features.

- **aaa authentication login**
- **access-restrict**
- **acl (ISAKMP)**
- **auto-update client**
- **backup-gateway**
- **banner**
- **browser-proxy**
- **clear crypto ctcp**
- **clear crypto session**
- **client authentication list**
- **client pki authorization list**
- **configuration url**
- **configuration version**
- **crypto aaa attribute list**
- **crypto ctcp**
- **crypto ipsec server send-update**

- **crypto isakmp client configuration browser-proxy**
- **crypto isakmp client configuration group**
- **crypto isakmp client firewall**
- **crypto logging ezvpn**
- **debug crypto ctcp**
- **debug crypto condition**
- **debug ip dns name-list**
- **debug ip dns view**
- **debug ip dns view-list**
- **dhcp server (isakmp)**
- **dhcp timeout**
- **domain (isakmp-group)**
- **firewall are-u-there**
- **firewall policy**
- **group-lock**
- **include-local-lan**
- **key (isakmp-group)**
- **max-logins**
- **max-users**
- **pfs**
- **policy**
- **pool (isakmp-group)**
- **proxy**
- **save-password**
- **show crypto ctcp**
- **show crypto debug-condition**
- **show crypto isakmp peers**
- **show crypto isakmp profile**
- **show crypto isakmp sa**
- **show crypto session**
- **show crypto session group**
- **show crypto session summary**
- **show ip dns name-list**
- **show ip dns view**
- **show ip dns view-list**
- **split-dns**
- **wins**
- **Glossary**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Feature Information for Easy VPN Server

Table 3 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 3 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 3 Feature Information for Easy VPN Server

Feature Name	Releases	Feature Information
Easy VPN Server	12.2(8)T	The Easy VPN Server feature introduces server support for the Cisco VPN Client Release 3.x and later software clients and Cisco VPN hardware clients (such as the Cisco 800, Cisco 900, Cisco 1700, VPN 3002, and PIX 501 devices). This feature allows a remote end user to communicate using IP Security (IPsec) with any Cisco IOS Virtual Private Network (VPN) gateway. Centrally managed IPsec policies are “pushed” to the client device by the server, minimizing configuration by the end user.
	12.3(2)T	RADIUS support for user profiles, user-based policy control, session monitoring for VPN group access, backup-gateway list, and PFS were added.
	12.3(7)T	The **netmask** command was integrated for use on the Easy VPN server. For information about configuring this command, see the following section: • Defining Group Policy Information for Mode Configuration Push, page 21
	12.4(2)T 12.2(33)SXH	The following feature was added in this release: • Banner, Auto-Update, and Browser Proxy Enhancements

Table 3 Feature Information for Easy VPN Server (continued)

Feature Name	Releases	Feature Information
	12.4(4)T 12.2(33)SXH	The following features were added in this release: • Configuration Management Enhancements (Pushing a Configuration URL Through a Mode-Configuration Exchange) • Per User AAA Policy Download with PKI • Syslog Message Enhancements • Network Admission Control for Easy VPN • Password Aging • Virtual IPsec Interface Support
	12.4(6)T	The Central Policy Push Firewall Policy Push feature was added.
	12.2(33)SRA	This feature was integrated into Cisco IOS Release 12.2(33)SRA.

Table 3 Feature Information for Easy VPN Server (continued)

Feature Name	Releases	Feature Information
	12.4(9)T	The following features were added in this release: • DHCP Client Proxy The following section provides information about this feature: – DHCP Client Proxy, page 11 • Virtual Tunnel Interface Per-User Attribute Support for Easy VPN Servers. – Virtual Tunnel Interface Per-User Attribute Support, page 13 • Split DNS The following section provides information about this feature: – Split DNS, page 18 • cTCP The following sections provide information about this feature: – cTCP, page 18 – Configuring cTCP, page 50 – cTCP Session: Example, page 70 • Per-User Attribute Support for Easy VPN Servers The following sections provide information about this feature: – Per-User Attribute Support for Easy VPN Servers, page 15 – Configuring Per-User Attributes on a Local Easy VPN AAA Server, page 38 – Per-User Attributes on an Easy VPN Server: Example, page 62 • VRF Assignment by a AAA Server The following sections provide information about this feature: – VRF Assignment by a AAA Server, page 18 – VRF Assignment by a AAA Server: Example, page 71 The following new commands were introduced: **crypto aaa attribute list**, **debug ip dns**, **dhcp-server (isakmp)**, **dhcp-timeout**, **show ip dns name-list, show ip dns view,** and **show ip dns view-list** The following commands were modified: **crypto isakmp client configuration group**

Table 3 Feature Information for Easy VPN Server (continued)

Feature Name	Releases	Feature Information
	12.4(11)T	The DHCP Client Proxy feature was updated to include manageability enhancements for remote access VPNs. The following commands were modified: **clear crypto session, crypto isakmp client configuration group, debug crypto condition, show crypto debug-condition, show crypto isakmp peers, show crypto isakmp profile, show crypto isakmp sa, show crypto session**
EasyVPN Server Enhancements	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Glossary

AAA—authentication, authorization, and accounting. Framework of security services that provides the method for identifying users (authentication), for remote access control (authorization), and for collecting and sending security server information used for billing, auditing, and reporting (accounting).

aggressive mode (AM)—Mode during Internet Key Exchange negotiation. Compared to main mode (MM), AM eliminates several steps, which makes it faster but less secure than MM. Cisco IOS software will respond in aggressive mode to an Internet Key Exchange (IKE) peer that initiates aggressive mode.

AV pair—attribute-value pair. Additional authentication and authorization information in the following format: Cisco:AVPair="protocol:attribute=value".

IKE—Internet Key Exchange. Hybrid protocol that implements Oakley key exchange and Skeme key exchange inside the ISAKMP framework. Although IKE can be used with other protocols, its initial implementation is with IPsec. IKE provides authentication of the IPsec peers, negotiates IPsec keys, and negotiates IPsec security associations.

IPsec—IP Security Protocol. Framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPsec provides these security services at the IP layer. IPsec uses IKE to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPsec. IPsec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

ISAKMP—Internet Security Association Key Management Protocol. Protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.

MM—main mode. Mode that is slower than aggressive mode but more secure and more flexible than aggressive mode because it can offer an IKE peer more security proposals. The default action for IKE authentication (Rivest, Shamir, and Adelman signature (rsa-sig), RSA encryption (rsa-encr), or preshared) is to initiate main mode.

policy push—Allows administrators to push policies that enforce security to the Cisco Easy VPN (software) Client and related firewall software.

reverse route injection (RRI)—Simplified network design for VPNs on which there is a requirement for redundancy or load balancing. RRI works with both dynamic and static crypto maps.

In the dynamic case, as remote peers establish IPsec security associations with an RRI enabled router, a static route is created for each subnet or host protected by that remote peer. For static crypto maps, a static route is created for each destination of an extended access-list rule.

SA—security association. Description of how two or more entities will utilize security services to communicate securely. For example, an IPsec SA defines the encryption algorithm (if used), the authentication algorithm, and the shared session key to be used during the IPsec connection.

Both IPsec and IKE require and use SAs to identify the parameters of their connections. IKE can negotiate and establish its own SA. The IPsec SA is established either by IKE or by manual user configuration.

VPN—Virtual Private Network. Framework that consists of multiple peers transmitting private data securely to one another over an otherwise public infrastructure. In this framework, inbound and outbound network traffic is protected using protocols that tunnel and encrypt all data. This framework permits networks to extend beyond their local topology, while remote users are provided with the appearance and functionality of a direct network connection.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Invalid Security Parameter Index Recovery

When an invalid security parameter index error (shown as “Invalid SPI”) occurs in IP Security (IPSec) packet processing, the feature allows for an Internet Key Exchange (IKE) security association (SA) to be established. The “IKE” module sends notification of the “Invalid SPI” error to the originating IPSec peer so that Security Association Databases (SADBs) can be resynchronized and successful packet processing can be resumed.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the “Feature Information for” section on page 17.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for

Before configuring the feature, you must have enabled Internet Key Exchange (IKE) and IPSec on your router.

Restrictions for

If an IKE SA is being initiated to notify an IPSec peer of an “Invalid SPI” error, there is the risk that a denial-of-service (DoS) attack can occur. The feature has a built-in mechanism to minimize such a risk, but because there is a risk, the feature is not enabled by default. You must enable the command using command-line interface (CLI).

Information About

To use the feature, you should understand the following concept.

How the Feature Works

An IPSec “black hole” occurs when one IPSec peer “dies” (for example, a peer can “die” if a reboot occurs or if an IPSec peer somehow gets reset). Because one of the peers (the receiving peer) is completely reset, it loses its IKE SA with the other peer. Generally, when an IPSec peer receives a packet for which it cannot find an SA, it tries to send an IKE “INVALID SPI NOTIFY” message to the data originator. This notification is sent using the IKE SA. If there is no IKE SA available, the receiving peer drops the packet.

Note A single security association (SA) has only two peers. However, a SADB can have multiple SAs, whereby each SA has an association with a different peer.

When an invalid security parameter index (SPI) is encountered, the Invalid Security Parameter Index feature provides for the setting up of an IKE SA with the originator of the data, and the IKE “INVALID SPI NOTIFY” message is sent. The peer that originated the data “sees” the “INVALID SPI NOTIFY” message and deletes the IPSec SA that has the invalid SPI. If there is further traffic from the originating peer, there will not be any IPSec SAs, and new SAs will be set up. Traffic will flow again. The default behavior (that is, without configuring the feature) is that the data packet that caused the invalid SPI error is dropped. The originating peer keeps on sending the data using the IPSec SA that has the invalid SPI, and the receiving peer keeps dropping the traffic (thus creating the “black hole”).

The IPSec module uses the IKE module to send an IKE “INVALID SPI NOTIFY” message to the other peer. Once the invalid SPI recovery is in place, there should not be any significant dropping of packets although the IPSec SA setup can itself result in the dropping of a few packets.

To configure your router for the feature, use the **crypto isakmp invalid-spi-recovery** command. The IKE SA will not be initiated unless you have configured this command.

How to Configure

This section contains the following procedure.

Configuring

To configure the feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp invalid-spi-recovery**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp invalid-spi-recovery** **Example:** `Router (config)# crypto isakmp invalid-spi-recovery`	Initiates the IKE module process whereby the IKE module notifies the receiving peer that an “Invalid SPI” error has occurred.

Verifying an Configuration

To determine the status of the IPSec SA for traffic between two peers, you can use the **show crypto ipsec sa** command. If the IPSec SA is available on one peer and not on the other, there is a “black hole” situation, in which case you will see the invalid SPI errors being logged for the receiving peer. If you turn console logging on or check the syslog server, you will see that these errors are also being logged.

Figure 1 shows the topology of a typical preshared configuration setup. Host 1 is the initiating peer (initiator), and Host 2 is the receiving peer (responder).

Figure 1 Preshared Configuration Topology

SUMMARY STEPS

To verify the preshared configuration, perform the following steps.

1. **Initiate the IKE and IPSec SAs between Host 1 and Host 2**
2. **Clear the IKE and IPSec SAs on Router B**
3. **Send traffic from Host 1 to Host 2 and ensure that IKE and IPSec SAs are correctly established**
4. **Check for an invalid SPI message on Router B**

DETAILED STEPS

Step 1 Initiate the IKE and IPSec SAs between Host 1 and Host 2

Router A

```
Router# show crypto isakmp sa

  f_vrf/i_vrf   dst            src             state       conn-id slot
       / 10.2.2.2              10.1.1.1        QM_IDLE              1       0
```

Router B

```
Router# show crypto isakmp sa

  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.1.1.1       10.2.2.2        QM_IDLE              1       0
```

Router A

```
Router# show crypto ipsec sa interface fastethernet0/0

interface: FastEthernet0/0
    Crypto map tag: testtag1, local addr. 10.1.1.1

   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
```

```
   current_peer: 10.2.2.2:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest: 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify: 10
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 10.1.1.1, remote crypto endpt.: 10.2.2.2
     path mtu 1500, media mtu 1500
     current outbound spi: 7AA69CB7

     inbound esp sas:
      spi: 0x249C5062(614223970)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5123, flow_id: 1, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4537831/3595)
        IV size: 8 bytes
        replay detection support: Y

     inbound ah sas:
      spi: 0xB16D1587(2976716167)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5121, flow_id: 1, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4537831/3595)
        replay detection support: Y

     inbound pcp sas:

     outbound esp sas:
      spi: 0x7AA69CB7(2057739447)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5124, flow_id: 2, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4537835/3595)
        IV size: 8 bytes
        replay detection support: Y

     outbound ah sas:
      spi: 0x1214F0D(18960141)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5122, flow_id: 2, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4537835/3594)
        replay detection support: Y

     outbound pcp sas:
```

Router B

```
Router# show crypto ipsec sa interface ethernet1/0

interface: Ethernet1/0
    Crypto map tag: testtag1, local addr. 10.2.2.2

   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
```

```
remote ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
current_peer: 10.1.1.1:500
  PERMIT, flags={origin_is_acl,}
 #pkts encaps: 10, #pkts encrypt: 10, #pkts digest: 10
 #pkts decaps: 10, #pkts decrypt: 10, #pkts verify: 10
 #pkts compressed: 0, #pkts decompressed: 0
 #pkts not compressed: 0, #pkts compr. failed: 0
 #pkts not decompressed: 0, #pkts decompress failed: 0

 #send errors 0, #recv errors 0

  local crypto endpt.: 10.2.2.2, remote crypto endpt.: 10.1.1.1
  path mtu 1500, media mtu 1500
  current outbound spi: 249C5062

  inbound esp sas:
   spi: 0x7AA69CB7(2057739447)
     transform: esp-des esp-sha-hmac ,
     in use settings ={Tunnel, }
     slot: 0, conn id: 5123, flow_id: 1, crypto map: testtag1
     crypto engine type: Hardware
     sa timing: remaining key lifetime (k/sec): (4421281/3593)
     IV size: 8 bytes
     replay detection support: Y

  inbound ah sas:
   spi: 0x1214F0D(18960141)
     transform: ah-sha-hmac ,
     in use settings ={Tunnel, }
     slot: 0, conn id: 5121, flow_id: 1, crypto map: testtag1
     crypto engine type: Hardware
     sa timing: remaining key lifetime (k/sec): (4421281/3593)
     replay detection support: Y

  inbound pcp sas:

  outbound esp sas:
   spi: 0x249C5062(614223970)
     transform: esp-des esp-sha-hmac ,
     in use settings ={Tunnel, }
     slot: 0, conn id: 5124, flow_id: 2, crypto map: testtag1
     crypto engine type: Hardware
     sa timing: remaining key lifetime (k/sec): (4421285/3593)
     IV size: 8 bytes
     replay detection support: Y

  outbound ah sas:
   spi: 0xB16D1587(2976716167)
     transform: ah-sha-hmac ,
     in use settings ={Tunnel, }
     slot: 0, conn id: 5122, flow_id: 2, crypto map: testtag1
     crypto engine type: Hardware
     sa timing: remaining key lifetime (k/sec): (4421285/3592)
     replay detection support: Y

  outbound pcp sas:
```

Step 2 Clear the IKE and IPSec SAs on Router B

```
Router# clear crypto isakmp

Router# clear crypto sa
```

```
Router# show crypto isakmp sa

  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.2.2.2.       10.1.1.1      MM_NO_STATE           1       0 (deleted)

Router# show crypto ipsec sa

interface: Ethernet1/0
    Crypto map tag: testtag1, local addr. 10.2.2.2


   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   current_peer: 10.1.1.1:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 10.2.2.2, remote crypto endpt.: 10.1.1.1
     path mtu 1500, media mtu 1500
     current outbound spi: 0

     inbound esp sas:

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:

     outbound ah sas:

     outbound pcp sas:
```

Step 3 Send traffic from Host 1 to Host 2 and ensure that new IKE and IPSec SAs are correctly established

```
ping
Protocol [ip]: ip
Target IP address: 10.0.2.2
Repeat count [5]: 30
Datagram size [100]: 100
Timeout in seconds [2]:
Extended commands [n]: no
Sweep range of sizes [n]: n
Type escape sequence to abort.
Sending 30, 100-byte ICMP Echos to 10.0.2.2, timeout is 2 seconds:
..!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 93 percent (28/30), round-trip min/avg/max = 1/3/8 ms

RouterB# show crypto isakmp sa

  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.1.1.1       10.2.2.2     QM_IDLE              3       0
       /        10.1.1.1       10.2.2.2      MM_NO_STATE          1       0 (deleted)

RouterB# show crypto ipsec sa

interface: Ethernet1/0
    Crypto map tag: testtag1, local addr. 10.2.2.2
```

```
   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   current_peer: 10.1.1.1:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 28, #pkts encrypt: 28, #pkts digest: 28
    #pkts decaps: 28, #pkts decrypt: 28, #pkts verify: 28
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0

    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 10.2.2.2, remote crypto endpt.: 10.1.1.1
     path mtu 1500, media mtu 1500
     current outbound spi: D763771F

     inbound esp sas:
      spi: 0xE7AB4256(3886760534)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5127, flow_id: 3, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502463/3596)
        IV size: 8 bytes
        replay detection support: Y

     inbound ah sas:
      spi: 0xF9205CED(4179647725)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5125, flow_id: 3, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502463/3596)
        replay detection support: Y

     inbound pcp sas:

     outbound esp sas:
      spi: 0xD763771F(3613619999)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5128, flow_id: 4, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502468/3596)
        IV size: 8 bytes
        replay detection support: Y

     outbound ah sas:
      spi: 0xEB95406F(3952427119)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5126, flow_id: 4, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502468/3595)
        replay detection support: Y

     outbound pcp sas:

RouterA# show crypto isakmp sa

  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.2.2.2       10.1.1.1      MM_NO_STATE           1       0 (deleted)
```

```
       /        10.2.2.2       10.1.1.1      QM_IDLE              2       0


               Check for an invalid SPI message on Router B
Router# show logging


Syslog logging: enabled (10 messages dropped, 13 messages rate-limited, 0 flushes, 0
overruns, xml disabled)
    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled
    Buffer logging: level debugging, 43 messages logged, xml disabled
    Logging Exception size (8192 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 72 message lines logged

Log Buffer (8000 bytes):

*Mar 24 20:55:45.739: %CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid
spi for
        destaddr=10.2.2.2, prot=51, spi=0x1214F0D(18960141), srcaddr=10.1.1.1
*Mar 24 20:55:47.743: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/255.255.255.255/0/0 (type=1),
    remote_proxy= 10.0.0.1/255.255.255.255/0/0 (type=1),
    protocol= AH, transform= ah-sha-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
*Mar 24 20:55:47.743: IPSEC(validate_proposal_request): proposal part #2,
  (key eng. msg.) INBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/255.255.255.255/0/0 (type=1),
    remote_proxy= 10.0.0.1/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-des esp-sha-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
*Mar 24 20:55:47.743: IPSEC(kei_proxy): head = testtag1, map->ivrf = , kei->ivrf =
*Mar 24 20:55:47.743: IPSEC(key_engine): got a queue event with 2 kei messages
*Mar 24 20:55:47.743: IPSEC(spi_response): getting spi 4179647725 for SA
        from 10.2.2.2        to 10.1.1.1     for prot 2
*Mar 24 20:55:47.747: IPSEC(spi_response): getting spi 3886760534 for SA
        from 10.2.2.2     to 10.1.1.1        for prot 3
*Mar 24 20:55:48.071: IPSec: Flow_switching Allocated flow for flow_id 939524099
*Mar 24 20:55:48.071: IPSec: Flow_switching Allocated flow for flow_id 939524100
*Mar 24 20:55:48.135: IPSEC(key_engine): got a queue event with 4 kei messages
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= AH, transform= ah-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xF9205CED(4179647725), conn_id= 939529221, keysize= 0, flags= 0x2
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= AH, transform= ah-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xEB95406F(3952427119), conn_id= 939529222, keysize= 0, flags= 0xA
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-des esp-sha-hmac ,
    lifedur= 3600s and 4608000kb,
```

```
    spi= 0xE7AB4256(3886760534), conn_id= 939529223, keysize= 0, flags= 0x2
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 10.2.2.2, remote= 10.1.1.1,
    local_proxy= 10.0.2.2/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-des esp-sha-hmac ,

    lifedur= 3600s and 4608000kb,
    spi= 0xD763771F(3613619999), conn_id= 939529224, keysize= 0, flags= 0xA
*Mar 24 20:55:48.139: IPSEC(kei_proxy): head = testtag1, map->ivrf = , kei->ivrf =
*Mar 24 20:55:48.139: IPSEC(mtree_add_ident): src 10.2.2.2, dest 10.1.1.1, dest_port 0

*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.1.1.1, sa_prot= 51,
    sa_spi= 0xF9205CED(4179647725),
    sa_trans= ah-sha-hmac , sa_conn_id= 939529221
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.2.2.2, sa_prot= 51,
    sa_spi= 0xEB95406F(3952427119),
    sa_trans= ah-sha-hmac , sa_conn_id= 939529222
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.1.1.1, sa_prot= 50,
    sa_spi= 0xE7AB4256(3886760534),
    sa_trans= esp-des esp-sha-hmac , sa_conn_id= 939529223
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.2.2.2, sa_prot= 50,
    sa_spi= 0xD763771F(3613619999),
    sa_trans= esp-des esp-sha-hmac , sa_conn_id= 939529224
ipseca-72a#
```

Configuration Examples for

This section provides the following configuration example.

- : Example, page 10

: Example

The following example shows that invalid security parameter index recovery has been configured on Router A and Router B. Figure 1 shows the topology used for this example.

Router A

```
Router# show running-config

Building configuration...

Current configuration : 2048 bytes
!
version 12.3
no service pad
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service tcp-small-servers
!
hostname ipseca-71a
!
logging queue-limit 100
```

```
no logging console
enable secret 5 $1$4GZB$L2YOmnenOCNAu0jgFxebT/
enable password lab
!
clock timezone PST -8

clock summer-time PDT recurring
ip subnet-zero
!
!
no ip domain lookup
!
ip cef
ip audit notify log
ip audit po max-events 100
mpls ldp logging neighbor-changes
no ftp-server write-enable
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
crypto isakmp policy 1
 authentication pre-share
 lifetime 180
crypto isakmp key 0 1234 address 10.2.2.2
crypto isakmp invalid-spi-recovery
!
!
crypto ipsec transform-set auth2 ah-sha-hmac esp-des esp-sha-hmac
!
crypto map testtag1 10 ipsec-isakmp
 set peer 10.2.2.2
 set transform-set auth2
 match address 150
!
!
controller ISA 5/1
!
!
interface FastEthernet0/0
 ip address 10.1.1.1 255.0.0.0
 no ip route-cache cef
 duplex full
 speed 100
 crypto map testtag1
!
interface FastEthernet0/1
 ip address 10.0.0.1 255.0.0.0
 no ip route-cache cef
 duplex auto
 speed auto
!
interface Serial1/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
 clockrate 128000
!
interface Serial1/1
 no ip address
```

```
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
 clockrate 128000
!

interface Serial1/2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
!
interface Serial1/3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no keepalive
 serial restart_delay 0
 clockrate 128000
!
ip classless
ip route 10.3.3.3 255.0.0.0 10.2.0.1
no ip http server
no ip http secure-server
!
!
access-list 150 permit ip host 10.0.0.1 host 10.0.2.2
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
call rsvp-sync
!
!
mgcp profile default
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 password lab
 login
!
!
end

ipseca-71a#
```

Router B

```
Router# show running-config

Building configuration...

Current configuration : 2849 bytes
!
version 12.3
no service pad
service timestamps debug datetime msec localtime
```

```
service timestamps log datetime msec localtime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname ipseca-72a
!

logging queue-limit 100
no logging console
enable secret 5 $1$kKqL$5Th5Qhw1ubDkkK90KWFxi1
enable password lab
!
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
!
!
no ip domain lookup
!
ip cef
ip audit notify log
ip audit po max-events 100
mpls ldp logging neighbor-changes
no ftp-server write-enable
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
mta receive maximum-recipients 0
!
!
crypto isakmp policy 1
 authentication pre-share
 lifetime 180
crypto isakmp key 0 1234 address 10.1.1.1
crypto isakmp invalid-spi-recovery
!
!
crypto ipsec transform-set auth2 ah-sha-hmac esp-des esp-sha-hmac
!
crypto map testtag1 10 ipsec-isakmp
 set peer 10.1.1.1
 set transform-set auth2
 match address 150
!
!
controller ISA 5/1
!
!
interface FastEthernet0/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/0
 ip address 10.2.2.2 255.0.0.0
 no ip route-cache cef
 duplex half
 crypto map testtag1
```

```
!
interface Ethernet1/1
 ip address 10.0.2.2 255.0.0.0
 no ip route-cache cef
 duplex half
!
interface Ethernet1/2
 no ip address

 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/4
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/5
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/6
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/7
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Serial3/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
!
interface Serial3/1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
 clockrate 128000
!
```

```
interface Serial3/2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
!
interface Serial3/3
 no ip address

 no ip route-cache
 no ip mroute-cache
 shutdown
 no keepalive
 serial restart_delay 0
 clockrate 128000
!
ip classless
ip route 10.0.0.0 255.0.0.0 10.2.0.1
no ip http server
no ip http secure-server
!
!
access-list 150 permit ip host 10.0.2.2 host 10.0.0.1
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password lab
 login
!
!
end
```

Additional References

The following sections provide references related to .

Related Documents

Related Topic	Document Title
Configuring IKE	"Configuring Internet Key Exchange Security Protocol" section of the *Cisco IOS Security Configuration Guide*
Configuring IPSec	"Part 4: IP Security and Encryption" of the *Cisco IOS Security Configuration Guide*
Interface commands	The *Cisco IOS Interface and Hardware Component Command Reference*, Release 12.3

Standards

Standards	Title
This feature has no new or modified standards.	—

MIBs

MIBs	MIBs Link
This feature has no new or modified MIBs.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
This feature has no new or modified RFCs.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features

- **crypto isakmp invalid-spi-recovery**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Feature Information for

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for

Feature Name	Releases	Feature Information
	12.3(2)T	This feature was introduced.
	12.2(18)SXE	This feature was integrated into Cisco IOS Release 12.2(18)SXE.
Invalid Special Parameter Index (SPI) Recovery	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

IP Security VPN Monitoring

The IP Security VPN Monitoring feature provides VPN session monitoring enhancements that will allow you to troubleshoot the Virtual Private Network (VPN) and monitor the end-user interface. Session monitoring enhancements include the following:

- Ability to specify an Internet Key Exchange (IKE) peer description in the configuration file
- Summary listing of crypto session status
- Syslog notification for crypto session up or down status
- Ability to clear both IKE and IP Security (IPSec) security associations (SAs) using one command-line interface (CLI)

Feature History for IP Security VPN Monitoring

Release	Modification
12.3(4)T	This feature was introduced.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for IP Security VPN Monitoring

- You should be familiar with IPSec and encryption.
- Your router must support IPSec, and before using the IP Security VPN Monitoring feature, you must have configured IPSec on your router.

Restrictions for IP Security VPN Monitoring

- You must be running Cisco IOS k8 or k9 crypto images on your router.

Information About IPSec VPN Monitoring

To troubleshoot the IPSec VPN and monitor the end-user interface, you should understand the following concepts:

Background: Crypto Sessions

A crypto session is a set of IPSec connections (flows) between two crypto endpoints. If the two crypto endpoints use IKE as the keying protocol, they are IKE peers to each other. Typically, a crypto session consists of one IKE security association (for control traffic) and at least two IPSec security associations (for data traffic—one per each direction). There may be duplicated IKE security associations (SAs) and IPSec SAs or duplicated IKE SAs or IPSec SAs for the same session in the duration of rekeying or because of simultaneous setup requests from both sides.

Per-IKE Peer Description

The Per-IKE Peer Description function allows you to enter a description of your choosing for an IKE peer. (Before Cisco IOS Release 12.3(4)T, you could use only the IP address or fully qualified domain name [FQDN] to identify the peer; there was no way to configure a description string.) The unique peer description, which can include up to 80 characters, can be used whenever you are referencing that particular IKE peer. To add the peer description, use the **description** command.

Note IKE peers that "sit" behind a Network Address Translation (NAT) device cannot be uniquely identified; therefore, they have to share the same peer description.

The primary application of this description field is for monitoring purposes (for example, when using **show** commands or for logging [syslog messages]). The description field is purely informational (for example, it cannot act as a substitute for the peer address or FQDN when defining crypto maps).

Summary Listing of Crypto Session Status

You can get a list of all the active VPN sessions by entering the **show crypto session** command. The listing will include the following:

- Interface
- IKE peer description, if available
- IKE SAs that are associated with the peer by whom the IPSec SAs are created
- IPSec SAs serving the flows of a session

Multiple IKE or IPSec SAs may be established for the same peer (for the same session), in which case IKE peer descriptions will be repeated with different values for the IKE SAs that are associated with the peer and for the IPSec SAs that are serving the flows of the session.

You can also use the **show crypto session detail** variant of this command to obtain more detailed information about the sessions.

Syslog Notification for Crypto Session Up or Down Status

The Syslog Notification for Crypto Session Up or Down Status function provides syslog notification every time the crypto session comes up or goes down.

The following is a sample syslog notification showing that a crypto session is up:

```
%CRYPTO-5-SESSION_STATUS: Crypto session is UP. Peer 10.6.6.1:500 fvrf=name10 ivrf=name20
Description: SJC24-2-VPN-Gateway Id: 10.5.5.2
```

The following is a sample syslog notification showing that a crypto session is down:

```
%CRYPTO-5-SESSION_STATUS: Crypto session is DOWN. Peer 10.6.6.1:500 fvrf=name10
ivrf=name20 Description: SJC24-2-VPN-Gateway Id: 10.5.5.2
```

IKE and IPSec Security Exchange Clear Command

In previous IOS versions, there was no single command to clear both IKE and IPSec connections (that is, SAs). Instead, you had to use the **clear crypto isakmp** command to clear IKE and the **clear crypto ipsec** command to clear IPSec. The new **clear crypto session** command allows you to clear both IKE and IPSec with a single command. To clear a specific crypto session or a subset of all the sessions (for example, a single tunnel to one remote site), you need to provide session-specific parameters, such as a local or remote IP address, a local or remote port, a front door VPN routing and forwarding (FVRF) name, or an inside VRF (IVRF) name. Typically, the remote IP address will be used to specify a single tunnel to be deleted.

If a local IP address is provided as a parameter when you use the **clear crypto session** command, all the sessions (and their IKE SAs and IPSec SAs) that share the IP address as a local crypto endpoint (IKE local address) will be cleared. If you do not provide a parameter when you use the **clear crypto session** command, all IPSec SAs and IKE SAs that are in the router will be deleted.

How to Configure IP Security VPN Monitoring

See the following sections for configuration tasks for this feature. Each task in the list is identified as either required or optional.

- Adding the Description of an IKE Peer, page 4 (optional)
- Verifying Peer Descriptions, page 5 (optional)
- Clearing a Crypto Session, page 6 (optional)

Adding the Description of an IKE Peer

To add the description of an IKE peer to an IPSec VPN session, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp peer** {**ip-address** *ip-address*}
4. **description**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto isakmp peer** {**ip-address** *ip-address*} **Example:** `Router (config)# crypto isakmp peer address 10.2.2.9`	Enables an IPSec peer for IKE querying of authentication, authorization, and accounting (AAA) for tunnel attributes in aggressive mode and enters ISAKMP peer configuration mode.
Step 4	**description** **Example:** `Router (config-isakmp-peer)# description connection from site A`	Adds a description for an IKE peer.

Verifying Peer Descriptions

To verify peer descriptions, use the **show crypto isakmp peer** command.

SUMMARY STEPS

1. **enable**
2. **show crypto isakmp peer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto isakmp peer** **Example:** `Router# show crypto isakmp peer`	Displays peer descriptions.

Examples

The following output example verifies that the description “connection from site A” has been added for IKE peer 10.2.2.9:

```
Router# show crypto isakmp peer

Peer: 10.2.2.9 Port: 500
 Description: connection from site A
  flags: PEER_POLICY
```

When the peer at address 10.2.2.9 connects and the session comes up, the syslog status will be shown as follows:

```
%CRYPTO-5-SESSION_STATUS: Crypto tunnel is UP. Peer 10.2.2.9:500 Description: connection
from site A Id: ezvpn
```

Clearing a Crypto Session

To clear a crypto session, use the **clear crypto session** command from the router command line. No configuration statements are required in the configuration file to use this command.

SUMMARY STEPS

1. **enable**
2. **clear crypto session**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`clear crypto session` **Example:** `Router# clear crypto session`	Deletes crypto sessions (IPSec and IKE SAs).

Configuration Examples for IP Security VPN Monitoring

This section provides the following configuration example:

- show crypto session Command Output: Examples, page 6

show crypto session Command Output: Examples

The following is sample output for the **show crypto session** output without the **detail** keyword:

```
Router# show crypto session

Crypto session current status

Interface: FastEthernet0/1
Session status: UP-ACTIVE
Peer: 172.0.0.2/500
   IKE SA: local 172.0.0.1/500 remote 172.0.0.2/500 Active
   IPSEC FLOW: permit ip 10.10.10.0/255.255.255.0 10.30.30.0/255.255.255.0
         Active SAs: 2, origin: crypto map
```

The following is sample output using the **show crypto session command and the detail** keyword:

```
Router# show crypto session detail

Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.1.1.3 port 500 fvrf: (none) ivrf: (none)
      Desc: this is my peer at 10.1.1.3:500 Green
```

```
    Phase1_id: 10.1.1.3
IKE SA: local 10.1.1.4/500 remote 10.1.1.3/500 Active
        Capabilities:(none) connid:3 lifetime:22:03:24
IPSEC FLOW: permit 47 host 10.1.1.4 host 10.1.1.3
      Active SAs: 0, origin: crypto map
      Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
      Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/0
IPSEC FLOW: permit ip host 10.1.1.4 host 10.1.1.3
      Active SAs: 4, origin: crypto map
      Inbound:  #pkts dec'ed 4 drop 0 life (KB/Sec) 4605665/2949
      Outbound: #pkts enc'ed 4 drop 1 life (KB/Sec) 4605665/2949
```

Additional References

The following sections provide references related to IP Security VPN Monitoring.

Related Documents

Related Topic	Document Title
IP security, encryption, and IKE	*"IP Security and Encryption" section of the Cisco IOS Security Configuration Guide*
Security commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features

- **clear crypto session**

- **description (isakmp peer)**
- **show crypto isakmp peer**
- **show crypto session**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

IPsec Anti-Replay Window: Expanding and Disabling

First Published: February 28, 2005
Last Updated: September 12, 2006

Cisco IP security (IPsec) authentication provides anti-replay protection against an attacker duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. The decryptor keeps track of which packets it has seen on the basis of these numbers. Currently, the default window size is 64 packets. Generally, this number (window size) is sufficient, but there are times when you may want to expand this window size. The IPsec Anti-Replay Window: Expanding and Disabling feature allows you to expand the window size, allowing the decryptor to keep track of more than 64 packets.

History for the IPsec Anti-Replay Window: Expanding and Disabling Feature

Release	Modification
12.3(14)T	This feature was introduced.
12.2(33)SRA	This feature was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF6	This feature was integrated into Cisco IOS Release 12.2(18)SXF6.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for IPsec Anti-Replay Window: Expanding and Disabling

- Before configuring this feature, you should have already created a crypto map or crypto profile.

Information About IPsec Anti-Replay Window: Expanding and Disabling

To configure the IPsec Anti-Replay Window: Expanding and Disabling feature, you should understand the following concept:

IPsec Anti-Replay Window

Cisco IPsec authentication provides anti-replay protection against an attacker duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. (Security association [SA] anti-replay is a security service in which the receiver can reject old or duplicate packets to protect itself against replay attacks.) The decryptor checks off the sequence numbers that it has seen before. The encryptor assigns sequence numbers in an increasing order. The decryptor remembers the value X of the highest sequence number that it has already seen. N is the window size, and the decryptor also remembers whether it has seen packets having sequence numbers from X-N+1 through X. Any packet with the sequence number X–N is discarded. Currently, N is set at 64, so only 64 packets can be tracked by the decryptor.

At times, however, the 64-packet window size is not sufficient. For example, Cisco quality of service (QoS) gives priority to high-priority packets, which could cause some low-priority packets to be discarded even though they could be one of the last 64 packets received by the decryptor. The IPsec Anti-Replay Window: Expanding and Disabling feature allows you to expand the window size, allowing the decryptor to keep track of more than 64 packets.

Increasing the anti-replay window size has no impact on throughput and security. The impact on memory is insignificant because only an extra 128 bytes per incoming IPsec SA is needed to store the sequence number on the decryptor. It is recommended that you use the full 1024 window size to eliminate any future anti-replay problems.

How to Configure IPsec Anti-Replay Window: Expanding and Disabling

This section contains the following procedures:

- Configuring IPsec Anti-Replay Window: Expanding and Disabling Globally, page 3 (optional)
- Configuring IPsec Anti-Replay Window: Expanding and Disabling on a Crypto Map, page 4 (optional)

Configuring IPsec Anti-Replay Window: Expanding and Disabling Globally

To configure IPsec Anti-Replay Window: Expanding and Disabling globally (so that it affects all SAs that are created— except for those that are specifically overridden on a per-crypto map basis), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec security-association replay window-size** [*N*]
4. **crypto ipsec security-association replay disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec security-association replay window-size** [*N*] **Example:** `Router (config)# crypto ipsec security-association replay window-size 256`	Sets the size of the SA replay window globally. **Note** Configure this command or the **crypto ipsec security-association replay disable** command. The two commands are not used at the same time.
Step 4	**crypto ipsec security-association replay disable** **Example:** `Router (config)# crypto ipsec security-association replay disable`	Disables checking globally. **Note** Configure this command or the **crypto ipsec security-association replay window-size** command. The two commands are not used at the same time.

Configuring IPsec Anti-Replay Window: Expanding and Disabling on a Crypto Map

To configure IPsec Anti-Replay Window: Expanding and Disabling on a crypto map so that it affects those SAs that have been created using a specific crypto map or profile, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* [**ipsec-isakmp**]
4. **set security-association replay window-size** [*N*]
5. **set security-association replay disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* [**ipsec-isakmp**] **Example:** `Router (config)# crypto map ETHO 17 ipsec-isakmp`	Enters crypto map configuration mode and creates a crypto profile that provides a template for configuration of dynamically created crypto maps.
Step 4	**set security-association replay window-size** [*N*] **Example:** `Router (crypto-map)# set security-association replay window-size 128`	Controls the SAs that are created using the policy specified by a particular crypto map, dynamic crypto map, or cyrpto profile. **Note** Configure this command or the **set security-association replay disable** command. The two commands are not used at the same time.
Step 5	**set security-association replay disable** **Example:** `Router (crypto-map)# set security-association replay disable`	Disables replay checking for a particular crypto map, dynamic crypto map, or crypto profile. **Note** Configure this command or the **set security-association replay window-size** command. The two commands are not used at the same time.

Troubleshooting Tips

- If your replay window size has not been set to a number that is high enough for the number of packets received, you will receive a system message such as the following:

  ```
  *Nov 17 19:27:32.279: %CRYPTO-4-PKT_REPLAY_ERR: decrypt: replay check failed
  connection id=1
  ```

  The above message is generated when a received packet is judged to be outside the anti-replay window.

Configuration Examples for IPsec Anti-Replay Window: Expanding and Disabling

This section includes the following configuration examples:

Global Expanding and Disabling of an Anti-Replay Window: Example

The following example shows that the anti-replay window size has been set globally to 1024:

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname VPN-Gateway1
!

boot-start-marker
boot-end-marker
!
!
clock timezone EST 0
no aaa new-model
ip subnet-zero
!
!
ip audit po max-events 100
no ftp-server write-enable
!
!
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key cisco123 address 192.165.201.2 !
crypto ipsec security-association replay window-size 1024 !
crypto ipsec transform-set basic esp-des esp-md5-hmac !
crypto map mymap 10 ipsec-isakmp
 set peer 192.165.201.2
 set transform-set basic
 match address 101
!
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Serial1/0
 ip address 192.165.200.2 255.255.255.252  serial restart-delay 0  crypto map mymap !
ip classless
ip route 0.0.0.0 0.0.0.0 192.165.200.1
no ip http server
no ip http secure-server
!
!
access-list 101 permit ip 192.168.1.0 0.0.0.255 172.16.2.0 0.0.0.255 access-list 101
remark Crypto ACL
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
!
```

```
!
end
```

Expanding and Disabling of an Anti-Replay Window for a Particular Crypto Map, Dynamic Crypto Map, or Crypto Profile: Example

The following example shows that anti-replay checking is disabled for IPsec connections to 172.17.150.2 but enabled (and the default window size is 64) for IPsec connections to 172.17.150.3 and 172.17.150.4:

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname networkserver1
!
enable secret 5 $1$KxKv$cbqKsZtQTLJLGPN.tErFZ1 enable password ww !
ip subnet-zero
!
cns event-service server

crypto isakmp policy 1
authentication pre-share

crypto isakmp key cisco170 address 172.17.150.2 crypto isakmp key cisco180 address
172.17.150.3 crypto isakmp key cisco190 address 172.17.150.4

crypto ipsec transform-set 170cisco esp-des esp-md5-hmac crypto ipsec transform-set
180cisco esp-des esp-md5-hmac crypto ipsec transform-set 190cisco esp-des esp-md5-hmac

crypto map ETH0 17 ipsec-isakmp
 set peer 172.17.150.2
 set security-association replay disable set transform-set 170cisco match address 170
crypto map ETH0 18 ipsec-isakmp  set peer 192.168.1.3 set transform-set 180cisco match
address 180 crypto map ETH0 19 ipsec-isakmp set peer 192.168.1.4 set transform-set
190cisco match address 190 !
interface Ethernet0
 ip address 172.17.150.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no mop enabled
 crypto map ETH0
!
interface Serial0
 ip address 172.16.160.1 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
!
ip classless
ip route 172.18.170.0 255.255.255.0 172.17.150.2 ip route 172.19.180.0 255.255.255.0
172.17.150.3 ip route 172.20.190.0 255.255.255.0 172.17.150.4 no ip http server !

access-list 170 permit ip 172.16.160.0 0.0.0.255 172.18.170.0 0.0.0.255 access-list 180
permit ip 172.16.160.0 0.0.0.255 172.19.180.0 0.0.0.255 access-list 190 permit ip
172.16.160.0 0.0.0.255 172.20.190.0 0.0.0.255 !
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
```

```
line con 0
transport input none
line aux 0
line vty 0 4
password ww
login
end
```

Additional References

The following sections provide references related to IPsec Anti-Replay Window: Expanding and Disabling.

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Security Command Reference, Release 12.3T*
IP security and encryption	"IP Security and Encryption" *section of Cisco IOS Security Configuration Guide, Release 12.3*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features

- **crypto ipsec security-association replay disable**
- **crypto ipsec security-association replay window-size**
- **set security-association replay disable**
- **set security-association replay window-size**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

IPSec Dead Peer Detection Periodic Message Option

First Published: May 1, 2004
Last Updated: August 21, 2007

The IPSec Dead Peer Detection Periodic Message Option feature allows you to configure your router to query the liveliness of its Internet Key Exchange (IKE) peer at regular intervals. The benefit of this approach over the default approach (on-demand dead peer detection) is earlier detection of dead peers.

History for IPSec Dead Peer Detection Periodic Message Option Feature

Release	Modification
12.3(7)T	This feature was introduced.
12.2(33)SRA	This feature was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This feature was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for IPSec Dead Peer Detection Periodic Message Option

Before configuring the IPSec Dead Peer Detection Periodic Message Option feature, you should have the following:

- Familiarity with configuring IP Security (IPSec).
- An IKE peer that supports DPD (dead peer detection). Implementations that support DPD include the Cisco VPN 3000 concentrator, Cisco PIX Firewall, Cisco VPN Client, and Cisco IOS software in all modes of operation—site-to-site, Easy VPN remote, and Easy VPN server.

Restrictions for IPSec Dead Peer Detection Periodic Message Option

Using periodic DPD potentially allows the router to detect an unresponsive IKE peer with better response time when compared to on-demand DPD. However, use of periodic DPD incurs extra overhead. When communicating to large numbers of IKE peers, you should consider using on-demand DPD instead.

Information About IPSec Dead Peer Detection Periodic Message Option

To configure IPSec Dead Peer Detection Periodic Message Option, you should understand the following concepts:

How DPD and Cisco IOS Keepalive Features Work

DPD and Cisco IOS keepalives function on the basis of the timer. If the timer is set for 10 seconds, the router will send a “hello” message every 10 seconds (unless, of course, the router receives a “hello” message from the peer). The benefit of IOS keepalives and periodic DPD is earlier detection of dead peers. However, IOS keepalives and periodic DPD rely on periodic messages that have to be sent with considerable frequency. The result of sending frequent messages is that the communicating peers must encrypt and decrypt more packets.

DPD also has an on-demand approach. The contrasting on-demand approach is the default. With on-demand DPD, messages are sent on the basis of traffic patterns. For example, if a router has to send outbound traffic and the liveliness of the peer is questionable, the router sends a DPD message to query the status of the peer. If a router has no traffic to send, it never sends a DPD message. If a peer is dead, and the router never has any traffic to send to the peer, the router will not find out until the IKE or IPSec security association (SA) has to be rekeyed (the liveliness of the peer is unimportant if the router is not trying to communicate with the peer). On the other hand, if the router has traffic to send to the peer, and the peer does not respond, the router will initiate a DPD message to determine the state of the peer.

Using the IPSec Dead Peer Detection Periodic Message Option

With the IPSec Dead Peer Detection Periodic Message Option feature, you can configure your router so that DPD messages are "forced" at regular intervals. This forced approach results in earlier detection of dead peers. For example, if a router has no traffic to send, a DPD message is still sent at regular intervals, and if a peer is dead, the router does not have to wait until the IKE SA times out to find out.

If you want to configure the DPD periodic message option, you should use the **crypto isakmp keepalive** command with the **periodic** keyword. If you do not configure the **periodic** keyword, the router defaults to the on-demand approach.

Note When the **crypto isakmp keepalive** command is configured, the Cisco IOS software negotiates the use of Cisco IOS keepalives or DPD, depending on which protocol the peer supports.

Using DPD and Cisco IOS Keepalive Features with Multiple Peers in the Crypto Map

DPD and IOS keepalive features can be used in conjunction with multiple peers in the crypto map to allow for stateless failover. DPD allows the router to detect a dead IKE peer, and when the router detects the dead state, the router deletes the IPSec and IKE SAs to the peer. If you configure multiple peers, the router will switch over to the next listed peer for a stateless failover.

Using DPD in an Easy VPN Remote Configuration

DPD can be used in an Easy VPN remote configuration. See the section "Configuring DPD for an Easy VPN Remote" section on page 5.

How to Configure IPSec Dead Peer Detection Periodic Message Option

This section contains the following prodedures:

Configuring a Periodic DPD Message

To configure a periodic DPD message, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp keepalive** *seconds* [*retries*] [**periodic** | **on-demand**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp keepalive** *seconds* [*retries*] [**periodic** | **on-demand**] **Example:** `Router (config)# crypto isakmp keepalive 10 periodic`	Allows the gateway to send DPD messages to the peer. • *seconds*—Number of seconds between DPD messages. • *retries*—(Optional) Number of seconds between DPD retries if the DPD message fails. • **periodic**—(Optional) DPD messages are sent at regular intervals. • **on-demand**—(Optional) DPD retries are sent on demand. This is the default behavior.

Configuring DPD and Cisco IOS Keepalives with Multiple Peers in the Crypto Map

To configure DPD and IOS keepalives to be used in conjunction with the crypto map to allow for stateless failover, perform the following steps. This configuration will cause a router to cycle through the peer list when it detects that the first peer is dead.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **set peer** {*host-name* [**dynamic**] | *ip-address*}

5. **set transform-set** *transform-set-name*
6. **match address** [*access-list-id* | *name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router (config)# crypto map green 1 ipsec-isakmp`	Enters crypto map configuration mode and creates or modifies a crypto map entry. • The **ipsec-isakmp** keyword indicates that IKE will be used to establish the IPSec SAs for protecting the traffic specified by this crypto map entry.
Step 4	**set peer** {*host-name* [**dynamic**] | *ip-address*} **Example:** `Router (config-crypto-map)# set peer 10.12.12.12`	Specifies an IPSec peer in a crypto map entry. • You can specify multiple peers by repeating this command.
Step 5	**set transform-set** *transform-set-name* **Example:** `Router (config-crypto-map)# set transform-set txfm`	Specifies which transform sets can be used with the crypto map entry. • You can specify more than one transform set name by repeating this command.
Step 6	**match address** [*access-list-id* | *name*] **Example:** `Router (config-crypto-map)# match address 101`	Specifies an extended access list for a crypto map entry.

Configuring DPD for an Easy VPN Remote

To configure DPD in an Easy VPN remote configuration, perform the following steps. This configuration also will cause a router to cycle through the peer list when it detects that the first peer is dead.

Note IOS keepalives are not supported for Easy VPN remote configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **crypto ipsec client ezvpn** *name*
4. **connect** {**auto** | **manual**}
5. **group** *group-name* **key** *group-key*
6. **mode** {**client** | **network-extension**}
7. **peer** {*ipaddress* | *hostname*}

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn ezvpn-config1`	Creates a Cisco Easy VPN remote configuration and enters the Cisco Easy VPN Remote configuration mode.
Step 4	**connect** {**auto** | **manual**} **Example:** `Router (config-crypto-ezvpn)# connect manual`	Manually establishes and terminates an IPSec VPN tunnel on demand. • The **auto** keyword option is the default setting.
Step 5	**group** *group-name* **key** *group-key* **Example:** `Router (config-crypto-ezvpn)# group unity key preshared`	Specifies the group name and key value for the Virtual Private Network (VPN) connection.
Step 6	**mode** {**client** | **network-extension**} **Example:** `Router (config-crypto-ezvpn)# mode client`	Specifies the VPN mode of operation of the router.
Step 7	**peer** {*ipaddress* | *hostname*} **Example:** `Router (config-crypto-ezvpn)# peer 10.10.10.10`	Sets the peer IP address or host name for the VPN connection. • A hostname can be specified only when the router has a DNS server available for host-name resolution. • This command can be repeated multiple times.

Verifying That DPD Is Enabled

DPD allows the router to clear the IKE state when a peer becomes unreachable. If DPD is enabled and the peer is unreachable for some time, you can use the **clear crypto session** command to manually clear IKE and IPSec SAs.

The **debug crypto isakmp** command can be used to verify that DPD is enabled.

SUMMARY STEPS

1. **enable**
2. **clear crypto session** [**local** *ip-address* [**port** *local-port*]] [**remote** *ip-address* [**port** *remote-port*]] | [**fvrf** *vrf-name*] [**ivrf** *vrf-name*]
3. **debug crypto isakmp**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto session** [**local** *ip-address* [**port** *local-port*]] [**remote** *ip-address* [**port** *remote-port*]]	[**fvrf** *vrf-name*] [**ivrf** *vrf-name*] **Example:** `Router# clear crypto session`	Deletes crypto sessions (IPSec and IKE SAs).
Step 3	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays messages about IKE events.

Configuration Examples for IPSec Dead Peer Detection Periodic Message Option

This section provides the following configuration examples:

Site-to-Site Setup with Periodic DPD Enabled: Example

The following configurations are for a site-to-site setup with no periodic DPD enabled. The configurations are for the IKE Phase 1 policy and for the IKE preshared key.

IKE Phase 1 Policy

```
crypto isakmp policy 1
  encryption 3des
```

```
  authentication pre-share
  group 2
!
```

IKE Preshared Key

```
crypto isakmp key kd94j1ksldz address 10.2.80.209 255.255.255.0
crypto isakmp keepalive 10 periodic
crypto ipsec transform-set esp-3des-sha esp-3des esp-sha-hmac
crypto map test 1 ipsec-isakmp
  set peer 10.2.80.209
  set transform-set esp-3des-sha
  match address 101
!
!
interface FastEthernet0
  ip address 10.1.32.14 255.255.255.0
  speed auto
  crypto map test
!
```

Easy VPN Remote with DPD Enabled: Example

The following configuration tells the router to send a periodic DPD message every 30 seconds. If the peer fails to respond to the DPD R_U_THERE message, the router will resend the message every 20 seconds (four transmissions altogether).

```
crypto isakmp keepalive 30 20 periodic
crypto ipsec client ezvpn ezvpn-config
  connect auto
  group unity key preshared
  mode client
  peer 10.2.80.209
!
!
interface Ethernet0
  ip address 10.2.3.4 255.255.255.0
  half-duplex
  crypto ipsec client ezvpn ezvpn-config inside
!
interface FastEthernet0
  ip address 10.1.32.14 255.255.255.0
  speed auto
  crypto ipsec client ezvpn ezvpn-config outside
```

Verifying DPD Configuration Using the debug crypto isakmp Command: Example

The following sample output from the **debug crypto isakmp** command verifies that IKE DPD is enabled:

```
*Mar 25 15:17:14.131: ISAKMP:(0:1:HW:2):IKE_DPD is enabled, initializing timers
```

To see that IKE DPD is enabled (and that the peer supports DPD): when periodic DPD is enabled, you should see the following debug messages at the interval specified by the command:

```
*Mar 25 15:18:52.107: ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:18:52.107: ISAKMP:(0:1:HW:2):purging node 899852982 *Mar 25 15:18:52.111:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_IM_ALIVE
*Mar 25 15:18:52.111: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
```

The above message corresponds to sending the DPD R_U_THERE message.

```
*Mar 25 15:18:52.123: ISAKMP (0:268435457): received packet from 10.2.80.209
dport 500 sport 500 Global (I) QM_IDLE
*Mar 25 15:18:52.123: ISAKMP: set new node -443923643 to QM_IDLE *Mar 25 15:18:52.131:
ISAKMP:(0:1:HW:2): processing HASH payload. message ID =
-443923643
*Mar 25 15:18:52.131: ISAKMP:(0:1:HW:2): processing NOTIFY R_U_THERE_ACK protocol 1
    spi 0, message ID = -443923643, sa = 81BA4DD4
*Mar 25 15:18:52.135: ISAKMP:(0:1:HW:2): DPD/R_U_THERE_ACK received from peer
10.2.80.209, sequence 0x9
*Mar 25 15:18:52.135: ISAKMP:(0:1:HW:2):deleting node -443923643 error FALSE
reason "informational (in) state 1"
*Mar 25 15:18:52.135: ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_PEER, IKE_INFO_NOTIFY *Mar
25 15:18:52.135: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
```

The above message corresponds to receiving the acknowledge (ACK) message from the peer.

```
Router#
*Mar 25 15:47:35.335: ISAKMP: set new node -90798077 to QM_IDLE *Mar 25 15:47:35.343:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:35.343: ISAKMP:(0:1:HW:2):purging node -90798077 *Mar 25 15:47:35.347:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_IM_ALIVE
*Mar 25 15:47:35.347: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

*Mar 25 15:47:36.611: ISAKMP:(0:1:HW:2):purging node 1515050537 *Mar 25 15:47:37.343:
ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:37.343: ISAKMP: set new node -1592471565 to QM_IDLE *Mar 25 15:47:37.351:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:37.351: ISAKMP:(0:1:HW:2):purging node -1592471565 *Mar 25 15:47:37.355:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:37.355: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

*Mar 25 15:47:39.355: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:39.355: ISAKMP: set new node 1758739401 to QM_IDLE *Mar 25 15:47:39.363:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:39.363: ISAKMP:(0:1:HW:2):purging node 1758739401 *Mar 25 15:47:39.367:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:39.367: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

*Mar 25 15:47:41.367: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
```

```
PEERS_ALIVE_TIMER
*Mar 25 15:47:41.367: ISAKMP: set new node 320258858 to QM_IDLE *Mar 25 15:47:41.375:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:41.379: ISAKMP:(0:1:HW:2):purging node 320258858 *Mar 25 15:47:41.379:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:41.379: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

*Mar 25 15:47:43.379: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:43.379: ISAKMP: set new node -744493014 to QM_IDLE *Mar 25 15:47:43.387:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:43.387: ISAKMP:(0:1:HW:2):purging node -744493014 *Mar 25 15:47:43.391:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:43.391: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

*Mar 25 15:47:45.391: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:45.391: ISAKMP:(0:1:HW:2):peer 10.2.80.209 not responding! *Mar 25
15:47:45.391: ISAKMP:(0:1:HW:2):peer does not do paranoid keepalives.

*Mar 25 15:47:45.391: ISAKMP:(0:1:HW:2):deleting SA reason "peers alive" state
(I) QM_IDLE       (peer 10.2.80.209) input queue 0
*Mar 25 15:47:45.395: ISAKMP: Unlocking IPSEC struct 0x81E5C4E8 from
delete_siblings, count 0
*Mar 25 15:47:45.395: %CRYPTO-5-SESSION_STATUS: Crypto tunnel is DOWN.  Peer
10.2.80.209:500       Id: 10.2.80.209
*Mar 25 15:47:45.399: ISAKMP: set new node -2061951065 to QM_IDLE *Mar 25 15:47:45.411:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:45.411: ISAKMP:(0:1:HW:2):purging node -2061951065 *Mar 25 15:47:45.411:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:45.411: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_DEST_SA

*Mar 25 15:47:45.415: ISAKMP:(0:1:HW:2):deleting SA reason "peers alive" state
(I) QM_IDLE       (peer 10.2.80.209) input queue 0
*Mar 25 15:47:45.415: ISAKMP: Unlocking IKE struct 0x81E5C4E8 for
isadb_mark_sa_deleted(), count 0
*Mar 25 15:47:45.415: ISAKMP: Deleting peer node by peer_reap for 10.2.80.209:
81E5C4E8
*Mar 25 15:47:45.415: ISAKMP:(0:1:HW:2):deleting node -1067612752 error TRUE
reason "peers alive"
*Mar 25 15:47:45.415: ISAKMP:(0:1:HW:2):deleting node -114443536 error TRUE
reason "peers alive"
*Mar 25 15:47:45.419: ISAKMP:(0:1:HW:2):deleting node 2116015069 error TRUE
reason "peers alive"
*Mar 25 15:47:45.419: ISAKMP:(0:1:HW:2):deleting node -1981865558 error TRUE
reason "peers alive"
*Mar 25 15:47:45.419: ISAKMP:(0:1:HW:2):Input = IKE_MESG_INTERNAL, IKE_PHASE1_DEL *Mar 25
15:47:45.419: ISAKMP:(0:1:HW:2):Old State = IKE_DEST_SA  New State =
IKE_DEST_SA

*Mar 25 15:47:45.419: ISAKMP: received ke message (4/1)
*Mar 25 15:47:45.419: ISAKMP: received ke message (3/1)
*Mar 25 15:47:45.423: ISAKMP: ignoring request to send delete notify (no ISAKMP
sa) src 10.1.32.14 dst 10.2.80.209 for SPI 0x3A7B69BF
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting SA reason "" state (I)
```

```
MM_NO_STATE (peer 10.2.80.209) input queue 0
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting node -1067612752 error FALSE
reason ""
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting node -114443536 error FALSE
reason ""
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting node 2116015069 error FALSE
reason ""
*Mar 25 15:47:45.427: ISAKMP:(0:1:HW:2):deleting node -1981865558 error FALSE
reason ""
*Mar 25 15:47:45.427: ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH *Mar 25
15:47:45.427: ISAKMP:(0:1:HW:2):Old State = IKE_DEST_SA  New State =
IKE_DEST_SA
```

The above message shows what happens when the remote peer is unreachable. The router sends one DPD R_U_THERE message and four retransmissions before it finally deletes the IPSec and IKE SAs.

DPD and Cisco IOS Keepalives Used in Conjunction with Multiple Peers in a Crypto Map: Example

The following example shows that DPD and Cisco IOS keepalives are used in conjunction with multiple peers in a crypto map configuration when IKE will be used to establish the security associations (SAs). In this example, an SA could be set up to the IPSec peer at 10.0.0.1, 10.0.0.2, or 10.0.0.3.

```
crypto map green 1 ipsec-isakmp
  set peer 10.0.0.1
  set peer 10.0.0.2
  set peer 10.0.0.3
  set transform-set txfm
  match address 101
```

DPD Used in Conjunction with Multiple Peers for an Easy VPN Remote: Example

The following example shows that DPD is used in conjunction with multiple peers in an Easy VPN remote configuration. In this example, an SA could be set up to the IPSec peer at 10.10.10.10, 10.2.2.2, or 10.3.3.3.

```
crypto ipsec client ezvpn ezvpn-config
  connect auto
  group unity key preshared
  mode client
  peer 10.10.10.10
  peer 10.2.2.2
  peer 10.3.3.3
```

Additional References

The following sections provide references related to IPSec Dead Peer Detection Periodic Message Option.

Related Documents

Related Topic	Document Title
Configuring IPSec	*"IP Security and Encryption" section of Cisco IOS Security Configuration Guide*
IPSec commands	*Cisco IOS Security Command Reference, Release 12.4 T*

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
DPD conforms to the Internet draft "draft-ietf-ipsec-dpd-04.txt," which is pending publication as an Informational RFC (a number has not yet been assigned).	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features

- **crypto isakmp keepalive**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

IPsec Preferred Peer

First Published: March 28, 2005
Last Updated: August 21, 2007

The IP Security (IPsec) Preferred Peer feature allows you to control the circumstances by which multiple peers on a crypto map are tried in a failover scenario.

This feature includes the following capabilities:

- Default peer configuration
- IPsec idle-timer usage with default peer

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for IPsec Preferred Peer" section on page 9.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for IPsec Preferred Peer

- You must have a properly defined, complete crypto map.

Restrictions for IPsec Preferred Peer

Default peer:

- This feature must be used in conjunction with Dead Peer Detection (DPD). It is most effective on a remote site running DPD in periodic mode. DPD detects the failure of a device quickly and resets the peer list so that the default peer is tried for the next attempted connection.
- Only one peer can be designated as the default peer in a crypto map.
- The default peer must be the first peer in the peer list.

IPsec idle-timer usage with default peer:

- This feature works only on the crypto map for which it is configured. You cannot configure the capability globally for all crypto maps.
- If there is a global idle timer, the crypto map idle-timer value must be different from the global value; otherwise, the idle timer is not added to the crypto map.

Information About IPsec Preferred Peer

To configure IPsec Preferred Peer, you need to understand the following concepts:

IPsec

IPsec is a framework of open standards developed by the Internet Engineering Task Force (IETF). IPsec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating Internet Protocol (IP) packets between participating IPsec devices (peers), such as Cisco routers.

IPsec provides the following network security services. These services are optional. In general, local security policy dictates the use of one or more of these services:

- Data Confidentiality—The IPsec sender can encrypt packets before transmitting them across a network.

- Data Integrity—The IPsec receiver can authenticate packets sent by the IPsec sender to ensure that the data has not been altered during transmission.
- Data Origin Authentication—The IPsec receiver can authenticate the source of the IPsec packets sent.
- Anti-Replay—The IPsec receiver can detect and reject replayed packets.

With IPsec, data can be transmitted across a public network without fear of observation, modification, or spoofing. This enables applications such as Virtual Private Networks (VPNs), including intranets, extranets, and remote user access.

IPsec provides secure tunnels between two peers, such as two routers. You define which packets are considered sensitive and should be sent through these secure tunnels, and you define the parameters that should be used to protect these sensitive packets, by specifying characteristics of these tunnels. When the IPsec peer sees such a sensitive packet, it sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer.

Dead Peer Detection

The VPN Client uses a keepalive mechanism called Dead Peer Detection (DPD) to check the availability of the VPN device on the other side of an IPsec tunnel. If the network is unusually busy or unreliable, you can increase the number of seconds that the VPN Client will wait before deciding whether the peer is no longer active.

Keepalive packets are not sent if traffic is received. This lowers the overhead associated with DPD, because on a heavily loaded network very few keepalive packets will be sent because traffic is being received on the tunnels. In addition, DPD sends keepalive packets only if there is user traffic to send (and no user traffic is received).

You can configure Internet Key Exchange (IKE) DPD so that DPD sends the keepalive packets whether or not there is outbound user data. That is, as long as there is no inbound user data, the keepalive packets are sent at the configured keepalive interval.

Default Peer Configuration

If a connection timeout occurs, the connection to the current peer is closed. The **set peer** command allows you to configure the first peer as the default peer. If there is a default peer, the next time a connection is initiated, the connection is directed to the default peer instead of to the next peer in the peer list. If the default peer is unresponsive, the next peer in the peer list becomes the current peer and future connections through the crypto map try that peer.

This capability is useful when traffic on a physical link stops due to the failure of a remote peer. DPD indicates that the remote peer is unavailable, but that peer remains the current peer.

A default peer facilitates the failover to a preferred peer that was previously unavailable, but has returned to service. Users can give preference to certain peers in the event of a failover. This is useful if the original failure was due to a network connectivity problem rather than failure of the remote peer.

Idle Timers

When a router running Cisco IOS software creates an IPsec security association (SA) for a peer, resources must be allocated to maintain the SA. The SA requires both memory and several managed timers. For idle peers, these resources are wasted. If enough resources are wasted by idle peers, the router could be prevented from creating new SAs with other peers.

IPsec SA idle timers increase the availability of resources by deleting SAs associated with idle peers. Because IPsec SA idle timers prevent the wasting of resources by idle peers, more resources are available to create new SAs when required.

If IPsec SA idle timers are not configured, only the global lifetimes for IPsec SAs are applied. SAs are maintained until the global timers expire, regardless of peer activity.

IPsec Idle-Timer Usage with Default Peer

If all connections to the current peer time out, the next time a connection is initiated it is directed to the default peer configured in the **set peer** command. If a default peer is not configured and there is a connection timeout, the current peer remains the one that timed out.

This enhancement helps facilitate a failover to a preferred peer that was previously unavailable but is in service now.

Peers on Crypto Maps

A crypto map set can contain multiple entries, each with a different access list. The router searches the crypto map entries in order, and attempts to match the packet to the access list specified in that entry.

When a packet matches a **permit** entry in a particular access list, and the corresponding crypto map entry is tagged as Cisco, connections are established with the remote peer as specified in the set peer statements within the crypto map.

How to Configure IPsec Preferred Peer

This section contains the following procedures:

- Configuring a Default Peer, page 4 (required)
- Configuring the Idle Timer, page 5 (optional)

Configuring a Default Peer

To configure a default peer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*]

4. **set peer** {*host-name* [**dynamic**] [**default**] | *ip-address* [**default**] }
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*] **Example:** `Router(config)# crypto map mymap 10 ipsec-isakmp`	Enters crypto map configuration mode. Creates or modifies a crypto map entry, creates a crypto profile that provides a template for configuration of dynamically created crypto maps, or configures a client accounting list.
Step 4	**set peer** {*host-name* [**dynamic**] [**default**] | *ip-address* [**default**] } **Example:** `Router(config-crypto-map)# set peer 10.0.0.2 default`	Specifies an IPsec peer in a crypto map entry. Ensures that the first peer specified is defined as the default peer.
Step 5	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits crypto map configuration mode and returns to global configuration mode.

Configuring the Idle Timer

To configure the idle timer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*]
4. **set security-association idletime** *seconds* [**default**]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*] **Example:** `Router(config)# crypto map mymap 10 ipsec-isakmp`	Enters crypto map configuration mode. Creates or modifies a crypto map entry, creates a crypto profile that provides a template for configuration of dynamically created crypto maps, or configures a client accounting list.
Step 4	**set security-association idletime** *seconds* [**default**] **Example:** `Router(config-crypto-map)# set security-association idletime 120 default`	Specifies the maximum amount of time for which the current peer can be idle before the default peer is used.
Step 5	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits crypto map configuration mode and returns to global configuration mode.

Configuration Examples for IPsec Preferred Peer

Configuring a Default Peer: Example

The following example shows that the first peer, at IP address 10.1.1.1, is the default peer:

```
crypto map tohub 1 ipsec-isakmp
 set peer 10.1.1.1 default
 set peer 10.2.2.2
```

Configuring the IPsec Idle Timer: Example

In the following example, if the current peer is idle for 120 seconds, the default peer 10.1.1.1 (which was specified in the **set peer** command) is used for the next attempted connection:

```
crypto map tohub 1 ipsec-isakmp
```

```
set peer 10.1.1.1 default
set peer 10.2.2.2
set security-association idletime 120 default
```

Additional References

The following sections provide references related to IPsec Preferred Peer.

Related Documents

Related Topic	Document Title
IPsec	*Cisco IOS Security Configuration Guide, Release 12.4* *Cisco IOS Security Command Reference, Release 12.4T*
Crypto map	*Cisco IOS Security Configuration Guide, Release 12.4* *Cisco IOS Security Command Reference, Release 12.4T*
DPD	*IPSec Dead Peer Detection Periodic Message Option*, Release 12.3(7)T *Cisco IOS Security Configuration Guide, Release 12.4*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features

- **set peer (IPsec)**
- **set security-association idle-time**

For information about these commands, see the Cisco IOS Security Command Reference at

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at

http://tools.cisco.com/Support/CLILookup or the Master Command List.

Feature Information for IPsec Preferred Peer

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for IPsec Preferred Peer

Feature Name	Releases	Feature Information
IPsec Preferred Peer	12.3(14)T 12.2(33)SRA 12.2(33)SXH	The IPsec Preferred Peer feature allows you to control the circumstances by which multiple peers on a crypto map are tried in a failover scenario. In 12.3(14)T, this feature was introduced. In 12.2(33)SRA, this feature, the **set peer** (IPsec) command, and the **set security-association idle-time** command were integrated into this release.
IPSEC Preferred Peer	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Glossary

crypto access list—A list that defines which IP traffic will be protected by crypto and which traffic will not be protected by crypto.

crypto map—A map that specifies which traffic should be protected by IPsec, where IPsec-protected traffic should be sent, and what IPsec transform sets should be applied to this traffic.

dead peer detection—A feature that allows the router to detect an unresponsive peer.

keepalive message—A message sent by one network device to inform another network device that the virtual circuit between the two is still active.

peer—Router or other device that participates in IPsec and IKE. In IPsec, peers are devices or entities that communicate securely either through the exchange of keys or the exchange of digital certificates.

SA—security association. An instance of security policy and keying material applied to a data flow. Both IKE and IPsec use SAs, although SAs are independent of one another. IPsec SAs are unidirectional and are unique in each security protocol. An IKE SA is used by IKE only, and unlike the IPsec SA, it is bidirectional. IKE negotiates and establishes SAs on behalf of IPsec. A user also can establish IPsec SAs manually. A set of SAs are needed for a protected data pipe, one per direction per protocol. For example, if you have a pipe that supports Encapsulating Security Payload (ESP) between peers, one ESP SA is required for each direction. SAs are identified uniquely by destination (IPsec endpoint) address, security protocol (AH or ESP), and security parameter index (SPI).

transform set—An acceptable combination of security protocols, algorithms, and other settings to apply to IPsec-protected traffic. During the IPsec security association negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

IPSec Security Association Idle Timers

When a router running the Cisco IOS software creates an IPsec security association (SA) for a peer, resources must be allocated to maintain the SA. The SA requires both memory and several managed timers. For idle peers, these resources are wasted. If enough resources are wasted by idle peers, the router could be prevented from creating new SAs with other peers. The IPsec Security Association Idle Timers feature introduces a configurable idle timer to monitor SAs for activity, allowing SAs for idle peers to be deleted. Benefits of this feature include:

- Increased availability of resources
- Improved scalability of Cisco IOS IPsec deployments

Feature Specifications for IPsec Security Association Idle Timers

Feature History	
Release	**Modification**
12.2(15)T	This feature was introduced.
12.3(14)T	The **set security-association idle-time** command was added, allowing for the configuration of an IPsec idle timer for a specified crypto map.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Supported Platforms	
Cisco 1700 series access routers, Cisco 2400 series integrated access devices, Cisco 2600 series multiservice platforms, Cisco 3600 series multiservice platforms, Cisco 3700 series multiservice access routers, Cisco 7100 series VPN routers, Cisco 7200 series routers, Cisco 7400 series routers, Cisco 7500 series routers, Cisco 801–804 ISDN routers, Cisco 805 serial router, Cisco 806 broadband router, Cisco 811, Cisco 813, Cisco 820, Cisco 827 ADSL router, Cisco 828 G.SHDSL router, Cisco 8850-RPM, Cisco 950, Cisco AS5350 universal gateway, Cisco AS5400 series universal gateways, Cisco integrated communications system 7750, Cisco MC3810 series multiservice access concentrators, Cisco ubr7200, Cisco ubr900 series cable access routers	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

Prerequisites for IPsec Security Association Idle Timers

You must configure Internet Key Exchange (IKE) as described in the "Configuring Internet Key Exchange Security Protocol" chapter of the *Cisco IOS Security Configuration Guide*, Release 12.2.

Information About IPsec Security Association Idle Timers

To configure the IPsec Security Association Idle Timers feature, you must understand the following concepts:

Lifetimes for IPsec Security Associations

The Cisco IOS software currently allows the configuration of lifetimes for IPsec SAs. Lifetimes can be configured globally or per crypto map. There are two lifetimes: a "timed" lifetime and a "traffic-volume" lifetime. A security association expires after the first of these lifetimes is reached.

IPsec Security Association Idle Timers

The IPsec SA idle timers are different from the global lifetimes for IPsec SAs. The expiration of the global lifetime is independent of peer activity. The IPsec SA idle timer allows SAs associated with inactive peers to be deleted before the global lifetime has expired.

If the IPsec SA idle timers are not configured, only the global lifetimes for IPsec SAs are applied. SAs are maintained until the global timers expire, regardless of peer activity.

Note If the last IPsec SA to a given peer is deleted due to idle timer expiration, the Internet Key Exchange (IKE) SA to that peer will also be deleted.

Benefits of IPsec Security Association Idle Timers

Increased Availability of Resources

Configuring the IPsec Security Association Idle Timers feature increases the availability of resources by deleting SAs associated with idle peers.

Improved Scalability of Cisco IOS IPsec Deployments

Because the IPsec Security Association Idle Timers feature prevents the wasting of resources by idle peers, more resources will be available to create new SAs as required.

How to Configure IPsec Security Association Idle Timers

Configuring the IPsec SA Idle Timer Globally

This task configures the IPsec SA idle timer globally. The idle timer configuration will be applied to all SAs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec security-association idle-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec security-association idle-time** *seconds* **Example:** `Router(config)# crypto ipsec security-association idle-time 600`	Configures the IPsec SA idle timer. • The *seconds* argument specifies the time, in seconds, that the idle timer will allow an inactive peer to maintain an SA. Valid values for the *seconds* argument range from 60 to 86400.

Configuring the IPsec SA Idle Timer per Crypto Map

This task configures the IPsec SA idle timer for a specified crypto map. The idle timer configuration will be applied to all SAs under the specified crypto map.

Note This configuration task was available effective with Cisco IOS Release 12.3(14)T.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-number* **ipsec-isakmp**
4. **set security-association idle-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-number* **ipsec-isakmp** **Example:** `Router(config)# crypto map test 1 ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode.
Step 4	**set security-association idle-time** *seconds* **Example:** `Router(config-crypto-map)# set security-association idle-time 600`	Specifies the maximum amount of time for which the current peer can be idle before the default peer is used. • The *seconds* argument is the number of seconds for which the current peer can be idle before the default peer is used. Valid values are 60 to 86400.

Configuration Examples for IPsec Security Association Idle Timers

Configuring the IPsec SA Idle Timer Globally Example

The following example globally configures the IPsec SA idle timer to drop SAs for inactive peers after 600 seconds:

```
crypto ipsec security-association idle-time 600
```

Configuring the IPsec SA Idle Timer per Crypto Map Example

The following example configures the IPsec SA idle timer for the crypto map named test to drop SAs for inactive peers after 600 seconds:

```
crypto map test 1 ipsec-isakmp
 set security-association idle-time 600
```

Note The above configuration was not available until Cisco IOS Release 12.3(14)T.

Additional References

For additional information related to IPsec Security Association Idle Timers, see the following sections:

Related Documents

Related Topic	Document Title
Additional information about configuring IKE	"Configuring Internet Key Exchange Security Protocol" chapter of the *Cisco IOS Security Configuration Guide*, Release 12.2
Additional information about configuring global lifetimes for IPsec SAs	"Configuring IPSec Network Security" chapter ofthe *Cisco IOS Security Configuration Guide*, Release 12.2
Additional Security commands	*Cisco IOS Security Command Reference*, Release 12.2 T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features

- **crypto ipsec security-association idle-time**
- **set security-association idle-time**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

IPSec VPN Accounting

The IPSec VPN Accounting feature allows for a session to be accounted for by indicating when the session starts and when it stops.

A VPN session is defined as an Internet Key Exchange (IKE) security association (SA) and the one or more SA pairs that are created by the IKE SA. The session starts when the first IP Security (IPSec) pair is created and stops when all IPSec SAs are deleted.

Session identifying information and session usage information is passed to the Remote Authentication Dial-In User Service (RADIUS) server via standard RADIUS attributes and vendor-specific attributes (VSAs).

Feature Specifications for IPSec VPN Accounting

Feature History	
Release	**Modification**
12.2(15)T	This feature was introduced.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Supported Platforms	
Cisco 2610–2613, Cisco 2620–Cisco 2621, Cisco 2650–Cisco 2651, Cisco 3620, Cisco 3640, Cisco 3660, Cisco 3725, Cisco 3745, Cisco 7100, Cisco 7200, Cisco 7400, Cisco ubr7100, Cisco ubr7200.	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for IPSec VPN Accounting

You need to understand how to configure RADIUS and authentication, authorization, and accounting (AAA) accounting. For information about configuring RADIUS and AAA, refer to the following documents:

- *Configuring Basic AAA RADIUS for Dial-In Clients*
- How Does RADIUS Work?
- The chapter "Configuring RADIUS" in the *Cisco IOS Security Configuration Guide*
- The chapter "RADIUS Commands" in the *Cisco IOS Security Command Reference*, Release 12.2
- The chapter "Configuring Accounting" in the *Cisco IOS Security Configuration Guide*, Release 12.2

You also need to know how to configure IPSec accounting. For information about configuring IPSec accounting, refer to the chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*, Release 12.2.

Information About IPSec VPN Accounting

To configure IPSec VPN accounting, you must understand the following concepts:

RADIUS Accounting

For many large networks, it is required that user activity be recorded for auditing purposes. The method that is used most is RADIUS accounting.

RADIUS accounting allows for a session to be accounted for by indicating when the session starts and when it stops. Additionally, session identifying information and session usage information will be passed to the RADIUS server via RADIUS attributes and VSAs.

RADIUS Start Accounting

The RADIUS Start packet contains many attributes that generally identify who is requesting the service and of what the property of that service consists. Table 1 represents the attributes required for the start.

Table 1 RADIUS Accounting Start Packet Attributes

RADIUS Attributes Value	Attribute	Description
1	user-name	Username used in extended authentication (XAUTH).The username may be NULL when XAUTH is not used.
4	nas-ip-address	Identifying IP address of the network access server (NAS) that serves the user. It should be unique to the NAS within the scope of the RADIUS server.
5	nas-port	Physical port number of the NAS that serves the user.
8	framed-ip-address	Private address allocated for the IP Security (IPSec) session.
40	acct-status-type	Status type. This attribute indicates whether this accounting request marks the beginning (start), the end (stop), or an update of the session.
41	acct-delay-time	Number of seconds the client has been trying to send a particular record.
44	acct-session-id	Unique accounting identifier that makes it easy to match start and stop records in a log file.
26	vrf-id	String that represents the name of the Virtual Route Forwarder (VRF).
26	isakmp-initiator-ip	Endpoint IP address of the remote Internet Key Exchange (IKE) initiator (V4).
26	isakmp-group-id	Name of the VPN group profile used for accounting.
26	isakmp-phase1-id	Phase 1 identification (ID) used by IKE (for example, domain name [DN], fully qualified domain name [FQDN], IP address) to help identify the session initiator.

RADIUS Stop Accounting

The RADIUS Stop packet contains many attributes that identify the usage of the session. Table 2 represents the additional attributes required for the RADIUS stop packet. It is possible that only the stop packet will be sent without the start if configured to do so. If only the stop packet is sent, this allows an easy way to reduce the number of records going to the AAA server.

Table 2 RADIUS Accounting Stop Packet Attributes

RADIUS Attributes Value	Attribute	Description
42	acct-input-octets	Number of octets that have been received from the Unity client over the course of the service that is being provided.
43	acct-output-octets	Number of octets that have been sent to the Unity client in the course of delivering this service.

Table 2 RADIUS Accounting Stop Packet Attributes (continued)

RADIUS Attributes Value	Attribute	Description
46	acct-session-time	Length of time (in seconds) that the Unity client has received service.
47	acct-input-packets	Quantity of packets that have been received from the Unity client in the course of delivering this service.
48	acct-output-packets	Quantity of packets that have been sent to the Unity client in the course of delivering this service.
49	acct-terminate-cause	For future use.
52	acct-input-gigawords	How many times the Acct-Input-Octets counter has wrapped around the 2^{32} (2 to the 32nd power) over the course of this service.
52	acct-output-gigawords	How many times the Acct-Input-Octets counter has wrapped around the 2^{32} (2 to the 32nd power) over the course of this service.

RADIUS Update Accounting

RADIUS accounting updates are supported. Packet and octet counts are shown in the updates. To learn more about AAA, refer to the following documents:

- Configuring Basic AAA RADIUS for Dial-In Clients
- The chapter "RADIUS Commands" in the *Cisco IOS Security Command Reference*, Release 12.2 T
- How to Assign Privilege Levels with TACACS+ and RADIUS
- Other AAA documentation at the Cisco.com website

IKE and IPSec Subsystem Interaction

Accounting Start

If IPSec accounting is configured, after IKE phases are complete, an accounting start record is generated for the session. New accounting records are not generated during a rekeying.

The following is an account start record that was generated on a router and that is to be sent to the AAA server that is defined:

```
*Aug 23 04:06:20.131: RADIUS(00000002): sending
*Aug 23 04:06:20.131: RADIUS(00000002): Send Accounting-Request to 10.1.1.4:1646 id 4, len
220
*Aug 23 04:06:20.131: RADIUS:  authenticator 38 F5 EB 46 4D BE 4A 6F - 45 EB EF 7D B7 19
FB 3F
*Aug 23 04:06:20.135: RADIUS:  Acct-Session-Id     [44]  10  "00000001"
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  31
*Aug 23 04:06:20.135: RADIUS:   Cisco AVpair       [1]   25  "isakmp-group-id=cclient"
*Aug 23 04:06:20.135: RADIUS:  Framed-IP-Address   [8]   6   10.13.13.1
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  20
*Aug 23 04:06:20.135: RADIUS:   Cisco AVpair       [1]   14  "vrf-id=cisco"
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  35
```

```
*Aug 23 04:06:20.135: RADIUS:   Cisco AVpair       [1]   29  "isakmp-initator-ip=11.1.2.2"
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  36
*Aug 23 04:06:20.135: RADIUS:   Cisco AVpair       [1]   30  "connect-progress=No
Progress"
*Aug 23 04:06:20.135: RADIUS:  User-Name           [1]   13  "joe@cclient"
*Aug 23 04:06:20.135: RADIUS:  Acct-Status-Type    [40]  6   Start                     [1]
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  25
*Aug 23 04:06:20.135: RADIUS:   cisco-nas-port     [2]   19  "FastEthernet0/0.1"
*Aug 23 04:06:20.135: RADIUS:  NAS-Port            [5]   6   0
*Aug 23 04:06:20.135: RADIUS:  NAS-IP-Address      [4]   6   10.1.1.147
*Aug 23 04:06:20.135: RADIUS:  Acct-Delay-Time     [41]  6   0
*Aug 23 04:06:20.139: RADIUS: Received from id 21645/4 10.1.1.4:1646, Accounting-response,
len 20
*Aug 23 04:06:20.139: RADIUS:  authenticator B7 E3 D0 F5 61 9A 89 D8 - 99 A6 8A 8A 98 79
9D 5D
```

Accounting Stop

An accounting stop packet is generated when there are no more flows (IPSec SA pairs) with the remote peer.

The accounting stop records contain the following information:

- Packets out
- Packets in
- Octets out
- Gigawords in
- Gigawords out

Below is an account start record that was generated on a router. The account start record is to be sent to the AAA server that is defined.

```
*Aug 23 04:20:16.519: RADIUS(00000003): Using existing nas_port 0
*Aug 23 04:20:16.519: RADIUS(00000003): Config NAS IP: 100.1.1.147
*Aug 23 04:20:16.519: RADIUS(00000003): sending
*Aug 23 04:20:16.519: RADIUS(00000003): Send Accounting-Request to 100.1.1.4:1646 id 19,
len 238
*Aug 23 04:20:16.519: RADIUS:  authenticator 82 65 5B 42 F0 3F 17 C3 - 23 F3 4C 35 A2 8A
3E E6
*Aug 23 04:20:16.519: RADIUS:  Acct-Session-Id     [44]  10  "00000002"
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  20
*Aug 23 04:20:16.519: RADIUS:   Cisco AVpair       [1]   14  "vrf-id=cisco"
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  35
*Aug 23 04:20:16.519: RADIUS:   Cisco AVpair       [1]   29  "isakmp-initator-ip=11.1.1.2"
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  36
*Aug 23 04:20:16.519: RADIUS:   Cisco AVpair       [1]   30  "connect-progress=No
Progress"
*Aug 23 04:20:16.519: RADIUS:  Acct-Session-Time   [46]  6   709
*Aug 23 04:20:16.519: RADIUS:  Acct-Input-Octets   [42]  6   152608
*Aug 23 04:20:16.519: RADIUS:  Acct-Output-Octets  [43]  6   152608
*Aug 23 04:20:16.519: RADIUS:  Acct-Input-Packets  [47]  6   1004
*Aug 23 04:20:16.519: RADIUS:  Acct-Output-Packets [48]  6   1004
*Apr 23 04:20:16.519: RADIUS:  Acct-Input-Giga-Word[52]  6   0
*Apr 23 04:20:16.519: RADIUS:  Acct-Output-Giga-Wor[53]  6   0
*Aug 23 04:20:16.519: RADIUS:  Acct-Terminate-Cause[49]  6   none                      [0]
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  32
*Aug 23 04:20:16.519: RADIUS:   Cisco AVpair       [1]   26  "disc-cause-ext=No Reason"
*Aug 23 04:20:16.519: RADIUS:  Acct-Status-Type    [40]  6   Stop                      [2]
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  25
*Aug 23 04:20:16.519: RADIUS:   cisco-nas-port     [2]   19  "FastEthernet0/0.1"
```

```
*Aug 23 04:20:16.519: RADIUS:  NAS-Port            [5]   6   0
*Aug 23 04:20:16.519: RADIUS:  NAS-IP-Address      [4]   6   100.1.1.147
*Aug 23 04:20:16.519: RADIUS:  Acct-Delay-Time     [41]  6   0
*Aug 23 04:20:16.523: RADIUS: Received from id 21645/19 100.1.1.4:1646,
Accounting-response, len 20
*Aug 23 04:20:16.523: RADIUS:  authenticator F1 CA C1 28 CE A0 26 C9 - 3E 22 C9 DA EA B8
22 A0
```

Accounting Updates

If accounting updates are enabled, accounting updates are sent while a session is “up.” The update interval is configurable. To enable the accounting updates, use the **aaa accounting update** command.

The following is an accounting update record that is being sent from the router:

```
Router#
*Aug 23 21:46:05.263: RADIUS(00000004): Using existing nas_port 0
*Aug 23 21:46:05.263: RADIUS(00000004): Config NAS IP: 100.1.1.147
*Aug 23 21:46:05.263: RADIUS(00000004): sending
*Aug 23 21:46:05.263: RADIUS(00000004): Send Accounting-Request to 100.1.1.4:1646 id 22,
len 200
*Aug 23 21:46:05.263: RADIUS:  authenticator 30 FA 48 86 8E 43 8E 4B - F9 09 71 04 4A F1
52 25
*Aug 23 21:46:05.263: RADIUS:  Acct-Session-Id     [44]  10  "00000003"
*Aug 23 21:46:05.263: RADIUS:  Vendor, Cisco       [26]  20
*Aug 23 21:46:05.263: RADIUS:   Cisco AVpair       [1]   14  "vrf-id=cisco"
*Aug 23 21:46:05.263: RADIUS:  Vendor, Cisco       [26]  35
*Aug 23 21:46:05.263: RADIUS:   Cisco AVpair       [1]   29  "isakmp-initator-ip=11.1.1.2"
*Aug 23 21:46:05.263: RADIUS:  Vendor, Cisco       [26]  36
*Aug 23 21:46:05.263: RADIUS:   Cisco AVpair       [1]   30  "connect-progress=No
Progress"
*Aug 23 21:46:05.263: RADIUS:  Acct-Session-Time   [46]  6   109
*Aug 23 21:46:05.263: RADIUS:  Acct-Input-Octets   [42]  6   608
*Aug 23 21:46:05.263: RADIUS:  Acct-Output-Octets  [43]  6   608
*Aug 23 21:46:05.263: RADIUS:  Acct-Input-Packets  [47]  6   4
*Aug 23 21:46:05.263: RADIUS:  Acct-Output-Packets [48]  6   4
*Aug 23 21:46:05.263: RADIUS:  Acct-Status-Type    [40]  6   Watchdog                  [3]
*Aug 23 21:46:05.263: RADIUS:  Vendor, Cisco       [26]  25
*Aug 23 21:46:05.263: RADIUS:   cisco-nas-port     [2]   19  "FastEthernet0/0.1"
*Aug 23 21:46:05.263: RADIUS:  NAS-Port            [5]   6   0
*Aug 23 21:46:05.263: RADIUS:  NAS-IP-Address      [4]   6   100.1.1.147
*Aug 23 21:46:05.263: RADIUS:  Acct-Delay-Time     [41]  6   0
*Aug 23 21:46:05.267: RADIUS: Received from id 21645/22 100.1.1.4:1646,
Accounting-response, len 20
*Aug 23 21:46:05.267: RADIUS:  authenticator 51 6B BB 27 A4 F5 D7 61 - A7 03 73 D3 0A AC
1C
```

How to Configure IPSec VPN Accounting

This section contains the following procedures:

Configuring IPSec VPN Accounting

To enable IPSec VPN Accounting, you need to perform the following required task:

Prerequisites

Before configuring IPSec VPN accounting, you must first configure IPSec. To learn about configuring IPSec, refer to the following documents:

- The chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*, Release 12.2
- Other IPSec documentation at the Cisco.com website

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** *list-name method*
5. **aaa authorization network** *list-name method*
6. **aaa accounting network** *list-name* **start-stop [broadcast] group** *group-name*
7. aaa session-id common
8. crypto isakmp profile *profile-name*
9. vrf *ivrf*
10. match identity group *group-name*
11. client authentication list *list-name*
12. isakmp authorization list *list-name*
13. client configuration address [**initiate** | respond]
14. accounting *list-name*
15. exit
16. crypto dynamic-map *dynamic-map-name dynamic-seq-num*
17. set transform-set *transform-set-name*
18. set isakmp-profile *profile-name*
19. reverse-route [**r**emote-peer]
20. exit
21. crypto map *map-name* ipsec-isakmp dynamic *dynamic-template-name*
22. radius-server host *ip-address* [auth-port *port-number]* [acct-port *port-number*]
23. **radius-server key** *string*
24. radius-server vsa send accounting
25. interface *interface-id*
26. crypto map *map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables periodic interim accounting records to be sent to the accounting server.
Step 4	**aaa authentication login** *list-name method* **Example:** `Router (config)# aaa authentication login cisco-client group radius`	Enforces authentication, authorization, and accounting (AAA) authentication for extended authorization (XAUTH) via RADIUS or local.
Step 5	**aaa authorization network** *list-name method* **Example:** `Router (config)# aaa authorization network cisco-client group radius`	Sets AAA authorization parameters on the remote client from RADIUS or local.
Step 6	**aaa accounting network list-name start-stop [broadcast] group** *group-name* **Example:** `Router (config)# aaa accounting network acc start-stop broadcast group radius`	Enables AAA accounting of requested services for billing or security purposes when you use RADIUS or TACACS+.
Step 7	**aaa session-id common** **Example:** `Router (config)# aaa session-id common`	Specifies whether the same session ID will be used for each AAA accounting service type within a call or whether a different session ID will be assigned to each accounting service type.
Step 8	**crypto isakmp profile** *profile-name* **Example:** `Route (config)# crypto isakmp profile cisco`	Audits IP security (IPSec) user sessions and enters isakmp-profile submode.
Step 9	**vrf** *ivrf* **Example:** `Router (conf-isa-prof)# vrf cisco`	Associates the on-demand address pool with a Virtual Private Network (VPN) routing and forwarding (VRF) instance name.

	Command or Action	Purpose
Step 10	**match identity group** *group-name* **Example:** `Router(conf-isa-prof)# match identity group cisco`	Matches an identity from a peer in an ISAKMP profile.
Step 11	**client authentication list** *list-name* **Example:** `Router(conf-isa-prof)# client authentication list cisco`	Configures Internet Key Exchange (IKE) extended authentication (XAUTH) in an Internet Security Association and Key Management Protocol (ISAKMP) profile.
Step 12	**isakmp authorization list** *list-name* **Example:** `Router(conf-isa-prof)# isakmp authorization list cisco-client`	Configures an IKE shared secret and other parameters using the AAA server in an ISAKMP profile. The shared secret and other parameters are generally pushed to the remote peer via mode configuration (MODECFG).
Step 13	**client configuration address [initiate | respond]** **Example:** `Router(conf-isa-prof)# client configuration address respond`	Configures IKE mode configuration (MODECFG) in the ISAKMP profile.
Step 14	**accounting** *list-name* **Example:** `Router(conf-isa-prof)# accounting acc`	Enables AAA accounting services for all peers that connect via this ISAKMP profile.
Step 15	**exit** **Example:** `Router(conf-isa-prof)# exit`	Exits isakmp-profile submode.
Step 16	**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **Example:** `Router(config)# crypto dynamic-map mymap 10 ipsec-isakmp`	Creates a dynamic crypto map template and enters the crypto map configuration command mode.
Step 17	**set transform-set** *transform-set-name* **Example:** `Router(config-crypto-map)# set transform-set aswan`	Specifies which transform sets can be used with the crypto map template.
Step 18	**set isakmp-profile** *profile-name* **Example:** `Router(config-crypto-map)# set isakmp-profile cisco`	Sets the ISAKMP profile name.

	Command or Action	Purpose
Step 19	**reverse-route [remote-peer]** **Example:** `Router(config-crypto-map)# reverse-route`	Allows routes (ip addresses) to be injected for destinations behind the VPN remote tunnel endpoint and may include a route to the tunnel endpoint itself (using the **remote-peer** keyword for the crypto map.
Step 20	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits dynamic crypto map configuration mode.
Step 21	`crypto map` *map-name* `ipsec-isakmp dynamic` *dynamic-template-name* **Example:** `Router(config)# crypto map mymap ipsec-isakmp dynamic dmap`	Enters crypto map configuration mode
Step 22	**radius-server host** *ip-address* [**auth-port** *port-number]* [**acct-port** *port-number*] **Example:** `Router(config)# radius-server host 172.16.1.4`	Specifies a RADIUS server host.
Step 23	**radius-server key** *string* **Example:** `Router(config)# radius-server key nsite`	Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.
Step 24	**radius-server vsa send accounting** **Example:** `Router(config)# radius-server vsa send accounting`	Configures the network access server to recognize and use vendor-specific attributes.
Step 25	**interface** *type slot/port* **Example:** `Router(config)# interface FastEthernet 1/0`	Configures an interface type and enters interface configuration mode.
Step 26	**crypto map** *map-name* **Example:** `Router(config-if)# crypto map mymap`	Applies a previously defined crypto map set to an interface.

Configuring Accounting Updates

To send accounting updates while a session is “up,” perform the following optional task:

Prerequisites

Before you configure accounting updates, you must first configure IPSec VPN accounting. See the section “Configuring IPSec VPN Accounting.”

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting update periodic** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa accounting update periodic** *number* **Example:** `Router (config)# aaa accounting update periodic 1-2147483647`	(Optional) Enables periodic interim accounting records to be sent to the accounting server.

Troubleshooting for IPSec VPN Accounting

To display messages about IPSec accounting events, perform the following optional task:

SUMMARY STEPS

1. **enable**
2. **debug crypto isakmp aaa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto isakmp aaa** **Example:** `Router# debug crypto isakmp aaa`	Displays messages about Internet Key Exchange (IKE) events. • The **aaa** keyword specifies accounting events.

Configuration Examples for IPSec VPN Accounting

Accounting and ISAKMP-Profile Example

The following example shows a configuration for supporting remote access clients with accounting and ISAKMP profiles:

```
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname sheep
!
aaa new-model
!
!
aaa accounting network ipsecaaa start-stop group radius
aaa accounting update periodic 1
aaa session-id common
ip subnet-zero
ip cef
!
!
no ip domain lookup
ip domain name cisco.com
ip name-server 172.29.2.133
ip name-server 172.29.11.48
!
!
crypto isakmp policy 1
authentication pre-share
group 2
!
crypto isakmp policy 10
hash md5
authentication pre-share
lifetime 200
crypto isakmp key cisco address 172.31.100.2

crypto iakmp client configuration group cclient
 key jegjegjhrg
 pool addressA

crypto-isakmp profile groupA
 vrf cisco
 match identity group cclient
 client authentication list cisco-client
 isakmp authorization list cisco-client
 client configuration address respond
 accounting acc
!
!
crypto ipsec transform-set esp-des-md5 esp-des esp-md5-hmac
!
crypto dynamic-map remotes 1
set peer 172.31.100.2
```

```
set security-association lifetime seconds 120
set transform-set esp-des-md5
reverse-route

!
crypto map test 10 ipsec-isakmp dynamic remotes
!
voice call carrier capacity active
!
interface Loopback0
ip address 10.20.20.20 255.255.255.0
no ip route-cache
no ip mroute-cache
!
interface FastEthernet0/0
ip address 10.2.80.203 255.255.255.0
no ip mroute-cache
load-interval 30
duplex full
!
interface FastEthernet1/0
ip address 192.168.219.2 255.255.255.0
no ip mroute-cache
duplex auto
speed auto
!
interface FastEthernet1/1
ip address 172.28.100.1 255.255.255.0
no ip mroute-cache
duplex auto
speed auto
crypto map test
!
no fair-queue
ip default-gateway 10.2.80.1
ip classless
ip route 10.0.0.0 0.0.0.0 10.2.80.1
ip route 10.20.0.0 255.0.0.0 10.2.80.56
ip route 10.10.10.0 255.255.255.0 172.31.100.2
ip route 10.0.0.2 255.255.255.255 10.2.80.73

ip local pool addressA 192.168.1.1 192.168.1.253
no ip http server
ip pim bidir-enable
!
!
ip access-list extended encrypt
permit ip host 10.0.0.1 host 10.5.0.1
!
access-list 101 permit ip host 10.20.20.20 host 10.10.10.10
!
!
radius-server host 172.27.162.206 auth-port 1645 acct-port 1646 key cisco123
radius-server retransmit 3
radius-server authorization permit missing Service-Type
radius-server vsa send accounting
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!
!
```

```
gatekeeper
shutdown
!
!
line con 0
exec-timeout 0 0
exec prompt timestamp
line aux 0
line vty 5 15
 ntp server 172.31.150.52
end
```

Accounting Without ISAKMP Profiles Example

The following example shows a full Cisco IOS configuration that supports accounting remote access peers when ISAKMP profiles are not used:

```
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname sheep
!
aaa new-model
!
!
aaa accounting network ipsecaaa start-stop group radius
aaa accounting update periodic 1
aaa session-id common
ip subnet-zero
ip cef
!
!
no ip domain lookup
ip domain name cisco.com
ip name-server 172.29.2.133
ip name-server 172.29.11.48
!
!
crypto isakmp policy 1
 authentication pre-share
 group 2
!
crypto isakmp policy 10
 hash md5
 authentication pre-share
 lifetime 200
crypto isakmp key cisco address 172.31.100.2
!
!
crypto ipsec transform-set esp-des-md5 esp-des esp-md5-hmac
!
crypto map test client accounting list ipsecaaa
crypto map test 10 ipsec-isakmp
 set peer 172.31.100.2
 set security-association lifetime seconds 120
 set transform-set esp-des-md5
 match address 101
!
voice call carrier capacity active
!
```

```
interface Loopback0
 ip address 10.20.20.20 255.255.255.0
 no ip route-cache
 no ip mroute-cache
!
interface FastEthernet0/0
 ip address 10.2.80.203 255.255.255.0
 no ip mroute-cache
 load-interval 30
 duplex full
!
interface FastEthernet1/0
 ip address 192.168.219.2 255.255.255.0
 no ip mroute-cache
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 172.28.100.1 255.255.255.0
 no ip mroute-cache
 duplex auto
 speed auto
 crypto map test
!
no fair-queue
ip default-gateway 10.2.80.1
ip classless
ip route 10.0.0.0 0.0.0.0 10.2.80.1
ip route 10.30.0.0 255.0.0.0 10.2.80.56
ip route 10.10.10.0 255.255.255.0 172.31.100.2
ip route 10.0.0.2 255.255.255.255 10.2.80.73
no ip http server
ip pim bidir-enable
!
!
ip access-list extended encrypt
 permit ip host 10.0.0.1 host 10.5.0.1
!
access-list 101 permit ip host 10.20.20.20 host 10.10.10.10
!
!
radius-server host 172.27.162.206 auth-port 1645 acct-port 1646 key cisco123
radius-server retransmit 3
radius-server authorization permit missing Service-Type
radius-server vsa send accounting
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 exec prompt timestamp
line aux 0
line vty 5 15
!
exception core-file ioscrypto/core/sheep-core
```

```
exception dump 172.25.1.129
ntp clock-period 17208229
ntp server 172.71.150.52
!
end
```

Additional References

For additional information related to IPSec VPN accounting, refer to the following references:

Related Documents

Related Topic	Document Title
Configuring AAA accounting	• The chapter "Configuring Accounting" in the *Cisco IOS Security Configuration Guide*, Release 12.2
Configuring IPSec VPN accounting	• The chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*, Release 12.2
Configuring basic AAA RADIUS	• *Configuring Basic AAA RADIUS for Dial-In Clients* • How Does RADIUS Work? • The chapter "Configuring RADIUS" in the *Cisco IOS Security Configuration Guide, Release 12.2* • The chapter "RADIUS Commands" in the *Security Command Reference*, Release 12.2 T
Configuring ISAKMP profiles	*VRF-Aware IPSec*, Cisco IOS Release 12.2(15)T feature module
Privilege levels with TACACS+ and RADIUS	How to Assign Privilege Levels with TACACS+ and RADIUS
IP security, RADIUS, and AAA commands	*Cisco IOS Security Command Reference, Release 12.2 T*

Standards

Standards	Title
None	

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs	Title
None	

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features

- **client authentication list**
- **client configuration address**
- **crypto isakmp profile**
- **crypto map (global IPSec)**
- **debug crypto isakmp**
- **isakmp authorization list**
- **match identity**
- **set isakmp-profile**
- **vrf**

For information about these commands, see the Cisco IOS Security Command Reference at

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at

http://tools.cisco.com/Support/CLILookup or the Master Command List.

Glossary

IKE—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IP security [IPSec]) that require keys. Before any IPSec traffic can be passed, each router, firewall, and host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a certification authority (CA) service.

IPSec—IP security. IPSec is A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

ISAKMP—Internet Security Association and Key Management Protocol. ISAKMP is an Internet IPSec protocol (RFC 2408) that negotiates, establishes, modifies, and deletes security associations. It also exchanges key generation and authentication data (independent of the details of any specific key generation technique), key establishment protocol, encryption algorithm, or authentication mechanism.

L2TP session—Layer 2 Transport Protocol. L2TP are communications transactions between the L2TP access concentrator (LAC) and the L2TP network server (LNS) that support tunneling of a single PPP connection. There is a one-to-one relationship among the PPP connection, L2TP session, and L2TP call.

NAS—network access server. A NAS is a Cisco platform (or collection of platforms, such as an AccessPath system) that interfaces between the packet world (for example, the Internet) and the circuit world (for example, the public switched telephone network [PSTN]).

PFS—perfect forward secrecy. PFS is a cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised because subsequent keys are not derived from previous keys.

QM—Queue Manager. The Cisco IP Queue Manager (IP QM) is an intelligent, IP-based, call-treatment and routing solution that provides powerful call-treatment options as part of the Cisco IP Contact Center (IPCC) solution.

RADIUS—Remote Authentication Dial-In User Service. RADIUS is a database for authenticating modem and ISDN connections and for tracking connection time.

RSA—Rivest, Shamir, and Adelman. Rivest, Shamir, and Adelman are the inventors of the Public-key cryptographic system that can be used for encryption and authentication.

SA—security association. A SA is an instance of security policy and keying material that is applied to a data flow.

TACACS+—Terminal Access Controller Access Control System Plus. TACACS+ is a security application that provides centralized validation of users attempting to gain access to a router or network access server.

TED—Tunnel Endpoint Discovery. TED is a Cisco IOS software feature that allows routers to discover IPSec endpoints.

VPN—Virtual Private Network. A VPN enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses "tunneling" to encrypt all information at the IP level.

VRF—A VPN routing/forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

VSA—vendor-specific attribute. A VSA is an attribute that has been implemented by a particular vendor. It uses the attribute Vendor-Specific to encapsulate the resulting AV pair: essentially, Vendor-Specific = protocol:attribute = value.

XAUTH—Extended authentication. XAUTH is an optional exchange between IKE Phase 1 and IKE Phase 2, in which the router demands additional authentication information in an attempt to authenticate the actual user (as opposed to authenticating the peer).

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

IPsec Virtual Tunnel Interface

First Published: October 18, 2004
Last Updated: June 11, 2008

IP security (IPsec) virtual tunnel interfaces (VTIs) provide a routable interface type for terminating IPsec tunnels and an easy way to define protection between sites to form an overlay network. IPsec VTIs simplify configuration of IPsec for protection of remote links, support multicast, and simplify network management and load balancing.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for IPsec Virtual Tunnel Interface" section on page 24.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for IPsec Virtual Tunnel Interface

IPsec Transform Set

The IPsec transform set must be configured in tunnel mode only.

IKE Security Association

The Internet Key Exchange (IKE) security association (SA) is bound to the VTI. Because IKE SA is bound to the VTI, the same IKE SA cannot be used for a crypto map.

IPsec SA Traffic Selectors

Static VTIs support only a single IPsec SA that is attached to the VTI interface. The traffic selector for the IPsec SA is always “IP any any.”

A dynamic VTI also is a point-point interface that supports only a single IPsec SA, but the dynamic VTI is flexible in that it can accept the IPsec selectors that are proposed by the initiator.

IPv4 and IPv6 Packets

This feature supports static VTIs that are configured to encapsulate IPv4 packets or IPv6 packets, but IPv4 packets cannot carry IPv6 packets, and IPv6 packets cannot carry IPv4 packets.

Proxy

Static VTIs support only the “IP any any” proxy.

Dynamic VTIs support only one proxy, which can be “IP any any” or any subset of it.

QoS Traffic Shaping

The shaped traffic is process switched.

Stateful Failover

IPsec stateful failover is not supported with IPsec VTIs.

Tunnel Protection

The **shared** keyword is not required and must not be configured when using the **tunnel mode ipsec ipv4** command for IPsec IPv4 mode.

Static VTIs Versus GRE Tunnels

The IPsec VTI is limited to IP unicast and multicast traffic only, as opposed to GRE tunnels, which have a wider application for IPsec implementation.

VRF-Aware IPsec Configuration

In VRF-aware IPsec configurations with either static or dynamic VTIs (DVTIs), the VRF must *not* be configured in the Internet Security Association and Key Management Protocol (ISAKMP) profile. Instead, the VRF must be configured on the tunnel interface for static VTIs. For DVTIs, you must apply VRF to the vtemplate using the **ip vrf forwarding** command.

Information About IPsec Virtual Tunnel Interface

The use of IPsec VTIs both greatly simplifies the configuration process when you need to provide protection for remote access and provides a simpler alternative to using generic routing encapsulation (GRE) or Layer 2 Tunneling Protocol (L2TP) tunnels for encapsulation and crypto maps with IPsec. A major benefit associated with IPsec VTIs is that the configuration does not require a static mapping of IPsec sessions to a physical interface. The IPsec tunnel endpoint is associated with an actual (virtual) interface. Because there is a routable interface at the tunnel endpoint, many common interface capabilities can be applied to the IPsec tunnel.

The IPsec VTI allows for the flexibility of sending and receiving both IP unicast and multicast encrypted traffic on any physical interface, such as in the case of multiple paths. Traffic is encrypted or decrypted when it is forwarded from or to the tunnel interface and is managed by the IP routing table. Using IP routing to forward the traffic to the tunnel interface simplifies the IPsec VPN configuration compared to the more complex process of using access control lists (ACLs) with the crypto map in native IPsec configurations. DVTIs function like any other real interface so that you can apply quality of service (QoS), firewall, and other security services as soon as the tunnel is active.

Without Virtual Private Network (VPN) Acceleration Module2+ (VAM2+) accelerating virtual interfaces, the packet traversing an IPsec virtual interface is directed to the router processor (RP) for encapsulation. This method tends to be slow and has limited scalability. In hardware crypto mode, all the IPsec VTIs are accelerated by the VAM2+ crypto engine, and all traffic going through the tunnel is encrypted and decrypted by the VAM2+.

The following sections provide details about the IPsec VTI:

Benefits of Using IPsec Virtual Tunnel Interfaces

IPsec VTIs allow you to configure a virtual interface to which you can apply features. Features for clear-text packets are configured on the VTI. Features for encrypted packets are applied on the physical outside interface.When IPsec VTIs are used, you can separate the application of features such as NAT, ACLs, and QoS and apply them to clear-text or encrypted text, or both. When crypto maps are used, there is no simple way to apply encryption features to the IPsec tunnel.

There are two types of VTI interfaces: static VTIs (SVTIs) and dynamic VTIs (DVTIs).

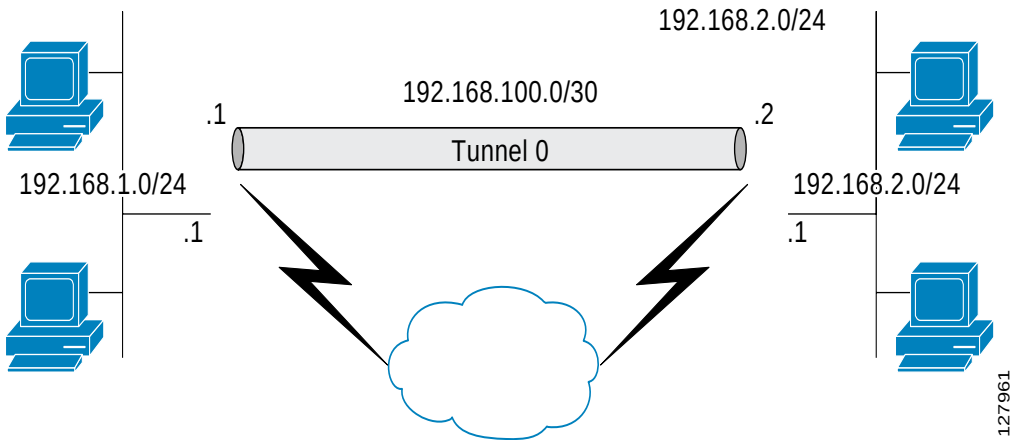
Static Virtual Tunnel Interfaces

SVTI configurations can be used for site-to-site connectivity in which a tunnel provides always-on access between two sites. The advantage of using SVTIs as opposed to crypto map configurations is that users can enable dynamic routing protocols on the tunnel interface without the extra 4 bytes required for GRE headers, thus reducing the bandwidth for sending encrypted data.

Additionally, multiple Cisco IOS software features can be configured directly on the tunnel interface and on the physical egress interface of the tunnel interface. This direct configuration allows users to have solid control on the application of the features in the pre- or post-encryption path.

Figure 1 illustrates how a static VTI is used.

Figure 1 IPsec Static VTI



The IPsec VTI supports native IPsec tunneling and exhibits most of the properties of a physical interface.

Dynamic Virtual Tunnel Interfaces

DVTIs can provide highly secure and scalable connectivity for remote-access VPNs. The DVTI technology replaces dynamic crypto maps and the dynamic hub-and-spoke method for establishing tunnels.

Dynamic VTIs can be used for both the server and remote configuration. The tunnels provide an on-demand separate virtual access interface for each VPN session. The configuration of the virtual access interfaces is cloned from a virtual template configuration, which includes the IPsec configuration and any Cisco IOS software feature configured on the virtual template interface, such as QoS, NetFlow, or ACLs.

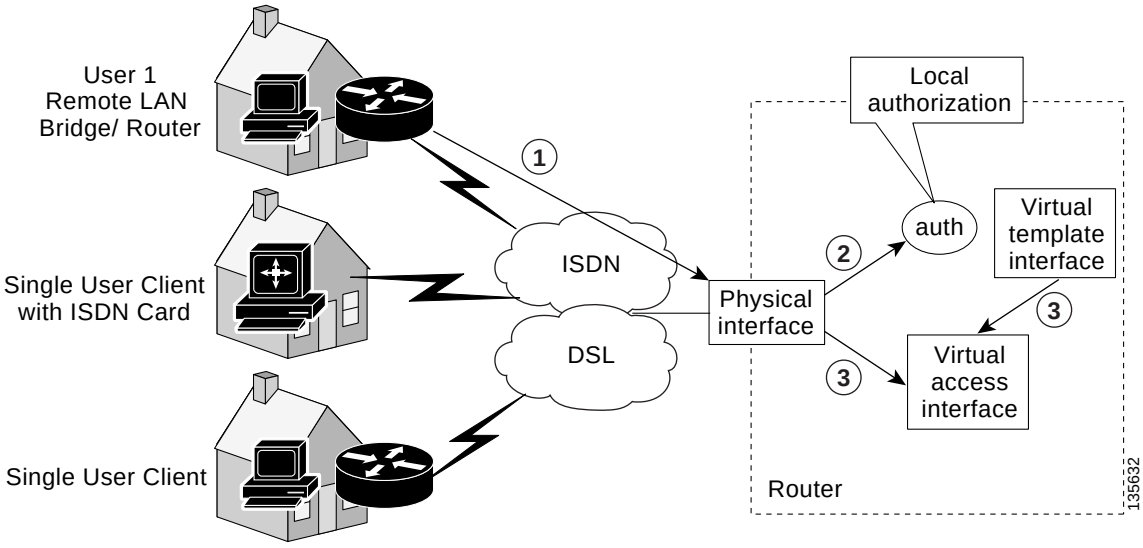
Dynamic VTIs function like any other real interface so that you can apply QoS, firewall, other security services as soon as the tunnel is active. QoS features can be used to improve the performance of various applications across the network. Any combination of QoS features offered in Cisco IOS software can be used to support voice, video, or data applications.

Dynamic VTIs provide efficiency in the use of IP addresses and provide secure connectivity. Dynamic VTIs allow dynamically downloadable per-group and per-user policies to be configured on a RADIUS server. The per-group or per-user definition can be created using extended authentication (Xauth) User or Unity group, or it can be derived from a certificate. Dynamic VTIs are standards based, so interoperability in a multiple-vendor environment is supported. IPsec DVTIs allow you to create highly secure connectivity for remote access VPNs and can be combined with Cisco Architecture for Voice, Video, and Integrated Data (AVVID) to deliver converged voice, video, and data over IP networks. The DVTI simplifies Virtual Private Network (VRF) routing and forwarding- (VRF-) aware IPsec deployment. The VRF is configured on the interface.

A DVTI requires minimal configuration on the router. A single virtual template can be configured and cloned.

The DVTI creates an interface for IPsec sessions and uses the virtual template infrastructure for dynamic instantiation and management of dynamic IPsec VTIs. The virtual template infrastructure is extended to create dynamic virtual-access tunnel interfaces. Dynamic VTIs are used in hub-and-spoke configurations. A single DVTI can support several static VTIs. Figure 2 illustrates the DVTI authentication path.

Figure 2 Dynamic IPsec VTI



The authentication shown in Figure 2 follows this path:

1. User 1 calls the router.
2. Router 1 authenticates User 1.
3. IPsec clones virtual access interface from virtual template interface.

Dynamic Virtual Tunnel Interface Life Cycle

IPsec profiles define policy for dynamic VTIs. The dynamic interface is created at the end of IKE Phase 1 and IKE Phase 1.5. The interface is deleted when the IPsec session to the peer is closed. The IPsec session is closed when both IKE and IPsec SAs to the peer are deleted.

Routing with IPsec Virtual Tunnel Interfaces

Because VTIs are routable interfaces, routing plays an important role in the encryption process. Traffic is encrypted only if it is forwarded out of the VTI, and traffic arriving on the VTI is decrypted and routed accordingly. VTIs allow you to establish an encryption tunnel using a real interface as the tunnel endpoint. You can route to the interface or apply services such as QoS, firewalls, network address translation, and Netflow statistics as you would to any other interface. You can monitor the interface, route to it, and it has an advantage over crypto maps because it is a real interface and provides the benefits of any other regular Cisco IOS interface.

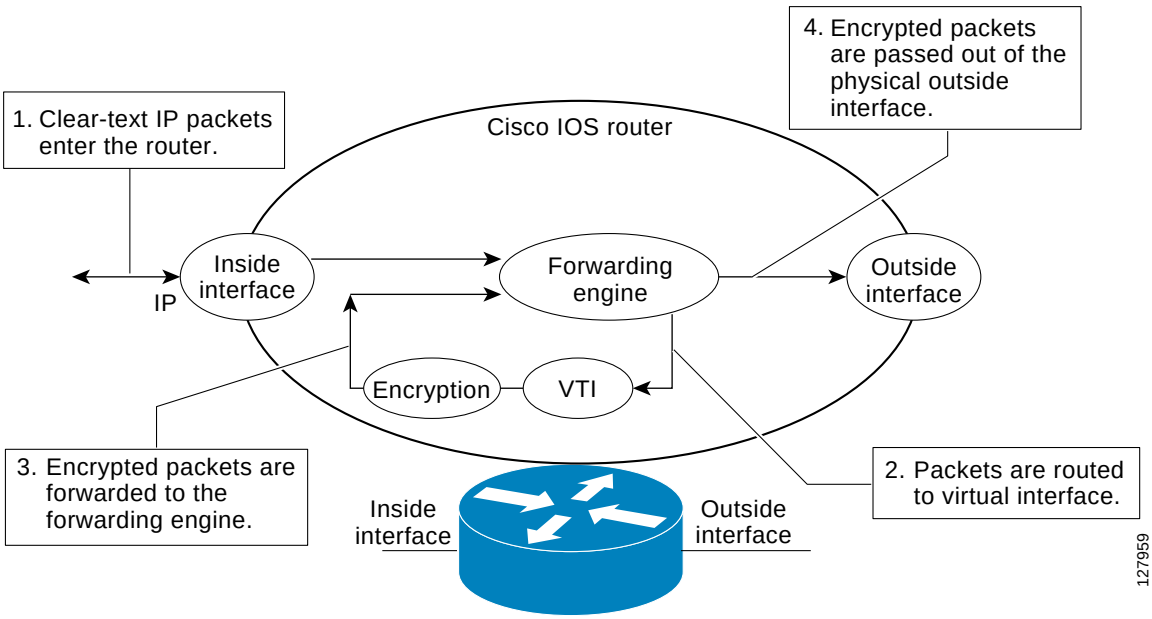
Note Dynamic routing can be used with SVTIs. Routing with DVTIs is **not** supported or recommended.

Traffic Encryption with the IPsec Virtual Tunnel Interface

When an IPsec VTI is configured, encryption occurs in the tunnel. Traffic is encrypted when it is forwarded to the tunnel interface. Traffic forwarding is handled by the IP routing table, and dynamic or static routing can be used to route traffic to the SVTI. DVTI uses reverse route injection to further simplify the routing configurations. Using IP routing to forward the traffic to encryption simplifies the IPsec VPN configuration because the use of ACLs with a crypto map in native IPsec configurations is not required. The IPsec virtual tunnel also allows you to encrypt multicast traffic with IPsec.

IPsec packet flow into the IPsec tunnel is illustrated in Figure 3.

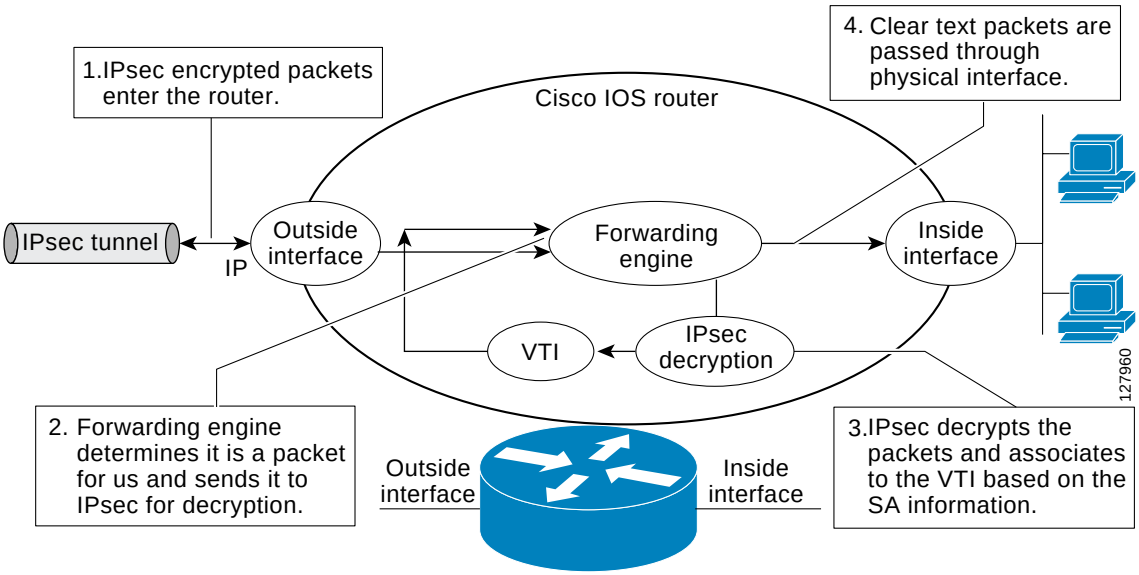
Figure 3 Packet Flow into the IPsec Tunnel



After packets arrive on the inside interface, the forwarding engine switches the packets to the VTI, where they are encrypted. The encrypted packets are handed back to the forwarding engine, where they are switched through the outside interface.

Figure 4 shows the packet flow out of the IPsec tunnel.

Figure 4 Packet Flow out of the IPsec Tunnel



How to Configure IPsec Virtual Tunnel Interface

Configuring Static IPsec Virtual Tunnel Interfaces

This configuration shows how to configure a static IPsec VTI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto IPsec profile** *profile-name*
4. **set transform-set** *transform-set-name*
5. **interface** *type number*
6. **ip address** *address mask*
7. **tunnel mode ipsec ipv4**
8. **tunnel source** *interface*
9. **tunnel destination** *ip-address*
10. **tunnel protection IPsec profile** *profile-name* [**shared**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto IPsec profile** *profile-name* **Example:** `Router(config)# crypto IPsec profile PROF`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers.
Step 4	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config)# set transform-set tset`	Specifies which transform sets can be used with the crypto map entry.
Step 5	**interface** *type number* **Example:** `Router(config)# interface tunnel0`	Specifies the interface on which the tunnel will be configured and enters interface configuration mode.
Step 6	**ip address** *address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.0`	Specifies the IP address and mask.
Step 7	**tunnel mode ipsec ipv4** **Example:** `Router(config-if)# tunnel mode ipsec ipv4`	Defines the mode for the tunnel.
Step 8	**tunnel source** *interface* **Example:** `Router(config-if)# tunnel source loopback0`	Specifies the tunnel source as a loopback interface.

	Command or Action	Purpose
Step 9	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 172.16.1.1`	Identifies the IP address of the tunnel destination.
Step 10	**tunnel protection IPsec profile** *profile-name* [**shared**] **Example:** `Router(config-if)# tunnel protection IPsec profile PROF`	Associates a tunnel interface with an IPsec profile.

Configuring Dynamic IPsec Virtual Tunnel Interfaces

This task shows how to configure a dynamic IPsec VTI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto IPsec profile** *profile-name*
4. **set transform-set** *transform-set-name*
5. **interface virtual-template** *number*
6. **tunnel mode** *mode*
7. **tunnel protection IPsec profile** *profile-name* [**shared**]
8. **exit**
9. **crypto isakamp profile** *profile-name*
10. **virtual-template** *template-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto IPsec profile** *profile-name* **Example:** `Router(config)# crypto IPsec profile PROF`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers.
Step 4	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config)# set transform-set tset`	Specifies which transform sets can be used with the crypto map entry.
Step 5	**interface virtual-template** *number* **Example:** `Router(config)# interface virtual-template 2`	Defines a virtual-template tunnel interface and enters interface configuration mode.
Step 6	**tunnel mode ipsec ipv4** **Example:** `Router(config-if)# tunnel mode ipsec ipv4`	Defines the mode for the tunnel.
Step 7	**tunnel protection IPsec profile** *profile-name* [**shared**] **Example:** `Router(config-if)# tunnel protection IPsec profile PROF`	Associates a tunnel interface with an IPsec profile.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 9	**crypto isakamp profile** *profile-name* **Example:** `Router(config)# crypto isakamp profile red`	Defines the ISAKAMP profile to be used for the virtual template.
Step 10	**virtual-template** *template-number* **Example:** `Router(config)# virtual-template 1`	Specifies the virtual template attached to the ISAKAMP profile.

Configuration Examples for IPsec Virtual Tunnel Interface

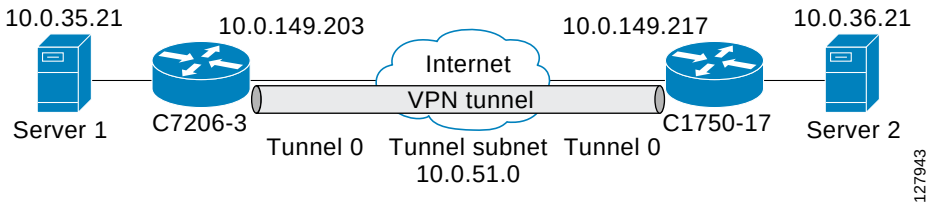
The following examples are provided to illustrate configuration scenarios for IPsec VTIs:

- Static Virtual Tunnel Interface with IPsec: Example, page 11
- VRF-Aware Static Virtual Tunnel Interface: Example, page 14
- Static Virtual Tunnel Interface with QoS: Example, page 14
- Static Virtual Tunnel Interface with Virtual Firewall: Example, page 15

Static Virtual Tunnel Interface with IPsec: Example

The following example configuration uses a preshared key for authentication between peers. VPN traffic is forwarded to the IPsec VTI for encryption and then sent out the physical interface. The tunnel on subnet 10 checks packets for IPsec policy and passes them to the Crypto Engine (CE) for IPsec encapsulation. Figure 5 illustrates the IPsec VTI configuration.

Figure 5 VTI with IPsec



C7206 Router Configuration

```
version 12.3

service timestamps debug datetime
service timestamps log datetime
hostname 7200-3
no aaa new-model
ip subnet-zero
ip cef
controller ISA 6/1
!
crypto isakmp policy 1
encr 3des
authentication pre-share
group 2
crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto IPsec transform-set T1 esp-3des esp-sha-hmac
crypto IPsec profile P1
set transform-set T1
!


interface Tunnel0
 ip address 10.0.51.203 255.255.255.0
 ip ospf mtu-ignore
 load-interval 30
 tunnel source 10.0.149.203
 tunnel destination 10.0.149.217
 tunnel mode IPsec ipv4
 tunnel protection IPsec profile P1
```

```
!
interface Ethernet3/0
 ip address 10.0.149.203 255.255.255.0
 duplex full
!
interface Ethernet3/3
 ip address 10.0.35.203 255.255.255.0
 duplex full
!
ip classless
ip route 10.0.36.0 255.255.255.0 Tunnel0
line con 0
line aux 0
line vty 0 4
end
```

C1750 Router Configuration

```
version 12.3

hostname c1750-17
no aaa new-model
ip subnet-zero
ip cef
crypto isakmp policy 1
encr 3des
authentication pre-share
group 2

crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto IPsec transform-set T1 esp-3des esp-sha-hmac
crypto IPsec profile P1
set transform-set T1
!
interface Tunnel0
 ip address 10.0.51.217 255.255.255.0
 ip ospf mtu-ignore
 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
!
interface FastEthernet0/0
 ip address 10.0.149.217 255.255.255.0
 speed 100
 full-duplex
!
interface Ethernet1/0
 ip address 10.0.36.217 255.255.255.0
 load-interval 30
 full-duplex
!


ip classless
ip route 10.0.35.0 255.255.255.0 Tunnel0
line con 0
line aux 0
line vty 0 4
end
```

Verifying the Results for the IPsec Static Virtual Tunnel Interface: Example

This section provides information that you can use to confirm that your configuration is working properly. In this display, Tunnel 0 is "up," and the line protocol is "up." If the line protocol is "down," the session is not active.

Verifying the C7206 Status

```
Router# show interface tunnel 0

Tunnel0 is up, line protocol is up
Hardware is Tunnel
Internet address is 10.0.51.203/24
MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
reliability 255/255, txload 103/255, rxload 110/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.0.149.203, destination 10.0.149.217
Tunnel protocol/transport IPsec/IP, key disabled, sequencing disabled
Tunnel TTL 255

Checksumming of packets disabled, fast tunneling enabled
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Tunnel protection via IPsec (profile "P1")
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 1/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
30 second input rate 13000 bits/sec, 34 packets/sec
30 second output rate 36000 bits/sec, 34 packets/sec
191320 packets input, 30129126 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
59968 packets output, 15369696 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out

Router# show crypto session

Crypto session current status
Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.0.149.217 port 500
IKE SA: local 10.0.149.203/500 remote 10.0.149.217/500 Active
IPsec FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
Active SAs: 4, origin: crypto map

Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C 10.0.35.0/24 is directly connected, Ethernet3/3
S 10.0.36.0/24 is directly connected, Tunnel0
```

```
C 10.0.51.0/24 is directly connected, Tunnel0
C 10.0.149.0/24 is directly connected, Ethernet3/0
```

VRF-Aware Static Virtual Tunnel Interface: Example

To add VRF to the static VTI example, include the **ipvrf** and **ip vrf forwarding** commands to the configuration as shown in the following example.

C7206 Router Configuration

```
hostname c7206
.
.
ip vrf sample-vti1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
.
.
interface Tunnel0
 ip vrf forwarding sample-vti1
 ip address 10.0.51.217 255.255.255.0
 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
.
.
!
end
```

Static Virtual Tunnel Interface with QoS: Example

You can apply any QoS policy to the tunnel endpoint by including the **service-policy** statement under the tunnel interface. The following example is policing traffic out the tunnel interface.

C7206 Router Configuration

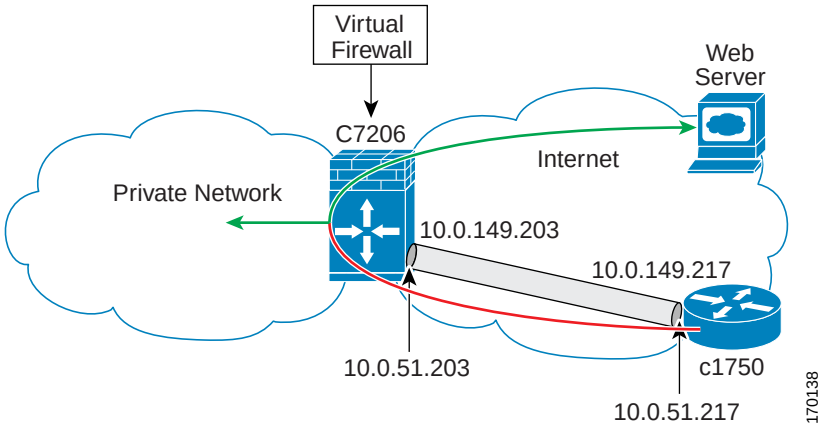
```
hostname c7206
.
.
class-map match-all VTI
 match any
!
policy-map VTI
  class VTI
  police cir 2000000
    conform-action transmit
    exceed-action drop
!
.
.
interface Tunnel0
 ip address 10.0.51.217 255.255.255.0
 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
```

```
 service-policy output VTI
!
.
.
!
end
```

Static Virtual Tunnel Interface with Virtual Firewall: Example

Applying th e virtual firewall to the static VTI tunnel allows traffic from the spoke to pass through the hub to reach the internet. Figure 6 illustrates a static VTI with the spoke protected inherently by the corporate firewall.

Figure 6 Static VTI with Virtual Firewall



The basic static VTI configuration has been modified to include the virtual firewall definition.

C7206 Router Configuration

```
hostname c7206
.
.
ip inspect max-incomplete high 1000000
ip inspect max-incomplete low 800000
ip inspect one-minute high 1000000
ip inspect one-minute low 800000
ip inspect tcp synwait-time 60
ip inspect tcp max-incomplete host 100000 block-time 2
ip inspect name IOSFW1 tcp timeout 300
ip inspect name IOSFW1 udp
!
.
.
interface GigabitEthernet0/1
 description Internet Connection
 ip address 172.18.143.246 255.255.255.0
 ip access-group 100 in
 ip nat outside
!
interface Tunnel0
 ip address 10.0.51.217 255.255.255.0
 ip nat inside
```

```
 ip inspect IOSFW1 in
 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
ip nat translation timeout 120
ip nat translation finrst-timeout 2
ip nat translation max-entries 300000
ip nat pool test1 10.2.100.1 10.2.100.50 netmask 255.255.255.0
ip nat inside source list 110 pool test1 vrf test-vti1 overload
!
access-list 100 permit esp any any
access-list 100 permit udp any eq isakmp any
access-list 100 permit udp any eq non500-isakmp any
access-list 100 permit icmp any any
access-list 110 deny   esp any any
access-list 110 deny   udp any eq isakmp any
access-list 110 permit ip any any
access-list 110 deny   udp any eq non500-isakmp any
!
end
```

Dynamic Virtual Tunnel Interface Easy VPN Server: Example

The following example illustrates the use of the DVTI Easy VPN server, which serves as an IPsec remote access aggregator. The client can be a home user running a Cisco VPN client or it can be a Cisco IOS router configured as an Easy VPN client.

C7206 Router Configuration

```
hostname c7206
!
aaa new-model
aaa authentication login local_list local
aaa authorization network local_list local
aaa session-id common
!
ip subnet-zero
ip cef
!
username cisco password 0 cisco123
!
controller ISA 1/1
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp client configuration group group1
 key cisco123
 pool group1pool
 save-password
!
crypto isakmp profile vpn1-ra
   match identity group group1
   client authentication list local_list
```

```
   isakmp authorization list local_list
   client configuration address respond
   virtual-template 1
!
crypto ipsec transform-set VTI-TS esp-3des esp-sha-hmac
!
crypto ipsec profile test-vti1
 set transform-set VTI-TS
!
interface GigabitEthernet0/1
 description Internet Connection
 ip address 172.18.143.246 255.255.255.0
!
interface GigabitEthernet0/2
 description Internal Network
 ip address 10.2.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered GigabitEthernet0/1
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
!
ip local pool group1pool 192.168.1.1 192.168.1.4
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
end
```

Verifying the Results for the Dynamic Virtual Tunnel Interface Easy VPN Server: Example

The following examples show that a dynamic VTI has been configured for an Easy VPN server.

```
Router# show running-config interface Virtual-Access2

Building configuration...

Current configuration : 250 bytes
!
interface Virtual-Access2
 ip unnumbered GigabitEthernet0/1
 ip virtual-reassembly
 tunnel source 172.18.143.246
 tunnel destination 172.18.143.208
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
 no tunnel protection ipsec initiate
end

Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.2.1.10 to network 0.0.0.0

     172.18.0.0/24 is subnetted, 1 subnets
C       172.18.143.0 is directly connected, GigabitEthernet0/1
```

```
     192.168.1.0/32 is subnetted, 1 subnets
S       192.168.1.1 [1/0] via 0.0.0.0, Virtual-Access2
     10.0.0.0/24 is subnetted, 1 subnets
C       10.2.1.0 is directly connected, GigabitEthernet0/2
S*   0.0.0.0/0 [1/0] via 172.18.143.1
```

Dynamic Virtual Tunnel Interface Easy VPN Client: Example

The following example shows how you can set up a router as the Easy VPN client. This example uses basically the same idea as the Easy VPN client that you can run from a PC to connect. In fact, the configuration of the Easy VPN server will work for the software client or the Cisco IOS client.

```
hostname c1841
!
no aaa new-model
!
ip cef
!
username cisco password 0 cisco123
!
crypto ipsec client ezvpn CLIENT
 connect manual
 group group1 key cisco123
 mode client
 peer 172.18.143.246
 virtual-interface 1
 username cisco password cisco123
 xauth userid mode local
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
interface FastEthernet0/0
 description Internet Connection
 ip address 172.18.143.208 255.255.255.0
 crypto ipsec client ezvpn CLIENT
!
interface FastEthernet0/1
 ip address 10.1.1.252 255.255.255.0
 crypto ipsec client ezvpn CLIENT inside
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
!
ip route 0.0.0.0 0.0.0.0 172.18.143.1 254
!
end
```

The client definition can be set up in many different ways. The mode specified with the **connect** command can be automatic or manual. If the connect mode is set to manual, the IPsec tunnel has to be initiated manually by a user.

Also note use of the **mode** command. The mode can be client, network-extension, or network-extension-plus. This example indicates client mode, which means that the client is given a private address from the server. Network-extension mode is different from client mode in that the client specifies for the server its attached private subnet. Depending on the mode, the routing table on either end will be slightly different. The basic operation of the IPSec tunnel remains the same, regardless of the specified mode.

Verifying the Results for the Dynamic Virtual Tunnel Interface Easy VPN Client: Example

The following examples illustrate different ways to display the status of the DVTI.

```
Router# show running-config interface Virtual-Access2

Building configuration...

Current configuration : 148 bytes
!
interface Virtual-Access2
 ip unnumbered Loopback1
 tunnel source FastEthernet0/0
 tunnel destination 172.18.143.246
 tunnel mode ipsec ipv4
end

Router# show running-config interface Loopback1

Building configuration...

Current configuration : 65 bytes
!
interface Loopback1
 ip address 192.168.1.1 255.255.255.255
end

Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 172.18.143.1 to network 0.0.0.0

     10.0.0.0/32 is subnetted, 1 subnets
C       10.1.1.1 is directly connected, Loopback0
     172.18.0.0/24 is subnetted, 1 subnets
C       172.18.143.0 is directly connected, FastEthernet0/0
     192.168.1.0/32 is subnetted, 1 subnets
C       192.168.1.1 is directly connected, Loopback1
S*   0.0.0.0/0 [1/0] via 0.0.0.0, Virtual-Access2

Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 6

Tunnel name : CLIENT
Inside interface list: FastEthernet0/1
Outside interface: Virtual-Access2 (bound to FastEthernet0/0)
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 192.168.1.1
Mask: 255.255.255.255
Save Password: Allowed
Current EzVPN Peer: 172.18.143.246
```

VRF-Aware IPsec with Dynamic VTI: Example

This example shows how to configure VRF-Aware IPsec to take advantage of the dynamic VTI:

```
hostname c7206
.
.
ip vrf test-vti1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
.
.
interface Virtual-Template1 type tunnel
 ip vrf forwarding test-vti1
 ip unnumbered Loopback0
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
!
.
.
end
```

Dynamic Virtual Tunnel Interface with Virtual Firewall: Example

The DVTI Easy VPN server can be configured behind a virtual firewall. Behind-the-firewall configuration allows users to enter the network, while the network firewall is protected from unauthorized access. The virtual firewall uses Context-Based Access Control (CBAC) and NAT applied to the Internet interface as well as to the virtual template.

```
hostname c7206
.
.
ip inspect max-incomplete high 1000000
ip inspect max-incomplete low 800000
ip inspect one-minute high 1000000
ip inspect one-minute low 800000
ip inspect tcp synwait-time 60
ip inspect tcp max-incomplete host 100000 block-time 2
ip inspect name IOSFW1 tcp timeout 300
ip inspect name IOSFW1 udp
!
.
.
interface GigabitEthernet0/1
 description Internet Connection
 ip address 172.18.143.246 255.255.255.0
 ip access-group 100 in
 ip nat outside
!
interface GigabitEthernet0/2
 description Internal Network
 ip address 10.2.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
 ip nat inside
 ip inspect IOSFW1 in
```

```
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
ip nat translation timeout 120
ip nat translation finrst-timeout 2
ip nat translation max-entries 300000
ip nat pool test1 10.2.100.1 10.2.100.50 netmask 255.255.255.0
ip nat inside source list 110 pool test1 vrf test-vti1 overload
!
access-list 100 permit esp any any
access-list 100 permit udp any eq isakmp any
access-list 100 permit udp any eq non500-isakmp any
access-list 100 permit icmp any any
access-list 110 deny   esp any any
access-list 110 deny   udp any eq isakmp any
access-list 110 permit ip any any
access-list 110 deny   udp any eq non500-isakmp any
!
end
```

Dynamic Virtual Tunnel Interface with QoS: Example

You can add QoS to the DVTI tunnel by applying the service policy to the virtual template. When the template is cloned to make the virtual-access interface, the service policy will be applied there. The following example shows the basic DVTI configuration with QoS added.

```
hostname c7206
.
.
class-map match-all VTI
 match any
!
policy-map VTI
  class VTI
  police cir 2000000
    conform-action transmit
    exceed-action drop
!
.
.
interface Virtual-Template1 type tunnel
 ip vrf forwarding test-vti1
 ip unnumbered Loopback0
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
 service-policy output VTI
!
.
.
!
end
```

Additional References

The following sections provide references related to IPsec virtual tunnel interface.

Related Documents

Related Topic	Document Title
IPsec, security issues	*Cisco IOS Security Configuration Guide*, Release 12.4
QoS, configuring	• Quality of Service (QoS) Support for Enhanced Easy VPN • *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.4T
Security commands	*Cisco IOS Security Command Reference*, Release 12.4T
VPN configuration	• *Cisco Easy VPN Remote* • *Easy VPN Server*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 2408	*Internet Security Association and Key Management Protocol*
RFC 2409	*The Internet Key Exchange (IKE)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features

- **crypto isakmp profile**
- **interface virtual-template**
- **show vtemplate**
- **tunnel mode**
- **virtual-template**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Feature Information for IPsec Virtual Tunnel Interface

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click Cancel at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for IPsec Virtual Tunnel Interfacel

Feature Name	Releases	Feature Configuration Information
Static IPsec VTIs	12.3(7)T 12.3(14)T 12.2(33)SRA 12.2(33)SXH	IPsec VTIs (VTIs) provide a routable interface type for terminating IPsec tunnels and an easy way to define protection between sites to form an overlay network. IPsec VTIs simplify configuration of IPsec for protection of remote links, support multicast, and simplify network management and load balancing.
Dynamic IPsec VTIs	12.3(7)T 12.3(14)T	Dynamic VTIs provide efficiency in the use of IP addresses and provide secure connectivity. Dynamic VTIs allow dynamically downloadable per-group and per-user policies to be configured on a RADIUS server. The per-group or per-user definition can be created using Xauth User or Unity group, or it can be derived from a certificate. Dynamic VTIs are standards based, so interoperability in a multiple-vendor environment is supported. IPsec dynamic VTIs allow you to create highly secure connectivity for remote access VPNs and can be combined with Cisco Architecture for Voice, Video, and Integrated Data (AVVID) to deliver converged voice, video, and data over IP networks. The dynamic VTI simplifies VRF-aware IPsec deployment. The VRF is configured on the interface.
IPSec Virtual Tunnel Interface	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Low Latency Queueing (LLQ) for IPSec Encryption Engines

Feature History

Release	Modification
12.2(13)T	This feature was introduced.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.

This feature module describes the Low Latency Queueing (LLQ) for IPSec encryption engines feature in Cisco IOS Release 12.2(13)T and 12.2(14)S. It includes the following sections:

Feature Overview

Low Latency Queueing (LLQ) for IPSec encryption engines helps reduce packet latency by introducing the concept of queueing before crypto engines. Prior to this, the crypto processing engine gave data traffic and voice traffic equal status. Administrators now designate voice traffic as priority. Data packets arriving at a router interface are directed into a data packet inbound queue for crypto engine processing. This queue is called the best effort queue. Voice packets arriving on a router interface are directed into

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

a priority packet inbound queue for crypto engine processing. This queue is called the priority queue. The crypto engine undertakes packet processing in a favorable ratio for voice packets. Voice packets are guaranteed a minimum processing bandwidth on the crypto engine.

Benefits

The Low Latency Queueing (LLQ) for IPSec encryption engines feature guarantees a certain level of crypto engine processing time for priority designated traffic.

Note On the Cisco 2600 platform, with the exception of the Cisco 2691 router, the CPU utilization maximizes out before the crypto engine becomes congested, so latency is not improved.

Better Voice Performance

Voice packets can be identified as priority, allowing the crypto engine to guarantee a certain percentage of processing bandwidth. This feature impacts the end user experience by assuring voice quality if voice traffic is directed onto a congested network.

Improved Latency and Jitters

Predictability is a critical component of network performance. The Low Latency Queueing (LLQ) for IPSec encryption engines feature delivers network traffic predictability relating to VPN. With this feature disabled, an end user employing an IP phone over VPN might experience jitter or latency, both symptoms of overall network latency and congestion. With this feature enabled, these undesirable characteristics are dissipated.

Restrictions

- No per-tunnel QoS policy. An interface QoS policy represents all tunnels.
- Assume the same IP precedence/DSCP marking for inbound and outbound voice packets.
- Assume the IP precedence/DSCP marking for voice packets are done at the source.
- Limited match criteria for voice traffic in the interface QoS policy.
- Assume call admission control is enforced within the enterprise.
- No strict error checking when aggregate policy's bandwidth exceeds crypto engine bandwidth. Only a warning is displayed but configuration is allowed.
- Assume voice packets are either all encrypted or unencrypted.

Related Features and Technologies

- CBWFQ
- Priority Queueing
- Weighted Fair Queueing

Related Documents

- *Quality of Service Solutions Command Reference*, Cisco IOS Release 12.2
- *Class-Based Weighted Fair Queueing* feature module, Cisco IOS Release 12.1
- *IP RTP Priority* feature module, Cisco IOS Release 12.0

Supported Platforms

12.2(14)S and higher

The LLQ for IPSec encryption engines feature is supported on the following platform:

- Cisco 7200 series

12.2(13)T

The LLQ for IPSec encryption engines feature is supported on all platforms using Cisco IOS Release 12.2(13)T or later, including:

- Cisco 2600 series
- Cisco 3600 series
- Cisco 7100 series
- Cisco 7200 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side-by-side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

- No new or modified standards are supported by this feature.

MIBs

- No new or modified standards are supported by this feature.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

- No new or modified RFCs are supported by this feature.

Prerequisites

To use this feature, you should be familiar with the following:

- Access control lists
- Bandwidth management
- CBWFQ

Configuration Tasks

To configure LLQ for IPSec encryption engines, perform the tasks described in the following section.

Note See the *Quality of Service Solutions Command Reference*, Cisco IOS Release 12.2, to learn more about configuring server policies on interfaces.

- Defining Class Maps (required)
- Configuring Class Policy in the Policy Map (required)
- Configuring Class Policy for a Priority Queue (required)
- Configuring Class Policy Using a Specified Bandwidth (optional)
- Configuring the Class-Default Class Policy (optional)
- Attaching the Service Policy (required)
- Verifying Configuration of Policy Maps and Their Classes (optional)

Defining Class Maps

To create a class map containing match criteria against which a packet is checked to determine if it belongs to a class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# class-map class-map-name`	Specifies the name of the class map to be created.
Step 2	`Router(config-cmap)# match access-group {access-group | name access-group-name}` or `Router(config-cmap)# match input-interface interface-name` or `Router(config-cmap)# match protocol protocol`	Specifies the name of the access control list (ACL) against whose contents packets are checked to determine if they belong to the class. Specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class. Specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class.

Configuring Class Policy in the Policy Map

To configure a policy map and create class policies that make up the service policy, begin with the **policy-map** command to specify the policy map name. Then use one or more of the following commands to configure the policy for a standard class or the default class:

- **priority**
- **bandwidth**
- **queue-limit** or **random-detect**
- **fair-queue** (for class-default class only)

For each class that you define, you can use one or more of the commands listed to configure the class policy. For example, you might specify bandwidth for one class and both bandwidth and queue limit for another class.

The default class of the policy map (commonly known as the class-default class) is the class to which traffic is directed if that traffic does not satisfy the match criteria of the other classes defined in the policy map.

You can configure class policies for as many classes as are defined on the router, up to the maximum of 64. However, the total amount of bandwidth allocated for all classes in a policy map must not exceed the minimum committed information rate (CIR) configured for the virtual circuit (VC) minus any bandwidth reserved by the **frame-relay voice bandwidth** and **frame-relay ip rtp priority** commands. If the minimum CIR is not configured, the bandwidth defaults to one half of the CIR. If all of the bandwidth is not allocated, the remaining bandwidth is allocated proportionally among the classes on the basis of their configured bandwidth.

To configure class policies in a policy map, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional.

Configuring Class Policy for a Priority Queue

To configure a policy map and give priority to a class within the policy map, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-cmap)# class class-name`	Specifies the name of a class to be created and included in the service policy.
Step 3	`Router(config-pmap-c)# priority bandwidth-kbps`	Creates a strict priority class and specifies the amount of bandwidth, in kbps, to be assigned to the class.

Configuring Class Policy Using a Specified Bandwidth

To configure a policy map and create class policies that make up the service policy, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-cmap)# class class-name`	Specifies the name of a class to be created and included in the service policy.
Step 3	`Router(config-pmap-c)# bandwidth bandwidth-kbps`	Specifies the amount of bandwidth to be assigned to the class, in kbps, or as a percentage of the available bandwidth. Bandwidth must be specified in kbps or as a percentage consistently across classes. (Bandwidth of the priority queue must be specified in kbps.)

To configure more than one class in the same policy map, repeat Step 2 and Step 3.

Configuring the Class-Default Class Policy

The class-default class is used to classify traffic that does not fall into one of the defined classes. Even though the class-default class is predefined when you create the policy map, you still have to configure it. If a default class is not configured, then traffic that does not match any of the configured classes is given best-effort treatment, which means that the network will deliver the traffic if it can, without any assurance of reliability, delay prevention, or throughput.

To configure a policy map and the class-default class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-cmap)# class class-default default-class-name`	Specifies the default class so that you can configure or modify its policy.
Step 3	`Router(config-pmap-c)# bandwidth bandwidth-kbps` `or` `Router(config-pmap-c)# fair-queue [number-of-dynamic-queues]`	Specifies the amount of bandwidth, in kbps, to be assigned to the class. Specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class. The number of dynamic queues is derived from the bandwidth of the interface.

Attaching the Service Policy

To attach a service policy to the output interface and enable LLQ for IPSec encryption engines, use the following command in map-class configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface using the LLQ for IPSec encryption engines.
Step 2	`Router(config-if)# service-policy output policy-map`	Attaches the specified service policy map to the output interface and enables LLQ for IPSec encryption engines.

Verifying Configuration of Policy Maps and Their Classes

To display the contents of a specific policy map or all policy maps configured on an interface, use the following commands in EXEC mode, as needed:

	Command	Purpose
Step 1	`Router# show frame-relay pvc dlci`	Displays statistics about the PVC and the configuration of classes for the policy map on the specified data-link connection identifier (DLCI).

	Command	Purpose
Step 2	`Router# show policy-map interface interface-name`	When LLQ is configured, displays the configuration of classes for all policy maps.
Step 3	`Router# show policy-map interface interface-name dlci dlci`	When LLQ is configured, displays the configuration of classes for the policy map on the specified DLCI.

Monitoring and Maintaining LLQ for IPSec Encryption Engines

To monitor and maintain LLQ for IPSec encryption engines, use the following command in EXEC mode:

	Command	Purpose
Step 1	`Router# show crypto eng qos`	Displays quality of service queueing statistics for LLQ for IPSec encryption engines.

For a more detailed list of commands that can be used to monitor LLQ for IPSec encryption engines, see the section "Verifying Configuration of Policy Maps and Their Classes"

Configuration Examples

This section provides the following configuration example:

- LLQ for IPSec Encryption Engines Example

LLQ for IPSec Encryption Engines Example

In the following example, a strict priority queue with a guaranteed allowed bandwidth of 50 kbps is reserved for traffic that is sent from the source address 10.10.10.10 to the destination address 10.10.10.20, in the range of ports 16384 through 20000 and 53000 through 56000.

First, the following commands configure access list 102 to match the desired voice traffic:

```
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 16384
20000
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000
56000
```

Next, the class map voice is defined, and the policy map called policy1 is created; a strict priority queue for the class voice is reserved, a bandwidth of 20 kbps is configured for the class bar, and the default class is configured for WFQ. The service-policy command then attaches the policy map to the fas0/0.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth 20
Router(config-pmap)# class class-default
Router(config-pmap-c)# fair-queue
Router(config)# interface fas0/0
```

```
Router(config-if)# service-policy output policy1
```

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **show crypto eng qos**

For information about these commands, see the Cisco IOS Security Command Reference at

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at

http://tools.cisco.com/Support/CLILookup or the Master Command List.

Glossary

IKE—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPSec). Before any IPSec traffic can be passed, each router/firewall/host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

IPSec—IP Security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

Pre-Fragmentation for IPSec VPNs

Feature History

Release	Modification
12.1(11b)E	This feature was introduced.
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This feature module describes the Pre-fragmentation for IPSec VPNs feature in Cisco IOS Release 12.2(13)T and 12.2(14)S. It includes the following sections:

Feature Overview

When a packet is nearly the size of the maximum transmission unit (MTU) of the outbound link of the encrypting router, and it is encapsulated with IPSec headers, it is likely to exceed the MTU of the outbound link. This causes packet fragmentation after encryption, which makes the decrypting router reassemble in the process path. Pre-fragmentation for IPSec VPNs increases the decrypting router's performance by enabling it to operate in the high performance CEF path instead of the process path.

This feature allows an encrypting router to predetermine the encapsulated packet size from information available in transform sets, which are configured as part of the IPSec security association (SA). If it is predetermined that the packet will exceed the MTU of the output interface, the packet is fragmented before encryption. This function avoids process level reassembly before decryption and helps improve decryption performance and overall IPSec traffic throughput.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Note The pre-fragmentation feature is turned off by default for tunnel interfaces. To receive pre-fragmentation performance benefits, turn pre-fragmentation on after insuring that the tunnel interfaces have the same MTU on both ends.

Benefits

Increased Performance

Delivers encryption throughput at maximum encryption hardware accelerator speeds. This performance increase is for near MTU-sized packets.

Uniform Fragmentation

Packets are fragmented into equally sized units to prevent further downstream fragmentation.

Interoperability

This feature is interoperable with all Cisco IOS platforms and a number of Cisco VPN clients.

Restrictions

Take the following information into consideration before this feature is configured:

- Pre-fragmentation for IPSec VPNs operates in IPSec tunnel mode and IPSec tunnel mode with GRE, but not with IPSec transport mode.
- Pre-fragmentation for IPSec VPNs configured on the decrypting router in a unidirectional traffic scenario does not improve the performance or change the behavior of either of the peers.
- Pre-fragmentation for IPSec VPNs occurs before the transform is applied if compression is turned on for outgoing packets.
- Pre-fragmentation for IPSec VPNs functionality depends on the egress interface **crypto ipsec df-bit** configuration and the incoming packet “do not fragment” (DF) bit state. See Table 1.

Table 1 Pre-Fragmentation for IPSec VPNs Dependencies

Pre-Fragmentation for IPSec VPNs Feature State (Enabled/Disabled)	Egress Interface “crypto ipsec df-bit” Configuration	Incoming Packet DF Bit State	Result
Enabled	crypto ipsec df-bit clear	0	Fragmentation occurs before encryption.
Enabled	crypto ipsec df-bit clear	1	Fragmentation occurs before encryption.
Disabled	crypto ipsec df-bit clear	0	Fragmentation occurs after encryption and packets are reassembled before decryption.
Disabled	crypto ipsec df-bit clear	1	Fragmentation occurs after encryption and packets are reassembled before decryption.
Enabled	crypto ipsec df-bit set	0	Fragmentation occurs before encryption.

Table 1 Pre-Fragmentation for IPSec VPNs Dependencies (continued)

Pre-Fragmentation for IPSec VPNs Feature State (Enabled/Disabled)	Egress Interface "crypto ipsec df-bit" Configuration	Incoming Packet DF Bit State	Result
Enabled	crypto ipsec df-bit set	1	Packets are dropped.
Disabled	crypto ipsec df-bit set	0	Fragmentation occurs after encryption and packets are reassembled before decryption.
Disabled	crypto ipsec df-bit set	1	Packets are dropped.
Enabled	crypto ipsec df-bit copy	0	Fragmentation occurs before encryption.
Enabled	crypto ipsec df-bit copy	1	Packets are dropped.
Disabled	crypto ipsec df-bit copy	0	Fragmentation occurs after encryption and packets are reassembled before decryption.
Disabled	crypto ipsec df-bit copy	1	Packets are dropped.

Supported Platforms

12.2(14)S and higher

The Pre-fragmentation for IPSec VPN feature is supported on the following platforms:

- Cisco 7200 series
- Cisco 7400 series

12.2(13)T

The Pre-fragmentation for IPSec VPN feature is supported on all platforms using Cisco IOS Release 12.2(13)T or higher, including:

- Cisco 1710
- Cisco 1720
- Cisco 1721
- Cisco 1751
- Cisco 1760
- Cisco 2600
- Cisco 2691
- Cisco 3620
- Cisco 3640
- Cisco 3660

- Cisco 3725
- Cisco 3745
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7400 series

12.1(11b)E

The Pre-fragmentation for IPSec VPN feature is supported on all platforms using Cisco IOS Release 12.1(11b)E or higher, including:

- Cisco 7100 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

- No new or modified standards are supported by this feature.

MIBs

- No new or modified standards are supported by this feature.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

- No new or modified RFCs are supported by this feature.

Configuration Tasks

See the following sections for configuration tasks for the Pre-fragmentation for IPSec VPNs feature. Each task in the list is identified as either required or optional.

- Configuring Pre-Fragmentation For IPSec VPNs (required)
- Verifying Pre-Fragmentation For IPSec VPNs (optional)

Configuring Pre-Fragmentation For IPSec VPNs

Pre-fragmentation for IPSec VPNs is globally enabled by default. To enable or disable pre-fragmentation for IPSec VPNs while in interface configuration mode, enter the commands in the following table. Use the **no** form of the commands to revert back to the default configuration, or use the commands themselves to enable configuration of the pre-fragmentation IPSec VPNs.

Note Manually enabling or disabling this feature will override the global configuration.

Command	Purpose
`Router(config-if)# crypto ipsec fragmentation before-encryption`	Enables pre-fragmentation for IPSec VPNs on the interface.
`Router(config-if)# crypto ipsec fragmentation after-encryption`	Disables pre-fragmentation for IPSec VPNs on the interface.
`Router(config)# crypto ipsec fragmentation before-encryption`	Enables pre-fragmentation for IPSec VPNs globally.
`Router(config)# crypto ipsec fragmentation after-encryption`	Disables pre-fragmentation for IPSec VPNs globally.

Verifying Pre-Fragmentation For IPSec VPNs

To verify that this feature is enabled, consult the interface statistics on the encrypting router and the decrypting router. If fragmentation occurs on the encrypting router, and no reassembly occurs on the decrypting router, fragmentation is happening before encryption, and thus the packets are not being reassembled before decryption. This means that the feature is enabled.

Note This method of verification does not apply to packets destined for the decrypting router.

Step 1 Enter the **show running-configuration** command on the encrypting router. If the feature is enabled, you will observe output similar to the following:

```
Router# show running-configuration
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key abcd123 address 25.0.0.7
!
!
crypto ipsec transform-set fooprime esp-3des esp-sha-hmac
!
crypto map bar 10 ipsec-isakmp
 set peer 25.0.0.7
 set transform-set fooprime
 match address 102
```

If the feature has been disabled, you will observe output similar to the following:

```
Router# show running-configuration
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key abcd123 address 25.0.0.7
!
!
crypto ipsec transform-set fooprime esp-3des esp-sha-hmac
crypto ipsec fragmentation after-encryption
!
crypto map bar 10 ipsec-isakmp
 set peer 25.0.0.7
 set transform-set fooprime
 match address 102
```

Step 2 Enter the **show running-configuration interface** *type number* command to display statistics for the encrypting router egress interface. If the feature is enabled, you will observe output similar to the following:

```
Router# show running-configuration interface fastethernet 0/0
interface FastEthernet0/0
 ip address 25.0.0.6 255.0.0.0
 no ip mroute-cache
 load-interval 30
 duplex full
 speed 100
 crypto map bar
```

If the feature has been disabled, you will observe output similar to the following:

Router# **show running-configuration interface fastethernet 0/0**

```
interface FastEthernet0/0
 ip address 25.0.0.6 255.0.0.0
 no ip mroute-cache
 load-interval 30
 duplex full
 speed 100
 crypto map bar
 crypto ipsec fragmentation after-encryption
```

Configuration Examples

This section provides the following configuration example:

- Enabling Pre-Fragmentation For IPSec VPNs Example

Enabling Pre-Fragmentation For IPSec VPNs Example

The following configuration example shows how to configure the Pre-Fragmentation for IPSec VPNs feature:

Note This feature does not show up in the running configuration in this example because the default global pre-fragmentation for IPSec VPNs feature is enabled. Pre-fragmentation for IPSec VPNs shows in the running configuration only when you explicitly enable the feature on the interface.

```
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key abcd123 address 25.0.0.7
!
!
crypto ipsec transform-set fooprime esp-3des esp-sha-hmac
!
crypto map bar 10 ipsec-isakmp
 set peer 25.0.0.7
 set transform-set fooprime
 match address 102
```

Command Reference

The following commands are introduced or modified in the feature or features

- **crypto ipsec fragmentation**
- **crypto ipsec fragmentation (interface configuration)**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Real-Time Resolution for IPSec Tunnel Peer

After a user specifies a host name (instead of an IP address) for remote IP Security (IPSec) peer, the Real-Time Resolution for IPSec Tunnel Peer feature allows the host name to be domain name server (DNS) resolved before the router establishes the IPSec tunnel. Thus, the router can immediately discover whether the IP address of the peer has changed.

Feature History for Real-Time Resolution for IPSec Tunnel Peer

Release	Modification
12.3(4)T	This feature was introduced.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Real-Time Resolution for IPSec Tunnel Peer

Secure DNS Requirement

It is recommended that you use this feature only with secure DNS and when the DNS responses can be authenticated. Otherwise, an attacker can spoof or forge DNS responses and have access to Internet Key Exchange (IKE) authentication data, such as a certificate. If an attacker has a certificate that is trusted by the initiating host, the attacker can successfully establish Phase 1 IKE security association (SA), or the attacker can try to guess the preshared key that is shared between the initiator and the actual responder.

DNS Initiator

DNS names resolution for remote IPSec peers will work only if they are used as an initiator. The first packet that is to be encrypted will trigger a DNS lookup; after the DNS lookup is complete, subsequent packets will trigger IKE.

Information About Real-Time Resolution for IPSec Tunnel Peer

To configure real-time resolution for your IPSec peer, you should understand the following concept:

- Benefits of Real-Time Resolution Via Secure DNS, page 2

Benefits of Real-Time Resolution Via Secure DNS

When specifying the host name of a remote IPSec peer via the **set peer** command, you can also issue the **dynamic** keyword, which defers DNS resolution of the host name until right before the IPSec tunnel has been established. Deferring resolution enables the Cisco IOS software to detect whether the IP address of the remote IPSec peer has changed. Thus, the software can contact the peer at the new IP address.

If the **dynamic** keyword is not issued, the host name is resolved immediately after it is specified. So, the Cisco IOS software cannot detect an IP address change and, therefore, attempts to connect to the IP address that it previously resolved.

DNS resolution assures users that their established IPSec tunnel is secure and authenticated.

How to Configure Real-Time Resolution

This section contains the following procedure:

- Configuring Real-Time Resolution for IPSec Peers, page 2

Configuring Real-Time Resolution for IPSec Peers

Use this task to configure a router to perform real-time DNS resolution with a remote IPSec peer; that is, the host name of peer is resolved via a DNS lookup right before the router establishes a connection (an IPSec tunnel) with the peer.

Prerequisites

Before creating a crypto map, you should perform the following tasks:

- Define Internet Security Association Key Management Protocol (ISAKMP) policies.
- Define IPSec transform sets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **match address** *access-list-id*
5. **set peer** {*host-name* [**dynamic**] | *ip-address*}
6. **set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router(config)# crypto map secure_b 10 ipsec-isakmp`	Specifies the crypto map entry to create (or modify) and enters crypto map configuration mode.
Step 4	**match address** *access-list-id* **Example:** `Router(config-crypto-m)# match address 140`	Names an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec in the context of this crypto map entry.

	Command or Action	Purpose
Step 5	**set peer** {*host-name* [**dynamic**] | *ip-address*} **Example:** `Router(config-crypto-m)# set peer b.cisco.com dynamic`	Specifies a remote IPSec peer. This is the peer to which IPSec-protected traffic can be forwarded. • **dynamic**—Allows the host name to be resolved via a DNS lookup just before the router establishes the IPSec tunnel with the remote peer. If this keyword is not specified, the host name will be resolved immediately after the host name is specified. Repeat for multiple remote peers.
Step 6	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config-crypto-m)# set transform-set myset`	Specifies which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first).

Troubleshooting Tips

To display crypto map configuration information, use the **show crypto map** command.

What to Do Next

You need to apply a crypto map set to each interface through which IPSec traffic will flow. Applying the crypto map set to an interface instructs the router to evaluate all the interface's traffic against the crypto map set and to use the specified policy during connection or security association (SA) negotiation on behalf of traffic to be protected by crypto.

Configuration Examples for Real-Time Resolution

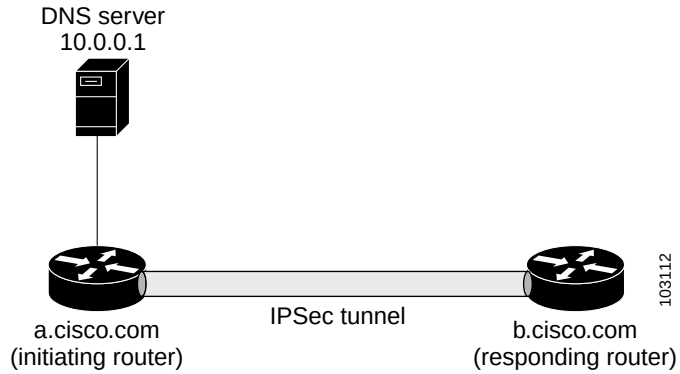
This section provides the following configuration example:

- Configuring Real-Time Resolution for an IPSec Peer: Example, page 4

Configuring Real-Time Resolution for an IPSec Peer: Example

Figure 1 and the following example illustrate how to create a crypto map that configures the host name of a remote IPSec peer to DNS resolved via a DNS lookup right before the Cisco IOS software attempts to establish a connection with that peer.

Figure 1 Real-Time Resolution Sample Topology




```
! Configure the initiating router.
hostname a.cisco.com
ip domain name cisco.com
ip name server 10.0.0.1
!
crypto map secure_b 10 ipsec-isakmp
  match address 140
  set peer b.cisco.com dynamic
  set transform-set xset
interface serial1
  ip address 30.0.0.1
  crypto map secure_b
access-list 140 permit ...
!
! Configure the responding router (the remote IPSec peer).
hostname b.cisco.com
!
crypto map secure_a 10 ipsec-isakmp
  match address 150
  set peer 30.0.0.1
  set transform-set
interface serial0/1
  ip address 40.0.0.1
  crypto map secure_a
access-list 150 ...

! DNS server configuration

b.cisco.com    40.0.0.1       # the address of serial0/1 of b.cisco.com
```

Additional References

The following sections provide references related to Real-Time Resolution for IPSec Tunnel Peer.

Related Documents

Related Topic	Document Title
Crypto maps	*The chapter "Configuring IPSec Network Security" in the Cisco IOS Security Configuration Guide*
ISAKMP policies	The chapter "Configuring Internet Key Exchange Security Protocol" in the *Cisco IOS Security Configuration Guide*
IPSec and IKE configuration commands	*Cisco IOS Security Command Reference, Release 12.3 T*

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features

- **set peer (IPSec)**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Reverse Route Injection

First Published: August 16, 2001
Last Updated: November 5, 2007

Reverse route injection (RRI) is the ability for static routes to be automatically inserted into the routing process for those networks and hosts protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities.

Each route is created on the basis of the remote proxy network and mask, with the next hop to this network being the remote tunnel endpoint. By using the remote Virtual Private Network (VPN) router as the next hop, the traffic is forced through the crypto process to be encrypted.

Enhancements to the default behavior of RRI, the addition of a route tag value, and enhancements to how RRI is configured were added to the Reverse Route Injection feature in Cisco IOS Release 12.3(14)T.

An enhancement was added in Cisco IOS Release 12.4(15)T that allows a distance metric to be set for routes that are created by a VPN process so that the dynamically learned route on a router can take precedence over a locally configured static route.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Reverse Route Injection" section on page 18.

Finding Support Information for Platforms and Cisco IOS Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Reverse Route Injection

- IP routing should be enabled and static routes should be redistributed if dynamic routing protocols are to be used to propagate RRI-generated static routes.

Restrictions for Reverse Route Injection

- If RRI is applied to a crypto map, that map must be unique to one interface on the router. In other words, the same crypto map cannot be applied to multiple interfaces. If more than one crypto map is applied to multiple interfaces, routes may not be cleaned up correctly. If multiple interfaces require a crypto map, each must use a uniquely defined map. This restriction applies only to RRI before Cisco IOS Release 12.3(14)T.
- For static crypto maps, routes are always present if RRI is configured on an applied crypto map. In Cisco IOS Release 12.3(14)T, the default behavior—of routes always being present for a static map—will not apply unless the **static keyword** is added to the **reverse-route** command.

Information About Reverse Route Injection

To configure the Reverse Route Injection enhancements, you should understand the following concepts:

Reverse Route Injection

RRI is the ability for static routes to be automatically inserted into the routing process for those networks and hosts that are protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities.

Each route is created on the basis of the remote proxy network and mask, with the next hop to this network being the remote tunnel endpoint. By using the remote VPN router as the next hop, the traffic is forced through the crypto process to be encrypted.

After the static route is created on the VPN router, this information is propagated to upstream devices, allowing them to determine the appropriate VPN router to which to send returning traffic in order to maintain IPsec state flows. Being able to determine the appropriate VPN router is particularly useful if multiple VPN routers are used at a site to provide load balancing or failover or if the remote VPN devices are not accessible via a default route. Routes are created in either the global routing table or the appropriate virtual route forwarding (VRF) table.

RRI is applied on a per-crypto map basis, whether this is via a static crypto map or a dynamic crypto map template. The default behavior for the two map types is as follows:

- In the case of a dynamic crypto map, routes are created upon the successful establishment of IPsec security associations (SAs) for those remote proxies. The next hop back to those remote proxies is via the remote VPN router whose address is learned and applied during the creation of the dynamic crypto map template. The routes are deleted after the SAs are deleted. In Cisco IOS Release 12.3(14)T, the creation of routes on the basis of IPsec source proxies on static crypto maps was added. This behavior became the default behavior on static maps and overrode the creation of routes on the basis of crypto ACLs (see the next bullet).
- For static crypto maps, routes are created on the basis of the destination information defined in the crypto access list. The next hop is taken from the first set peer statement that is attached to the crypto map. If at any time, RRI, the peer, or the access list is removed from the crypto map, routes will be deleted. This behavior changes with the addition of the RRI enhancements, as explained in the sections below.

Enhancements to Reverse Route Injection in Cisco IOS Release 12.4(15)T

The following enhancements have been added to the Reverse Route Injection feature in Cisco IOS Release 12.4(15)T:

RRI Distance Metric

In general, a static route is created having an administrative distance of 1, which means that static routes always have precedence in the routing table. In some scenarios, however, it is required that dynamically learned routes take precedence over static routes, with the static route being used in the absence of a dynamically learned route. The addition of the **set reverse-route distance** command under either a crypto map or IPsec profile allows you to specify a different distance metric for VPN-created routes so that those routes will be in effect only if a dynamic or more favored route becomes unavailable.

Gateway Option

This RRI gateway option is relevant to the crypto map only.

This option allows you to configure unique next hops or gateways for remote tunnel endpoints. The option is identical to the way the **reverse-route remote-peer** {*ip-address*} command worked prior to Cisco IOS Release 12.3(14)T in that two routes are created for each VPN tunnel. The first route is to the destination-protected subnet via the remote tunnel endpoint. The second route specifies the next hop to be taken to reach this tunnel endpoint. This RRI gateway option allows specific default paths to be specified for specific groups of VPN connections on platforms that support recursive route lookups.

Note In 12.4(15)T and later releases, the **gateway** keyword option replaces the **reverse-route remote-peer** command (with no *ip-address*). Due to changes to Cisco Express Forwarding (CEF), an interface as a next-hop cannot be used without also adding a next-hop IP address.

Support for RRI on IPsec Profiles

Previously RRI was available for crypto map configurations only. Cisco IOS Release 12.4(15)T introduces support for relevant RRI options on IPsec profiles that are predominantly used for virtual tunnel interfaces. On tunnel interfaces, only the distance metric and tag options are useful with the generic RRI capability.

Note It is not necessary to specifically enable RRI on dynamic virtual interfaces for Easy VPN clients. Route support is enabled by default. It is necessary to specify tag or distance metric values if these are required.

Tag Option Configuration Changes

The tag option was introduced in 12.3(14)T for crypto maps. This option is now supported with IPsec profiles under the **set reverse-route tag** command syntax. The **set reverse-route tag** command is also available under the crypto map for uniformity although the legacy **reverse-route tag** command is no longer supported.

show crypto route Command

The **show crypto route** command displays routes that are created through IPsec via RRI or Easy VPN virtual tunnel interfaces (VTIs). The routes are displayed in one table. To see sample output for the **show crypto route** command, see the section "**show crypto route Command Output: Example**."

How to Configure Reverse Route Injection

The following sections show how to configure reverse route injection for Cisco IOS software before Release 12.4(15)T and for Release 12.4(15)T.

- Configuring RRI on Crypto Maps for Cisco IOS Releases Prior to 12.4(15)T, page 4
- Configuring RRI with Enhancements Added in Cisco IOS Release 12.4(15)T, page 6

Configuring RRI on Crypto Maps for Cisco IOS Releases Prior to 12.4(15)T

This section includes the following tasks:

- Configuring RRI Under a Static Crypto Map, page 4
- Configuring RRI Under a Dynamic Map Template, page 5

Configuring RRI Under a Static Crypto Map

To configure RRI under a static crypto map for Cisco IOS software prior to Release 12.4(15)T, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **crypto map {***map-name}* {*seq-name}* **ipsec-isakmp**
4. **reverse-route** [**static** | **tag** *tag-id* [**static**] | **remote-peer** [**static**] | **remote-peer** *ip-address* [**static**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`crypto map {`*map-name}* `{`*seq-name}* `ipsec-isakmp` **Example:** `Router (config)# crypto map mymap 1 ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode.
Step 4	`reverse-route` [`static` | `tag` *tag-id* [`static`] | `remote-peer` [`static`] | `remote-peer` *ip-address* [`static`]] **Example:** `Router (config-crypto-map)# reverse-route remote peer 10.1.1.1`	Creates source proxy information for a crypto map entry.

Configuring RRI Under a Dynamic Map Template

To configure RRI under a dynamic map template for Cisco IOS software prior to Release 12.4(15)T, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic-map** *dynamic-map-name dynamic-seq-name*
4. **reverse-route** [**static** | **tag** *tag-id* [**static**] | **remote-peer** [**static**] | **remote-peer** *ip-address* [**static**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto dynamic-map** *dynamic-map-name dynamic-seq-name* **Example:** `Router (config)# crypto dynamic-map mymap 1`	Creates a dynamic crypto map entry and enters the crypto map configuration command mode.
Step 4	**reverse-route** [**static** | **tag** *tag-id* [**static**] | **remote-peer** [**static**] | **remote-peer** *ip-address* [**static**]] **Example:** `Router (config-crypto-map)# reverse-route remote peer 10.1.1.1`	Creates source proxy information for a crypto map entry.

Configuring RRI with Enhancements Added in Cisco IOS Release 12.4(15)T

The following sections show how to configure RRI with the enhancements that were added in Cisco IOS Release 12.4(15)T:

- Configuring RRI with Enhancements Under a Static Crypto Map, page 6
- Configuring RRI with Enhancements Under a Dynamic Map Template, page 7
- Configuring a RRI Distance Metric Under an IPsec Profile, page 8
- Verifying Routes That Are Created Through IPsec via RRI or Easy VPN VTIs, page 9

Configuring RRI with Enhancements Under a Static Crypto Map

To configure RRI with enhancements under a static crypto map (for Cisco IOS Release 12.4(15)T and later releases), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-name* **ipsec-isakmp**
4. **reverse-route** [**static** | **remote-peer** *ip-address* [**gateway**] [**static**]]
5. **set reverse-route** [**distance** *number* | **tag** *tag-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-name* **ipsec-isakmp** **Example:** `Router (config)# crypto map mymap 1 ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode.
Step 4	**reverse-route** [**static** | **remote-peer** *ip-address* [**gateway**] [**static**]] **Example:** `Router (config-crypto-map)# reverse-route`	Creates source proxy information for a crypto map entry. **Note** The **gateway** keyword can be added to enable the dual route functionality for default gateway support.
Step 5	**set reverse-route** [**distance** *number* | **tag** *tag-id*] **Example:** `Router (config-crypto-map)# set reverse-route distance 20`	Specifies a distance metric to be used or a tag value to be associated with these routes.

Configuring RRI with Enhancements Under a Dynamic Map Template

To configure RRI with enhancements under a dynamic map template (for Cisco IOS Release 12.4(15)T and later releases), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic-map** *dynamic-map-name dynamic-seq-name*
4. **reverse-route** [**static** | **remote-peer** *ip-address* [**gateway**] [**static**]]
5. **set reverse-route** [**distance** *number* | **tag** *tag-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto dynamic-map** *dynamic-map-name dynamic-seq-name* **Example:** `Router (config)# crypto dynamic-map mymap 1`	Creates a dynamic crypto map entry and enters the crypto map configuration command mode.
Step 4	**reverse-route** [**static** | **remote-peer** *ip-address* [**gateway**] [**static**]] **Example:** `Router (config-crypto-map)# reverse-route remote peer 10.1.1.1 gateway`	Creates source proxy information for a crypto map entry.
Step 5	**set reverse-route** [**distance** *number* | **tag** *tag-id*] **Example:** `Router (config-crypto-map)# set reverse-route distance 20`	Specifies a distance metric to be used or a tag value to be associated with these routes.

Configuring a RRI Distance Metric Under an IPsec Profile

To configure a RRI distance metric under an IPsec profile for Cisco IOS Release 12.4(15)T and later releases, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *name*
4. **set reverse-route** [**distance** *number* | **tag** *tag-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec profile** *name* **Example:** `Router (config)# crypto ipsec profile myprofile`	Creates or modifies an IPsec profile and enters IPsec profile configuration mode.
Step 4	**set reverse-route** [**distance** *number* | **tag** *tag-id*] **Example:** `Router (config-crypto-profile)# set reverse-route distance 20`	Defines a distance metric for each static route or tags a reverse route injection- (RRI-) created route. • **distance**—Defines a distance metric for each static route. • **tag**—Sets a tag value that can be used as a "match" value for controlling distribution using route maps.

Verifying Routes That Are Created Through IPsec via RRI or Easy VPN VTIs

To display routes that are created through IPsec via RRI or Easy VPN VTIs, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto route** **Example:** `Router# show crypto route`	Displays routes that are created through IPsec via RRI or Easy VPN VTIs.

Troubleshooting Tips

To observe the behavior of RRI and its relationship to the creation and deletion of an IPsec SA, you can use the **debug crypto ipsec** command (see the *Cisco IOS Debug Command Reference*, Release 12.4T).

Configuration Examples for Reverse Route Injection

This section contains the following sections:

Configuring RRI Prior to Cisco IOS Release 12.3(14)T: Examples

The following are examples of RRI configurations and output before Cisco IOS Release 12.3(14))T:

Configuring RRI When Crypto ACLs Exist: Example

The following example shows that all remote VPN gateways connect to the router via 192.168.0.3. RRI is added on the static crypto map, which creates routes on the basis of the source network and source netmask that are defined in the crypto access control list (ACL):

```
crypto map mymap 1 ipsec-isakmp
 set peer 10.1.1.1
 reverse-route
 set transform-set esp-3des-sha
 match address 102

Interface FastEthernet 0/0
 ip address 192.168.0.2 255.255.255.0
 standby name group1
 standby ip 192.168.0.3
 crypto map mymap redundancy group1

access-list 102 permit ip 192.168.1.0 0.0.0.255 10.0.0.0 0.0.255.255
```

In Cisco IOS Release 12.3(14)T and later releases, for the static map to retain this same behavior of creating routes on the basis of crypto ACL content, the **static** keyword is required, that is, **reverse-route static**.

Note The **reverse-route** command in this situation creates routes that are analogous to the following static route command-line interface (CLI) commands (**ip route**):

Remote Tunnel Endpoint

```
ip route 10.1.1.1 255.255.255.255 192.168.1.1
```

VPNSM

ip route 10.1.1.1 255.255.255.255 vlan0.1

Configuring RRI When Two Routes Are Created, One for the Remote Endpoint and One for Route Recursion: Example

In the following example, two routes are created, one for the remote endpoint and one for route recursion to the remote endpoint via the interface on which the crypto map is configured:

```
reverse-route remote-peer
```

Configuring RRI with Enhancements Added in Cisco IOS Release 12.3(14)T: Examples

The following are examples of configurations and output for RRI enhancements that were added in Cisco IOS Release 12.3(14)T.

- Configuring RRI When Crypto ACLs Exist: Example, page 11
- Configuring RRI with Route Tags: Example, page 11
- Configuring RRI for One Route to the Remote Proxy via a User-Defined Next Hop: Example, page 12

Configuring RRI When Crypto ACLs Exist: Example

The following example shows that RRI has been configured for a situation in which there are existing ACLs:

```
crypto map mymap 1 ipsec-isakmp
   set peer 172.17.11.1
   reverse-route static
   set transform-set esp-3des-sha
   match address 101

access-list 101 permit ip 192.168.1.0 0.0.0.255 172.17.11.0 0.0.0.255
```

Configuring RRI with Route Tags: Example

The following example shows how RRI-created routes can be tagged with a tag number and then used by a routing process to redistribute those tagged routes via a route map:

```
crypto dynamic-map ospf-clients 1
 reverse-route tag 5

router ospf 109
 redistribute rip route-map rip-to-ospf

route-map rip-to-ospf permit
 match tag 5
 set metric 5
 set metric-type type1

Router# show ip ospf topology
```

```
P 10.81.7.48/29, 1 successors, FD is 2588160, tag is 5
     via 192.168.82.25 (2588160/2585600), FastEthernet0/1
```

Configuring RRI for One Route to the Remote Proxy via a User-Defined Next Hop: Example

Note This option is applicable only to crypto maps.

The preceding example shows that one route has been created to the remote proxy via a user-defined next hop. This next hop should not require a recursive route lookup unless it will recurse to a default route.

```
reverse-route remote-peer 10.4.4.4
```

The preceding example yields the following prior to Cisco IOS Release 12.3(14)T:

```
10.0.0.0/24 via 10.1.1.1 (in the VRF table if VRFs are configured)
10.1.1.1/32 via 10.4.4.4 (in the global route table)
```

And this result occurs with RRI enhancements:

```
10.0.0.0/24 via 10.4.4.4 (in the VRF table if VRFs are configured, otherwise in the global
table)
```

Configuring RRI with Enhancements Added in Cisco IOS Release 12.4(15)T: Examples

The following are examples of configurations and output for RRI enhancements that were added in Cisco IOS Release 12.4(15)T.

Configuring a RRI Distance Metric Under a Crypto Map: Example

The following configuration shows a server and client configuration for which a RRI distance metric has been set under a crypto map:

Server

```
crypto dynamic-map mymap
 set security-association lifetime seconds 300
 set transform-set 3dessha
 set isakmp-profile profile1
 set reverse-route distance 20
 reverse-route
```

Client

```
crypto ipsec client ezvpn ez
 connect auto
 group cisco key cisco
```

```
 mode client
 peer 10.0.0.119
 username XXX password XXX
 xauth userid mode local
```

Configuring RRI with Route Tags: Example

The following example shows how RRI-created routes can be tagged with a tag number and then used by a routing process to redistribute those tagged routes via a route map:

```
crypto dynamic-map ospf-clients 1
 set reverse-route tag 5

router ospf 109
 redistribute rip route-map rip-to-ospf

route-map rip-to-ospf permit
 match tag 5
 set metric 5
 set metric-type type1

Router# show ip ospf topology

P 10.81.7.48/29, 1 successors, FD is 2588160, tag is 5
  via 192.168.82.25 (2588160/2585600), FastEthernet0/1
```

debug and show Command Output for a RRI Distance Metric Configuration Under a Crypto Map: Example

The following are **debug** and **show** command output for a RRI distance metric configuration under a crypto map on a server:

```
Router# debug crypto ipsec

00:23:37: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 10.0.0.119, remote= 10.0.0.14,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 192.168.6.1/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-sha-hmac  (Tunnel),
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x0
00:23:37: IPSEC(key_engine): got a queue event with 1 KMI message(s)
00:23:37: IPSEC(rte_mgr): VPN Route Event create routes for peer or rekeying for
 10.0.0.128
00:23:37: IPSEC(rte_mgr): VPN Route Refcount 1 FastEthernet0/0
00:23:37: IPSEC(rte_mgr): VPN Route Added 192.168.6.1 255.255.255.255 via 10.0.0.14 in IP
DEFAULT TABLE with tag 0 distance 20
00:23:37: IPSEC(policy_db_add_ident): src 0.0.0.0, dest 192.168.6.1, dest_port 0

Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.0.0.14 to network 0.0.0.0
```

```
C    192.200.200.0/24 is directly connected, Loopback0
     10.20.20.20/24 is subnetted, 1 subnets
C       10.30.30.30 is directly connected, Loopback4
C    192.168.5.0/24 is directly connected, Loopback3
     10.20.20.20/24 is subnetted, 2 subnets
S       10.3.1.0 [1/0] via 10.0.0.113
C       10.20.20.20 is directly connected, FastEthernet0/0
     192.168.6.0/32 is subnetted, 1 subnets
S       192.168.6.1 [20/0] via 10.0.0.14
C    192.168.3.0/24 is directly connected, Loopback2
     10.15.0.0/24 is subnetted, 1 subnets
C       10.15.0.0 is directly connected, Loopback6
S*   0.0.0.0/0 [1/0] via 10.0.0.14
```

Configuring a RRI Distance Metric for a VTI: Example

The following configuration shows a server and client configuration in which a RRI distance metric has been set for a VTI:

Server Configuration

```
crypto isakmp profile profile1
 keyring mykeyring
 match identity group cisco
 client authentication list authenlist
 isakmp authorization list autholist
 client configuration address respond
 virtual-template 1
crypto ipsec profile vi
 set transform-set 3dessha
 set reverse-route distance 20
 set isakmp-profile profile1
!
interface Virtual-Template1 type tunnel
 ip unnumbered
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile vi
```

Client Configuration

```
crypto ipsec client ezvpn ez
 connect auto
 group cisco key cisco
 mode client
 peer 10.0.0.119
 username XXX password XXX
 virtual-interface 1
```

debug and show Command Output for a RRI Metric Configuration Having a VTI: Example

The following are **debug** and **show** command output for a RRI metric configuration for a VTI on a server:

```
Router# debug crypto ipsec

00:47:56: IPSEC(key_engine): got a queue event with 1 KMI message(s)
00:47:56: Crypto mapdb : proxy_match
        src addr     : 0.0.0.0
        dst addr     : 192.168.6.1
        protocol     : 0
        src port     : 0
```

```
        dst port     : 0
00:47:56: IPSEC(crypto_ipsec_sa_find_ident_head): reconnecting with the same pro
xies and peer 10.0.0.14
00:47:56: IPSEC(rte_mgr): VPN Route Event create routes for peer or rekeying for
 10.0.0.14
00:47:56: IPSEC(rte_mgr): VPN Route Refcount 1 Virtual-Access2
00:47:56: IPSEC(rte_mgr): VPN Route Added 192.168.6.1 255.255.255.255 via Virtua
l-Access2 in IP DEFAULT TABLE with tag 0 distance 20
00:47:56: IPSEC(policy_db_add_ident): src 0.0.0.0, dest 192.168.6.1, dest_port 0

00:47:56: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.0.0.110, sa_proto= 50,
    sa_spi= 0x19E1175C(434181980),
    sa_trans= esp-3des esp-sha-hmac , sa_conn_id= 87
00:47:56: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.0.0.14, sa_proto= 50,
    sa_spi= 0xADC90C5(182227141),
    sa_trans= esp-3des esp-sha-hmac , sa_conn_id= 88
00:47:56: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access2, chang
ed state to up
00:47:56: IPSEC(key_engine): got a queue event with 1 KMI message(s)
00:47:56: IPSEC(key_engine_enable_outbound): rec'd enable notify from ISAKMP
00:47:56: IPSEC(key_engine_enable_outbound): enable SA with spi 182227141/50
00:47:56: IPSEC(update_current_outbound_sa): updated peer 10.0.0.14 current outb
ound sa to SPI ADC90C5

Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.0.0.14 to network 0.0.0.0

C    192.200.200.0/24 is directly connected, Loopback0
     10.20.20.20/24 is subnetted, 1 subnets
C       10.30.30.30 is directly connected, Loopback4
C    192.168.5.0/24 is directly connected, Loopback3
     10.20.20.20/24 is subnetted, 2 subnets
S       10.3.1.0 [1/0] via 10.0.0.113
C       10.20.20.20 is directly connected, FastEthernet0/0
     192.168.6.0/32 is subnetted, 1 subnets
S       192.168.6.1 [20/0] via 0.0.0.0, Virtual-Access2
C    192.168.3.0/24 is directly connected, Loopback2
     10.15.0.0/24 is subnetted, 1 subnets
C       10.15.0.0 is directly connected, Loopback6
S*   0.0.0.0/0 [1/0] via 10.0.0.14
```

show crypto route Command Output: Example

The following output example displays routes, in one table, that are created through IPsec via RRI or Easy VPN VTIs:

```
Router# show crypto route

VPN Routing Table: Shows RRI and VTI created routes
Codes: RRI - Reverse-Route, VTI- Virtual Tunnel Interface
       S - Static Map ACLs
```

```
Routes created in table GLOBAL DEFAULT
192.168.6.2/255.255.255.255 [0/0] via 10.0.0.133
                                  on Virtual-Access3 RRI
10.1.1.0/255.255.255.0 [10/0] via Virtual-Access2 VTI
192.168.6.1/255.255.255.255 [0/0] via Virtual-Access2 VTI
```

Additional References

The following sections provide references related to Reverse Route Injection enhancements.

Related Documents

Related Topic	Document Title
Cisco IOS Security commands	*Cisco IOS Security Command Reference, Release 12.4T*
Other Cisco IOS commands	*Cisco IOS Command Reference*, Release 12.4T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features

- **reverse-route**
- **set reverse-route**
- **show crypto route**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Feature Information for Reverse Route Injection

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Reverse Route Injection

Feature Name	Releases	Feature Information
Reverse Route Injection	12.1(9)E 12.2(8)T 12.2(8)YE	Reverse route injection (RRI) is the ability for static routes to be automatically inserted into the routing process for those networks and hosts protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities. Each route is created on the basis of the remote proxy network and mask, with the next hop to this network being the remote tunnel endpoint. By using the remote Virtual Private Network (VPN) router as the next hop, the traffic is forced through the crypto process to be encrypted. The following sections provide information about this feature: • "Reverse Route Injection" section on page 2 The following commands were introduced or modified by this feature: **reverse-route**.
Reverse Route Remote Peer Options	12.2(13)T 12.2(14)S	An enhancement was added to RRI to allow you to specify an interface or address as the explicit next hop to the remote VPN device. This functionality allows the overriding of a default route to properly direct outgoing encrypted packets. The following sections provide information about the remote peer options: • "Enhancements to Reverse Route Injection in Cisco IOS Release 12.4(15)T" section on page 3.

Table 1 Feature Information for Reverse Route Injection (continued)

Feature Name	Releases	Feature Information
Reverse Route Injection Enhancements	12.3(14)T 12.2(33)SRA 12.2(33)SXH	The following enhancements were added to the Reverse Route Injection feature: • The default behavior of static crypto maps will be the same as that of dynamic crypto maps unless the **reverse-route** command and **static** keyword are used. • A route tag value was added for any routes that are created using RRI. • RRI can be configured on the same crypto map that is applied to multiple router interfaces. • RRI configured with the **reverse-route remote-peer** {*ip-address}* command, keyword, and argument will create one route instead of two. The following sections provide information about the Reverse Route Injection enhancements: • "Reverse Route Injection" section on page 2 • "Configuring RRI on Crypto Maps for Cisco IOS Releases Prior to 12.4(15)T" section on page 4 • "Configuring RRI with Enhancements Added in Cisco IOS Release 12.4(15)T" section on page 6 • "Configuring RRI When Crypto ACLs Exist: Example" section on page 10 • "Configuring RRI with Route Tags: Example" section on page 11 • "Configuring RRI for One Route to the Remote Proxy via a User-Defined Next Hop: Example" section on page 12 The following command was modified by these feature enhancements: **reverse-route**.
Gateway Option	12.4(15)T	This option allows you to configure unique next hops or gateways for remote tunnel endpoints. The following section provides information about the Gateway Option: • "Gateway Option" section on page 3

Table 1 Feature Information for Reverse Route Injection (continued)

Feature Name	Releases	Feature Information
RRI Distance Metric	12.4(15)T	This enhancement allows you to define a metric distance for each static route. The following sections provide information about the RRI distance metric enhancement. • "RRI Distance Metric" section on page 3 • "Configuring a RRI Distance Metric Under an IPsec Profile" section on page 8 • "Configuring a RRI Distance Metric Under a Crypto Map: Example" section on page 12 • "debug and show Command Output for a RRI Metric Configuration Having a VTI: Example" section on page 14 The following commands were introduced or modified by this feature: **reverse-route**, **set reverse-route**.
show crypto route Command	12.4(15)T	This command displays routes that are created through IPsec via RRI or Easy VPN VTIs.
Support for RRI on IPsec Profiles	12.4(15)T	This feature provides support for relevant RRI options on IPsec profiles that are predominantly used by VTIs. The following section provides information about the Support for RRI on IPsec Profiles feature: • "Support for RRI on IPsec Profiles" section on page 4
Tag Option Configuration Changes	12.4(15)T	The tag option is now supported with IPsec profiles under the **set reverse-route tag** command. The following section provides information about this feature enhancement: • "Tag Option Configuration Changes" section on page 4
Reverse Route Injection (RRI)	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

SafeNet IPSec VPN Client Support

The SafeNet IPSec VPN Client Support feature allows you to limit the scope of an Internet Security Association and Key Management Protocol (ISAKMP) profile or ISAKMP keyring configuration to a local termination address or interface. The benefit of this feature is that different customers can use the same peer identities and ISAKMP keys by using different local termination addresses.

History for the SafeNet IPSec VPN Client Support Feature

Release	Modification
12.3(14)T	This feature was introduced.
12.2(18)SXE	This feature was integrated into Cisco IOS Release 12.2(18)SXE.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for SafeNet IPSec VPN Client Support

- You must understand how to configure ISAKMP profiles and ISAKMP keyrings.

Restrictions for SafeNet IPSec VPN Client Support

- The local address option works only for the primary address of an interface.
- If an IP address is provided, the administrator has to ensure that the connection of the peer terminates to the address that is provided.
- If the IP address does not exist on the device, or if the interface does not have an IP address, the ISAKMP profile or ISAKMP keyring will be effectively disabled.

Information About SafeNet IPSec VPN Client Support

Before configuring SafeNet IPSec VPN Client Support, you should understand the following concepts:

- ISAKMP Profile and ISAKMP Keyring Configurations: Background, page 2
- Local Termination Address or Interface, page 2

ISAKMP Profile and ISAKMP Keyring Configurations: Background

Prior to Cisco IOS Release 12.3(14)T, ISAKMP-profile and ISAKMP-keyring configurations could be only global, meaning that the scope of these configurations could not be limited by any locally defined parameters (VRF instances were an exception). For example, if an ISAKMP keyring contained a preshared key for address 10.11.12.13, the same key would be used if the peer had the address 10.11.12.13, irrespective of the interface or local address to which the peer was connected. There are situations, however, in which users prefer that associate keyrings be bound not only with virtual route forwarding (VRF) instances but also to a particular interface. For example, if instead of VRF instances, there are virtual LANS, and the Internet Key Exchange (IKE) is negotiated with a group of peers using one fixed virtual LAN (VLAN) interface. Such a group of peers uses a single preshared key, so if keyrings could be bound to an interface, it would be easy to define a wildcard key without risking that the keys would also be used for other customers.

Sometimes the identities of the peer are not in the control of the administrator, and even if the same peer negotiates for different customers, the local termination address is the only way to distinguish the peer. After such a distinction is made, if the traffic is sent to different VRF instances, configuring an ISAKMP profile is the only way to distinguish the peer. Unfortunately, when the peer uses an identical identity for all such situations, the ISAKMP profile cannot distinguish among the negotiations. For such scenarios, it would be beneficial to bind ISAKMP profiles to a local termination address. If a local termination address could be assigned, identical identities from the peer would not be a problem.

Local Termination Address or Interface

Effective with Cisco IOS Release 12.3(14)T, the SafeNet IPSec VPN Client Support feature allows you to limit the scope of ISAKMP profiles and ISAKMP keyrings to a local termination address or interface.

Benefit of SafeNet IPSec VPN Client Support

The benefit of this feature is that different customers can use the same peer identities and ISAKMP keys by using different local termination addresses.

How to Configure SafeNet IPSec VPN Client Support

This section contains the following procedures. The first two configurations are independent of each other.

Limiting an ISAKMP Profile to a Local Termination Address or Interface

To configure an ISAKMP profile and limit it to a local termination address or interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name*
4. **keyring** *keyring-name*
5. **match identity address** *address*
6. local-address {*interface-name* | *ip-address* [*vrf-tag*]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto isakmp profile** *profile-name* **Example:** `Router (config)# crypto isakmp profile profile1`	Defines an ISAKMP profile and enters ISAKMP profile configuration mode.
Step 4	**keyring** *keyring-name* **Example:** `Router (conf-isa-profile)# keyring keyring1`	(Optional) Configures a keyring with an ISAKMP profile. • A keyring is not needed inside an ISAKMP profile for local termination to work. Local termination works even if Rivest, Shamir, and Adelman (RSA) certificates are used.
Step 5	**match identity address** *address* **Example:** `Router (conf-isa-profile)# match identity address 10.0.0.0 255.0.0.0`	Matches an identity from a peer in an ISAKMP profile.
Step 6	**local-address** {*interface-name* | *ip-address* [*vrf-tag*]} **Example:** `Router (conf-isa-profile)# local-address serial2/0`	Limits the scope of an ISAKMP profile or an ISAKMP keyring configuration to a local termination address or interface.

Limiting a Keyring to a Local Termination Address or Interface

To configure an ISAKMP keyring and limit its scope to a local termination address or interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto keyring** *keyring-name*
4. local-address {*interface-name* | *ip-address* [*vrf-tag*]}
5. pre-shared-key address *address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto keyring** *keyring-name* **Example:** `Router (config)# crypto keyring keyring1`	Defines a crypto keyring to be used during IKE authentication and enters keyring configuration mode.
Step 4	**local-address** {*interface-name* | *ip-address* [*vrf-tag*]} **Example:** `Router (conf-keyring)# local-address serial2/0`	Limits the scope of an ISAKMP profile or an ISAKMP keyring configuration to a local termination address or interface.
Step 5	**pre-shared-key address** *address* **Example:** `Router (conf-keyring)# pre-shared-key address 10.0.0.1`	Defines a preshared key to be used for IKE authentication.

Monitoring and Maintaining SafeNet IPSec VPN Client Support

The following **debug** and **show** commands may be used to monitor and maintain the configuration in which you limited the scope of an ISAKMP profile or ISAKMP keyring to a local termination address or interface.

SUMMARY STEPS

1. **enable**
2. **debug crypto isakmp**
3. **show crypto isakmp profile**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays messages about IKE events.
Step 3	**show crypto isakmp profile** **Example:** `Router# show crypto isakmp profile`	Lists all the ISAKMP profiles that are defined on a router.

Examples

debug crypto isakmp Command Output for an ISAKMP Keyring That Is Bound to Local Termination Addresses: Example

You have an ISAKMP configuration as follows (the address of serial2/0 is 10.0.0.1, and the address of serial2/1 is 10.0.0.2),

```
crypto keyring keyring1
! Scope of the keyring is limited to interface serial2/0.
  local-address serial2/0
  ! The following is the key string used by the peer.
  pre-shared-key address 10.0.0.3 key somerandomkeystring
crypto keyring keyring2
  local-address serial2/1
  ! The following is the keystring used by the peer coming into serial2/1.
   pre-shared-key address 10.0.0.3 key someotherkeystring
```

and if the connection is coming into serial2/0, keyring1 is chosen as the source of the preshared key (and keyring2 is ignored because it is bound to serial2/1), you would see the following output:

```
Router# debug crypto isakmp

*Feb 11 15:01:29.595: ISAKMP:(0:0:N/A:0):Keyring keyring2 is bound to
  10.0.0.0, skipping
*Feb 11 15:01:29.595: ISAKMP:(0:0:N/A:0):Looking for a matching key for
  10.0.0.3 in keyring1
*Feb 11 15:01:29.595: ISAKMP:(0:0:N/A:0): : success
*Feb 11 15:01:29.595: ISAKMP:(0:0:N/A:0):found peer pre-shared key
  matching 10.0.0.3
*Feb 11 15:01:29.595: ISAKMP:(0:0:N/A:0): local preshared key found
```

debug crypto isakmp Command Output for an ISAKMP ProfileThat Is Bound to a Local Termination Address: Example

If you have the following configuration,

```
crypto isakmp profile profile1
```

```
  keyring keyring1
  match identity address 10.0.0.0 255.0.0.0
  local-address serial2/0
crypto isakmp profile profile2
  keyring keyring1
  keyring keyring2
  self-identity fqdn
  match identity address 10.0.0.1 255.255.255.255
  local-address serial2/1
```

and the connection is coming through the local terminal address serial2/0, you will see the following output:

```
Router# debug crypto isakmp

*Feb 11 15:01:29.935: ISAKMP:(0:0:N/A:0):

Profile profile2 bound to 10.0.0.0 skipped

*Feb 11 15:01:29.935: ISAKMP:(0:1:SW:1):: peer matches profile1 profile
```

show crypto isakmp profile Command Output: Example

The following is an example of typical **show** command output for an ISAKMP profile that is bound to serial2/0:

```
Router# show crypto isakmp profile

ISAKMP PROFILE profile1
  Identities matched are:
    ip-address 10.0.0.0 255.0.0.0
  Certificate maps matched are:
  keyring(s): keyring1
  trustpoint(s): <all>
  Interface binding: serial2/0 (10.20.0.1:global)
```

Troubleshooting SafeNet IPSec VPN Client Support

If an ISAKMP profile or ISAKMP keyring fails to be selected, you should double-check the local-address binding in the ISAKMP profile or ISAKMP keyring configuration and follow the output of the IKE debugs to determine whether the peer is correctly terminating on the address. You may remove the local-address binding (to make the scope of the profile or keyring global) and check to determine whether the profile or keyring is selected to confirm the situation.

Configuration Examples for SafeNet IPSec VPN Client Support

This section contains the following configuration, **debug** command, and **show** command examples.

ISAKMP Profile Bound to a Local Interface: Example

The following example shows that the ISAKMP profile is bound to a local interface:

```
crypto isakmp profile profile1
  keyring keyring1
  match identity address 10.0.0.0 255.0.0.0
 local-address serial2/0
```

ISAKMP Keyring Bound to a Local Interface: Example

The following example shows that the ISAKMP keyring is bound only to interface serial2/0:

```
crypto keyring
  local-address serial2/0
 pre-shared-key address 10.0.0.1
```

ISAKMP Keyring Bound to a Local IP Address: Example

The following example shows that the ISAKMP keyring is bound only to IP address 10.0.0.2:

```
crypto keyring keyring1
  local-address 10.0.0.2
 pre-shared-key address 10.0.0.2 key
```

ISAKMP Keyring Bound to an IP Address and Limited to a VRF: Example

The following example shows that an ISAKMP keyring is bound to IP address 10.34.35.36 and that the scope is limited to VRF examplevrf1:

```
ip vrf examplevrf1
  rd 12:3456
crypto keyring ring1
  local-address 10.34.35.36 examplevrf1
interface ethernet2/0
  ip vrf forwarding examplevrf1
  ip address 10.34.35.36 255.255.0.0
```

Additional References

The following sections provide references related to SafeNet IPSec VPN Client Support.

Related Documents

Related Topic	Document Title
Configuring ISAKMP profiles and ISAKMP keyrings	*VRF-Aware IPSec*
Security commands	*Cisco IOS Security Command Reference*, *Release 12.3T*

Standards

Standard	Title
No new or modified standards are supported by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features

- **local-address**

For information about these commands, see the Cisco IOS Security Command Reference at

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

VRF-Aware IPSec

The VRF-Aware IPSec feature introduces IP Security (IPSec) tunnel mapping to Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). Using the VRF-Aware IPSec feature, you can map IPSec tunnels to Virtual Routing and Forwarding (VRF) instances using a single public-facing address.

Feature Specifications for VRF-Aware IPSec

Feature History	
Release	**Modification**
12.2(15)T	This feature was introduced.
Supported Platforms	
Cisco 1710, Cisco 1760, Cisco 2610-Cisco 2613, Cisco 2620-Cisco 2621, Cisco 2650-Cisco 2651, Cisco 3620, Cisco 3640, Cisco 3660, Cisco 7100, Cisco 7200, Cisco 7400, Cisco 870 Series	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for VRF-Aware IPSec

- If you are configuring VRF-Aware IPSec using a crypto map configuration and the Inside VRF (IVRF) is not the same as the Front Door VRF (FVRF), this feature is not interoperable with unicast reverse path forwarding (uRPF) if uRPF is enabled on the crypto map interface. If your network requires uRPF, it is recommended that you use Virtual Tunnel Interface (VTI) for IPSec instead of crypto maps.
- The VRF-Aware IPSec feature does not allow IPSec tunnel mapping between VRFs. For example, it does not allow IPSec tunnel mapping from VRF vpn1 to VRF vpn2.

Information About VRF-Aware IPSec

The VRF-Aware IPSec feature maps an IPSec tunnel to a MPLS VPN. To configure and use the feature, you need to understand the following concepts:

- VRF Instance, page 2
- MPLS Distribution Protocol, page 2
- VRF-Aware IPSec Functional Overview, page 2

VRF Instance

A VRF instance is a per-VPN routing information repository that defines the VPN membership of a customer site attached to the Provider Edge (PE) router. A VRF comprises an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table. A separate set of routing and CEF tables is maintained for each VPN customer.

MPLS Distribution Protocol

The MPLS distribution protocol is a high-performance packet-forwarding technology that integrates the performance and traffic management capabilities of data link layer switching with the scalability, flexibility, and performance of network-layer routing.

VRF-Aware IPSec Functional Overview

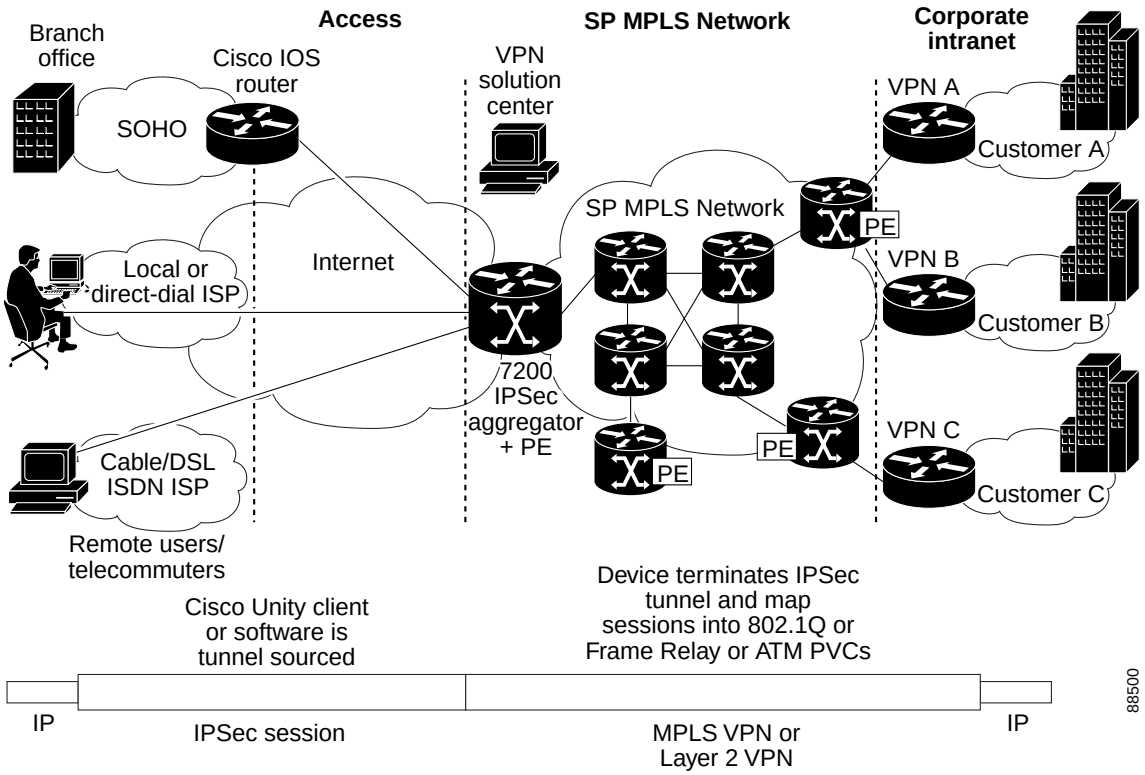
Front Door VRF (FVRF) and and Inside VRF (IVRF) are central to understanding the feature.

Each IPSec tunnel is associated with two VRF domains. The outer encapsulated packet belongs to one VRF domain, which we shall call the FVRF, while the inner, protected IP packet belongs to another domain called the IVRF. Another way of stating the same thing is that the local endpoint of the IPSec tunnel belongs to the FVRF while the source and destination addresses of the inside packet belong to the IVRF.

One or more IPSec tunnels can terminate on a single interface. The FVRF of all these tunnels is the same and is set to the VRF that is configured on that interface. The IVRF of these tunnels can be different and depends on the VRF that is defined in the Internet Security Association and Key Management Protocol (ISAKMP) profile that is attached to a crypto map entry.

Figure 96 is an illustration of a scenario showing IPSec to MPLS and Layer 2 VPNs.

Figure 96 IPSec to MPLS and Layer 2 VPNs



Packet Flow into the IPSec Tunnel

- A VPN packet arrives from the Service Provider MPLS backbone network to the PE and is routed through an interface facing the Internet.
- The packet is matched against the Security Policy Database (SPD), and the packet is IPSec encapsulated. The SPD includes the IVRF and the access control list (ACL).
- The IPSec encapsulated packet is then forwarded using the FVRF routing table.

Packet Flow from the IPSec Tunnel

- An IPSec-encapsulated packet arrives at the PE router from the remote IPSec endpoint.
- IPSec performs the Security Association (SA) lookup for the Security Parameter Index (SPI), destination, and protocol.
- The packet is decapsulated using the SA and is associated with IVRF.
- The packet is further forwarded using the IVRF routing table.

How to Configure VRF-Aware IPSec

This section contains the following procedures:

Configuring Crypto Keyrings

A crypto keyring is a repository of preshared and Rivest, Shamir, and Adelman (RSA) public keys. There can be zero or more keyrings on the Cisco IOS router.

Perform the following optional task to configure a crypto keyring.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto keyring** *keyring-name* [**vrf** *fvrf-name*]
4. **description** *string* (Optional)
5. **pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname*} **key** *key (Optional)*
6. **rsa-pubkey** {**address** *address* | **name** *fqdn*} [**encryption** | **signature**] (Optional)
7. **address** *ip-address (Optional)*
8. **serial-number** *serial-number (Optional)*
9. **key-string**
10. **text**
11. quit
12. exit
13. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto keyring** *keyring-name* [**vrf** *fvrf-name*] **Example:** `Router (config)# crypto keyring VPN1`	Defines a keyring with *keyring-name* as the name of the keyring and enters keyring configuration mode. • (Optional) The **vrf** keyword and *fvrf-name* argument imply that the keyring is bound to Front Door Virtual Routing and Forwarding (FVRF). The key in the keyring is searched if the local endpoint is in FVRF. If **vrf** is not specified, the keyring is bound to the global.
Step 4	**description** *string* `Router (config-keyring)# description The keys for VPN1`	(Optional) Specifies a one-line description of the keyring.
Step 5	**pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname*} **key** *key* **Example:** `Router (config-keyring)# pre-shared-key address 10.72.23.11 key VPN1`	(Optional) Defines a preshared key by address or host name.
Step 6	**rsa-pubkey** {**address** *address* | **name** *fqdn*} [**encryption** | **signature**] **Example:** `Router(config-keyring)# rsa-pubkey name host.vpn.com`	(Optional) Defines a Rivest, Shamir, and Adelman (RSA) public key by address or host name and enters rsa-pubkey configuration mode. • By default, the key is used for signature. • The optional **encryption** keyword specifies that the key should be used for encryption. The optional **signature** keyword specifies that the key should be used for signature. By default, the key is used for signature.
Step 7	**address** *ip-address* **Example:** `Router(config-pubkey-key)# address 10.5.5.1`	(Optional) Defines the RSA public key IP address.
Step 8	**serial-number** *serial-number* **Example:** `Router(config-pubkey-key)# serial-number 1000000`	(Optional) Specifies the serial number of the public key. The value is from 0 through infinity.

	Command or Action	Purpose
Step 9	**key-string** **Example:** `Router (config-pubkey-key)# key-string`	Enters into the text mode in which you define the public key.
Step 10	**text** **Example:** `Router (config-pubkey)# 00302017 4A7D385B 1234EF29 335FC973`	Specifies the public key. **Note** Only one public key may be added in this step.
Step 11	**quit** **Example:** `Router (config-pubkey)# quit`	Quits to the public key configuration mode.
Step 12	**exit** **Example:** `Router (config-pubkey)# exit`	Exits to the keyring configuration mode.
Step 13	**exit** **Example:** `Router(config-keyring)# exit#`	Exits to global configuration mode.

Configuring ISAKMP Profiles

An ISAKMP profile is a repository for IKE Phase 1 and IKE Phase 1.5 configuration for a set of peers. An ISAKMP profile defines items such as keepalive, trustpoints, peer identities, and XAUTH AAA list during the IKE Phase 1 and Phase 1.5 exchange. There can be zero or more ISAKMP profiles on the Cisco IOS router.

Note

- If traffic from the router to a certification authority (CA) (for authentication, enrollment, or for obtaining a certificate revocation list [CRL]) or to an Lightweight Directory Access Protocol (LDAP) server (for obtaining a CRL) needs to be routed via a VRF, the **vrf** command must be added to the trustpoint. Otherwise, such traffic will use the default routing table.
- If a profile does not specify one or more trustpoints, all trustpoints in the router will be used to attempt to validate the certificate of the peer (Internet Key Exchange (IKE) main mode or signature authentication). If one or more trustpoints are specified, only those trustpoints will be used.

Restriction

A router initiating IKE and a router responding to the IKE request should have symmetrical trustpoint configurations. For example, a responding router (in IKE Main Mode) performing RSA signature encryption and authentication might use trustpoints that were defined in the global configuration when sending the CERT-REQ payloads. However, the router might use a restricted list of trustpoints that were defined in the ISAKMP profile for the certificate verification. If the peer (the IKE initiator) is configured

to use a certificate whose trustpoint is in the global list of the responding router but not in ISAKMP profile of the responding router, the certificate will be rejected. (However, if the initiating router does not know about the trustpoints in the global configuration of the responding router, the certificate can still be authenticated.)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name*
4. **description** *string* (Optional)
5. **vrf** i*vrf-name (Optional)*
6. **keepalive** *seconds* **retry** *retry-seconds* (Optional)
7. **self-identity** {**address** | **fqdn** | **user-fqdn** *user-fqdn*} (Optional)
8. **keyring** *keyring-name (Optional)*
9. **ca trust-point** *trustpoint-name* (Optional)
10. **match identity** {**group** *group-name* | **address** *address* [*mask*] [*fvrf]* | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name}*
11. **client configuration address** {**initiate** | **respond**} (Optional)
12. **client authentication list** *list-name (Optional)*
13. **isakmp authorization list** *list-name* (Optional)
14. **initiate mode aggressive**
15. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp profile** *profile-name* **Example:** `Router (config)# crypto isakmp profile vpnprofile`	Defines an Internet Security Association and Key Management Protocol (ISAKMP) profile and enters into isakmp profile configuration mode.

	Command or Action	Purpose
Step 4	**description** *string* **Example:** `Router (conf-isa-prof)# description configuration for VPN profile`	(Optional) Specifies a one-line description of an ISAKMP profile.
Step 5	**vrf** i*vrf-name* **Example:** `Router (conf-isa-prof)# vrf VPN1`	(Optional) Maps the IPSec tunnel to a Virtual Routing and Forwarding (VRF) instance. **Note** The VRF also serves as a selector for matching the Security Policy Database (SPD). If the VRF is not specified in the ISAKMP profile, the IVRF of the IPSec tunnel will be the same as its FVRF.
Step 6	**keepalive** *seconds* **retry** *retry-seconds* **Example:** `Router (conf-isa-prof)# keepalive 60 retry 5`	(Optional) Allows the gateway to send dead peer detection (DPD) messages to the peer. • If not defined, the gateway uses the global configured value. • *seconds*—Number of seconds between DPD messages. The range is from 10 to 3600 seconds. • **retry** *retry-seconds*—Number of seconds between retries if the DPD message fails. The range is from 2 to 60 seconds.
Step 7	**self-identity** {**address** | **fqdn** | **user-fqdn** *user-fqdn*} **Example:** `Router (conf-isa-prof)# self-identity address`	(Optional) Specifies the identity that the local Internet Key Exchange (IKE) should use to identify itself to the remote peer. • If not defined, IKE uses the global configured value. • **address**—Uses the IP address of the egress interface. • **fqdn—**Uses the fully qualified domain name (FQDN) of the router. • **user-fqdn**—Uses the specified value.
Step 8	**keyring** *keyring-name* **Example:** `Router (conf-isa-prof)# keyring VPN1`	(Optional) Specifies the keyring to use for Phase 1 authentication. • If the keyring is not specified, the global key definitions are used.
Step 9	**ca trust-point** {*trustpoint-name*} **Example:** `Router (conf-isa-prof)# ca trustpoint VPN1-trustpoint`	(Optional) Specifies a trustpoint to validate a Rivest, Shamir, and Adelman (RSA) certificate. • If no trustpoint is specified in the ISAKMP profile, all the trustpoints that are configured on the Cisco IOS router are used to validate the certificate.

	Command or Action	Purpose
Step 10	**match identity** {**group** *group-name* | **address** *address* [*mask*] [*fvrf]* | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name*} **Example:** `Router (conf-isa-prof)# match identity address 10.1.1.1`	Specifies the client IKE Identity (ID) that is to be matched. • **group** *group-name*—Matches the *group-name* with the ID type ID_KEY_ID. It also matches the *group-name* with the Organizational Unit (OU) field of the Distinguished Name (DN).· • **address** *address* [*mask*] *fvrf* —Matches the *address* with the ID type ID_IPV4_ADDR. The *mask* argument can be used to specify a range of addresses. The *fvrf argument* specifies that the address is in Front Door Virtual Routing and Forwarding (FVRF) • **host** *hostname*—Matches the *hostname* with the ID type ID_FQDN. • **host domain** *domainname*—Matches the *domainname* to the ID type ID_FQDN whose domain name is the same as the *domainname*. Use this command to match all the hosts in the domain. • **user** *username*—Matches the *username* with the ID type ID_USER_FQDN· • **user domain** *domainname*—Matches the ID type ID_USER_FQDN whose domain name matches the *domainname*.
Step 11	**client configuration address** {**initiate** | **respond**} **Example:** `Router (conf-isa-prof)# client configuration address initiate`	(Optional) Specifies whether to initiate the mode configuration exchange or responds to mode configuration requests.
Step 12	**client authentication list** *list-name* **Example:** `Router (conf-isa-prof)# client authentication list xauthlist`	(Optional) Authentication, authorization, and accounting (AAA) to use for authenticating the remote client during the extended authentication (XAUTH) exchange.
Step 13	**isamkp authorization list** *list-name* **Example:** `Router (conf-isa-prof)# isakmp authorization list ikessaaalist`	(Optional) Network authorization server for receiving the Phase 1 preshared key and other attribute-value (AV) pairs.
Step 14	**initiate mode aggressive** **Example:** `Router (conf-isa-prof)# initiate mode aggressive`	(Optional) Initiates aggressive mode exchange. • If not specified, IKE always initiates Main Mode exchange.
Step 15	**exit** **Example:** `Router (conf-isa-prof)# exit`	Exits to global configuration mode.

What to Do Next

Go to the section "Configuring an ISAKMP Profile on a Crypto Map."

Configuring an ISAKMP Profile on a Crypto Map

An ISAKMP profile must be applied to the crypto map. The IVRF on the ISAKMP profile is used as a selector when matching the VPN traffic. If there is no IVRF on the ISAKMP profile, the IVRF will be equal to the FVRF. Perform this required task to configure an ISAKMP profile on a crypto map.

Prerequisites

Before configuring an ISAKMP profile on a crypto map, you must first have configured your router for basic IPSec.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name* **isakmp-profile** *isakmp-profile-name (Optional)*
4. **set isakmp-profile** *profile-name (Optional)*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name* **isakmp-profile** *isakmp-profile-name* **Example:** `Router (config)# crypto map vpnmap isakmp-profile vpnprofile`	(Optional) Specifies the Internet Key Exchange and Key Management Protocol (ISAKMP) profile for the crypto map set and enters crypto map configuration mode. • The ISAKMP profile will be used during IKE exchange.

	Command or Action	Purpose
Step 4	**set isakmp-profile** *profile-name* **Example:** `Router (config-crypto-map)# set isakmp-profile vpnprofile`	(Optional) Specifies the ISAKMP profile to use when the traffic matches the crypto map entry.
Step 5	**exit** **Example:** `Router (config-crypto-map)# exit`	Exits to global configuration mode.

Configuring to Ignore Extended Authentication During IKE Phase 1 Negotiation

To ignore XAUTH during an IKE Phase 1 negotiation, use the **no crypto xauth** command. Use the **no crypto xauth** command if you do not require extended authentication for the Unity clients.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no crypto xauth** *interface*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no crypto xauth** *interface* **Example:** `Router(config)# no crypto xauth ethernet0`	Ignores XAUTH proposals for requests that are destined to the IP address of the interface. By default, Internet Key Exchange (IKE) processes XAUTH proposals.

Verifying VRF-Aware IPSec

To verify your VRF-Aware IPSec configurations, use the following **show** commands. These **show** commands allow you to list configuration information and security associations (SAs):

SUMMARY STEPS

- **enable**
- **show crypto ipsec sa** [**map** *map-name* | **address** | **identity | interface** *interface* | **peer** [**vrf** *fvrf-name*] **address | vrf** *ivrf-name*] **[detail]**
- **show crypto isakmp key**
- **show crypto isakmp profile**
- **show crypto key pubkey-chain rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto ipsec sa** [**map** *map-name* | **address** | **identity	interface** *interface* | **peer** [**vrf** *fvrf-name*] **address	vrf** *ivrf-name*] **[detail]** **Example:** `Router# show crypto ipsec sa vrf vpn1`	Allows you to view the settings used by current security associations (SAs).
Step 3	**show crypto isakmp key** **Example:** `Router# show crypto isakmp key`	Lists all the keyrings and their preshared keys. • Use this command to verify your crypto keyring configuration.
Step 4	**show crypto isakmp profile** **Example:** `Router# show crypto isakmp profile`	Lists all ISAKMP profiles and their configurations.
Step 5	**show crypto key pubkey-chain rsa** **Example:** `Router# show crypto key pubkey-chain rsa`	Views the Rivest, Shamir, and Adelman (RSA) public keys of the peer that are stored on your router. • The output is extended to show the keyring to which the public key belongs.

Clearing Security Associations

The following **clear** commands allow you to clear SAs.

SUMMARY STEPS

- **enable**
- **clear crypto sa** [**counters** | **map** *map-name* | **peer** [**vrf** *fvrf-name*] *address* | **spi** *address* {**ah** | **esp**} *spi* | **vrf** *ivrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto sa** [**counters	map** *map-name* **| peer** [**vrf** *fvrf-name*] *address* **| spi** *address* {**ah	esp**} *spi* **| vrf** *ivrf-name*] **Example:** `Router# clear crypto sa vrf VPN1`	Clears the IPSec security associations (SAs).

Troubleshooting VRF-Aware IPSec

To troubleshoot VRF-Aware IPSec, use the following **debug** commands:

SUMMARY STEPS

1. **enable**
2. **debug crypto ipsec**
3. **debug crypto isakmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto ipsec** **Example:** `Router# debug crypto ipsec`	Displays IP security (IPSec) events.
Step 3	**debug crypto isakmp** **Example:** `Router(config)# debug crypto isakmp`	Displays messages about Internet Key Exchange (IKE) events.

Debug Examples for VRF-Aware IPSec

The following sample debug outputs are for a VRF-aware IPSec configuration:

IPSec PE

```
Router# debug crypto ipsec

Crypto IPSEC debugging is on
IPSEC-PE#debug crypto isakmp
Crypto ISAKMP debugging is on
IPSEC-PE#debug crypto isakmp d
04:31:28: ISAKMP (0:12): purging SA., sa=6482B354, delme=6482B354
04:31:28: ISAKMP: Unlocking IKE struct 0x63C142F8 for declare_sa_dead(), count 0
IPSEC-PE#debug crypto isakmp detail
Crypto ISAKMP internals debugging is on
IPSEC-PE#
IPSEC-PE#
IPSEC-PE#
04:32:07: ISAKMP: Deleting peer node by peer_reap for 10.1.1.1: 63C142F8
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B DC887D4E
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.68.1.1
04:32:55: ISAKMP cookie AA8F7B41 49A60E88
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B DBC8E125
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 B4BDB5B7
04:32:55: ISAKMP (0:0): received packet from 10.1.1.1 dport 500 sport 500 Global (N) NEW
SA
04:32:55: ISAKMP: local port 500, remote port 500
04:32:55: ISAKMP: hash from 729FA94 for 619 bytes
04:32:55: ISAKMP: Packet hash:
64218CC0:                   B91E2C70 095A1346          9.,p.Z.F
64218CD0: 0EDB4CA6 8A46784F B314FD3B 00        .[L&.FxO.};.
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:32:55: ISAKMP cookie AA8F7B41 F7ACF384
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:32:55: ISAKMP cookie AA8F7B41 0C07C670
04:32:55: ISAKMP: insert sa successfully sa = 6482B354
04:32:55: ISAKMP (0:13): processing SA payload. message ID = 0
```

```
04:32:55: ISAKMP (0:13): processing ID payload. message ID = 0
04:32:55: ISAKMP (0:13): peer matches vpn2-ra profile
04:32:55: ISAKMP: Looking for a matching key for 10.1.1.1 in default
04:32:55: ISAKMP: Created a peer struct for 10.1.1.1, peer port 500
04:32:55: ISAKMP: Locking peer struct 0x640BBB18, IKE refcount 1 for
crypto_ikmp_config_initialize_sa
04:32:55: ISAKMP (0:13): Setting client config settings 648252B0
04:32:55: ISAKMP (0:13): (Re)Setting client xauth list  and state
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 157 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v3
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 123 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v2
04:32:55: ISAKMP (0:13) Authentication by xauth preshared
04:32:55: ISAKMP (0:13): Checking ISAKMP transform 1 against priority 1 policy
04:32:55: ISAKMP:      encryption 3DES-CBC
04:32:55: ISAKMP:      hash SHA
04:32:55: ISAKMP:      default group 2
04:32:55: ISAKMP:      auth XAUTHInitPreShared
04:32:55: ISAKMP:      life type in seconds
04:32:55: ISAKMP:      life duration (VPI) of  0x0 0x20 0xC4 0x9B
04:32:55: ISAKMP (0:13): atts are acceptable. Next payload is 3
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 157 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v3
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 123 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v2
04:32:55: ISAKMP (0:13): processing KE payload. message ID = 0
04:32:55: ISAKMP (0:13): processing NONCE payload. message ID = 0
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID is DPD
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 175 mismatch
04:32:55: ISAKMP (0:13): vendor ID is XAUTH
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): claimed IOS but failed authentication
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID is Unity
04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_AM_EXCH
04:32:55: ISAKMP (0:13): Old State = IKE_READY  New State = IKE_R_AM_AAA_AWAIT

04:32:55: ISAKMP cookie gen for src 11.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 7AE6E1DF
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 4 AA 31 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP: got callback 1
04:32:55: ISAKMP (0:13): SKEYID state generated
04:32:55: ISAKMP: Unity/DPD ID: vendor_id_payload:
        next: 0xD, reserved: 0x0, len 0x14
04:32:55: ISAKMP: Unity/DPD ID payload dump:
63E66D70:                            0D000014              ....
63E66D80: 12F5F28C 457168A9 702D9FE2 74CC0100  .ur.Eqh)p-.btL..
63E66D90: 00                                   .
04:32:55: ISAKMP: Unity/DPD ID: vendor_id_payload:
        next: 0xD, reserved: 0x0, len 0x14
04:32:55: ISAKMP: Unity/DPD ID payload dump:
63E66D90: 0D000014 AFCAD713 68A1F1C9 6B8696FC  ..../JW.h!qIk..|
63E66DA0: 77570100 00                         wW...
04:32:55: ISAKMP (0:13): constructed NAT-T vendor-03 ID
04:32:55: ISAKMP (0:13): SA is doing pre-shared key authentication plus XAUTH using id
type ID_IPV4_ADDR
```

```
04:32:55: ISAKMP (13): ID payload
        next-payload : 10
        type         : 1
        addr         : 172.16.1.1
        protocol     : 17
        port         : 0
        length       : 8
04:32:55: ISAKMP (13): Total payload length: 12
04:32:55: ISAKMP (0:13): constructed HIS NAT-D
04:32:55: ISAKMP (0:13): constructed MINE NAT-D
04:32:55: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
AG_INIT_EXCH
04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, PRESHARED_KEY_REPLY
04:32:55: ISAKMP (0:13): Old State = IKE_R_AM_AAA_AWAIT  New State = IKE_R_AM2

04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B D99DA70D
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 9C69F917
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 5 21FF 1 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B 00583224
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 C1B006EE
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 5 21FF 1 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
AG_INIT_EXCH
04:32:55: ISAKMP: hash from 7003A34 for 132 bytes
04:32:55: ISAKMP: Packet hash:
64218CC0:                   D1202D99 2BB49D38          Q -.+4.8
64218CD0: B8FBB1BE 7CDC67D7 4E26126C 63        8{1>|\gWN&.lc
04:32:55: ISAKMP (0:13): processing HASH payload. message ID = 0
04:32:55: ISAKMP:received payload type 17
04:32:55: ISAKMP (0:13): Detected NAT-D payload
04:32:55: ISAKMP (0:13): recalc my hash for NAT-D
04:32:55: ISAKMP (0:13): NAT match MINE hash
04:32:55: ISAKMP:received payload type 17
04:32:55: ISAKMP (0:13): Detected NAT-D payload
04:32:55: ISAKMP (0:13): recalc his hash for NAT-D
04:32:55: ISAKMP (0:13): NAT match HIS hash
04:32:55: ISAKMP (0:13): processing NOTIFY INITIAL_CONTACT protocol 1
        spi 0, message ID = 0, sa = 6482B354
04:32:55: ISAKMP (0:13): Process initial contact,
bring down existing phase 1 and 2 SA's with local 172.16.1.1 remote 10.1.1.1 remote port
500
04:32:55: ISAKMP (0:13): returning IP addr to the address pool
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 05D315C5
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B 041A85A6
04:32:55: ISAKMP (0:13): SA has been authenticated with 10.1.1.1
04:32:55: ISAKMP: Trying to insert a peer 172.16.1.1/10.1.1.1/500/,  and inserted
successfully.
04:32:55: ISAKMP: set new node -803402627 to CONF_XAUTH
04:32:55: IPSEC(key_engine): got a queue event...
04:32:55: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) QM_IDLE
04:32:55: ISAKMP (0:13): purging node -803402627
04:32:55: ISAKMP: Sending phase 1 responder lifetime 86400

04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_AM_EXCH
```

```
04:32:55: ISAKMP (0:13): Old State = IKE_R_AM2  New State = IKE_P1_COMPLETE

04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.168.1.1
04:32:55: ISAKMP cookie AA8F7B41 25EEF256
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP (0:13): Need XAUTH
04:32:55: ISAKMP (0:13): Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
04:32:55: ISAKMP (0:13): Old State = IKE_P1_COMPLETE  New State =
IKE_XAUTH_AAA_START_LOGIN_AWAIT

04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 2CCFA491
04:32:55: ISAKMP:        isadb_post_process_list: crawler: B 27FF 12 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP: got callback 1
04:32:55: ISAKMP: set new node -1447732198 to CONF_XAUTH
04:32:55: ISAKMP/xauth: request attribute XAUTH_USER_NAME_V2
04:32:55: ISAKMP/xauth: request attribute XAUTH_USER_PASSWORD_V2
04:32:55: ISAKMP (0:13): initiating peer config to 10.1.1.1. ID = -1447732198
04:32:55: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
CONF_XAUTH
04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, IKE_AAA_START_LOGIN
04:32:55: ISAKMP (0:13): Old State = IKE_XAUTH_AAA_START_LOGIN_AWAIT  New State =
IKE_XAUTH_REQ_SENT

04:33:00: ISAKMP (0:13): retransmitting phase 2 CONF_XAUTH    -1447732198 ...
04:33:00: ISAKMP (0:13): incrementing error counter on sa: retransmit phase 2
04:33:00: ISAKMP (0:13): incrementing error counter on sa: retransmit phase 2
04:33:00: ISAKMP (0:13): retransmitting phase 2 -1447732198 CONF_XAUTH
04:33:00: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
CONF_XAUTH
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 124D4618
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 B0C91917
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 0E294692
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 091A7695
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
CONF_XAUTH
04:33:03: ISAKMP: hash from 7292D74 for 92 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   84A1AF24 5D92B116          .!/$].1.
64218CD0: FC2C6252 A472C5F8 152AC860 63        |,bR$rEx.*H`c
04:33:03: ISAKMP (0:13): processing transaction payload from 11.1.1.1. message ID =
-1447732198
04:33:03: ISAKMP: Config payload REPLY
04:33:03: ISAKMP/xauth: reply attribute XAUTH_USER_NAME_V2
04:33:03: ISAKMP/xauth: reply attribute XAUTH_USER_PASSWORD_V2
04:33:03: ISAKMP (0:13): deleting node -1447732198 error FALSE reason "done with xauth
request/reply exchange"
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_CFG_REPLY
04:33:03: ISAKMP (0:13): Old State = IKE_XAUTH_REQ_SENT  New State =
IKE_XAUTH_AAA_CONT_LOGIN_AWAIT
```

```
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 A1B3E684
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 12 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP: got callback 1
04:33:03: ISAKMP: set new node 524716665 to CONF_XAUTH
04:33:03: ISAKMP (0:13): initiating peer config to 10.1.1.1. ID = 524716665
04:33:03: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
CONF_XAUTH
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, IKE_AAA_CONT_LOGIN
04:33:03: ISAKMP (0:13): Old State = IKE_XAUTH_AAA_CONT_LOGIN_AWAIT  New State =
IKE_XAUTH_SET_SENT
004:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 5C83A09D
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 2BEBEFD4
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B DA00A46B
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 FDD27773
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
CONF_XAUTH
04:33:03: ISAKMP: hash from 7292A34 for 68 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   5034B99E B8BA531F          P49.8:S.
64218CD0: 6267B8BD F3006989 DC118796 63        bg8=s.i.\...c
04:33:03: ISAKMP (0:13): processing transaction payload from 11.1.1.1. message ID =
524716665
04:33:03: ISAKMP: Config payload ACK
04:33:03: ISAKMP (0:13):        XAUTH ACK Processed
04:33:03: ISAKMP (0:13): deleting node 524716665 error FALSE reason "done with
transaction"
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_CFG_ACK
04:33:03: ISAKMP (0:13): Old State = IKE_XAUTH_SET_SENT  New State = IKE_P1_COMPLETE

04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 E0BB50E9
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP (0:13): Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
04:33:03: ISAKMP (0:13): Old State = IKE_P1_COMPLETE  New State = IKE_P1_COMPLETE

04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 7794EF6E
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 C035AAE5
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B F1FCC25A
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 31744F44
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
```

```
04:33:03:        crawler his_cookie E46E088D F207FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
QM_IDLE
04:33:03: ISAKMP: set new node -1639992295 to QM_IDLE
04:33:03: ISAKMP: hash from 7293A74 for 100 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   9D7DF4DF FE3A6403          .}t_~:d.
64218CD0: 3F1D1C59 C5D138CE 50289B79 07        ?..YEQ8NP(.y.
04:33:03: ISAKMP (0:13): processing transaction payload from 10.1.1.1. message ID =
-1639992295
04:33:03: ISAKMP: Config payload REQUEST
04:33:03: ISAKMP (0:13): checking request:
04:33:03: ISAKMP:    IP4_ADDRESS
04:33:03: ISAKMP:    IP4_NETMASK
04:33:03: ISAKMP:    IP4_DNS
04:33:03: ISAKMP:    IP4_DNS
04:33:03: ISAKMP:    IP4_NBNS
04:33:03: ISAKMP:    IP4_NBNS
04:33:03: ISAKMP:    SPLIT_INCLUDE
04:33:03: ISAKMP:    DEFAULT_DOMAIN
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_CFG_REQUEST
04:33:03: ISAKMP (0:13): Old State = IKE_P1_COMPLETE  New State =
IKE_CONFIG_AUTHOR_AAA_AWAIT

04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 B02E0D67
04:33:03: ISAKMP:        isadb_post_process_list: crawler: C 27FF 12 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP: got callback 1
04:33:03: ISAKMP (0:13): attributes sent in message:
04:33:03:         Address: 10.2.0.0
04:33:03: ISAKMP (0:13): allocating address 10.4.1.4
04:33:03: ISAKMP: Sending private address: 10.4.1.4
04:33:03: ISAKMP: Sending DEFAULT_DOMAIN default domain name: vpn2.com
04:33:03: ISAKMP (0:13): responding to peer config from 10.1.1.1. ID = -1639992295
04:33:03: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
CONF_ADDR
04:33:03: ISAKMP (0:13): deleting node -1639992295 error FALSE reason ""
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, IKE_AAA_GROUP_ATTR
04:33:03: ISAKMP (0:13): Old State = IKE_CONFIG_AUTHOR_AAA_AWAIT  New State =
IKE_P1_COMPLETE

04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 881D5411
04:33:03: ISAKMP cookie gen for src 11.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 6FD82541
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 8A94C1BE
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 F3BA766D
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F207FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
QM_IDLE
04:33:03: ISAKMP: set new node 17011691 to QM_IDLE
04:33:03: ISAKMP: hash from 70029F4 for 540 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   AFBA30B2 55F5BC2D          /:02Uu<-
64218CD0: 3A86B1C9 00D2F5BA 77BF5589 07        :.1I.Ru:w?U..
```

```
04:33:03: ISAKMP (0:13): processing HASH payload. message ID = 17011691
04:33:03: ISAKMP (0:13): processing SA payload. message ID = 17011691
04:33:03: ISAKMP (0:13): Checking IPSec proposal 1
04:33:03: ISAKMP: transform 1, ESP_3DES
04:33:03: ISAKMP:   attributes in transform:
04:33:03: ISAKMP:      encaps is 1
04:33:03: ISAKMP:      SA life type in seconds
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x20 0xC4 0x9B
04:33:03: ISAKMP:      SA life type in kilobytes
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
04:33:03: ISAKMP:      authenticator is HMAC-SHA
04:33:03: ISAKMP (0:13): atts are acceptable.
04:33:03: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-sha-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn1, kei->ivrf = vpn2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn2, kei->ivrf = vpn2
04:33:03: IPSEC(validate_transform_proposal): transform proposal not supported for
identity:
    {esp-3des esp-sha-hmac }
04:33:03: ISAKMP (0:13): IPSec policy invalidated proposal
04:33:03: ISAKMP (0:13): Checking IPSec proposal 2
04:33:03: ISAKMP: transform 1, ESP_3DES
04:33:03: ISAKMP:   attributes in transform:
04:33:03: ISAKMP:      encaps is 1
04:33:03: ISAKMP:      SA life type in seconds
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x20 0xC4 0x9B
04:33:03: ISAKMP:      SA life type in kilobytes
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
04:33:03: ISAKMP:      authenticator is HMAC-MD5
04:33:03: ISAKMP (0:13): atts are acceptable.
04:33:03: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-md5-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn1, kei->ivrf = vpn2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn2, kei->ivrf = vpn2
04:33:03: ISAKMP (0:13): processing NONCE payload. message ID = 17011691
04:33:03: ISAKMP (0:13): processing ID payload. message ID = 17011691
04:33:03: ISAKMP (0:13): processing ID payload. message ID = 17011691
04:33:03: ISAKMP (0:13): asking for 1 spis from ipsec
04:33:03: ISAKMP (0:13): Node 17011691, Input = IKE_MESG_FROM_PEER, IKE_QM_EXCH
04:33:03: ISAKMP (0:13): Old State = IKE_QM_READY  New State = IKE_QM_SPI_STARVE
04:33:03: IPSEC(key_engine): got a queue event...
04:33:03: IPSEC(spi_response): getting spi 2749516541 for SA
        from 172.18.1.1     to 10.1.1.1        for prot 3
04:33:03: ISAKMP: received ke message (2/1)
04:33:04: ISAKMP (13): ID payload
        next-payload : 5
        type         : 1
        addr         : 10.4.1.4
        protocol     : 0
        port         : 0
04:33:04: ISAKMP (13): ID payload
        next-payload : 11
        type         : 4
        addr         : 0.0.0.0
```

```
        protocol     : 0
        port         : 0
04:33:04: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) QM_IDLE
04:33:04: ISAKMP (0:13): Node 17011691, Input = IKE_MESG_FROM_IPSEC, IKE_SPI_REPLY
04:33:04: ISAKMP (0:13): Old State = IKE_QM_SPI_STARVE  New State = IKE_QM_R_QM2
04:33:04: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:04: ISAKMP cookie 3123100B 93DE46D2
04:33:04: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:04: ISAKMP cookie AA8F7B41 088A0A16
04:33:04: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:04:       crawler my_cookie AA8F7B41 F7ACF384
04:33:04:        crawler his_cookie E46E088D F227FE4D
04:33:04: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:04: ISAKMP cookie 3123100B A8F23F73
04:33:04: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:04: ISAKMP cookie AA8F7B41 93D8D879
04:33:04: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:04:       crawler my_cookie AA8F7B41 F7ACF384
04:33:04:        crawler his_cookie E46E088D F227FE4D
04:33:04: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
QM_IDLE
04:33:04: ISAKMP: hash from 7290DB4 for 60 bytes
04:33:04: ISAKMP: Packet hash:
64218CC0:                   4BB45A92 7181A2F8          K4Z.q."x
64218CD0: 73CC12F8 091875C0 054F77CD 63        sL.x..u@.OwMc
04:33:04: ISAKMP: Locking peer struct 0x640BBB18, IPSEC refcount 1 for for stuff_ke
04:33:04: ISAKMP (0:13): Creating IPSec SAs
04:33:04:         inbound SA from 10.1.1.1 to 172.18.1.1 (f/i)  0/ 2
        (proxy 10.4.1.4 to 0.0.0.0)
04:33:04:         has spi 0xA3E24AFD and conn_id 5127 and flags 2
04:33:04:         lifetime of 2147483 seconds
04:33:04:         lifetime of 4608000 kilobytes
04:33:04:         has client flags 0x0
04:33:04:         outbound SA from 172.18.1.1     to 10.1.1.1        (f/i)  0/ 2 (proxy
0.0.0.0         to 10.4.1.4       )
04:33:04:         has spi 1343294712 and conn_id 5128 and flags A
04:33:04:         lifetime of 2147483 seconds
04:33:04:         lifetime of 4608000 kilobytes
04:33:04:         has client flags 0x0
04:33:04: ISAKMP (0:13): deleting node 17011691 error FALSE reason "quick mode done
(await)"
04:33:04: ISAKMP (0:13): Node 17011691, Input = IKE_MESG_FROM_PEER, IKE_QM_EXCH
04:33:04: ISAKMP (0:13): Old State = IKE_QM_R_QM2  New State = IKE_QM_PHASE2_COMPLETE
04:33:04: IPSEC(key_engine): got a queue event...
04:33:04: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-md5-hmac ,
    lifedur= 2147483s and 4608000kb,
    spi= 0xA3E24AFD(2749516541), conn_id= 5127, keysize= 0, flags= 0x2
04:33:04: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-md5-hmac ,
    lifedur= 2147483s and 4608000kb,
    spi= 0x50110CF8(1343294712), conn_id= 5128, keysize= 0, flags= 0xA
04:33:04: IPSEC(kei_proxy): head = ra, map->ivrf = vpn1, kei->ivrf = vpn2
04:33:04: IPSEC(kei_proxy): head = ra, map->ivrf = vpn2, kei->ivrf = vpn2
04:33:04: IPSEC(rte_mgr): VPN Route Added 10.4.1.4 255.255.255.255 via 10.1.1.1 in vpn2
04:33:04: IPSEC(add mtree): src 0.0.0.0, dest 10.4.1.4, dest_port 0

04:33:04: IPSEC(create_sa): sa created,
```

```
  (sa) sa_dest= 172.18.1.1, sa_prot= 50,
    sa_spi= 0xA3E24AFD(2749516541),
    sa_trans= esp-3des esp-md5-hmac , sa_conn_id= 5127
04:33:04: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.1.1.1, sa_prot= 50,
    sa_spi= 0x50110CF8(1343294712),
    sa_trans= esp-3des esp-md5-hmac , sa_conn_id= 5128
04:33:53: ISAKMP (0:13): purging node -1639992295
04:33:54: ISAKMP (0:13): purging node 17011691
```

Configuration Examples for VRF-Aware IPSec

The following examples show how to configure VRF-Aware IPSec:

Static IPSec-to-MPLS VPN Example

The following sample shows a static configuration that maps IPSec tunnels to MPLS VPNs. The configurations map IPSec tunnels to MPLS VPNs “VPN1” and “VPN2.” Both of the IPSec tunnels terminate on a single public-facing interface.

IPSec PE Configuration

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip vrf vpn2
 rd 101:1
 route-target export 101:1
 route-target import 101:1
!
crypto keyring vpn1
  pre-shared-key address 172.16.1.1 key vpn1
crypto keyring vpn2
  pre-shared-key address 10.1.1.1 key vpn2
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp profile vpn1
 vrf vpn1
 keyring vpn1
 match identity address 172.16.1.1 255.255.255.255
!
crypto isakmp profile vpn2
 vrf vpn2
 keyring vpn2
 match identity address 10.1.1.1 255.255.255.255
!
```

```
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
crypto ipsec transform-set vpn2 esp-3des esp-md5-hmac
!
crypto map crypmap 1 ipsec-isakmp
 set peer 172.16.1.1
 set transform-set vpn1
 set isakmp-profile vpn1
 match address 101
crypto map crypmap 3 ipsec-isakmp
 set peer 10.1.1.1
 set transform-set vpn2
 set isakmp-profile vpn2
 match address 102
!
interface Ethernet1/1
 ip address 172.17.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map crypmap
!
ip route 172.16.1.1 255.255.255.255 172.168.1.2
ip route 10.1.1.1 255.255.255.255 172.18.1.2
ip route vrf vpn1 10.2.0.0 255.255.0.0 172.18.1.2 global
ip route vrf vpn2 10.2.0.0 255.255.0.0 172.18.1.2 global
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
access-list 102 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
```

IPSec Customer Provided Edge (CPE) Configuration for VPN1

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
crypto isakmp key vpn1 address 172.18.1.1
!
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map vpn1 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn1
 match address 101
!
interface FastEthernet1/0
 ip address 172.16.1.1 255.255.255.0
 crypto map vpn1
!
interface FastEthernet1/1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
```

IPSec CPE Configuration for VPN2

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp key vpn2 address 172.18.1.1
```

```
!
!
crypto ipsec transform-set vpn2 esp-3des esp-md5-hmac
!
crypto map vpn2 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn2
 match address 101
!
interface FastEthernet0
 ip address 10.1.1.1 255.255.255.0
 crypto map vpn2
!
interface FastEthernet1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
```

IPSec-to-MPLS VPN Using RSA Encryption Example

The following example shows an IPSec-to-MPLS configuration using RSA encryption:

PE Router Configuration

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
crypto isakmp policy 10
 authentication rsa-encr
!
crypto keyring vpn1
 rsa-pubkey address 172.16.1.1 encryption
  key-string
   305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00DBF381 00DDECC8
   DC4AA490 40320C52 9912D876 EB36717C 63DCA95C 7E5EC02A 84F276CE 292B42D7
   D664F324 3726F4E0 39D33093 ECB81B95 482511A5 F064C4B3 D5020301 0001
   quit
!
crypto isakmp profile vpn1
 vrf vpn1
 keyring vpn1
 match identity address 172.16.1.1 255.255.255.255
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map crypmap 1 ipsec-isakmp
 set peer 172.16.1.1
 set transform-set vpn1
 set isakmp-profile vpn1
 match address 101
!
interface Ethernet1/1
 ip address 172.17.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map crypmap
!
ip route 172.16.1.1 255.255.255.255 172.18.1.2
```

```
ip route vrf vpn1 10.2.0.0 255.255.0.0 172.18.1.2 global
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
```

IPSec CPE Configuration for VPN1

```
crypto isakmp policy 10
 authentication rsa-encr
!
crypto key pubkey-chain rsa
 addressed-key 172.18.1.1 encryption
 key-string
  3082011B 300D0609 2A864886 F70D0101 01050003 82010800 30820103 0281FB00
  C90CC78A 6002BDBA 24683396 B7D7877C 16D08C47 E00C3C10 63CF13BC 4E09EA23
  92EB8A48 4113F5A4 8796C8BE AD7E2DC1 3B0742B6 7118CE7C 1B0E21D1 AA9724A4
  4D74FCEA 562FF225 A2B11F18 E53C4415 61C3B741 3A06E75D B4F9102D 6163EE40
  16C68FD7 6532F660 97B59118 9C8DE3E5 4E2F2925 BBB87FCB 95223D4E A5E362DB
  215CB35C 260080805 17BBE1EF C3050E13 031F3D5B 5C22D16C FC8B1EC5 074F07A5
  D050EC80 7890D9C5 EC20D6F0 173FE2BA 89F5B5F9 2EADC9A6 D461921E 3D5B60016
  ABB8B6B9 E2124A21 93F0E4AE B487461B E7F1F1C4 032A0B0E 80DC3E15 CB268EC9
  5D76B9BD 3C78CB75 CE9F68C6 484D6573 CBC3EB59 4B5F3999 8F9D0203 010001
  quit
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map vpn1 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn1
 match address 101
!
interface FastEthernet1/0
 ip address 172.16.1.1 255.255.255.0
 crypto map vpn1
!
interface FastEthernet1/1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
```

IPSec-to-MPLS VPN with RSA Signatures Example

The following shows an IPSec-to-MPLS VPN configuration using RSA signatures:

PE Router Configuration

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
crypto ca trustpoint bombo
 enrollment url http://172.31.68.59:80
 crl optional
!
crypto ca certificate chain bombo
 certificate 03C0
 308203BF 308202A7 A0030201 02020203 C0300D06 092A8648 86F70D01 01050500
 . . .
 quit
 certificate ca 01
 30820379 30820261 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
```

```
 . . .
 quit
!
crypto isakmp profile vpn1
 vrf vpn1
 ca trust-point bombo
 match identity address 172.16.1.1 255.255.255.255
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map crypmap 1 ipsec-isakmp
 set peer 172.16.1.1
 set transform-set vpn1
 set isakmp-profile vpn1
 match address 101
!
interface Ethernet1/1
 ip address 172.31.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map crypmap
!
ip route 172.16.1.1 255.255.255.255 172.18.1.2
ip route vrf vpn1 10.2.0.0 255.255.0.0 172.18.1.2 global
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
!
```

IPSec CPE Configuration for VPN1

```
crypto ca trustpoint bombo
 enrollment url http://172.31.68.59:80
 crl optional
 !
crypto ca certificate chain bombo
 certificate 03BF
  308203BD 308202A5 A0030201 02020203 BF300D06 092A8648 86F70D01 01050500
  . . .
  quit
 certificate ca 01
  30820379 30820261 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  . . .
  quit
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map vpn1 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn1
 match address 101
!
interface FastEthernet1/0
 ip address 172.16.1.1 255.255.255.0
 crypto map vpn1
!
interface FastEthernet1/1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
```

IPSec Remote Access-to-MPLS VPN Example

The following shows an IPSec remote access-to-MPLS VPN configuration. The configuration maps IPSec tunnels to MPLS VPNs. The IPSec tunnels terminate on a single public-facing interface.

PE Router Configuration

```
aaa new-model
!
aaa group server radius vpn1
 server-private 10.1.1.1 auth-port 1645 acct-port 1646 timeout 5 retransmit 3 key vpn1
!
aaa group server radius vpn2
 server-private 10.1.1.1 auth-port 1645 acct-port 1646 timeout 5 retransmit 3 key vpn2
!
aaa authorization network aaa-list group radius
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip vrf vpn2
 rd 101:1
 route-target export 101:1
 route-target import 101:1
!
crypto isakmp profile vpn1-ra
   vrf vpn1
   match identity group vpn1-ra
   client authentication list vpn1
   isakmp authorization list aaa-list
   client configuration address initiate
   client configuration address respond
crypto isakmp profile vpn2-ra
   vrf vpn2
   match identity group vpn2-ra
   client authentication list vpn2
   isakmp authorization list aaa-list
   client configuration address initiate
   client configuration address respond
!
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
crypto ipsec transform-set vpn2 esp-3des esp-md5-hmac
!
crypto dynamic-map vpn1 1
 set transform-set vpn1
 set isakmp-profile vpn1-ra
 reverse-route
!
crypto dynamic-map vpn2 1
 set transform-set vpn2
 set isakmp-profile vpn2-ra
 reverse-route
!
!
crypto map ra 1 ipsec-isakmp dynamic vpn1
crypto map ra 2 ipsec-isakmp dynamic vpn2
!
interface Ethernet1/1
 ip address 172.17.1.1 255.255.0.0
 tag-switching ip
```

```
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map ra
!
ip local pool vpn1-ra 10.4.1.1 10.4.1.254 group vpn1-ra
ip local pool vpn2-ra 10.4.1.1 10.4.1.254 group vpn2-ra
!
```

Upgrade from Previous Versions of the Cisco Network-Based IPSec VPN Solution

The VRF-Aware IPSec feature in the Cisco network-based IPSec VPN solution release 1.5 requires that you change your existing configurations.

The sample configurations that follow indicate the changes you must make to your existing configurations. These samples include the following:

- Site-to-Site Configuration Upgrade, page 28
- Remote Access Configuration Upgrade, page 29
- Combination Site-to-Site and Remote Access Configuration Upgrade, page 31

Site-to-Site Configuration Upgrade

The following configurations show the changes that are necessary for a site-to-site configuration upgrade from a previous version of the network-based IPSec VPN solution to the Cisco network-based IPSec VPN solution release 1.5:

Previous Version Site-to-Site Configuration

```
crypto isakmp key VPN1 address 172.21.25.74
 crypto isakmp key VPN2 address 172.21.21.74
!
 crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 match address 101
!
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 match address 102
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
  encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

New Version Site-to-Site Configuration

The following is an upgraded version of the same site-to-site configuration to the Cisco network-based IPSec VPN solution release 1.5 solution:

Note You must change to keyrings. The VRF-Aware IPSec feature requires that keys be associated with a VRF if the IKE local endpoint is in the VRF.

```
crypto keyring VPN1-KEYS vrf VPN1
 pre-shared-key address 172.21.25.74 key VPN1
!
 crypto keyring VPN2-KEYS vrf VPN2
  pre-shared-key address 172.21.21.74 key VPN2
!
 crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
  set transform-set VPN1
 match address 101
!
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
  set transform-set VPN2
 match address 102
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

Remote Access Configuration Upgrade

The following configurations show the changes that are necessary for a remote access configuration upgrade from a previous version of the network-based IPSec VPN solution to the Cisco network-based IPSec VPN solution release 1.5:

Previous Version Remote Access Configuration

```
crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
 crypto dynamic-map VPN1-RA 1
```

```
  set transform-set VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
  reverse-route
!
!
 crypto map VPN1 client authentication list VPN1-RA-LIST
 crypto map VPN1 isakmp authorization list VPN1-RA-LIST
 crypto map VPN1 client configuration address initiate
 crypto map VPN1 client configuration address respond
 crypto map VPN1 10 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 client authentication list VPN2-RA-LIST
 crypto map VPN2 isakmp authorization list VPN2-RA-LIST
 crypto map VPN2 client configuration address initiate
 crypto map VPN2 client configuration address respond
 crypto map VPN2 10 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
  encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

New Version Remote Access Configuration

In the following instance, there is no upgrade; it is recommended that you change to the following configuration:

```
crypto isakmp client configuration group VPN1-RA-GROUP
  key VPN1-RA
  pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto isakmp profile VPN1-RA
 match identity group VPN1-RA-GROUP
  client authentication list VPN1-RA-LIST
 isakmp authorization list VPN1-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto isakmp profile VPN2-RA
 match identity group VPN2-RA-GROUP
 client authentication list VPN2-RA-LIST
 isakmp authorization list VPN2-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
 crypto dynamic-map VPN1-RA 1
```

```
 set transform-set VPN1-RA
 set isakmp-profile VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 set isakmp-profile VPN2-RA
 reverse-route
!
 crypto map VPN1 10 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 10 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

Combination Site-to-Site and Remote Access Configuration Upgrade

The following configurations show the changes that are necessary for a site-to-site and remote access configuration upgrade from a previous version of the network-based IPSec VPN solution to the Cisco network-based IPSec VPN solution release 1.5:

Previous Version Site-to-Site and Remote Access Configuration

```
crypto isakmp key VPN1 address 172.21.25.74 no-xauth
crypto isakmp key VPN2 address 172.21.21.74 no-xauth
!
 crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
 crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
 crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 reverse-route
!
 crypto map VPN1 client authentication list VPN1-RA-LIST
 crypto map VPN1 isakmp authorization list VPN1-RA-LIST
 crypto map VPN1 client configuration address initiate
```

```
 crypto map VPN1 client configuration address respond
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 match address 101
 crypto map VPN1 20 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 client authentication list VPN2-RA-LIST
 crypto map VPN2 isakmp authorization list VPN2-RA-LIST
 crypto map VPN2 client configuration address initiate
 crypto map VPN2 client configuration address respond
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 match address 102
 crypto map VPN2 20 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

New Version Site-to-Site and Remote Access Configuration

You must upgrade to this configuration:

Note For site-to-site configurations that do not require XAUTH, configure an ISAKMP profile without XAUTH configuration. For remote access configurations that require XAUTH, configure an ISAKMP profile with XAUTH.

```
crypto keyring VPN1-KEYS vrf VPN1
 pre-shared-key address 172.21.25.74 key VPN1
!
 crypto keyring VPN2-KEYS vrf VPN2
 pre-shared-key address 172.21.21.74 key VPN2
!
 crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto isakmp profile VPN1
 keyring VPN1-KEYS
 match identity address 172.21.25.74 VPN1
!
 crypto isakmp profile VPN2
 keyring VPN2-KEYS
 match identity address 172.21.21.74 VPN2
!
 crypto isakmp profile VPN1-RA
```

```
 match identity group VPN1-RA-GROUP
 client authentication list VPN1-RA-LIST
 isakmp authorization list VPN1-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto isakmp profile VPN2-RA
 match identity group VPN2-RA-GROUP
 client authentication list VPN2-RA-LIST
 isakmp authorization list VPN2-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
 crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
 crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 set isakmp-profile VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 set isakmp-profile VPN2-RA
 reverse-route
!
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 set isakmp-profile VPN1
 match address 101
 crypto map VPN1 20 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 set isakmp-profile VPN2
 match address 102
 crypto map VPN2 20 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

Additional References

For additional information related to VRF-Aware IPSec, refer to the following references:

Related Documents

Related Topic	Document Title
IPSec configuration tasks	The chapter "Configuring Security for VPNs with IPSec" in the *Cisco IOS Security Configuration Guide*
IPSec commands	*Cisco IOS Security Command Reference*
IKE Phase 1 and Phase 2, agressive mode, and main mode	The chapter "Configuring Internet Key Exchange for IPSec VPNs" in the *Cisco IOS Security Configuration Guide*
IKE dead peer detection	*Easy VPN Server*

Standards

Standards[1]	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

1. Not all supported standards are listed.

MIBs

MIBs[1]	MIBs Link
• No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

1. Not all supported MIBs are listed.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[1]	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and lots more. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module:

New Commands

- **address**
- **ca trust-point**
- **client authentication list**
- **client configuration address**
- **crypto isakmp profile**
- **crypto keyring**
- **crypto map isakmp-profile**
- **initiate-mode**
- **isakmp authorization list**
- **keepalive (isakmp profile)**

- **keyring**
- **key-string**
- **match identity**
- **no crypto xauth**
- **pre-shared-key**
- **quit**
- **rsa-pubkey**
- **self-identity**
- **serial-number**
- **set isakmp-profile**
- **show crypto isakmp key**
- **show crypto isakmp profile**
- **vrf**

Modified Commands

- **clear crypto sa**
- **crypto isakmp peer**
- **crypto map isakmp-profile**
- **show crypto dynamic-map**
- **show crypto ipsec sa**
- **show crypto isakmp sa**
- **show crypto map (IPSec)**

For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Glossary

CA—certification authority. CA is an entity that issues digital certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate.

CLI—command-line-interface. CLI is an interface that allows the user to interact with the operating system by entering commands and optional arguments. The UNIX operating system and DOS provide CLIs.

client—Corresponding IPSec IOS peer of the UUT in the Multi Protocol Label Switching (MPLS) network.

dead peer—IKE peer that is no longer reachable.

DN—Distinguished Name. A DN is the global, authoritative name of an entry in the Open System Interconnection (OSI Directory [X.500]).

FQDN—fully qualified domain name. A FQDN is the full name of a system rather than just its host name. For example, aldebaran is a host name, and aldebaran.interop.com is an FQDN.

FR—Frame Relay. FR is an industry-standard, switch-data-link-layer protocol that handles multiple virtual circuits using high-level data link (HDLC) encapsulation between connected devices. Frame Relay is more efficient than X.25, the protocol for which it generally is considered a replacement.

FVRF—Front Door Virtual Routing and Forwarding (VRF) repository. FVRF is the VRF used to route the encrypted packets to the peer.

IDB—Interface descriptor block. An IDB subblock is an area of memory that is private to an application. This area stores private information and states variables that an application wants to associate with an IDB or an interface. The application uses the IDB to register a pointer to its subblock, not to the contents of the subblock itself.

IKE—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPSec) that require keys. Before any IPSec traffic can be passed, each router, firewall, and host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

IKE keepalive—Bidirectional mechanism for determining the liveliness of an IKE peer.

IPSec—Security protocol for IP.

IVRF—Inside Virtual Routing and Forwarding. IVRF is the VRF of the plaintext packets.

MPLS—Multiprotocol Label Switching. MPLS is a switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

RSA—Rivest, Shamir, and Adelman are the inventors of the RSA technique. The RSA technique is a public-key cryptographic system that can be used for encryption and authentication.

SA—Security Association. SA is an instance of security policy and keying material applied to a data flow.

VPN—Virtual Private Network. A VPN enables IP traffic to travel securely over a public TCP or IP network by encrypting all traffic from one network to another. A VPN uses "tunneling" to encrypt all information at the IP level.

VRF—Virtual Route Forwarding. VRF is A VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

XAUTH—Extended authentication. XAUTH is an optional exchange between IKE Phase 1 and IKE Phase 2, in which the router demands additional authentication information in an attempt to authenticate the actual user (as opposed to authenticating the peer).

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

Public Key Infrastructure (PKI)

Implementing and Managing PKI Features Roadmap

This roadmap lists the features documented in the *Cisco IOS Security Configuration Guide* and maps them to the modules in which they appear.

Roadmap History

This roadmap was first published on May 2, 2005, and last updated on May 2, 2005.

Feature and Release Support

Table 56 lists public key infrastructure (PKI) feature support for the following Cisco IOS software release trains:

- Cisco IOS Releases 12.2T, 12.3, and 12.3T

Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table. *Not all features may be supported in your Cisco IOS software release*.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 56 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Table 56 Supported PKI Features

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Releases 12.2T, 12.3, and 12.3T			
12.3(14)T	Administrative Secure Device Provisioning Introducer	This feature allows you to act as an administrative introducer to introduce a device into a PKI network and then provide a username as the device name for the record locator in the AAA database.	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI"
12.3(14)T	Persistent Self-Signed Certificates	This feature allows users the HTTPS server to generate and save a self-signed certificate in the router's startup configuration. Thus, future SSL handshakes between the client and the HTTPS server can use the same self-signed certificate without user intervention.	"Configuring Certificate Enrollment for a PKI"
12.3(14)T	Secure Device Provisioning Certificate-Based Authorization	This feature allows certificates issued by other authority (CA) servers to be used for SDP introductions.	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI"
12.3(14)T	Subordinate Certificate Server	This enhancement allows you to configure a subordinate certificate server to grant all or certain SCEP or manual certificate requests.	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment"
12.3(14)T	USB Storage	This feature explains how to store RSA keys on a device external to the router via a USB eToken. The SmartCard technology (which is owned by Aladdin Knowledge Systems) in a USB key form factor (also referred to as a USB eToken) provides secure configuration distribution and allows users to store PKI credentials, such as RSA keys, for deployment.	"Storing PKI Credentials External to the Router"
12.3(11)T	The Certificate Server Auto Archive enhancement	This enhancement enables the CA certificate and CA key to be backed up automatically just once after they are generated by the certificate server. As a result, it is not necessary to generate an exportable CA key if CA backup is desirable.	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment"
12.3(11)T	PKI AAA Authorization Using the Entire Subject Name	This feature provides users with the ability to query the AAA server using the entire subject name from the certificate as a unique AAA username.	"Configuring Revocation and Authorization of Certificates in a PKI"
12.3(11)T	PKI Status	This enhancement added the **status** keyword to the **show crypto pki trustpoints** command, which allows you to view the current status of the trustpoint. Prior to this enhancement, you had to issue the **show crypto pki certificates** and the **show crypto pki timers** commands for the current status.	"Configuring Certificate Enrollment for a PKI" and "Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment"
12.3(11)T	Reenroll Using Existing Certificates	This feature allows users to reenroll a router with a Cisco IOS CA via existing certificates from a third-party vendor CA.	"Configuring Certificate Enrollment for a PKI"
12.3(8)T	Easy Secure Device Deployment	This feature introduces support for SDP (formerly called EzSDD), which offers a web-based enrollment interface that enables network administrators to deploy new devices in large networks.	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI"

Table 56 Supported PKI Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.3(8)T	Easy Secure Device Deployment AAA Integration	This feature integrates an external AAA database, allowing the introducer to be authenticated against a AAA database instead of having to use the enable password of the local Cisco certificate server.	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI"
12.3(7)T	The Certificate Server Registration Authority (RA) Mode enhancement	A certificate server can be configured to run in RA mode.	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment"
12.3(7)T	The "crypto pki" commands should be a synonym for "crypto ca" commands	This enhancement changes all commands that begin as "crypto ca" to "crypto pki." Although the router will still accept crypto ca, all output will be read back as crypto pki.	All modules that contain crypto ca commands.
12.3(7)T	Key Rollover for Certificate Renewal	This feature allows the certificate renewal request to be made before the certificate expires and retains the old key and certificate until the new certificate is available.	"Configuring Certificate Enrollment for a PKI"
12.3(7)T	PKI: Query Multiple Servers During Certificate Revocation Check	This feature introduces the ability for Cisco IOS software to make multiple attempts to retrieve the CRL, allowing operations to continue when a particular server is not available. In addition, the ability to override the CDPs in a certificate with a manually configured CDP has been introduced. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.	"Configuring Revocation and Authorization of Certificates in a PKI"
12.3(7)T	Protected Private Key Storage	This feature allows a user to encrypt and lock the RSA private keys that are used on a Cisco IOS router, thereby, preventing unauthorized use of the private keys.	"Deploying RSA Keys Within a PKI"
12.3(4)T	Import of RSA Key Pair and Certificates in PEM Format	This feature allows customers to use PEM-formatted files to import or export RSA key pairs. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS routers instead of generating new keys. Also, customers can issue certificate requests and receive issued certificates in PEM-formatted files.	"Deploying RSA Keys Within a PKI" and "Configuring Certificate Enrollment for a PKI"
12.3(4)T	Using Certificate ACLs to Ignore Revocation Check and Expired Certificates	This feature allows a certificate that meets specified criteria to be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. Certificate ACLs are used to specify the criteria that the certificate must meet to be accepted or to avoid revocation checking. In addition, if AAA communication is protected by a certificate, this feature provides for the AAA checking of the certificate to be ignored.	"Configuring Revocation and Authorization of Certificates in a PKI"

Table 56 Supported PKI Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.3(4)T	Cisco IOS Certificate Server	This feature introduces support for the Cisco IOS CS, which offers users a CA that is directly integrated with Cisco IOS software to more easily deploy basic PKI networks.	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment"
12.3(4)T	Direct HTTP Enrollment with CA Servers	This feature allows users to configure an enrollment profile if their CA server does not support SCEP and they do not want to use an RA as a proxy. The enrollment profile allows users to send HTTP requests directly to the CA server instead of the RA proxy.	"Configuring Certificate Enrollment for a PKI"
12.3(2)T	Online Certificate Status Protocol (OCSP)	This feature allows users to enable OCSP instead of CRLs to check certificate status. Unlike CRLs, which provide only periodic certificate status, OCSP can provide timely information regarding the status of a certificate.	"Configuring Revocation and Authorization of Certificates in a PKI"
12.3(1)	PKI Integration with AAA Server	This feature provides additional scalability for authorization by generating a AAA username from the certificate presented by the peer. A AAA server is queried to determine whether the certificate is authorized for use by the internal component. The authorization is indicated by a component-specified label that must be present in the AV pair for the user.	"Configuring Revocation and Authorization of Certificates in a PKI"
12.2(15)T	Certificate Security Attribute-Based Access Control	Under the IPSec protocol, CA interoperability permits Cisco IOS devices and a CA to communicate so that the Cisco IOS device can obtain and use digital certificates from the CA. Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action. This feature adds fields to the certificate that allow specifying an ACL, to create a certificate-based ACL.	"Configuring Revocation and Authorization of Certificates in a PKI"
12.2(15)T	Exporting and Importing RSA Keys	This feature allows you to transfer security credentials between devices by exporting and importing RSA keys. The key pair that is shared between two devices will allow one device to immediately and transparently take over the functionality of the other router.	"Deploying RSA Keys Within a PKI"
12.2(15)T	Multiple-Tier CA Hierarchy	This enhancement enables users to set up a PKI in a hierarchical framework to support multiple CAs. Within a hierarchical PKI, all enrolled peers can validate the certificate of one another as long as the peers share a trusted root CA certificate or a common subordinate CA.	"Configuring Certificate Enrollment for a PKI"
12.2(13)T	Manual Certificate Enrollment (TFTP Cut-and-Paste)	This feature allows users to generate a certificate request and accept CA certificates as well as the router's certificates via a TFTP server or manual cut-and-paste operations.	"Configuring Certificate Enrollment for a PKI"
12.2(8)T	Certificate Autoenrollment	This feature introduces certificate autoenrollment, which allows the router to automatically request a certificate from the CA that is using the parameters in the configuration.	"Configuring Certificate Enrollment for a PKI"

Table 56 Supported PKI Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(8)T	Certificate Enrollment Enhancements	This feature introduces five new **crypto ca trustpoint** subcommands that provide new options for certificate requests and allow users to specify fields in the configuration instead of having to go through prompts.	"Configuring Certificate Enrollment for a PKI"
12.2(8)T	Multiple RSA Key Pair Support	This feature allows a user to configure a router to have multiple RSA key pairs. Thus, the Cisco IOS software can maintain a different key pair for each identity certificate.	"Deploying RSA Keys Within a PKI"
12.2(8)T	Trustpoint CLI	This feature introduces the **crypto ca trustpoint** command, which adds support for trustpoint CAs.	"Configuring Certificate Enrollment for a PKI"

Cisco IOS PKI Overview: Understanding and Planning a PKI

Cisco IOS public key infrastructure (PKI) provides certificate management to support security protocols such as IP Security (IPSec), secure shell (SSH), and secure socket layer (SSL).

This module identifies and describes concepts that are needed to understand, plan for, and implement a PKI.

Module History

This module was first published on May 2, 2005, and last updated on July 17, 2008.

Contents

Information About Cisco IOS PKI

Before implementing a basic PKI, you should understand the following concepts:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

What Is Cisco IOS PKI?

A PKI is composed of the following entities:

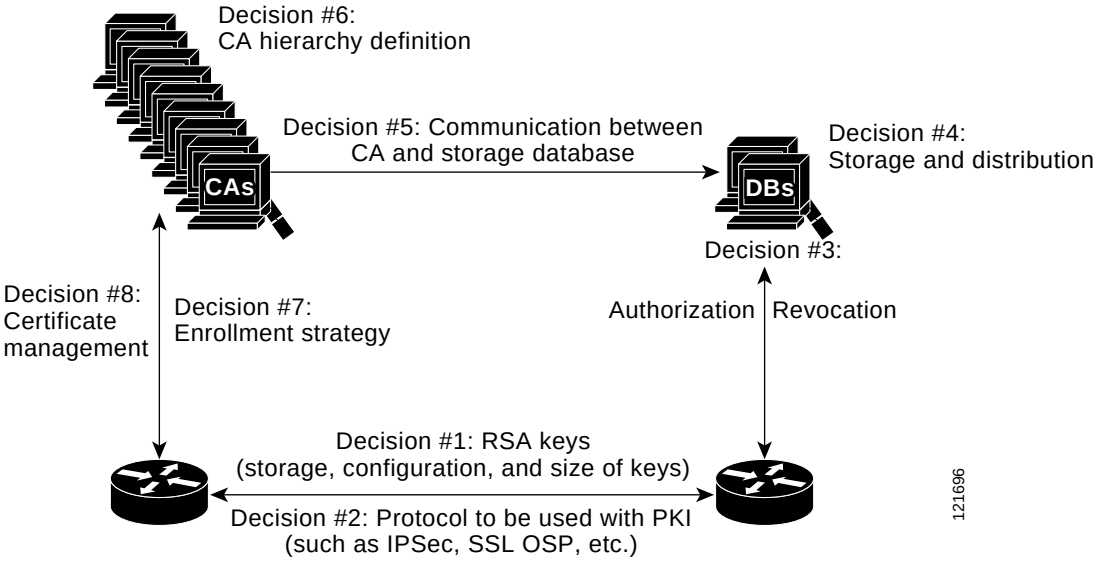
- Peers communicating on a secure network
- At least one certification authority (CA) that grants and maintains certificates
- Digital certificates, which contain information such as the certificate validity period, peer identity information, encryptions keys that are used for secure communications, and the signature of the issuing CA
- An optional registration authority (RA) to offload the CA by processing enrollment requests
- A distribution mechanism (such as Lightweight Directory Access Protocol [LDAP] or HTTP) for certificate revocation lists (CRLs)

PKI provides customers with a scalable, secure mechanism for distributing, managing, and revoking encryption and identity information in a secured data network. Every entity (a person or a device) participating in the secured communicated is enrolled in the PKI in a process where the entity generates an Rivest, Shamir, and Adelman (RSA) key pair (one private key and one public key) and has their identity validated by a trusted entity (also known as a CA or trustpoint).

After each entity enrolls in a PKI, every peer (also known as an end host) in a PKI is granted a digital certificate that has been issued by a CA. When peers must negotiate a secured communication session, they exchange digital certificates. Based on the information in the certificate, a peer can validate the identity of another peer and establish an encrypted session with the public keys contained in the certificate.

Although you can plan for and set up your PKI in a number of different ways, Figure 97 shows the major components that make up a PKI and suggests an order in which each decision within a PKI can be made. Figure 97 is a suggested approach; you can choose to set up your PKI from a different perspective.

Figure 97 Deciding How to Set Up Your PKI



RSA Keys Overview

An RSA key pair consists of a public key and a private key. When setting up your PKI, you must include the public key in the certificate enrollment request. After the certificate has been granted, the public key will be included in the certificate so that peers can use it to encrypt data that is sent to the router. The private key is kept on the router and used both to decrypt the data sent by peers and to digitally sign transactions when negotiating with peers.

RSA key pairs contain a key modulus value. The modulus determines the size of the RSA key. The larger the modulus, the more secure the RSA key. However, keys with large modulus values take longer to generate, and encryption and decryption operations take longer with larger keys.

What Are CAs?

A CA, also known as a trustpoint, manages certificate requests and issues certificates to participating network devices. These services (managing certificate requests and issuing certificates) provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

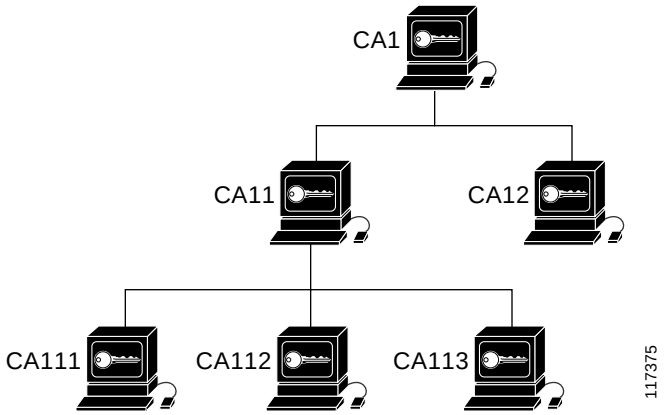
You can use a CA provided by a third-party CA vendor, or you can use an “internal” CA, which is the Cisco IOS Certificate Server.

Hierarchical PKI: Multiple CAs

PKI can be set up in a hierarchical framework to support multiple CAs. At the top of the hierarchy is a root CA, which holds a self-signed certificate. The trust within the entire hierarchy is derived from the RSA key pair of the root CA. The subordinate CAs within the hierarchy can be enrolled with either the root CA or with another subordinate CA. These enrollment options are how multiple tiers of CAs are configured. Within a hierarchical PKI, all enrolled peers, can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA.

Figure 98 shows the enrollment relationships among CAs within a three-tiered hierarchy.

Figure 98 Three-Tiered CA Hierarchy Sample Topology



Each CA corresponds to a trustpoint. For example, CA11 and CA12 are subordinate CAs, holding CA certificates that have been issued by CA1; CA111, CA112, and CA113 are also subordinate CAs, but their CA certificates have been issued by CA11.

When to Use Multiple CAs

Multiple CAs provide users with added flexibility and reliability. For example, subordinate CAs can be placed in branch offices while the root CA is at the office headquarters. Also, different granting policies can be implemented per CA, so you can set up one CA to automatically grant certificate requests while another CA within the hierarchy requires each certificate request to be manually granted.

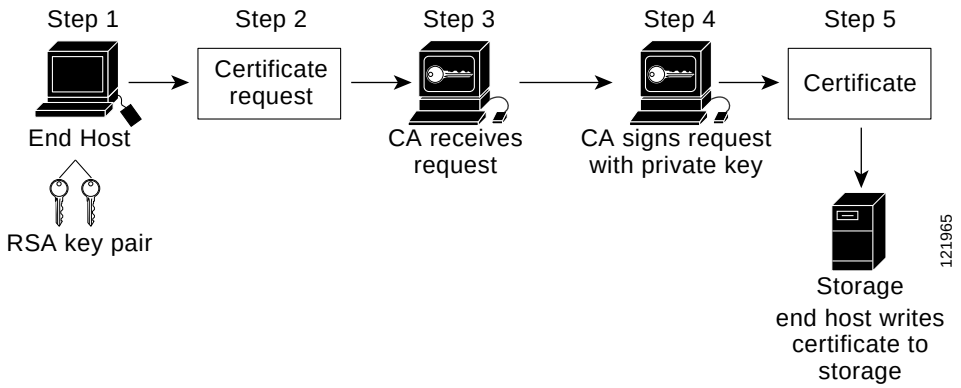
Scenarios in which at least a two-tier CA is recommended are as follows:

- Large and very active networks in which a large number of certificates are revoked and reissued. A multiple tier CA helps to control the size of the CRLs.
- When online enrollment protocols are used, the root CA can be kept offline with the exception of issuing subordinate CA certificates. This scenario provides added security for the root CA.

Certificate Enrollment: How It Works

Certificate enrollment is the process of obtaining a certificate from a CA. Each end host that wants to participate in the PKI must obtain a certificate. Certificate enrollment occurs between the end host requesting the certificate and the CA. Figure 99 and the following steps describe the certificate enrollment process.

Figure 99 Certificate Enrollment Process



1. The end host generates an RSA key pair.
2. The end host generates a certificate request and forwards it to the CA (or the RA, if applicable).
3. The CA receives the certificate enrollment request, and, depending on your network configuration, one of the following options occurs:
   c. Manual intervention is required to approve the request.
   d. The end host is configured to automatically request a certificate from the CA. Thus, operator intervention is no longer required at the time the enrollment request is sent to the CA server.

Note If you configure the end host to automatically request certificates from the CA, you should have an additional authorization mechanism.

4. After the request is approved, the CA signs the request with its private key and returns the completed certificate to the end host.
5. The end host writes the certificate to a storage area such as NVRAM.

Certificate Enrollment Via Secure Device Provisioning

Secure Device Provisioning (SDP) is a web-based certificate enrollment interface that can be used to easily deploy PKI between two end devices, such as a Cisco IOS client and a Cisco IOS certificate server.

SDP (also referred to as Trusted Transitive Introduction [TTI]) is a communication protocol that provides a bidirectional introduction between two end entities, such as a new network device and a Virtual Private Network (VPN). SDP involves the following three entities:

- Introducer—A mutually trusted device that introduces the petitioner to the registrar. The introducer can be a device user, such as a system administrator.
- Petitioner—A new device that is joined to the secure domain.
- Registrar—A certificate server or other server that authorizes the petitioner.

SDP is implemented over a web browser in three phases—welcome, introduction, and completion. Each phase is shown to the user via a web page. For more information on each phase and how SDP works, see the “Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI” module.

Certificate Revocation: Why It Occurs

After each participant has successfully enrolled in the PKI, the peers are ready to begin negotiations for a secure connection with each other. Thus, the peers present their certificates for validation followed by a revocation check. After the peer verifies that the other peer’s certificate was issued by an authenticated CA, the CRL or Online Certificate Status Protocol (OCSP) server is checked to ensure that the certificate has not been revoked by the issuing CA. The certificate usually contains a certificate distribution point (CDP) in the form of a URL. Cisco IOS software uses the CDP to locate and retrieve the CRL. If the CDP server does not respond, the Cisco IOS software reports an error, which may result in the peer’s certificate being rejected.

Planning for a PKI

Planning for a PKI requires evaluating the requirements and expected use for each of the PKI components shown in Figure 97. It is recommended that you (or the network administrator) thoroughly plan the PKI before beginning any PKI configuration.

Although there are a number of approaches to consider when planning the PKI, this document begins with peer-to-peer communication and proceeds as shown in Figure 97. However you or the network administrator choose to plan the PKI, understand that certain decisions influence other decisions within the PKI. For example, the enrollment and deployment strategy could influence the planned CA hierarchy. Thus, it is important to understand how each component functions within the PKI and how certain component options are dependent upon decisions made earlier in the planning process.

Where to Go Next

As suggested in Figure 97, you begin to configure a PKI by setting up and deploying RSA keys. For more information, see the module "Deploying RSA Keys Within a PKI."

Additional References

The following sections provide references related to Cisco IOS PKI.

Related Documents

Related Topic	Document Title
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
Certificate enrollment: supported methods, enrollment profiles, configuration tasks	"Configuring Certificate Enrollment for a PKI" module
Certificate revocation and authorization: configuration tasks	"Configuring Revocation and Authorization of Certificates in a PKI" module
Cisco IOS certificate server overview information and configuration tasks	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" module
Secure Device Provisioning: functionality overview and configuration tasks	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI" module
Storing RSA keys and certificates on a USB eToken	"Storing PKI Credentials" module

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2459	Internet X.509 Public Key Infrastructure Certificate and CRL Profile
RFC 2511	Internet X.509 Certificate Request Message Format
RFC 2527	Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework
RFC 2528	Internet X.509 Public Key Infrastructure
RFC 2559	Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2
RFC 2560	X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP

RFCs	Title
RFC 2585	Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP
RFC 2587	Internet X.509 Public Key Infrastructure LDAPv2 Schema
RFC 2875	Diffie-Hellman Proof-of-Possession Algorithms
RFC 3029	Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Glossary

CDP—certificate distribution point. Field within a digital certificate containing information that describes how to retrieve the CRL for the certificate. The most common CDPs are HTTP and LDAP URLs. A CDP may also contain other types of URLs or an LDAP directory specification. Each CDP contains one URL or directory specification.

certificates—Electronic documents that bind a user's or device's name to its public key. Certificates are commonly used to validate a digital signature.

CRL—certificate revocation list. Electronic document that contains a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when the certificate was issued and when it expires. A new CRL is issued when the current CRL expires.

CA—certification authority. Service responsible for managing certificate requests and issuing certificates to participating IPSec network devices. This service provides centralized key management for the participating devices and is explicitly trusted by the receiver to validate identities and to create digital certificates.

peer certificate—Certificate presented by a peer, which contains the peer's public key and is signed by the trustpoint CA.

PKI—public key infrastructure. System that manages encryption keys and identity information for components of a network that participate in secured communications.

RA—registration authority. Server that acts as a proxy for the CA so that CA functions can continue when the CA is offline. Although the RA is often part of the CA server, the RA could also be an additional application, requiring an additional device to run it.

RSA keys—Public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Deploying RSA Keys Within a PKI

First Published: May 2, 2005
Last Updated: November 17, 2006

This module explains how to set up and deploy Rivest, Shamir, and Adelman (RSA) keys within a public key infrastructure (PKI). An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router; that is, the end host must generate a pair of RSA keys and exchange the public key with the certification authority (CA) to obtain a certificate and enroll in a PKI.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for RSA Keys Within a PKI" section on page 20.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring RSA Keys for a PKI

- Before setting up and deploying RSA keys for a PKI, you should be familiar with the module "Cisco IOS PKI Overview: Understanding and Planning a PKI."
- As of Cisco IOS Release 12.3(7)T, all commands that begin as "crypto ca" have been changed to begin as "crypto pki." Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

Information About RSA Keys Configuration

To deploy RSA keys within a PKI, you should understand the following concepts:

RSA Keys Overview

An RSA key pair consists of a public key and a private key. When setting up your PKI, you must include the public key in the certificate enrollment request. After the certificate has been granted, the public key will be included in the certificate so that peers can use it to encrypt data that is sent to the router. The private key is kept on the router and used both to decrypt the data sent by peers and to digitally sign transactions when negotiating with peers.

RSA key pairs contain a key modulus value. The modulus determines the size of the RSA key. The larger the modulus, the more secure the RSA key. However, keys with large modulus values take longer to generate, and encryption and decryption operations take longer with larger keys.

If you want a modulus value between 512 and 1024, enter an integer value that is a multiple of 64. If you want a value higher than 1024, enter 1536 or 2048. If you enter a value greater than 512, key generation may take a minute or longer.

Note As of Cisco IOS Release 12.4(11)T, peer *public* RSA key modulus values up to 4096 bits are automatically supported.

The largest private RSA key modulus is 2048 bits. Therefore, the largest RSA private key a router may generate or import is 2048 bits.

The recommended modulus value for a CA is 2048 bits; the recommended modulus value for a client is 1024 bits.

Usage RSA Keys Versus General-Purpose RSA Keys

There are two mutually exclusive types of RSA key pairs—usage keys and general-purpose keys. When you generate RSA key pairs (via the **crypto key generate rsa** command), you will be prompted to select either usage keys or general-purpose keys.

Usage RSA Keys

Usage keys consist of two RSA key pairs—one RSA key pair is generated and used for encryption and one RSA key pair is generated and used for signatures. With usage keys, each key is not unnecessarily exposed. (Without usage keys, one key is used for both authentication methods, increasing the exposure of that key.)

General-Purpose RSA Keys

General-purpose keys consist of only one RSA key pair that used for both encryption and signatures. General-purpose key pairs are used more frequently than usage key pairs.

Reasons to Store Multiple RSA Keys on a Router

Configuring multiple RSA key pairs allows the Cisco IOS software to maintain a different key pair for each CA with which it is dealing or the software can maintain multiple key pairs and certificates with the same CA. Thus, the Cisco IOS software can match policy requirements for each CA without compromising the requirements specified by the other CAs, such as key length, key lifetime, and general-purpose versus usage keys.

Named key pairs (which are specified via the **label** *key-label* option) allow you to have multiple RSA key pairs, enabling the Cisco IOS software to maintain a different key pair for each identity certificate.

Benefits of Exportable RSA Keys

Caution Exportable RSA keys should be carefully evaluated before use because using exportable RSA keys introduces the risk that these keys might be exposed.

Any existing RSA keys are *not* exportable. New keys are generated as nonexportable by default. It is not possible to convert an existing nonexportable key to an exportable key.

As of Cisco IOS Release 12.2(15)T, users can share the private RSA key pair of a router with standby routers, therefore transferring the security credentials between networking devices. The key pair that is shared between two routers will allow one router to immediately and transparently take over the functionality of the other router. If the main router were to fail, the standby router could be dropped into the network to replace the failed router without the need to regenerate keys, reenroll with the CA, or manually redistribute keys.

Exporting and importing an RSA key pair also enables users to place the same RSA key pair on multiple routers so that all management stations using Secure Shell (SSH) can be configured with a single public RSA key.

Exportable RSA Keys in PEM-Formatted Files

Using privacy-enhanced mail (PEM)-formatted files to import or export RSA keys can be helpful for customers who are running Cisco IOS software Release 12.3(4)T or later and who are using secure socket layer (SSL) or secure shell (SSH) applications to manually generate RSA key pairs and import the keys back into their PKI applications. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS routers instead of generating new keys.

Passphrase Protection While Importing and Exporting RSA Keys

You have to include a passphrase to encrypt the PKCS12 file or the PEM file that will be exported, and when the PKCS12 or PEM file is imported, the same passphrase has to be entered to decrypt it. Encrypting the PKCS12 or PEM file when it is being exported, deleted, or imported protects the file from unauthorized access and use while it is being transported or stored on an external device.

The passphrase can be any phrase that is at least eight characters in length; it can include spaces and punctuation, excluding the question mark (?), which has special meaning to the Cisco IOS parser.

How to Convert an Exportable RSA Key Pair to a Nonexportable RSA Key Pair

Passphrase protection protects the external PKCS12 or PEM file from unauthorized access and use. To prevent an RSA key pair from being exported, it must be labeled “nonexportable.” To convert an exportable RSA key pair into a nonexportable key pair, the key pair must be exported and then reimported without specifying the “exportable” keyword.

How to Set Up and Deploy RSA Keys Within a PKI

This section contains the following procedures:

Generating an RSA Key Pair

Perform this task to manually generate an RSA key pair.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** {**general-keys** | **usage-keys**} [**label** *key-label*] [**modulus** *modulus-size*] [**exportable**]
4. **exit**
5. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa** {**general-keys** | **usage-keys**} [**label** *key-label*] [**modulus** *modulus-size*] [**exportable**] **Example:** `Router(config)# crypto key generate rsa general-keys modulus 360`	Generates RSA key pairs. • If a *key-label* argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the router, is used.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	`show crypto key mypubkey rsa` **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.

What to Do Next

After you have successfully generated an RSA key pair, you can proceed to any of the additional tasks in this module to generate additional RSA key pairs, perform export and import of RSA key pairs, or configure additional security parameters for the RSA key pair (such as encrypting or locking the private key).

Generating and Storing Multiple RSA Key Pairs

Perform this task to configure the router to generate and store multiple RSA key pairs and associate the key pairs with a trustpoint.

A trustpoint (also known as a CA) manages certificate requests and issues certificates to participating network devices. These services provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

Prerequisites

You must have already generated an RSA key pair as shown in the task "Generating an RSA Key Pair."

SUMMARY STEPS

1. **crypto pki trustpoint** *name*
2. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]
3. **exit**
4. exit
5. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`crypto pki trustpoint name` **Example:** `Router(config)# crypto pki trustpoint fancy-ca`	Creates a trustpoint and enters ca-trustpoint configuration mode.
Step 2	`rsakeypair key-label [key-size [encryption-key-size]]` **Example:** `Router(ca-trustpoint)# rsakeypair fancy-keys`	Specifies the key pair that is to be used with the trustpoint. • Specify the *key-size* argument for generating the key and specify the *encryption-key-size* argument to request separate encryption, signature keys, and certificates.
Step 3	`exit` **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 4	`exit` **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	`show crypto key mypubkey rsa` **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.

Exporting and Importing RSA Keys

This section contains the following tasks that can be used for exporting and importing RSA keys. Whether you are using PKCS12 files or PEM files, exportable RSA keys allow you to use existing RSA keys on Cisco IOS routers instead of having to generate new RSA keys if the main router were to fail.

- Exporting and Importing RSA Keys in PKCS12 Files, page 7
- Exporting and Importing RSA Keys in PEM-Formatted Files, page 8

Exporting and Importing RSA Keys in PKCS12 Files

Exporting and importing RSA key pairs enables users to transfer security credentials between devices. The key pair that is shared between two devices allows one device to immediately and transparently take over the functionality of the other router.

Prerequisites for Exporting and Importing RSA Key in PKCS12 Files

You must generate an RSA key pair and mark it "exportable" as specified in the task "Generating an RSA Key Pair."

Restrictions for Exporting and Importing RSA Keys in PKCS12 Files

- You cannot export RSA keys that existed on the router before your system was upgraded to Cisco IOS Release 12.2(15)T or later. You have to generate new RSA keys and label them as "exportable" after you upgrade the Cisco IOS software.
- When you import a PKCS12 file that was generated by a third-party application, the PKCS12 file must include a CA certificate.
- If you want reexport an RSA key pair after you have already exported the key pair and imported them to a target router, you must specify the **exportable** keyword when you are importing the RSA key pair.
- The largest RSA key a router may import is 2048-bits.

SUMMARY STEPS

1. **crypto pki trustpoint** *name*
2. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]
3. **exit**
4. **crypto pki export** *trustpointname* **pkcs12** *destination-url passphrase*
5. **crypto pki import** *trustpointname* **pkcs12** *source-url passphrase*
6. **exit**
7. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`crypto pki trustpoint name` **Example:** `Router(config)# crypto pki trustpoint my-ca`	Creates the trustpoint name that is to be associated with the RSA key pair and enters ca-trustpoint configuration mode.
Step 2	`rsakeypair key-label [key-size [encryption-key-size]]` **Example:** `Router(ca-trustpoint)# rsakeypair my-keys`	Specifies the key pair that is to be used with the trustpoint.
Step 3	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 4	**crypto pki export** *trustpointname* **pkcs12** *destination-url passphrase* **Example:** `Router(config)# crypto pki export my-ca pkcs12 tftp://tftpserver/my-keys PASSWORD`	Exports the RSA keys via the trustpoint name. **Note** You can export the trustpoint using any of the following file system types: flash, FTP, null, NVRAM, remote file copying (RCP), SCP, system, TFTP, Webflash, Xmodem, or Ymodem.
Step 5	`crypto pki import trustpointname pkcs12 source-url passphrase` **Example:** `Router(config)# crypto pki import my-ca pkcs12 tftp://tftpserver/my-keys PASSWORD`	Imports the RSA keys to the target router.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 7	`show crypto key mypubkey rsa` **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router.

Exporting and Importing RSA Keys in PEM-Formatted Files

Perform this task to export or import RSA key pairs in PEM files.

Prerequisites for Exporting and Importing RSA Keys in PEM-Formatted Files

You must generate an RSA key pair and mark it “exportable” as specified in the task “Generating an RSA Key Pair.”

Restrictions for Exporting and Importing RSA Keys in PEM Formatted Files

- You cannot export and import RSA keys that were generated without an exportable flag before your system was upgraded to Cisco IOS Release 12.3(4)T or a later release. You have to generate new RSA keys after you upgrade the Cisco IOS software.
- The largest RSA key a router may import is 2048 bits.

SUMMARY STEPS

1. **crypto key generate rsa** {**usage-keys** | **general-keys**} **label** *key-label* [**exportable**]
2. **crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase*
3. **crypto key import rsa** *key-label* **pem** [**usage-keys**] {**terminal** | **url** *url*} [**exportable**] *passphrase*

4. **exit**
5. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto key generate rsa** {**usage-keys** | **general-keys**} **label** *key-label* [**exportable**] **Example:** `Router(config)# crypto key generate rsa general-keys label mykey exportable`	Generates RSA key pairs. To use PEM files, the RSA key pair must be labeled exportable.
Step 2	**crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase* **Example:** `Router(config)# crypto key export rsa mycs pem url nvram: 3des PASSWORD`	Exports the generated RSA key pair. **Tip** Be sure to keep the PEM file safe. For example, you may want to store it on another backup router.
Step 3	**crypto key import rsa** *key-label* **pem** [**usage-keys**] {**terminal** | **url** *url*} [**exportable**] *passphrase* **Example:** `Router(config)# crypto key import rsa mycs2 pem url nvram: PASSWORD`	Imports the generated RSA key pair. **Note** If you do not want the key to be exportable from your CA, import it back to the CA after it has been exported as a nonexportable key pair. Thus, the key cannot be taken off again.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	`show crypto key mypubkey rsa` **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router.

Encrypting and Locking Private Keys on a Router

Digital signatures are used to authenticate one device to another device. To use digital signatures, private information (the private key) must be stored on the device that is providing the signature. The stored private information may aid an attacker who steals the hardware device that contains the private key; for example, a thief might be able to use the stolen router to initiate a secure connection to another site by using the RSA private keys stored in the router.

Note RSA keys are lost during password recovery operations. If you lose your password, the RSA keys will be deleted when you perform the password recovery operation. (This function prevents an attacker from performing password recovery and then using the keys.)

To protect the private RSA key from an attacker, a user can encrypt the private key that is stored in NVRAM via a passphrase. Users can also "lock" the private key, which blocks new connection attempts from a running router and protects the key in the router if the router is stolen by an attempted attacker.

Perform this task to encrypt and lock the private key that is saved to NVRAM.

Prerequisites

Before encrypting or locking a private key, you should perform the following tasks:

- Generate an RSA key pair as shown in the task "Generating an RSA Key Pair."
- Optionally, you can authenticate and enroll each router with the CA server.

Note The RSA keys must be unlocked while enrolling the CA. The keys can be locked while authenticating the router with the CA because the private key of the router is not used during authentication.

Restrictions for Encrypting and Locking Private Keys

Backward Compatibility Restriction

Any image prior to Cisco IOS Release 12.3(7)T does not support encrypted keys. To prevent your router from losing all encrypted keys, ensure that only unencrypted keys are written to NVRAM before booting an image prior to Cisco IOS Release 12.3(7)T.

If you must download an image prior to Cisco IOS Release 12.3(7)T, decrypt the key and immediately save the configuration so the downloaded image does not overwrite the configuration.

Interaction with Applications

An encrypted key is not effective after the router boots up until you manually unlock the key (via the **crypto key unlock rsa** command). Depending on which key pairs are encrypted, this functionality may adversely affect applications such as IP security (IPsec), SSH, and SSL; that is, management of the router over a secure channel may not be possible until the necessary key pair is unlocked.

SUMMARY STEPS

1. **crypto key encrypt** [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase*
2. **exit**
3. **show crypto key mypubkey rsa**
4. **crypto key lock rsa** [**name** *key-name*] **passphrase** *passphrase*
5. **show crypto key mypubkey rsa**
6. **crypto key unlock rsa** [**name** *key-name*] **passphrase** *passphrase*
7. **configure terminal**
8. **crypto key decrypt** [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto key encrypt** [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase* **Example:** `Router(config)# crypto key encrypt write rsa name pki.company.com passphrase password`	Encrypts the RSA keys. After this command is issued, the router can continue to use the key; the key remains unlocked. **Note** If the **write** keyword is not issued, the configuration must be manually written to NVRAM; otherwise, the encrypted key will be lost next time the router is reloaded.
Step 2	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 3	`show crypto key mypubkey rsa` **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Shows that the private key is encrypted (protected) and unlocked. **Note** You can also use this command to verify that applications such as Internet Key Exchange (IKE) and SSH are properly working after the key has been encrypted.
Step 4	**crypto key lock rsa** [**name** *key-name*] **passphrase** *passphrase* **Example:** `Router# crypto key lock rsa name pki.company.com passphrase password`	(Optional) Locks the encrypted private key on a running router. **Note** After the key is locked, it cannot be used to authenticate the router to a peer device. This behavior disables any IPSec or SSL connections that use the locked key. Any existing IPSec tunnels created on the basis of the locked key will be closed. If all RSA keys are locked, SSH will automatically be disabled.
Step 5	`show crypto key mypubkey rsa` **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Shows that the private key is protected and locked. The output will also show failed connection attempts via applications such as IKE, SSH, and SSL.
Step 6	**crypto key unlock rsa** [**name** *key-name*] **passphrase** *passphrase* **Example:** `Router# crypto key unlock rsa name pki.company.com passphrase password`	(Optional) Unlocks the private key. **Note** After this command is issued, you can continue to establish IKE tunnels.

	Command or Action	Purpose
Step 7	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 8	**crypto key decrypt** [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase* **Example:** `Router(config)# crypto key decrypt write rsa name pki.company.com passphrase password`	(Optional) Deletes the encrypted key and leaves only the unencrypted key. **Note** The **write** keyword immediately saves the unencrypted key to NVRAM. If the **write** keyword is not issued, the configuration must be manually written to NVRAM; otherwise, the key will remain encrypted the next time the router is reloaded.

Removing RSA Key Pair Settings

You might want to remove an RSA key pair for one of the following reasons:

- During manual PKI operations and maintenance, old RSA keys can be removed and replaced with new keys.
- An existing CA is replaced and the new CA requires newly generated keys; for example, the required key size might have changed in an organization so you would have to delete the old 1024-bit keys and generate new 2048-bit keys.

Perform this task to remove all RSA keys or the specified RSA key pair that has been generated by your router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key zeroize rsa** [*key-pair-label*]
4. **exit**
5. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key zeroize rsa** [*key-pair-label*] **Example:** `Router(config)# crypto key zeroize rsa fancy-keys`	Deletes RSA key pairs from your router. • If the *key-pair-label* argument is not specified, all RSA keys that have been generated by your router will be deleted.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	`show crypto key mypubkey rsa` **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.

Configuration Examples for RSA Key Pair Deployment

This section contains the following configuration examples:

- Generating and Specifying RSA Keys: Example, page 14
- Exporting and Importing RSA Keys: Examples, page 14
- Encrypting and Locking Private Keys on a Router: Examples, page 18

Generating and Specifying RSA Keys: Example

The following example is a sample trustpoint configuration that shows how to generate and specify the RSA key pair “exampleCAkeys”:

```
crypto key generate rsa general-purpose exampleCAkeys
crypto ca trustpoint exampleCAkeys
 enroll url http://exampleCAkeys/certsrv/mscep/mscep.dll
 rsakeypair exampleCAkeys 1024 1024
```

Exporting and Importing RSA Keys: Examples

This section contains the following configuration examples:

Exporting and Importing RSA Keys in PKCS12 Files: Example

In the following example, an RSA key pair “mynewkp” is generated on Router A, and a trustpoint name “mynewtp” is created and associated with the RSA key pair. The trustpoint is exported to a TFTP server, so that it can be imported on Router B. By importing the trustpoint “mynewtp” to Router B, the user has imported the RSA key pair “mynewkp” to Router B.

Router A

```
crypto key generate rsa general label mykeys exportable
! The name for the keys will be:mynewkp
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]:
% Generating 512 bit RSA keys ...[OK]
!
crypto pki trustpoint mynewtp
 rsakeypair mykeys
 exit

crypto pki export mytp pkcs12 flash:myexport companyname
Destination filename [myexport]?
Writing pkcs12 file to tftp:/mytftpserver/myexport
CRYPTO_PKI:Exported PKCS12 file successfully.
Verifying checksum... OK (0x3307)
!
Feb 18 17:30:09 GMT:%CRYPTO-6-PKCS12EXPORT_SUCCESS:PKCS #12 Successfully Exported.
```

Router B

```
crypto pki import mynewtp pkcs12 flash:myexport companyname
Source filename [myexport]?
CRYPTO_PKI:Imported PKCS12 file successfully.

!
Feb 18 18:07:50 GMT:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.
```

Generating, Exporting, Importing, and Verifying RSA Keys in PEM Files: Example

The following example shows how to generate, export, bring the key back (import), and verify the status of the RSA key pair “mycs”:

```
! Generate the key pair
!
Router(config)# crypto key generate rsa general-purpose label mycs exportable
The name for the keys will be: mycs

Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys ...[OK]
!
```

```
! Archive the key pair to a remote location, and use a good password.
!
Router(config)# crypto key export rsa mycs pem url nvram:3des PASSWORD

% Key name:mycs
Usage:General Purpose Key
Exporting public key...
Destination filename [mycs.pub]?
Writing file to nvram:mycs.pub
Exporting private key...
Destination filename [mycs.prv]?
Writing file to nvram:mycs.prv
!
! Import the key as a different name.
!
Router(config)# crypto key import rsa mycs2 pem url nvram:mycs PASSWORD

% Importing public key or certificate PEM file...
Source filename [mycs.pub]?
Reading file from nvram:mycs.pub
% Importing private key PEM file...
Source filename [mycs.prv]?
Reading file from nvram:mycs.prv% Key pair import succeeded.
!
! After the key has been imported, it is no longer exportable.
!
! Verify the status of the key.
!
Router# show crypto key mypubkey rsa

% Key pair was generated at:18:04:56 GMT Jun 6 2003
Key name:mycs
Usage:General Purpose Key
Key is exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253
9C30C12E 295AB73F B1DF9FAD 86F88192 7D4FA4D2 8BA7FB49 9045BAB9 373A31CB
A6B1B8F4 329F2E7E 8A50997E AADBCFAA 23C29E19 C45F4F05 DBB2FA51 4B7E9F79
A1095115 759D6BC3 5DFB5D7F BCF655BF 6317DB12 A8287795 7D8DC6A3 D31B2486
C9C96D2C 2F70B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
% Key pair was generated at:18:17:25 GMT Jun 6 2003
Key name:mycs2
Usage:General Purpose Key
Key is not exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253
9C30C12E 295AB73F B1DF9FAD 86F88192 7D4FA4D2 8BA7FB49 9045BAB9 373A31CB
A6B1B8F4 329F2E7E 8A50997E AADBCFAA 23C29E19 C45F4F05 DBB2FA51 4B7E9F79
A1095115 759D6BC3 5DFB5D7F BCF655BF 6317DB12 A8287795 7D8DC6A3 D31B2486
C9C96D2C 2F70B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
```

Exporting Router RSA Key Pairs and Certificates from PEM Files: Example

The following example shows how to generate and export the RSA key pair “aaa” and certificates of the router in PEM files that are associated with the trustpoint “mycs.” This example also shows PEM-formatted files, which include PEM boundaries before and after the base64-encoded data, that are used by other SSL and SSH applications.

```
Router(config)# crypto key generate rsa general-keys label aaa exportable

The name for the keys will be:aaa
```

```
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose
Keys. Choosing a key modulus greater than 512 may take a few minutes.
!
How many bits in the modulus [512]:
% Generating 512 bit RSA keys ...[OK]
!
Router(config)# crypto pki trustpoint mycs
Router(ca-trustpoint)# enrollment url http://mycs
Router(ca-trustpoint)# rsakeypair aaa
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate mycs
Certificate has the following attributes:
Fingerprint:C21514AC 12815946 09F635ED FBB6CF31
% Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.
!
Router(config)# crypto pki enroll mycs
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this password to the CA
Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please make a note of it.

Password:
Re-enter password:

% The fully-qualified domain name in the certificate will be: Router
% The subject name in the certificate will be:host.company.com
% Include the router serial number in the subject name? [yes/no]: n
% Include an IP address in the subject name? [no]: n
Request certificate from CA? [yes/no]: y
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the fingerprint.

Router(config)# Fingerprint:8DA777BC 08477073 A5BE2403 812DD157

00:29:11:%CRYPTO-6-CERTRET:Certificate received from Certificate Authority

Router(config)# crypto ca export aaa pem terminal 3des password
% CA certificate:
-----BEGIN CERTIFICATE-----
MIICAzCCAa2gAwIBAgIBATANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJVUzES
<snip>
waDeNOSI3WlDa0AWq5DkVBkxwgn0TqIJXJOCttjHnWHK1LMcMVGn
-----END CERTIFICATE-----

% Key name:aaa
Usage:General Purpose Key
-----BEGIN RSA PRIVATE KEY-----
Proc-Type:4,ENCRYPTED
DEK-Info:DES-EDE3-CBC,ED6B210B626BC81A

Urguv0jnjwOgowWVUQ2XR5nbzzYHI2vGLunpH/IxIsJuNjRVjbAAUpGk7VnPCT87
<snip>
kLCOtxzEv7JHc72gMku9uUlrLSnFH5slzAtoC0czfU4=
-----END RSA PRIVATE KEY-----

% Certificate:
-----BEGIN CERTIFICATE-----
MIICTjCCAfigAwIBAgICIQUwDQYJKoZIhvcNAQEFBQAwTjELMAkGA1UEBhMCVVMx
<snip>
6xlBaIsuMxnHmr89KkKkYlU6
```

```
-----END CERTIFICATE-----
```

Importing Router RSA Key Pairs and Certificate from PEM Files: Example

The following example shows how to import the RSA key pairs and certificate to the trustpoint “ggg” from PEM files via TFTP:

```
Router(config)# crypto pki import ggg pem url tftp://10.1.1.2/username/msca password
% Importing CA certificate...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.ca]?
Reading file from tftp://10.1.1.2/username/msca.ca
Loading username/msca.ca from 10.1.1.2 (via Ethernet0):!
[OK - 1082 bytes]

% Importing private key PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.prv]?
Reading file from tftp://10.1.1.2/username/msca.prv
Loading username/msca.prv from 10.1.1.2 (via Ethernet0):!
[OK - 573 bytes]

% Importing certificate PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.crt]?
Reading file from tftp://10.1.1.2/username/msca.crt
Loading username/msca.crt from 10.1.1.2 (via Ethernet0):!
[OK - 1289 bytes]
% PEM files import succeeded.
Router(config)#
```

Encrypting and Locking Private Keys on a Router: Examples

This section contains the following configuration examples:

- Configuring and Verifying an Encrypted Key: Example, page 18
- Configuring and Verifying a Locked Key: Example, page 19

Configuring and Verifying an Encrypted Key: Example

The following example shows how to encrypt the RSA key “pki-123.company.com.” Thereafter, the **show crypto key mypubkey rsa** command is issued to verify that the RSA key is encrypted (protected) and unlocked.

```
Router(config)# crypto key encrypt rsa name pki-123.company.com passphrase password
Router(config)# exit
Router# show crypto key mypubkey rsa

% Key pair was generated at:00:15:32 GMT Jun 25 2003
Key name:pki-123.company.com
Usage:General Purpose Key
*** The key is protected and UNLOCKED. ***
Key is not exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E0CC9A 1D23B52C
CD00910C ABD392AE BA6D0E3F FC47A0EF 8AFEE340 0EC1E62B D40E7DCC
23C4D09E
03018B98 E0C07B42 3CFD1A32 2A3A13C0 1FF919C5 8DE9565F 1F020301 0001
```

```
% Key pair was generated at:00:15:33 GMT Jun 25 2003
Key name:pki-123.company.com.server
Usage:Encryption Key
Key is exportable.
Key Data:
307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00D3491E 2A21D383
854D7DA8 58AFBDAC 4E11A7DD E6C40AC6 66473A9F 0C845120 7C0C6EC8 1FFF5757
3A41CE04 FDCB40A4 B9C68B4F BC7D624B 470339A3 DE739D3E F7DDB549 91CD4DA4
DF190D26 7033958C 8A61787B D40D28B8 29BCD0ED 4E6275C0 6D020301 0001
Router#
```

Configuring and Verifying a Locked Key: Example

The following example shows how to lock the key "pki-123.company.com." Thereafter, the **show crypto key mypubkey rsa** command is issued to verify that the key is protected (encrypted) and locked.

```
Router# crypto key lock rsa name pki-123.company.com passphrase password
!
Router# show crypto key mypubkey rsa

% Key pair was generated at:20:29:41 GMT Jun 20 2003
Key name:pki-123.company.com
Usage:General Purpose Key
*** The key is protected and LOCKED. ***
Key is exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D7808D C5FF14AC
0D2B55AC 5D199F2F 7CB4B355 C555E07B 6D0DECBE 4519B1F0 75B12D6F 902D6E9F
B6FDAD8D 654EF851 5701D5D7 EDA047ED 9A2A619D 5639DF18 EB020301 0001
```

Where to Go Next

After you have generated an RSA key pair, you should set up the trustpoint. If you have already set up the trustpoint, you should authenticate and enroll the routers in a PKI. For information on enrollment, see the module "Configuring Certificate Enrollment for a PKI."

Additional References

The following sections provide references related to configuring RSA keys for a PKI.

Related Documents

Related Topic	Document Title
Overview of PKI, including RSA keys, certificate enrollment, and CAs	"Cisco IOS PKI Overview: Understanding and Planning a PKI" module
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and technical documentation. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for RSA Keys Within a PKI

Table 57 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the *"Implementing and Managing PKI Features Roadmap"*.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 57 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 57 Feature Information for RSA Keys Within a PKI

Feature Name	Software Releases	Feature Configuration Information
Cisco IOS 4096-Bit Public Key Support	12.4(12)T	This feature introduces Cisco IOS 4096-bit public key support. The following section provides information about this feature: • RSA Keys Overview
Exporting and Importing RSA Keys	12.2(15)T Cisco IOS XE Release 2.1	This feature allows you to transfer security credentials between devices by exporting and importing RSA keys. The key pair that is shared between two devices will allow one device to immediately and transparently take over the functionality of the other router. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following sections provide information about this feature: • Benefits of Exportable RSA Keys • Exporting and Importing RSA Keys in PKCS12 Files The following commands were introduced or modified by this feature: **crypto ca export pkcs12**, **crypto ca import pkcs12**, **crypto key generate rsa (IKE)**
Import of RSA Key Pair and Certificates in PEM Format	12.3(4)T Cisco IOS XE Release 2.1	This feature allows customers to use PEM-formatted files to import or export RSA key pairs. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS routers instead of generating new keys. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following sections provide information about this feature: • Benefits of Exportable RSA Keys • Exporting and Importing RSA Keys in PEM-Formatted Files The following commands were introduced by this feature: **crypto ca export pem**, **crypto ca import pem**, **crypto key export pem**, **crypto key import pem**

Table 57 Feature Information for RSA Keys Within a PKI (continued)

Feature Name	Software Releases	Feature Configuration Information
Multiple RSA Key Pair Support	12.2(8)T Cisco IOS XE Release 2.1	This feature allows a user to configure a router to have multiple RSA key pairs. Thus, the Cisco IOS software can maintain a different key pair for each identity certificate. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following sections provide information about this feature: • Reasons to Store Multiple RSA Keys on a Router • Generating and Storing Multiple RSA Key Pairs The following commands were introduced or modified by this feature: **crypto key generate rsa**, **crypto key zeroize rsa**, **rsakeypair**
Protected Private Key Storage	12.3(7)T Cisco IOS XE Release 2.1	This feature allows a user to encrypt and lock the RSA private keys that are used on a Cisco IOS router, thereby, preventing unauthorized use of the private keys. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers. The following section provides information about this feature: • Encrypting and Locking Private Keys on a Router The following commands were introduced or modified by this feature: **crypto key decrypt rsa**, **crypto key encrypt rsa**, **crypto key lock rsa**, **crypto key unlock rsa**, **show crypto key mypubkey rsa**

Configuring Authorization and Revocation of Certificates in a PKI

First Published: May 2, 2005
Last Updated: June 19, 2006

This module describes how to configure authorization and revocation of certificates in a public key infrastructure (PKI).

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Certificate Authorization and Revocation" section on page 40.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Authorization and Revocation of Certificates

Plan Your PKI Strategy

Tip It is strongly recommended that you plan your entire PKI strategy before you begin to deploy actual certificates.

Authorization and revocation can occur only after you or a network administrator have completed the following tasks:

- Configured the CA.
- Enrolled peer devices with the CA.
- Identified and configured the protocol (such as IP Security [IPsec] or secure socket layer [SSL]) that is to be used for peer-to-peer communication.

You should decide which authorization and revocation strategy you are going to configure before enrolling peer devices because the peer device certificates might have to contain authorization and revocation-specific information.

"crypto ca" to "crypto pki" CLI Change

As of Cisco IOS Release 12.3(7)T, all commands that begin as "crypto ca" have been changed to begin as "crypto pki." Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

Information About Authorization and Revocation of Certificates

Before configuring certificate authorization and revocation, you should understand the following concepts:

- PKI Authorization, page 2
- PKI and AAA Server Integration for Certificate Status, page 3
- CRLs or OCSP Server: Choosing a Certificate Revocation Mechanism, page 4
- When to Use Certificate-Based ACLs for Authorization or Revocation, page 7
- PKI Certificate Chain Validation, page 8

PKI Authorization

PKI authentication does not provide authorization. Current solutions for authorization are specific to the router that is being configured, although a centrally managed solution is often required.

There is not a standard mechanism by which certificates are defined as authorized for some tasks and not for others. This authorization information can be captured in the certificate itself if the application is aware of the certificate-based authorization information. But this solution does not provide a simple mechanism for real-time updates to the authorization information and forces each application to be aware of the specific authorization information embedded in the certificate.

When the certificate-based ACL mechanism is configured as part of the trustpoint authentication, the application is no longer responsible for determining this authorization information, and it is no longer possible to specify for which application the certificate is authorized. In some cases, the certificate-based ACL on the router gets so large that it cannot be managed. Additionally, it is beneficial to retrieve certificate-based ACL indications from an external server. (For more information on using certificate-based ACLs for authentication, see the section "When to Use Certificate-Based ACLs for Authorization or Revocation.")

Current solutions to the real-time authorization problem involve specifying a new protocol and building a new server (with associated tasks, such as management and data distribution).

PKI and AAA Server Integration for Certificate Status

Integrating your PKI with an authentication, authorization, and accounting (AAA) server provides an alternative online certificate status solution that leverages the existing AAA infrastructure. Certificates can be listed in the AAA database with appropriate levels of authorization. For components that do not explicitly support PKI-AAA, a default label of "all" from the AAA server provides authorization. Likewise, a label of "none" from the AAA database indicates that the specified certificate is not valid. (The absence of any application label is equivalent, but "none" is included for completeness and clarity). If the application component does support PKI-AAA, the component may be specified directly; for example, the application component could be "ipsec," "ssl," or "osp." (ipsec=IP Security, ssl=Secure Sockets Layer, and osp=Open Settlement Protocol.)

Note

- Currently, no application component supports specification of the application label.
- There may be a time delay when accessing the AAA server. If the AAA server is not available, the authorization fails.

RADIUS or TACACS+: Choosing a AAA Server Protocol

The AAA server can be configured to work with either the RADIUS or TACACS+ protocol. When you are configuring the AAA server for the PKI integration, you must set the RADIUS or TACACS attributes that are required for authorization.

If the RADIUS protocol is used, the password that is configured for the username in the AAA server should be set to "cisco," which is acceptable because the certificate validation provides authentication and the AAA database is only being used for authorization. When the TACACS protocol is used, the password that is configured for the username in the AAA server is irrelevant because TACACS supports authorization without requiring authentication (the password is used for authentication).

In addition, if you are using TACACS, you must add a PKI service to the AAA server. The custom attribute "cert-application=all" is added under the PKI service for the particular user or usergroup to authorize the specific username.

Attribute-Value Pairs for PKI and AAA Server Integration

Table 1 lists the attribute-value (AV) pairs that are to be used when setting up PKI integration with a AAA server. (Note the values shown in the table are possible values.) The AV pairs must match the client configuration. If they do not match, the peer certificate is not authorized.

Note Users can sometimes have AV pairs that are different from those of every other user. As a result, a unique username is required for each user. The **all** parameter (within the **authorization username** command) specifies that the entire subject name of the certificate will be used as the authorization username.

Table 1 AV Pairs That Must Match

AV Pair	Value
cisco-avpair=pki:cert-application=all	Valid values are “all” and “none.”
cisco-avpair=pki:cert-trustpoint=msca	The value is a Cisco IOS command-line interface (CLI) configuration trustpoint label. **Note** The cert-trustpoint AV pair is normally optional. If it is specified, the Cisco IOS router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.
cisco-avpair=pki:cert-serial=16318DB7000100001671	The value is a certificate serial number. **Note** The cert-serial AV pair is normally optional. If it is specified, the Cisco IOS router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.
cisco-avpair=pki:cert-lifetime-end=1:00 jan 1, 2003	The cert-lifetime-end AV pair is available to artificially extend a certificate lifetime beyond the time period that is indicated in the certificate itself. If the cert-lifetime-end AV pair is used, the cert-trustpoint and cert-serial AV pairs must also be specified. The value must match the following form: hours:minutes month day, year. **Note** Only the first three characters of a month are used: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. If more than three characters are entered for the month, the remaining characters are ignored (for example Janxxxx).

CRLs or OCSP Server: Choosing a Certificate Revocation Mechanism

After a certificate is validated as a properly signed certificate, a certificate revocation method is performed to ensure that the certificate has not been revoked by the issuing CA. Cisco IOS software supports two revocation mechanisms—certificate revocation lists (CRLs) and Online Certificate Status Protocol (OCSP). (Cisco IOS software also supports AAA integration for certificate checking; however, additional authorization functionality is included. For more information on PKI and AAA certificate authorization and status check, see the section “PKI and AAA Server Integration for Certificate Status.”)

The following sections explain how each revocation mechanism works:

- What Is a CRL?, page 5
- What Is OCSP?, page 6

What Is a CRL?

A certificate revocation list (CRL) contains a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when each certificate was issued and when it expires.

CAs publish new CRLs periodically or when a certificate for which the CA is responsible has been revoked. By default, a new CRL will be downloaded after the currently cached CRL expires. An administrator may also configure the duration for which CRLs are cached in router memory or disable CRL caching completely. The CRL caching configuration will apply to all CRLs associated with a trustpoint.

When the CRL expires, the router deletes it from its cache. A new CRL is downloaded when a certificate is presented for verification; however, if a newer version of the CRL that lists the certificate under examination is on the server but the router is still using the CRL in its cache, the router will not know that the certificate has been revoked. The certificate will pass the revocation check even though it should have been denied.

When a CA issues a certificate, the CA can include in the certificate the CRL distribution point (CDP) for that certificate. Cisco IOS client devices use CDPs to locate and load the correct CRL. The Cisco IOS client supports multiple CDPs, but the Cisco IOS CA currently supports only one CDP; however, third-party vendor CAs may support multiple CDPs or different CDPs per certificate. If a CDP is not specified in the certificate, the client device will use the default Simple Certificate Enrollment Protocol (SCEP) method to retrieve the CRL. (The CDP location can be specified via the **cdp-url** command.)

When implementing CRLs, you should consider the following design considerations:

- CRL lifetimes and the security association (SA) and Internet Key Exchange (IKE) lifetimes

  The CRL lifetime determines the length of time between CA-issued updates to the CRL. (The default CRL lifetime value, which is 168 hours [1 week], can be changed via the **lifetime crl** command.)

- The method and location of the CDP
  - The method determines how the CRL is retrieved; some possible choices include HTTP, Lightweight Directory Access Protocol (LDAP), SCEP, or TFTP.

    HTTP, TFTP, and LDAP are the most commonly used methods. Although Cisco IOS software defaults to SCEP, an HTTP CDP is recommended for large installations using CRLs because HTTP can be made highly scalable.
  - The location determines from where the CRL is retrieved; for example, you can specify the server and file path from which to retrieve the CRL.

Querying All CDPs During Revocation Check

When a CDP server does not respond to a request, the Cisco IOS software reports an error, which may result in the peer's certificate being rejected. To prevent a possible certificate rejection and if there are multiple CDPs in a certificate, the Cisco IOS software will attempt to use the CDPs in the order in which they appear in the certificate. The router will attempt to retrieve a CRL using each CDP URL or directory specification. If an error occurs using a CDP, an attempt will be made using the next CDP.

Note Prior to Cisco IOS Release 12.3(7)T, the Cisco IOS software makes only one attempt to retrieve the CRL, even when the certificate contains more than one CDP.

Tip Although the Cisco IOS software will make every attempt to obtain the CRL from one of the indicated CDPs, it is recommended that you use an HTTP CDP server with high-speed redundant HTTP servers to avoid application timeouts because of slow CDP responses.

What Is OCSP?

OCSP is an online mechanism that is used to determine certificate validity and provides the following flexibility as a revocation mechanism:

- OCSP can provide real-time certificate status checking.
- OCSP allows the network administrator to specify a central OCSP server, which can service all devices within a network.
- OCSP also allows the network administrator the flexibility to specify multiple OCSP servers, either per client certificate or per group of client certificates.
- OCSP server validation is usually based on the root CA certificate or a valid subordinate CA certificate, but may also be configured so that external CA certificates or self-signed certificates may be used. Using external CA certificates or self-signed certificates allows the OCSP servers certificate to be issued and validated from an alternative PKI hierarchy.

A network administrator can configure an OCSP server to collect and update CRLs from different CA servers. The devices within the network can rely on the OCSP server to check the certificate status without retrieving and caching each CRL for every peer. When peers have to check the revocation status of a certificate, they send a query to the OCSP server that includes the serial number of the certificate in question and an optional unique identifier for the OCSP request, or a nonce. The OCSP server holds a copy of the CRL to determine if the CA has listed the certificate as being revoked; the server then responds to the peer including the nonce. If the nonce in the response from the OCSP server does not match the original nonce sent by the peer, the response is considered invalid and certificate verification fails. The dialog between the OCSP server and the peer consumes less bandwidth than most CRL downloads.

If the OCSP server is using a CRL, CRL time limitations will be applicable; that is, a CRL that is still valid might be used by the OCSP server although a new CRL has been issued by the CRL containing additional certificate revocation information. Because fewer devices are downloading the CRL information on a regular basis, you can decrease the CRL lifetime value or configure the OCSP server not to cache the CRL. For more information, check your OCSP server documentation.

When to Use an OCSP Server

OCSP may be more appropriate than CRLs if your PKI has any of the following characteristics:

- Real-time certificate revocation status is necessary. CRLs are updated only periodically and the latest CRL may not always be cached by the client device. For example, if a client does not yet have the latest CRL cached and a newly revoked certificate is being checked, that revoked certificate will successfully pass the revocation check.

- There are a large number of revoked certificates or multiple CRLs. Caching a large CRL consumes large portions of Cisco IOS memory and may reduce resources available to other processes.
- CRLs expire frequently, causing the CDP to handle a larger load of CRLs.

Note As of Cisco IOS Release 12.4(9)T or later, an administrator may configure CRL caching, either by disabling CRL caching completely or setting a maximum lifetime for a cached CRL per trustpoint.

When to Use Certificate-Based ACLs for Authorization or Revocation

Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action.

Because certificate-based ACLs are configured on the device, they do not scale well for large numbers of ACLs; however, certificate-based ACLs do provide very granular control of specific device behavior. Certificate-based ACLs are also leveraged by additional features to help determine when PKI components such as revocation, authorization, or a trustpoint should be used. They provide a general mechanism allowing users to select a specific certificate or a group of certificates that are being validated for either authorization or additional processing.

Certificate-based ACLs specify one or more fields within the certificate and an acceptable value for each specified field. You can specify which fields within a certificate should be checked and which values those fields may or may not have.

There are six logical tests for comparing the field with the value—equal, not equal, contains, does not contain, less than, and greater than or equal. If more than one field is specified within a single certificate-based ACL, the tests of all of the fields within the ACL must succeed to match the ACL. The same field may be specified multiple times within the same ACL. More than one ACL may be specified, and ACL will be processed in turn until a match is found or all of the ACLs have been processed.

Ignore Revocation Checks Using a Certificate-Based ACL

Certificate-based ACLs can be configured to instruct your router to ignore the revocation check and expired certificates of a valid peer. Thus, a certificate that meets the specified criteria can be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. You can also use a certificate-based ACL to ignore the revocation check when the communication with a AAA server is protected with a certificate.

Ignoring Revocation Lists

To allow a trustpoint to enforce CRLs except for specific certificates, enter the **match certificate** command with the **skip revocation-check** keyword. This type of enforcement is most useful in a hub-and-spoke configuration in which you also want to allow direct spoke-to-spoke connections. In pure hub-and-spoke configurations, all spokes connect only to the hub, so CRL checking is necessary only on the hub. For one spoke to communicate directly with another spoke, the **match certificate** command with the **skip revocation-check** keyword can be used for neighboring peer certificates instead of requiring a CRL on each spoke.

Ignoring Expired Certificates

To configure your router to ignore expired certificates, enter the **match certificate** command with the **allow expired-certificate** keyword. This command has the following purposes:

- If the certificate of a peer has expired, this command may be used to “allow” the expired certificate until the peer can obtain a new certificate.
- If your router clock has not yet been set to the correct time, the certificate of a peer will appear to be not yet valid until the clock is set. This command may be used to allow the certificate of the peer even though your router clock is not set.

Note

- If Network Time Protocol (NTP) is available only via the IPSec connection (usually via the hub in a hub-and-spoke configuration), the router clock can never be set. The tunnel to the hub cannot be “brought up” because the certificate of the hub is not yet valid.
- “Expired” is a generic term for a certificate that is expired or that is not yet valid. The certificate has a start and end time. An expired certificate, for purposes of the ACL, is one for which the current time of the router is outside the start and end times specified in the certificate.

Skipping the AAA Check of the Certificate

If the communication with an AAA server is protected with a certificate, and you want to skip the AAA check of the certificate, use the **match certificate** command with the **skip authorization-check** keyword. For example, if a virtual private network (VPN) tunnel is configured so that all AAA traffic goes over that tunnel, and the tunnel is protected with a certificate, you can use the **match certificate** command with the **skip authorization-check** keyword to skip the certificate check so that the tunnel can be established.

The **match certificate** command and the **skip authorization-check** keyword should be configured after PKI integration with an AAA server is configured.

Note If the AAA server is available only via an IPSec connection, the AAA server cannot be contacted until after the IPSec connection is established. The IPSec connection cannot be “brought up” because the certificate of the AAA server is not yet valid.

PKI Certificate Chain Validation

A certificate chain establishes a sequence of trusted certificates —from a peer certificate to the root CA certificate. Within a PKI hierarchy, all enrolled peers can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA. Each CA corresponds to a trustpoint.

When a certificate chain is received from a peer, the default processing of a certificate chain path continues until the first trusted certificate, or trustpoint, is reached. In Cisco IOS Release 12.4(6)T and later releases, an administrator may configure the level to which a certificate chain is processed on all certificates including subordinate CA certificates.

Configuring the level to which a certificate chain is processed allows for the reauthentication of trusted certificates, the extension of a trusted certificate chain, and the completion of a certificate chain that contains a gap.

Reauthentication of Trusted Certificates

The default behavior is for the router to remove any trusted certificates from the certificate chain sent by the peer before the chain is validated. An administrator may configure certificate chain path processing so that the router does not remove CA certificates that are already trusted before chain validation, so that all certificates in the chain are re-authenticated for the current session.

Extending the Trusted Certificate Chain

The default behavior is for the router to use its trusted certificates to extend the certificate chain if there are any missing certificates in the certificate chain sent by the peer. The router will validate only certificates in the chain sent by the peer. An administrator may configure certificate chain path processing so that the certificates in the peer's certificate chain and the router's trusted certificates are validated to a specified point.

Completing Gaps in a Certificate Chain

An administrator may configure certificate chain processing so that if there is a gap in the configured Cisco IOS trustpoint hierarchy, certificates sent by the peer can be used to complete the set of certificates to be validated.

Note If the trustpoint is configured to require parent validation and the peer does not provide the full certificate chain, the gap cannot be completed and the certificate chain is rejected and invalid.

Note It is a configuration error if the trustpoint is configured to require parent validation and there is no parent trustpoint configured. The resulting certificate chain gap cannot be completed and the subordinate CA certificate cannot be validated. The certificate chain is invalid.

How to Configure Authorization and Revocation of Certificates for Your PKI

This section contains the following procedures:

Configuring PKI Integration with a AAA Server

Perform this task to generate a AAA username from the certificate presented by the peer and specify which fields within a certificate should be used to build the AAA database username.

Restrictions When Using the Entire Subject Name for PKI Authorization

The following restrictions should be considered when using the **all** keyword as the subject name for the **authorization username** command:

- Some AAA servers limit the length of the username (for example, to 64 characters). As a result, the entire certificate subject name cannot be longer than the limitation of the server.
- Some AAA servers limit the available character set that may be used for the username (for example, a space [ ] and an equal sign [=] may not be acceptable). You cannot use the **all** keyword for a AAA server having such a character-set limitation.

- The **subject-name** command in the trustpoint configuration may not always be the final AAA subject name. If the fully qualified domain name (FQDN), serial number, or IP address of the router are included in a certificate request, the subject name field of the issued certificate will also have these components. To turn off the components, use the **fqdn**, **serial-number**, and **ip-address** commands with the **none** keyword.
- CA servers sometimes change the requested subject name field when they issue a certificate. For example, CA servers of some vendors switch the relative distinguished names (RDNs) in the requested subject names to the following order: CN, OU, O, L, ST, and C. However, another CA server might append the configured LDAP directory root (for example, O=cisco.com) to the end of the requested subject name.
- Depending on the tools you choose for displaying a certificate, the printed order of the RDNs in the subject name could be different. Cisco IOS software always displays the least significant RDN first, but other software, such as Open Source Secure Socket Layer (OpenSSL), does the opposite. Therefore, if you are configuring a AAA server with a full distinguished name (DN) (subject name) as the corresponding username, ensure that the Cisco IOS software style (that is, with the least significant RDN first) is used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authorization network** *listname* [*method*]
5. **crypto pki trustpoint** *name*
6. **enrollment url** *url*
7. **revocation-check** *method*
8. **exit**
9. **authorization username** {**subjectname** *subjectname*}
10. **authorization list** *listname*
11. **tacacs-server host** *hostname* [**key** *string*]

    or

    radius-server host *hostname* [**key** *string*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa authorization network** *listname* [*method*] **Example:** `Router (config)# aaa authorization network maxaaa group tacacs+`	Sets the parameters that restrict user access to a network. • *method*—Can be **group radius**, **group tacacs+**, or **group group-name**.
Step 5	**crypto pki trustpoint** *name* **Example:** `Route (config)# crypto pki trustpoint msca`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 6	**enrollment url** *url* **Example:** `Router (ca-trustpoint)# enrollment url http://caserver.mycompany.com`	Specifies the enrollment parameters of your CA. • The *url* argument is the URL of the CA to which your router should send certificate requests.
Step 7	**revocation-check** *method* **Example:** `Router (ca-trustpoint)# revocation-check crl`	(Optional) Checks the revocation status of a certificate.
Step 8	**exit** **Example:** `Router (ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 9	**authorization username** {**subjectname** *subjectname*} **Example:** `Router (config)# authorization username subjectname serialnumber`	Sets parameters for the different certificate fields that are used to build the AAA username. The *subjectname* argument can be any of the following: • **all**—Entire distinguished name (subject name) of the certificate. • **commonname**—Certification common name. • **country**—Certificate country. • **email**—Certificate e-mail. • **ipaddress**—Certificate IP address. • **locality**—Certificate locality. • **organization**—Certificate organization. • **organizationalunit**—Certificate organizational unit. • **postalcode**—Certificate postal code. • **serialnumber**—Certificate serial number. • **state**—Certificate state field. • **streetaddress**—Certificate street address. • **title**—Certificate title. • **unstructuredname**—Certificate unstructured name.
Step 10	**authorization list** *listname* **Example:** `Route (config)# authorization list maxaaa`	Specifies the AAA authorization list.
Step 11	**tacacs-server host** *hostname* [**key** *string*] **Example:** `Router(config)# tacacs-server host 192.0.2.2 key a_secret_key` or **radius-server host** *hostname* [**key** *string*] **Example:** `Router(config)# radius-server host 192.0.2.1 key another_secret_key`	Specifies a TACACS+ host. or Specifies a RADIUS host.

Troubleshooting Tips

To display debug messages for the trace of interaction (message type) between the CA and the router, use the **debug crypto pki transactions** command. (See the sample output, which shows a successful PKI integration with AAA server exchange and a failed PKI integration with AAA server exchange.)

Successful Exchange

```
Router# debug crypto pki transactions

Apr 22 23:15:03.695: CRYPTO_PKI: Found a issuer match
Apr 22 23:15:03.955: CRYPTO_PKI: cert revocation status unknown.
Apr 22 23:15:03.955: CRYPTO_PKI: Certificate validated without revocation check
```

Each line that shows “CRYPTO_PKI_AAA” indicates the state of the AAA authorization checks. Each of the AAA AV pairs is indicated, and then the results of the authorization check are shown.

```
Apr 22 23:15:04.019: CRYPTO_PKI_AAA: checking AAA authorization (ipsecca_script_aaalist,
PKIAAA-L, <all>)
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-serial" = "15DE")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: authorization passed
Apr 22 23:12:30.327: CRYPTO_PKI: Found a issuer match
```

Failed Exchange

```
Router# debug crypto pki transactions

Apr 22 23:11:13.703: CRYPTO_PKI_AAA: checking AAA authorization =
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-application" = “all”)
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint"= “CA1”)
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-serial" = “233D”)
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: parsed cert-lifetime-end as: 21:30:00
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: timezone specific extended
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end is expired
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end check failed.
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: authorization failed
```

In the above failed exchange, the certificate has expired.

Configuring a Revocation Mechanism for PKI Certificate Status Checking

Perform this task to set up a CRL as the certificate revocation mechanism—CRLs or OCSP—that is used to check the status of certificates in a PKI.

The revocation-check Command

Use the **revocation-check** command to specify at least one method (OCSP, CRL, or skip the revocation check) that is to be used to ensure that the certificate of a peer has not been revoked. For multiple methods, the order in which the methods are applied is determined by the order specified via this command.

If your router does not have the applicable CRL and is unable to obtain one or if the OCSP server returns an error, your router will reject the peer’s certificate—unless you include the **none** keyword in your configuration. If the **none** keyword is configured, a revocation check will not be performed and the certificate will always be accepted.

Nonces and Peer Communications with OCSP Servers

When using OCSP, nonces, unique identifiers for OCSP requests, are sent by default during peer communications with your OCSP server. The use of nonces offers a more secure and reliable communication channel between the peer and OCSP server.

If your OCSP server does not support nonces, you may disable the sending of nonces. For more information, check your OCSP server documentation.

Prerequisites

- Before issuing any client certificates, the appropriate settings on the server (such as setting the CDP) should be configured.
- When configuring an OCSP server to return the revocation status for a CA server, the OCSP server must be configured with an OCSP response signing certificate that is issued by that CA server. Ensure that the signing certificate is in the correct format, or the router will not accept the OCSP response. See your OCSP manual for additional information.

Restrictions

- OCSP transports messages over HTTP, so there may be a time delay when you access the OCSP server.
- If the OCSP server depends on normal CRL processing to check revocation status, the same time delay that affects CRLs will also apply to OCSP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **ocsp url** *url*
5. **revocation-check** *method1* [*method2* [*method3*]]
6. **ocsp disable-nonce**
7. **exit**
8. **exit**
9. **show crypto pki certificates**
10. **show crypto pki trustpoints** [**status** | *label* [**status**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint hazel`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**ocsp url** *url* **Example:** `Router(ca-trustpoint)# ocsp url http://ocsp-server`	(Optional) Specifies the URL of an OCSP server so that the trustpoint can check the certificate status. This URL will override the URL of the OCSP server (if one exists) in the Authority Info Access (AIA) extension of the certificate.
Step 5	**revocation-check** *method1* [*method2* [*method3*]] **Example:** `Router(ca-trustpoint)# revocation-check ocsp none`	Checks the revocation status of a certificate. • **crl**—Certificate checking is performed by a CRL. This is the default option. • **none**—Certificate checking is ignored. • **ocsp**—Certificate checking is performed by an OCSP server. If a second and third method are specified, each method will be used only if the previous method returns an error, such as a server being down.
Step 6	`ocsp disable-nonce` **Example:** `Router(ca-trustpoint)# ocsp disable-nonce`	(Optional) Specifies that a nonce, or an OCSP request unique identifier, will not be sent during peer communications with the OCSP server.
Step 7	**exit** **Example:** `Router(ca-trustpoint)# exit`	Returns to global configuration mode.
Step 8	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 9	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates.
Step 10	**show crypto pki trustpoints** [**status** | *label* [**status**]] **Example:** `Router# show crypto pki trustpoints`	Displays information about the trustpoint configured in router.

Configuring Certificate Authorization and Revocation Settings

Perform this task to specify a certificate-based ACL, to ignore revocation checks or expired certificates, to manually override the default CDP location, to manually override the OCSP server setting, to configure CRL caching, or to set session acceptance or rejection based on a certificate serial number, as appropriate.

Configuring Certificate-Based ACLs to Ignore Revocation Checks

To configure your router to use certificate-based ACLs to ignore revocation checks and expired certificates, perform the following steps:

- Identify an existing trustpoint or create a new trustpoint to be used when verifying the certificate of the peer. Authenticate the trustpoint if it has not already been authenticated. The router may enroll with this trustpoint if you want. Do not set optional CRLs for the trustpoint if you plan to use the **match certificate** command and **skip revocation-check** keyword.
- Determine the unique characteristics of the certificates that should not have their CRL checked and of the expired certificates that should be allowed.
- Define a certificate map to match the characteristics identified in the prior step.
- You can add the **match certificate** command and **skip revocation-check** keyword and the **match certificate command** and **allow expired-certificate** keyword to the trustpoint that was created or identified in the first step.

Manually Overriding CDPs in a Certificate

Users can override the CDPs in a certificate with a manually configured CDP. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.

Manually Overriding the OCSP Server Setting in a Certificate

Administrators can override the OCSP server setting specified in the Authority Information Access (AIA) field of the client certificate or set by the issuing the **ocsp url** command. One or more OCSP servers may be manually specified, either per client certificate or per group of client certificates by the

match certificate override ocsp command. The **match certificate override ocsp** command overrides the client certificate AIA field or the **ocsp url** command setting if a client certificate is successfully matched to a certificate map during the revocation check.

Note Only one OCSP server can be specified per client certificate.

Configuring CRL Cache Control

By default, a new CRL will be downloaded after the currently cached CRL expires. Administrators can either configure the maximum amount of time in minutes a CRL remains in the cache by issuing the **crl cache delete-after** command or disable CRL caching by issuing the **crl cache none** command. Only the **crl-cache delete-after** command or the **crl-cache none** command may be specified. If both commands are entered for a trustpoint, the last command executed will take effect and a message will be displayed.

Neither the **crl-cache none** command nor the **crl-cache delete-after** command affects the currently cached CRL. If you configure the **crl-cache none** command, all CRLs downloaded after this command is issued will not be cached. If you configure the **crl-cache delete-after** command, the configured lifetime will only affect CRLs downloaded after this command is issued.

This functionality is useful is when a CA issues CRLs with no expiration date or with expiration dates days or weeks ahead.

Configuring Certificate Serial Number Session Control

A certificate serial number can be specified to allow a certificate validation request to be accepted or rejected by the trustpoint for a session. A session may be rejected, depending on certificate serial number session control, even if a certificate is still valid. Certificate serial number session control may be configured by using either a certificate map with the **serial-number** field or an AAA attribute, with the **cert-serial-not** command.

Using certificate maps for session control allows an administrator to specify a single certificate serial number. Using the AAA attribute allows an administrator to specify one or more certificate serial numbers for session control.

Prerequisites

- The trustpoint should be defined and authenticated before attaching certificate maps to the trustpoint.
- The certificate map must be configured before the CDP override feature can be enabled or the **serial-number** command is issued.
- The PKI and AAA server integration must be successfully completed to use AAA attributes as described in "PKI and AAA Server Integration for Certificate Status."

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki certificate map** *label sequence-number*
4. *field-name match-criteria match-value*

5. **exit**
6. **crypto pki trustpoint** *name*
7. **crl-cache none**
8. **crl-cache delete-after** *time*
9. **match certificate** *certificate-map-label* [**allow expired-certificate** | **skip revocation-check** | **skip authorization-check**]
10. **match certificate** *certificate-map-label* **override cdp** {**url** | **directory**} *string*
11. **match certificate** *certificate-map-label* **override ocsp** [**trustpoint** *trustpoint-label*] *sequence-number* **url** *ocsp-url*
12. **exit**
13. **aaa new-model**
14. **aaa attribute list** *list-name*
15. **attribute type** {*name*} {*label*}
16. **exit**
17. **exit**
18. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki certificate map** *label sequence-number* **Example:** `Router(config)# crypto pki certificate map Group 10`	Defines values in a certificate that should be matched or not matched and enters ca-certificate-map configuration mode.

	Command or Action	Purpose
Step 4	*field-name match-criteria match-value* **Example:** `Router(ca-certificate-map)# subject-name co MyCompany`	Specifies one or more certificate fields together with their matching criteria and the value to match. The *field-name* is one of the following case-insensitive name strings or a date: • **alt-subject-name** • **expires-on** • **issuer-name** • **name** • serial-number • **subject-name** • **unstructured-subject-name** • **valid-start** **Note** Date field format is dd mm yyyy hh:mm:ss or mmm dd yyyy hh:mm:ss. The *match-criteria* is one of the following logical operators: • **co** —contains (valid only for name fields and serial number field) • **eq** —equal (valid for name, serial number, and date fields) • **ge** —greater than or equal (valid only for date fields) • **lt** —less than (valid only for date fields) • **nc** —does not contain (valid only for name fields and serial number field) • **ne** —not equal (valid for name, serial number, and date fields) The *match-value* is the name or date to test with the logical operator assigned by match-criteria. **Note** Use this command only when setting up a certificate-based ACL—not when setting up a certificate-based ACL to ignore revocation checks or expired certificates.
Step 5	**exit** **Example:** `Router(ca-certificate-map)# exit`	Returns to global configuration mode.
Step 6	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint Access2`	Declares the trustpoint, given name and enters ca-trustpoint configuration mode.

	Command or Action	Purpose
Step 7	**crl-cache none** **Example:** `Router(ca-trustpoint)# crl-cache none`	(Optional) Disables CRL caching completely for all CRLs associated with the trustpoint. The **crl-cache none** command does not affect any currently cached CRLs. All CRLs downloaded after this command is configured will not be cached.
Step 8	**crl-cache delete-after** *time* **Example:** `Router(ca-trustpoint)# crl-cache delete-after 2`	(Optional) Specifies the maximum time CRLs will remain in the cache for all CRLs associated with the trustpoint. • *time*—The amount of time in minutes before the CRL is deleted. The **crl-cache delete-after** command does not affect any currently cached CRLs. The configured lifetime will only affect CRLs downloaded after this command is configured.
Step 9	**match certificate** *certificate-map-label* [**allow expired-certificate** | **skip revocation-check** | **skip authorization-check**] **Example:** `Router(ca-trustpoint)# match certificate Group skip revocation-check`	(Optional) Associates the certificate-based ACL (that was defined via the **crypto pki certificate map** command) to a trustpoint. • *certificate-map-label*—Must match the *label* argument specified via the **crypto pki certificate map** command. • **allow expired-certificate**—Ignores expired certificates. • **skip revocation-check**—Allows a trustpoint to enforce CRLs except for specific certificates. • **skip authorization-check**—Skips the AAA check of a certificate when PKI integration with an AAA server is configured.
Step 10	**match certificate** *certificate-map-label* **override cdp** {**url** | **directory**} *string* **Example:** `Router(ca-trustpoint)# match certificate Group1 override cdp url http://server.cisco.com`	(Optional) Manually overrides the existing CDP entries for a certificate with a URL or directory specification. • *certificate-map-label*—A user-specified label that must match the *label* argument specified in a previously defined **crypto pki certificate map** command. • **url**—Specifies that the certificate's CDPs will be overridden with an HTTP or LDAP URL. • **directory**—Specifies that the certificate's CDPs will be overridden with an LDAP directory specification. • *string*—The URL or directory specification. **Note** Some applications may time out before all CDPs have been tried and will report an error message. The error message will not affect the router, and the Cisco IOS software will continue attempting to retrieve a CRL until all CDPs have been tried.

	Command or Action	Purpose
Step 11	**match certificate** *certificate-map-label* **override ocsp** [**trustpoint** *trustpoint-label*] *sequence-number* **url** *ocsp-url* **Example:** `Router(ca-trustpoint)# match certificate mycertmapname override ocsp trustpoint mytp 15 url http://192.0.2.2`	(Optional) Specifies an OCSP server, either per client certificate or per group of client certificates, and may be issued more than once to specify additional OCSP servers and client certificate settings including alternative PKI hierarchies. • *certificate-map-label*—The name of an existing certificate map. • **trustpoint**—The trustpoint to be used when validating the OCSP server certificate. • *sequence-number*—The order the **match certificate override ocsp** command statements apply to the certificate being verified. Matches are performed from the lowest sequence number to the highest sequence number. If more than one command is issued with the same sequence number, it overwrites the previous OCSP server override setting. • **url**—The URL of the OCSP server. When the certificate matches a configured certificate map, the AIA field of the client certificate and any previously issued **ocsp url** command settings are overwritten with the specified OCSP server. If no map-based match occurs, one of the following two cases will continue to apply to the client certificate. • If OCSP is specified as the revocation method, the AIA field value will continue to apply to the client certificate. • If the **ocsp url** configuration exists, the **ocsp url** configuration settings will continue to apply to the client certificates.
Step 12	**exit** **Example:** `Router(ca-trustpoint)# exit`	Returns to global configuration mode.
Step 13	**aaa new-model** **Example:** `Router(config)# aaa new-model`	(Optional) Enables the AAA access control model.
Step 14	**aaa attribute list** *list-name* **Example:** `Router(config)# aaa attribute list crl`	(Optional) Defines an AAA attribute list locally on a router and enters config-attr-list configuration mode.

	Command or Action	Purpose
Step 15	**attribute type** {*name*}{*value*} **Example:** `Router(config-attr-list)# attribute type cert-serial-not 6C4A`	(Optional) Defines an AAA attribute type that is to be added to an AAA attribute list locally on a router. To configure certificate serial number session control, an administrator may specify a specific certificate in the *value* field to be accepted or rejected based on its serial number where *name* is set to **cert-serial-not**. If the serial number of the certificate matches the serial number specified by the attribute type setting, the certificate will be rejected. For a full list of available AAA attribute types, execute the **show aaa attributes** command.
Step 16	**exit** **Example:** `Router(ca-trustpoint)# exit` **Example:** `Router(config-attr-list)# exit`	Returns to global configuration mode.
Step 17	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 18	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays the components of the certificates installed on the router if the CA certificate has been authenticated.

Examples

The following is a sample OCSP response when signing a certificate. The OCSP-related extensions are in bold.

```
Certificate:
        Data:
            Version: v3
            Serial Number:0x14
            Signature Algorithm:MD5withRSA - 1.2.840.113549.1.1.4
            Issuer:CN=CA server,OU=PKI,O=Cisco Systems
            Validity:
                Not Before:Thursday, August 8, 2002 4:38:05 PM PST
                Not After:Tuesday, August 7, 2003 4:38:05 PM PST
            Subject:CN=OCSP server,OU=PKI,O=Cisco Systems
            Subject Public Key Info:
                Algorithm:RSA - 1.2.840.113549.1.1.1
                Public Key:
                    Exponent:65537
                    Public Key Modulus:(1024 bits) :
                        <snip>

            Extensions:
                Identifier:Subject Key Identifier - 2.5.29.14
                    Critical:no
```

```
                    Key Identifier:
                        <snip>
                Identifier:Authority Key Identifier - 2.5.29.35
                    Critical:no
                    Key Identifier:
                        <snip>

                Identifier:OCSP NoCheck:- 1.3.6.1.5.5.7.48.1.5
                     Critical:no
                Identifier:Extended Key Usage:- 2.5.29.37
                     Critical:no
                     Extended Key Usage:
                     OCSPSigning
                Identifier:CRL Distribution Points - 2.5.29.31
                    Critical:no
                    Number of Points:1
                    Point 0
                        Distribution Point:
[URIName:ldap://CA-server/CN=CA server,OU=PKI,O=Cisco Systems]
        Signature:
            Algorithm:MD5withRSA - 1.2.840.113549.1.1.4
            Signature:
       <snip>
```

The following example shows an excerpt of the running configuration output when adding a **match certificate override ocsp** command to the beginning of an existing sequence:

```
match certificate map3 override ocsp 5 url http://192.0.2.3/
show running-configuration
.
.
.
        match certificate map3 override ocsp 5 url http://192.0.2.3/
        match certificate map1 override ocsp 10 url http://192.0.2.1/
        match certificate map2 override ocsp 15 url http://192.0.2.2/
```

The following example shows an excerpt of the running configuration output when an existing **match certificate override ocsp** command is replaced and a trustpoint is specified to use an alternative PKI hierarchy:

```
match certificate map4 override ocsp trustpoint tp4 10 url http://192.0.2.4/newvalue
show running-configuration
.
.
.
        match certificate map3 override ocsp trustpoint tp3 5 url http://192.0.2.3/
        match certificate map1 override ocsp trustpoint tp1 10 url http://192.0.2.1/
        match certificate map4 override ocsp trustpoint tp4 10 url
              http://192.0.2.4/newvalue
        match certificate map2 override ocsp trustpoint tp2 15 url http://192.0.2.2/
```

Troubleshooting Tips

If you ignored revocation check or expired certificates, you should carefully check your configuration. Verify that the certificate map properly matches either the certificate or certificates that should be allowed or the AAA checks that should be skipped. In a controlled environment, try modifying the certificate map and determine what is not working as expected.

Configuring Certificate Chain Validation

Perform this task to configure the processing level for the certificate chain path of your peer certificates.

Prerequisites

- The device must be enrolled in your PKI hierarchy.
- The appropriate key pair must be associated with the certificate.

Restrictions

- A trustpoint associated with the root CA cannot be configured to be validated to the next level.

  The **chain-validation** command is configured with the **continue** keyword for the trustpoint associated with the root CA, an error message will be displayed and the chain validation will revert to the default **chain-validation** command setting.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **chain-validation** [{**stop** | **continue**} [*parent-trustpoint*]]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint ca-sub1`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**chain-validation** [{**stop** | **continue**} [*parent-trustpoint*]] **Example:** `Router(ca-trustpoint)# chain-validation continue ca-sub1`	Configures the level to which a certificate chain is processed on all certificates including subordinate CA certificates. • Use the **stop** keyword to specify that the certificate is already trusted. This is the default setting. • Use the **continue** keyword to specify that the that the subordinate CA certificate associated with the trustpoint must be validated. • The *parent-trustpoint* argument specifies the name of the parent trustpoint the certificate must be validated against.
Step 5	**exit** **Example:** `Router(ca-trustpoint)# exit`	Returns to global configuration mode

Configuration Examples for Setting Up Authorization and Revocation of Certificates

This section contains the following configuration examples:

Configuring and Verifying PKI AAA Authorization: Examples

This section provides configuration examples of PKI AAA authorizations:

Router Configuration: Example

The following **show running-config** command output shows the working configuration of a router that is set up to authorize VPN connections using the PKI Integration with AAA Server feature:

```
Router# show running-config

Building configuration...
!
version 12.3
!
hostname router7200router7200
!
aaa new-model
!
!
aaa authentication login default group tacacs+
aaa authentication login no_tacacs enable
aaa authentication ppp default group tacacs+
aaa authorization exec ACSLab group tacacs+
aaa authorization network ACSLab group tacacs+
aaa accounting exec ACSLab start-stop group tacacs+
aaa accounting network default start-stop group ACSLab
aaa session-id common
!
ip domain name company.com
!
crypto pki trustpoint EM-CERT-SERV
 enrollment url http://192.0.2.33:80
 serial-number
 crl optional
 rsakeypair STOREVPN 1024
 auto-enroll
 authorization list ACSLab
!
crypto pki certificate chain EM-CERT-SERV
 certificate 04
  30820214 3082017D A0030201 02020104 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 454D2D43 4552542D 53455256 301E170D 30343031
  31393232 30323535 5A170D30 35303131 38323230 3235355A 3030312E 300E0603
  55040513 07314437 45424434 301C0609 2A864886 F70D0109 02160F37 3230302D
  312E6772 696C2E63 6F6D3081 9F300D06 092A8648 86F70D01 01010500 03818D00
  30818902 818100BD F3B837AA D925F391 2B64DA14 9C2EA031 5A7203C4 92F8D6A8
  7D2357A6 BCC8596F A38A9B10 47435626 D59A8F2A 123195BB BE5A1E74 B1AA5AE0
  5CA162FF 8C3ACA4F B3EE9F27 8B031642 B618AE1B 40F2E3B4 F996BEFE 382C7283
  3792A369 236F8561 8748AA3F BC41F012 B859BD9C DB4F75EE 3CEE2829 704BD68F
  FD904043 0F555702 03010001 A3573055 30250603 551D1F04 1E301C30 1AA018A0
  16861468 7474703A 2F2F3633 2E323437 2E313037 2E393330 0B060355 1D0F0404
  030205A0 301F0603 551D2304 18301680 1420FC4B CF0B1C56 F5BD4C06 0AFD4E67
  341AE612 D1300D06 092A8648 86F70D01 01040500 03818100 79E97018 FB955108
  12F42A56 2A6384BC AC8E22FE F1D6187F DA5D6737 C0E241AC AAAEC75D 3C743F59
  08DEEFF2 0E813A73 D79E0FA9 D62DC20D 8E2798CD 2C1DC3EC 3B2505A1 3897330C
```

```
  15A60D5A 8A13F06D 51043D37 E56E45DF A65F43D7 4E836093 9689784D C45FD61D
  EC1F160C 1ABC8D03 49FB11B1 DA0BED6C 463E1090 F34C59E4
  quit
 certificate ca 01
  30820207 30820170 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 454D2D43 4552542D 53455256 301E170D 30333132
  31363231 34373432 5A170D30 36313231 35323134 3734325A 30173115 30130603
  55040313 0C454D2D 43455254 2D534552 5630819F 300D0609 2A864886 F70D0101
  01050003 818D0030 81890281 8100C14D 833641CF D784F516 DA6B50C0 7B3CB3C9
  589223AB 99A7DC14 04F74EF2 AAEEE8F5 E3BFAE97 F2F980F7 D889E6A1 2C726C69
  54A29870 7E7363FF 3CD1F991 F5A37CFF 3FFDD3D0 9E486C44 A2E34595 C2D078BB
  E9DE981E B733B868 AA8916C0 A8048607 D34B83C0 64BDC101 161FC103 13C06500
  22D6EE75 7D6CF133 7F1B515F 32830203 010001A3 63306130 0F060355 1D130101
  FF040530 030101FF 300E0603 551D0F01 01FF0404 03020186 301D0603 551D0E04
  16041420 FC4BCF0B 1C56F5BD 4C060AFD 4E67341A E612D130 1F060355 1D230418
  30168014 20FC4BCF 0B1C56F5 BD4C060A FD4E6734 1AE612D1 300D0609 2A864886
  F70D0101 04050003 81810085 D2E386F5 4107116B AD3AC990 CBE84063 5FB2A6B5
  BD572026 528E92ED 02F3A0AE 1803F2AE AA4C0ED2 0F59F18D 7B50264F 30442C41
  0AF19C4E 70BD3CB5 0ADD8DE8 8EF636BD 24410DF4 DB62DAFC 67DA6E58 3879AA3E
  12AFB1C3 2E27CB27 EC74E1FC AEE2F5CF AA80B439 615AA8D5 6D6DEDC3 7F9C2C79
  3963E363 F2989FB9 795BA8
  quit
!
!
crypto isakmp policy 10
 encr 3des
 group 2
!
!
crypto ipsec transform-set ISC_TS_1 esp-3des esp-sha-hmac
!
crypto ipsec profile ISC_IPSEC_PROFILE_2
 set security-association lifetime kilobytes 530000000
 set security-association lifetime seconds 14400
 set transform-set ISC_TS_1
!
!
controller ISA 1/1
!
!
interface Tunnel0
 description MGRE Interface provisioned by ISC
 bandwidth 10000
 ip address 192.0.2.172 255.255.255.0
 no ip redirects
 ip mtu 1408
 ip nhrp map multicast dynamic
 ip nhrp network-id 101
 ip nhrp holdtime 500
 ip nhrp server-only
 no ip split-horizon eigrp 101
 tunnel source FastEthernet2/1
 tunnel mode gre multipoint
 tunnel key 101
 tunnel protection ipsec profile ISC_IPSEC_PROFILE_2
!
interface FastEthernet2/0
 ip address 192.0.2.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet2/1
 ip address 192.0.2.2 255.255.255.0
 duplex auto
```

```
 speed auto
!
!
tacacs-server host 192.0.2.55 single-connection
tacacs-server directed-request
tacacs-server key company lab
!
ntp master 1
!
end
```

Debug of a Successful PKI AAA Authorization: Example

The following **show debugging** command output shows a successful authorization using the PKI Integration with AAA Server feature:

```
Router# show debugging

General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
Crypto PKI Trans debugging is on

Router#
May 28 19:36:11.117: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:36:12.789: CRYPTO_PKI: Found a issuer match
May 28 19:36:12.805: CRYPTO_PKI: cert revocation status unknown.
May 28 19:36:12.805: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:36:12.813: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.company.com,
<all>)
May 28 19:36:12.813: AAA/BIND(00000042): Bind i/f
May 28 19:36:12.813: AAA/AUTHOR (0x42): Pick method list 'ACSLab'
May 28 19:36:12.813: TPLUS: Queuing AAA Authorization request 66 for processing
May 28 19:36:12.813: TPLUS: processing authorization request id 66
May 28 19:36:12.813: TPLUS: Protocol set to None .....Skipping
May 28 19:36:12.813: TPLUS: Sending AV service=pki
May 28 19:36:12.813: TPLUS: Authorization request created for 66(POD5.company.com)
May 28 19:36:12.813: TPLUS: Using server 192.0.2.55
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT/203A4628: Started 5 sec timeout
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT: wrote entire 46 bytes request
May 28 19:36:12.813: TPLUS: Would block while reading pak header
May 28 19:36:12.817: TPLUS(00000042)/0/READ: read entire 12 header bytes (expect 27 bytes)
May 28 19:36:12.817: TPLUS(00000042)/0/READ: read entire 39 bytes response
May 28 19:36:12.817: TPLUS(00000042)/0/203A4628: Processing the reply packet
May 28 19:36:12.817: TPLUS: Processed AV cert-application=all
May 28 19:36:12.817: TPLUS: received authorization response for 66: PASS
May 28 19:36:12.817: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
May 28 19:36:12.817: CRYPTO_PKI_AAA: authorization passed
Router#
Router#
May 28 19:36:18.681: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 101: Neighbor 192.0.2.171 (Tunnel0) is
up: new adjacency
Router#

Router# show crypto isakmp sa

dst             src             state          conn-id slot
192.0.2.22    192.0.2.102   QM_IDLE             84    0
```

Debugs of a Failed PKI AAA Authorization: Example

The following **show debugging** command output shows that the router is not authorized to connect using VPN. The messages are typical of those that you might see in such a situation.

In this example, the peer username was configured as not authorized, by moving the username to a Cisco Secure ACS group called VPN_Router_Disabled in Cisco Secure ACS. The router, router7200.company.com, has been configured to check with a Cisco Secure ACS AAA server prior to establishing a VPN connection to any peer.

```
Router# show debugging

General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
  Crypto PKI Trans debugging is on

Router#
May 28 19:48:29.837: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:48:31.509: CRYPTO_PKI: Found a issuer match
May 28 19:48:31.525: CRYPTO_PKI: cert revocation status unknown.
May 28 19:48:31.525: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:48:31.533: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.company.com,
<all>)
May 28 19:48:31.533: AAA/BIND(00000044): Bind i/f
May 28 19:48:31.533: AAA/AUTHOR (0x44): Pick method list 'ACSLab'
May 28 19:48:31.533: TPLUS: Queuing AAA Authorization request 68 for processing
May 28 19:48:31.533: TPLUS: processing authorization request id 68
May 28 19:48:31.533: TPLUS: Protocol set to None .....Skipping
May 28 19:48:31.533: TPLUS: Sending AV service=pki
May 28 19:48:31.533: TPLUS: Authorization request created for 68(POD5.company.com)
May 28 19:48:31.533: TPLUS: Using server 192.0.2.55
May 28 19:48:31.533: TPLUS(00000044)/0/NB_WAIT/203A4C50: Started 5 sec timeout
May 28 19:48:31.533: TPLUS(00000044)/0/NB_WAIT: wrote entire 46 bytes request
May 28 19:48:31.533: TPLUS: Would block while reading pak header
May 28 19:48:31.537: TPLUS(00000044)/0/READ: read entire 12 header bytes (expect 6 bytes)
May 28 19:48:31.537: TPLUS(00000044)/0/READ: read entire 18 bytes response
May 28 19:48:31.537: TPLUS(00000044)/0/203A4C50: Processing the reply packet
May 28 19:48:31.537: TPLUS: received authorization response for 68: FAIL
May 28 19:48:31.537: CRYPTO_PKI_AAA: authorization declined by AAA, or AAA server not
found.
May 28 19:48:31.537: CRYPTO_PKI_AAA: No cert-application attribute found. Failing.
May 28 19:48:31.537: CRYPTO_PKI_AAA: authorization failed
May 28 19:48:31.537: CRYPTO_PKI: AAA authorization for list 'ACSLab', and user
'POD5.company.com' failed.
May 28 19:48:31.537: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.0.2.162 is
bad: certificate invalid
May 28 19:48:39.821: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:48:41.481: CRYPTO_PKI: Found a issuer match
May 28 19:48:41.501: CRYPTO_PKI: cert revocation status unknown.
May 28 19:48:41.501: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:48:41.505: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.company.com,
<all>)
May 28 19:48:41.505: AAA/BIND(00000045): Bind i/f
May 28 19:48:41.505: AAA/AUTHOR (0x45): Pick method list 'ACSLab'
May 28 19:48:41.505: TPLUS: Queuing AAA Authorization request 69 for processing
May 28 19:48:41.505: TPLUS: processing authorization request id 69
May 28 19:48:41.505: TPLUS: Protocol set to None .....Skipping
May 28 19:48:41.505: TPLUS: Sending AV service=pki
May 28 19:48:41.505: TPLUS: Authorization request created for 69(POD5.company.com)
May 28 19:48:41.505: TPLUS: Using server 198.168.244.55
```

```
May 28 19:48:41.509: TPLUS(00000045)/0/IDLE/63B22834: got immediate connect on new 0
May 28 19:48:41.509: TPLUS(00000045)/0/WRITE/63B22834: Started 5 sec timeout
May 28 19:48:41.509: TPLUS(00000045)/0/WRITE: wrote entire 46 bytes request
May 28 19:48:41.509: TPLUS(00000045)/0/READ: read entire 12 header bytes (expect 6 bytes)
May 28 19:48:41.509: TPLUS(00000045)/0/READ: read entire 18 bytes response
May 28 19:48:41.509: TPLUS(00000045)/0/63B22834: Processing the reply packet
May 28 19:48:41.509: TPLUS: received authorization response for 69: FAIL
May 28 19:48:41.509: CRYPTO_PKI_AAA: authorization declined by AAA, or AAA server not
found.
May 28 19:48:41.509: CRYPTO_PKI_AAA: No cert-application attribute found. Failing.
May 28 19:48:41.509: CRYPTO_PKI_AAA: authorization failed
May 28 19:48:41.509: CRYPTO_PKI: AAA authorization for list 'ACSLab', and user
'POD5.company.com' failed.
May 28 19:48:41.509: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.0.2.162 is
bad: certificate invalid
Router#

Router# show crypto iskmp sa

dst             src             state          conn-id slot
192.0.2.2       192.0.2.102   MM_KEY_EXCH         95    0
```

Configuring a Revocation Mechanism: Examples

This section contains the following configuration examples that can be used when specifying a revocation mechanism for your PKI:

Configuring an OCSP Server: Example

The following example shows how to configure the router to use the OCSP server that is specified in the AIA extension of the certificate:

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check ocsp
```

Specifying a CRL and Then an OCSP Server: Example

The following example shows how to configure the router to download the CRL from the CDP. If the CRL is unavailable, the OCSP server that is specified in the AIA extension of the certificate will be used. If both options fail, certificate verification will also fail.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check crl ocsp
```

Specifying an OCSP Server: Example

The following example shows how to configure your router to use the OCSP server at the HTTP URL “http://myocspserver:81.” If the server is down, the revocation check will be ignored.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
```

Disabling Nonces in Communications with the OCSP Server: Example

The following example shows communications when a nonce, or a unique identifier for the OCSP request, is disabled for communications with the OCSP server:

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
Router(ca-trustpoint)# ocsp disable-nonce
```

Configuring a Hub Router at a Central Site for Certificate Revocation Checks: Example

The following example shows a hub router at a central site that is providing connectivity for several branch offices to the central site.

The branch offices are also able to communicate directly with each other using additional IPSec tunnels between the branch offices.

The CA publishes CRLs on an HTTP server at the central site. The central site checks CRLs for each peer when setting up an IPSec tunnel with that peer.

The example does not show the IPSec configuration—only the PKI-related configuration is shown.

Home Office Hub Configuration

```
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
```

Central Site Hub Router

```
Router# show crypto ca certificate

Certificate
  Status: Available
  Certificate Serial Number: 2F62BE14000000000CA0
  Certificate Usage: General Purpose
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    Name: Central VPN Gateway
    cn=Central VPN Gateway
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 00:43:26 GMT Sep 26 2003
    end   date: 00:53:26 GMT Sep 26 2004
    renew date: 00:00:00 GMT Jan 1 1970
  Associated Trustpoints: VPN-GW
```

```
CA Certificate
  Status: Available
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
  Certificate Usage: Signature
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    cn=Central Certificate Authority
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017
  Associated Trustpoints: VPN-GW
```

Trustpoint on the Branch Office Router

```
crypto pki trustpoint home-office
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none

ip-address none
 subject-name o=Home Office Inc,cn=Branch 1
 revocation-check crl
```

A certificate map is entered on the branch office router.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
branch1(config)# crypto pki certificate map central-site 10
branch1(ca-certificate-map)#
```

The output from the **show certificate** command on the central site hub router shows that the certificate was issued by the following:

cn=Central Certificate Authority

o=Home Office Inc

These two lines are combined into one line using a comma (,) to separate them, and the original lines are added as the first criteria for a match.

```
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home
Office Inc
!The above line wrapped but should be shown on one line with the line above it.
```

The same combination is done for the subject name from the certificate on the central site router (note that the line that begins with “Name:” is not part of the subject name and must be ignored when creating the certificate map criteria). This is the subject name to be used in the certificate map.

cn=Central VPN Gateway

o=Home Office Inc

```
Router (ca-certificate-map)# subject-name eq cn=central vpn gateway, o=home office inc
```

Now the certificate map is added to the trustpoint that was configured earlier.

```
Router (ca-certificate-map)# crypto pki trustpoint home-office
Router (ca-trustpoint)# match certificate central-site skip revocation-check
Router (ca-trustpoint)# exit
Router (config)# exit
```

The configuration is checked (most of configuration is not shown).

```
Router# write term
!Many lines left out
.
.
.
crypto pki trustpoint home-office
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Branch 1
 revocation-check crl
 match certificate central-site skip revocation-check
!
!
crypto pki certificate map central-site 10
 issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
 subject-name eq cn = central vpn gateway, o = home office inc
!many lines left out
```

Note that the issuer-name and subject-name lines have been reformatted to make them consistent for later matching with the certificate of the peer.

If the branch office is checking the AAA, the trustpoint will have lines similar to the following:

```
crypto pki trustpoint home-office
 auth list allow_list
 auth user subj commonname
```

After the certificate map has been defined as was done above, the following command is added to the trustpoint to skip AAA checking for the central site hub.

```
match certificate central-site skip authorization-check
```

In both cases, the branch site router has to establish an IPSec tunnel to the central site to check CRLs or to contact the AAA server. However, without the **match certificate** command and **central-site skip authorization-check (argument and keyword)**, the branch office cannot establish the tunnel until it has checked the CRL or the AAA server. (The tunnel will not be established unless the **match certificate** command and **central-site skip authorization-check** argument and keyword are used.)

The **match certificate** command and **allow expired-certificate** keyword would be used at the central site if the router at a branch site had an expired certificate and it had to establish a tunnel to the central site to renew its certificate.

Trustpoint on the Central Site Router

```
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
```

Trustpoint on the Branch 1 Site Router

```
Router# show crypto ca certificate

Certificate
  Status: Available
  Certificate Serial Number: 2F62BE14000000000CA0
```

```
  Certificate Usage: General Purpose
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    Name: Branch 1 Site
    cn=Branch 1 Site
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 00:43:26 GMT Sep 26 2003
    end   date: 00:53:26 GMT Oct 3 2003
    renew date: 00:00:00 GMT Jan 1 1970
  Associated Trustpoints: home-office
CA Certificate
  Status: Available
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
  Certificate Usage: Signature
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    cn=Central Certificate Authority
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017
  Associated Trustpoints: home-office
```

A certificate map is entered on the central site router.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router (config)# crypto pki certificate map branch1 10
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home
Office Inc
!The above line wrapped but should be part of the line above it.
Router (ca-certificate-map)# subject-name eq cn=Brahcn 1 Site,o=home office inc
```

The certificate map is added to the trustpoint.

```
Router (ca-certificate-map)# crypto pki trustpoint VPN-GW
Router (ca-trustpoint)# match certificate branch1 allow expired-certificate
Router (ca-trustpoint)# exit
Router (config) #exit
```

The configuration should be checked (most of the configuration is not shown).

```
Router# write term

!many lines left out

crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
 match certificate branch1 allow expired-certificate
!
!
```

```
crypto pki certificate map central-site 10
 issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
 subject-name eq cn = central vpn gateway, o = home office inc
! many lines left out
```

The **match certificate** command and **branch1 allow expired-certificate** (argument and keyword) and the certificate map should be removed as soon as the branch router has a new certificate.

Configuring Certificate Authorization and Revocation Settings: Examples

This section contains the following configuration examples that can be used when specifying a CRL cache control setting or certificate serial number session control:

- Configuring CRL Cache Control, page 36
- Configuring Certificate Serial Number Session Control, page 37

Configuring CRL Cache Control

The following example shows how to disable CRL caching for all CRLs associated with the CA1 trustpoint:

```
crypto pki trustpoint CA1
 enrollment url http://CA1:80
 ip-address FastEthernet0/0
 crl query ldap://ldap_CA1
 revocation-check crl
 crl-cache none
```

The current CRL is still cached immediately after executing the example configuration shown above:

```
Router# show crypto pki crls
CRL Issuer Name:
    cn=name Cert Manager,ou=pki,o=company.com,c=US
    LastUpdate: 18:57:42 GMT Nov 26 2005
    NextUpdate: 22:57:42 GMT Nov 26 2005
    Retrieved from CRL Distribution Point:
      ldap://ldap.company.com/CN=name Cert Manager,O=company.com
```

When the current CRL expires, a new CRL is then downloaded to the router at the next update. The **crl-cache none** command takes effect and all CRLs for the trustpoint are no longer cached; caching is disabled. You can verify that no CRL is cached by executing the **show crypto pki crls** command. No output will be shown because there are no CRLs cached.

The following example shows how to configure the maximum lifetime of 2 minutes for all CRLs associated with the CA1 trustpoint:

```
crypto pki trustpoint CA1
 enrollment url http://CA1:80
 ip-address FastEthernet0/0
 crl query ldap://ldap_CA1
 revocation-check crl
 crl-cache delete-after 2
```

The current CRL is still cached immediately after executing the example configuration above for setting the maximum lifetime of a CRL:

```
Router# show crypto pki crls
CRL Issuer Name:
    cn=name Cert Manager,ou=pki,o=company.com,c=US
```

```
    LastUpdate: 18:57:42 GMT Nov 26 2005
    NextUpdate: 22:57:42 GMT Nov 26 2005
    Retrieved from CRL Distribution Point:
      ldap://ldap.company.com/CN=name Cert Manager,O=company.com
```

When the current CRL expires, a new CRL is downloaded to the router at the next update and the **crl-cache delete-after** command takes effect. This newly cached CRL and all subsequent CRLs will be deleted after a maximum lifetime of 2 minutes.

You can verify that the CRL will be cached for 2 minutes by executing the **show crypto pki crls** command. Note that the NextUpdate time is 2 minutes after the LastUpdate time.

```
Router# show crypto pki crls
CRL Issuer Name:
    cn=name Cert Manager,ou=pki,o=company.com,c=US
    LastUpdate: 22:57:42 GMT Nov 26 2005


    NextUpdate: 22:59:42 GMT Nov 26 2005
    Retrieved from CRL Distribution Point:
      ldap://ldap.company.com/CN=name Cert Manager,O=company.com
```

Configuring Certificate Serial Number Session Control

The following example shows the configuration of certificate serial number session control using a certificate map for the CA1 trustpoint:

```
crypto pki trustpoint CA1
 enrollment url http://CA1
 chain-validation stop
 crl query ldap://ldap_server
 revocation-check crl
 match certificate crl
!
crypto pki certificate map crl 10
 serial-number co 279d
```

Note If the *match-criteria* value is set to **eq** (equal) instead of **co** (contains), the serial number must match the certificate map serial number *exactly*, including any spaces.

The following example shows the configuration of certificate serial number session control using AAA attributes. In this case, all valid certificates will be accepted if the certificate does not have the serial number “4ACA.”

```
crypto pki trustpoint CA1
 enrollment url http://CA1
 ip-address FastEthernet0/0
 crl query ldap://ldap_CA1
 revocation-check crl
 aaa new-model
!
aaa attribute list crl
attribute-type aaa-cert-serial-not 4ACA
```

The server log shows that the certificate with the serial number “4ACA” was rejected. The certificate rejection is shown in bold.

```
.
.
.
```

```
Dec 3 04:24:39.051: CRYPTO_PKI: Trust-Point CA1 picked up
Dec 3 04:24:39.051: CRYPTO_PKI: locked trustpoint CA1, refcount is 1
Dec 3 04:24:39.051: CRYPTO_PKI: unlocked trustpoint CA1, refcount is 0
Dec 3 04:24:39.051: CRYPTO_PKI: locked trustpoint CA1, refcount is 1
Dec 3 04:24:39.135: CRYPTO_PKI: validation path has 1 certs
Dec 3 04:24:39.135: CRYPTO_PKI: Found a issuer match
Dec 3 04:24:39.135: CRYPTO_PKI: Using CA1 to validate certificate
Dec 3 04:24:39.135: CRYPTO_PKI: Certificate validated without revocation check
Dec 3 04:24:39.135: CRYPTO_PKI: Selected AAA username: 'PKIAAA'
Dec 3 04:24:39.135: CRYPTO_PKI: Anticipate checking AAA list:'CRL'
Dec 3 04:24:39.135: CRYPTO_PKI_AAA: checking AAA authorization (CRL, PKIAAA-L1, <all>)
Dec 3 04:24:39.135: CRYPTO_PKI_AAA: pre-authorization chain validation status (0x4)
Dec 3 04:24:39.135: AAA/BIND(00000021): Bind i/f
Dec 3 04:24:39.135: AAA/AUTHOR (0x21): Pick method list 'CRL'
.
.
.

Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-serial-not" = "4ACA")
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: cert-serial doesn't match ("4ACA" != "4ACA")
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: post-authorization chain validation status (0x7)
Dec 3 04:24:39.175: CRYPTO_PKI: AAA authorization for list 'CRL', and user 'PKIAAA'
failed.
Dec 3 04:24:39.175: CRYPTO_PKI: chain cert was anchored to trustpoint CA1, and chain
validation result was: CRYPTO_PKI_CERT_NOT_AUTHORIZED
Dec 3 04:24:39.175: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.0.2.43 is
bad: certificate invalid
Dec 3 04:24:39.175: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Main mode failed with peer
at 192.0.2.43
.
.
.
```

Configuring Certificate Chain Validation: Examples

This section contains the following configuration examples that can be used to specify the level of certificate chain processing for your device certificates:

Configuring Certificate Chain Validation from Peer to Root CA

In the following configuration example, all of the certificates will be validated—the peer, SubCA11, SubCA1, and RootCA certificates.

```
crypto pki trustpoint RootCA
 enrollment terminal
 chain-validation stop
 revocation-check none
 rsakeypair RootCA

crypto pki trustpoint SubCA1
 enrollment terminal
 chain-validation continue RootCA
 revocation-check none
```

```
 rsakeypair SubCA1

crypto pki trustpoint SubCA11
 enrollment terminal
 chain-validation continue SubCA1
 revocation-check none
 rsakeypair SubCA11
```

Configuring Certificate Chain Validation from Peer to Subordinate CA

In the following configuration example, the following certificates will be validated—the peer and SubCA1 certificates.

```
crypto pki trustpoint RootCA
 enrollment terminal
 chain-validation stop
 revocation-check none
 rsakeypair RootCA

crypto pki trustpoint SubCA1
 enrollment terminal
 chain-validation continue RootCA
 revocation-check none
 rsakeypair SubCA1


crypto pki trustpoint SubCA11
 enrollment terminal
 chain-validation continue SubCA1
 revocation-check none
 rsakeypair SubCA11
```

Configuring Certificate Chain Validation Through a Gap

In the following configuration example, SubCA1 is not in the configured Cisco IOS hierarchy but is expected to have been supplied in the certificate chain presented by the peer.

If the peer supplies the SubCA1 certificate in the presented certificate chain, the following certificates will be validated—the peer, SubCA11, and SubCA1 certificates.

If the peer does not supply the SubCA1 certificate in the presented certificate chain, the chain validation will fail.

```
crypto pki trustpoint RootCA
 enrollment terminal
 chain-validation stop
 revocation-check none
 rsakeypair RootCA

crypto pki trustpoint SubCA11
 enrollment terminal
 chain-validation continue RootCA
 revocation-check none
 rsakeypair SubCA11
```

Additional References

The following sections provide references related to PKI certificate authorization and revocation.

Related Documents

Related Topic	Document Title
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference, Release 12.4*
Overview of PKI, including RSA keys, certificate enrollment, and CAs	"Cisco IOS PKI Overview: Understanding and Planning a PKI" module
RSA key generation and deployment	"Deploying RSA Keys Within a PKI" module
Certificate enrollment: supported methods, enrollment profiles, configuration tasks	"Configuring Certificate Enrollment for a PKI" module
Cisco IOS certificate server overview information and configuration tasks	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" module

Technical Assistance

Description	Link
The Cisco Technical Support Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Certificate Authorization and Revocation

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation. For information on a feature in this technology that is not documented here, see the Implementing and Managing PKI Features Roadmap.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for PKI Certificate Authorization and Revocation

Feature Name	Software Releases	Feature Configuration Information
Cache Control Enhancements for Certification Revocation Lists	12.4(9)T	This feature provides users the ability to disable CRL caching or to specify the maximum lifetime for which a CRL will be cached in router memory. It also provides functionality to configure certificate serial number session control. The following sections provide information about this feature: • What Is a CRL? • Configuring Certificate Authorization and Revocation Settings • Configuring Certificate Authorization and Revocation Settings: Examples The following commands were introduced or modified by this feature: **crl-cache delete-after, crl-cache none, crypto pki certificate map**
Certificate-Complete Chain Validation	12.4(6)T	This feature provides users the ability to configure the level to which a certificate chain is processed on all certificates including subordinate CA certificates. The following sections provide information about this feature: • PKI Certificate Chain Validation • Configuring Certificate Chain Validation • Configuring Certificate Chain Validation: Examples The following command was introduced by this feature: **chain-validation**
OCSP - Server Certification from Alternate Hierarchy	12.4(6)T	This feature provides users with the flexibility to specify multiple OCSP servers, either per client certificate or per group of client certificates, and provides the capability for OCSP server validation based on external CA certificates or self-signed certificates. The following sections provide information about this feature: • What Is OCSP? • Configuring Certificate Authorization and Revocation Settings The following command was introduced by this feature: **match certificate override ocsp**

Table 2 Feature Information for PKI Certificate Authorization and Revocation (continued)

Feature Name	Software Releases	Feature Configuration Information
Optional OCSP Nonce	12.2(33)SR 12.4(4)T	This feature provides users with the ability to configure the sending of a nonce, or unique identifier for an OCSP request, during OCSP communications. The following sections provide information about this feature: • What Is OCSP? • Configuring a Revocation Mechanism for PKI Certificate Status Checking • Disabling Nonces in Communications with the OCSP Server: Example
Certificate Security Attribute-Based Access Control	12.2(15)T	Under the IPsec protocol, CA interoperability permits Cisco IOS devices and a CA to communicate so that the Cisco IOS device can obtain and use digital certificates from the CA. Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action. This feature adds fields to the certificate that allow specifying an ACL, creating a certificate-based ACL. The following sections provide information about this feature: • When to Use Certificate-Based ACLs for Authorization or Revocation • Configuring Certificate Authorization and Revocation Settings The following commands were introduced or modified by this feature: **crypto pki certificate map**, **crypto pki trustpoint**, **match certificate**
Online Certificate Status Protocol (OCSP)	12.3(2)T	This feature allows users to enable OCSP instead of CRLs to check certificate status. Unlike CRLs, which provide only periodic certificate status, OCSP can provide timely information regarding the status of a certificate. The following sections provide information about this feature: • CRLs or OCSP Server: Choosing a Certificate Revocation Mechanism • Configuring a Revocation Mechanism for PKI Certificate Status Checking The following commands were introduced by this feature: **ocsp url**, **revocation-check**

Table 2 Feature Information for PKI Certificate Authorization and Revocation (continued)

Feature Name	Software Releases	Feature Configuration Information
PKI AAA Authorization Using the Entire Subject Name	12.3(11)T	This feature provides users with the ability to query the AAA server using the entire subject name from the certificate as a unique AAA username. The following sections provide information about this feature: • Attribute-Value Pairs for PKI and AAA Server Integration • Configuring PKI Integration with a AAA Server The following command was modified by this feature: **authorization username**
PKI Integration with AAA Server	12.3(1)	This feature provides additional scalability for authorization by generating a AAA username from the certificate presented by the peer. A AAA server is queried to determine whether the certificate is authorized for use by the internal component. The authorization is indicated by a component-specified label that must be present in the AV pair for the user. The following sections provide information about this feature: • PKI and AAA Server Integration for Certificate Status • Configuring PKI Integration with a AAA Server The following commands were introduced by this feature: **authorization list**, **authorization username**
PKI: Query Multiple Servers During Certificate Revocation Check	12.3(7)T	This feature introduces the ability for Cisco IOS software to make multiple attempts to retrieve the CRL, allowing operations to continue when a particular server is not available. In addition, the ability to override the CDPs in a certificate with a manually configured CDP has been introduced. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP. The following sections provide information about this feature: • Querying All CDPs During Revocation Check • Manually Overriding CDPs in a Certificate The following command was introduced by this feature: **match certificate override cdp**

Table 2 Feature Information for PKI Certificate Authorization and Revocation (continued)

Feature Name	Software Releases	Feature Configuration Information
PKI: Query Multiple Servers During Certificate Revocation Check	12.3(7)T	This feature introduces the ability for Cisco IOS software to make multiple attempts to retrieve the CRL, allowing operations to continue when a particular server is not available. In addition, the ability to override the CDPs in a certificate with a manually configured CDP has been introduced. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP. The following sections provide information about this feature: • Querying All CDPs During Revocation Check • Manually Overriding CDPs in a Certificate The following command was introduced by this feature: **match certificate override cdp**
Using Certificate ACLs to Ignore Revocation Check and Expired Certificates	12.3(4)T	This feature allows a certificate that meets specified criteria to be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. Certificate ACLs are used to specify the criteria that the certificate must meet to be accepted or to avoid revocation checking. In addition, if AAA communication is protected by a certificate, this feature provides for the AAA checking of the certificate to be ignored. The following sections provide information about this feature: • Ignore Revocation Checks Using a Certificate-Based ACL • Configuring Certificate-Based ACLs to Ignore Revocation Checks The following command was modified by this feature: **match certificate**
Certificate - Security Attribute-Based Access Control	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
OCSP (Online Certificate Status Protocol)	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Optional OCSP Nonce	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Table 2 Feature Information for PKI Certificate Authorization and Revocation (continued)

Feature Name	Software Releases	Feature Configuration Information
PKI AAA Authorization Using the Entire Subject Name	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
PKI Integration with AAA Server	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Query Mode Definition Per Trustpoint	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Query Multiple Servers during Certificate Revocation Check	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Configuring Certificate Enrollment for a PKI

First Published: May 2, 2005
Last Updated: August 21, 2007

Certificate enrollment, which is the process of obtaining a certificate from a certification authority (CA), occurs between the end host that requests the certificate and the CA. Each peer that participates in the public key infrastructure (PKI) must enroll with a CA. This module describes the different methods available for certificate enrollment and how to set up each method for a participating PKI peer.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for PKI Certificate Enrollment" section on page 34.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for PKI Certificate Enrollment

Before configuring peers for certificate enrollment, you should have the following items:

- A generated Rivest, Shamir, and Adelman (RSA) key pair to enroll and a PKI in which to enroll.
- Your CA should be authenticated.
- Familiarity with the module "Cisco IOS PKI Overview: Understanding and Planning a PKI."

Note As of Cisco IOS Release 12.3(7)T, all commands that begin with "**crypto ca**" have been changed to begin with "**crypto pki**." Although the router will still accept **crypto ca** commands, all output will be read back as **crypto pki**.

Information About Certificate Enrollment for a PKI

Before configuring peers to request a certificate and enroll in the PKI, you should understand the following concepts:

What Are CAs?

A CA manages certificate requests and issues certificates to participating network devices. These services (managing certificate requests and issuing certificates) provide centralized key management for the participating devices to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

You can use the Cisco IOS certificate server or a CA provided by a third-party CA vendor.

Hierarchical PKI: Multiple CAs

A PKI can be set up in a hierarchical framework to support multiple CAs. At the top of the hierarchy is a root CA, which holds a self-signed certificate. The trust within the entire hierarchy is derived from the RSA key pair of the root CA. The subordinate CAs within the hierarchy can be enrolled with either the root CA or with another subordinate CA. Multiple tiers of CAs are configured by either the root CA or with another subordinate CA. Within a hierarchical PKI, all enrolled peers, can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA.

When to Use Multiple CAs

Multiple CAs provide users with added flexibility and reliability. For example, subordinate CAs can be placed in branch offices while the root CA is at the office headquarters. Also, different granting policies can be implemented per CA, so you can set up one CA to automatically grant certificate requests while another CA within the hierarchy requires each certificate request to be manually granted.

Scenarios in which at least a two-tier CA is recommended are as follows:

- Large and very active networks in which a large number of certificates are revoked and reissued. A multiple tier CA helps to control the size of the certificate revocation lists (CRLs).
- When online enrollment protocols are used, the root CA can be kept offline except to issue subordinate CA certificates. This scenario provides added security for the root CA.

Authentication of the CA

The certificate of the CA must be authenticated before the device will be issued its own certificate and before certificate enrollment can occur. Authentication of the CA typically occurs only when you initially configure PKI support at your router. To authenticate the CA, issue the **crypto pki authenticate** command, which authenticates the CA to your router by obtaining the self-signed certificate of the CA that contains the public key of the CA.

Authentication via the fingerprint Command

After Cisco IOS Release 12.3(12), you can issue the **fingerprint** command to preenter a fingerprint that can be matched against the fingerprint of a CA certificate during authentication.

If a fingerprint is not preentered for a trustpoint, and if the authentication request is interactive, you must verify the fingerprint that is displayed during authentication of the CA certificate. If the authentication request is noninteractive, the certificate will be rejected without a preentered fingerprint.

Note If the authentication request is made using the command-line interface (CLI), the request is an interactive request. If the authentication request is made using HTTP or another management tool, the request is a noninteractive request.

Supported Certificate Enrollment Methods

Cisco IOS software supports the following methods to obtain a certificate from a CA:

- Simple Certificate Enrollment Protocol (SCEP)—A Cisco developed enrollment protocol that uses HTTP to communicate with the CA or registration authority (RA). SCEP is the most commonly used method for sending and receiving requests and certificates.

  Note To take advantage of automated certificate and key rollover functionality, you must be running a CA that supports rollover and SCEP must be used as your client enrollment method.

  If you are running a Cisco IOS CA, you must be running Cisco IOS Release 12.4(2)T or a later release for rollover support.

- PKCS12—The router imports certificates in PKCS12 format from an external server.
- IOS File System (IFS)—The router uses any file system that is supported by Cisco IOS software (such as TFTP, FTP, flash, and NVRAM) to send a certificate request and to receive the issued certificate. Users may enable IFS certificate enrollment when their CA does not support SCEP.

Note Prior to Cisco IOS Release 12.3(4)T, only the TFTP file system is supported within IFS.

- Manual cut-and-paste—The router displays the certificate request on the console terminal, allowing the user to enter the issued certificate on the console terminal. A user may manually cut-and-paste certificate requests and certificates when there is no network connection between the router and CA.
- Enrollment profiles—The router sends HTTP-based enrollment requests directly to the CA server instead of to the RA-mode CS. Enrollment profiles can be used if a CA server does not support SCEP.
- Self-signed certificate enrollment for a trustpoint—The secure HTTP (HTTPS) server generates a self-signed certificate that is to be used during the secure socket layer (SSL) handshake, establishing a secure connection between the HTTPS server and the client. The self-signed certificate is then saved in the router's startup configuration (NVRAM). The saved, self-signed certificate can then be used for future SSL handshakes, eliminating the user intervention that was necessary to accept the certificate every time the router reloaded.

Note To take advantage of autoenrollment and auto reenrollment, do not use either TFTP or manual cut-and-paste enrollment as your enrollment method. Both TFTP and manual cut-and-paste enrollment methods are manual enrollment processes, requiring user input.

Registration Authorities (RA)

A Cisco IOS certificate server can be configured to run in RA mode. An RA offloads authentication and authorization responsibilities from a CA. When the RA receives a SCEP or manual enrollment request, the administrator can either reject or grant it on the basis of local policy. If the request is granted, it will be forwarded to the issuing CA, and the CA can be configured to automatically generate the certificate and return it to the RA. The client can later retrieve the granted certificate from the RA.

Automatic Certificate Enrollment

Certificate autoenrollment allows the CA client to automatically request a certificate from its CA sever. This automatic router request eliminates the need for operator intervention when the enrollment request is sent to the CA server. Automatic enrollment is performed on startup for any trustpoint CA that is configured and that does not have a valid client certificate. When the certificate expires, a new certificate is automatically requested.

Note When automatic enrollment is configured, clients automatically request client certificates. The CA server performs its own authorization checks; if these checks include a policy to automatically issue certificates, all clients will automatically receive certificates, which is not very secure. Thus, automatic certificate enrollment should be combined with additional authentication and authorization mechanisms (such as Secure Device Provisioning (SDP), leveraging existing certificates, and one-time passwords).

Automated Client Certificate and Key Rollover

By default, the automatic certificate enrollment function requests a new client certificate and keys from the CS before the client's current certificate expires. Certificate and key rollover allows the certificate renewal rollover request to be made before the certificate expires by retaining the current key and certificate until the new, or rollover, certificate is available. After a specified amount of time, the rollover certificate and keys will become the active certificate and keys. The expired certificate and keys are immediately deleted upon rollover and removed from the certificate chain and CRL.

The setup for automatic rollover is twofold: CA clients must be automatically enrolled and the client's CAs must be automatically enrolled and have the **auto-rollover** command enabled. For more information on configuring your CA servers for automatic certificate rollover see the section "Automatic CA Certificate and Key Rollover" in the chapter "Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment."

An optional renewal percentage parameter can be used with the **auto-enroll** command to allow a new certificate to be requested when a specified percentage of the lifetime of the certificate has passed. For example, if the renewal percentage is configured as 90 and the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. In order for automatic rollover to occur, the renewal percentage must be less than 100.

Tip If CA autoenrollment is not enabled, you may manually initiate rollover on an existing client with the **crypto pki enroll** command if the expiration time of the current client certificate is equal to or greater than the expiration time of the corresponding CA certificate.

The client will initiate the rollover process, which only occurs if the server is configured for automated rollover and has an available rollover server certificate.

Note A key pair is also sent if configured by the **auto-enroll re-generate** command and keyword. It is recommended that a new key pair be issued for security reasons.

Certificate Enrollment Profiles

Enrollment profiles allow users to specify certificate authentication, enrollment, and reenrollment parameters when prompted. The values for these parameters are referenced by two templates that make up the profile. One template contains parameters for the HTTP request that is sent to the CA server to obtain the certificate of the CA (also known as certificate authentication); the other template contains parameters for the HTTP request that is sent to the CA for certificate enrollment.

Configuring two templates enables users to specify different URLs or methods for certificate authentication and enrollment; for example, authentication (getting the certificate of the CA) can be performed via TFTP (using the **authentication url** command) while enrollment can be performed manually (using the **enrollment terminal** command).

Prior to Cisco IOS Release 12.3(11)T, certificate requests could be sent only in a PKCS10 format; however, an additional parameter has now been added to the profile, allowing users to specify the PKCS7 format for certificate renewal requests.

Note A single enrollment profile can have up to three separate sections for each task—certificate authentication, enrollment, and reenrollment.

How to Configure Certificate Enrollment for a PKI

This section contains the following enrollment option procedures. If you configure enrollment or autoenrollment (the first task), you cannot configure manual certificate enrollment. Also, if you configure TFTP or manual cut-and-paste certificate enrollment, you cannot configure autoenrollment, auto reenrollment, an enrollment profile, nor can you utilize the automated CA certificate rollover capability.

- Configuring Certificate Enrollment or Autoenrollment, page 6
- Configuring Manual Certificate Enrollment, page 11
- Configuring a Persistent Self-Signed Certificate for Enrollment via SSL, page 17
- Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment, page 21

Configuring Certificate Enrollment or Autoenrollment

Perform this task to configure certificate enrollment for clients participating in your PKI.

Prerequisites for Autoenrollment

Before configuring automatic certificate enrollment requests, you should ensure that all necessary enrollment information is configured.

Prerequisites for Enabling Automated Client Certificate and Key Rollover

CA client support for certificate rollover is automatically enabled when using autoenrollment. For automatic CA certificate rollover to run successfully, the following prerequisites are applicable:

- Your network devices must support shadow PKI.
- Your clients must be running Cisco IOS Release 12.4(2)T or a later release.
- The client's CS must support automatic rollover. See the section "Automatic CA Certificate and Key Rollover" in the chapter "Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" for more information on CA server automatic rollover configuration.

Prerequisites for Specifying Autoenrollment Initial Key Generation Location

To specify the location of the autoenrollment initial key generation, you must be running Cisco IOS Release 12.4(11)T or a later release.

Restrictions for Autoenrollment

RSA Key Pair Restriction for Autoenrollment

Trustpoints configured to generate a new key pair using the **regenerate** command or the **regenerate** keyword of the **auto-enroll** command must not share key pairs with other trustpoints. To give each trustpoint its own key pair, use the **rsakeypair** command in ca-trustpoint configuration mode. Sharing key pairs among regenerating trustpoints is not supported and will cause loss of service on some of the trustpoints because of key and certificate mismatches.

Restrictions for Automated Client Certificate and Key Rollover

In order for clients to run automatic CA certificate rollover successfully, the following restrictions are applicable:

- SCEP must be used to support rollover. Any device that enrolls with the PKI using an alternative to SCEP as the certificate management protocol or mechanism (such as enrollment profiles, manual enrollment, or TFTP enrollment) will not be able to take advantage of the rollover functionality provided by SCEP.
- If the configuration cannot be saved to the startup configuration after a shadow certificate is generated, rollover will not occur.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]
5. **subject-name** [*x.500-name*]
6. **ip address** {*ip address* | *interface* | **none**}
7. **serial-number** [**none**]
8. **auto-enroll** [*percent*] [**regenerate**]
9. **usage** *method1* [*method2* [*method3*]]
10. **password** *string*
11. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]
12. **fingerprint** *ca-fingerprint*
13. **on** *devicename:*
14. **exit**
15. **crypto pki authenticate** *name*
16. **exit**
17. **copy system:running-config nvram:startup-config**
18. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**] **Example:** `Router(ca-trustpoint)# enrollment url http://cat.example.com`	Specifies the URL of the CA on which your router should send certificate requests. • **mode**—Specifies RA mode if your CA system provides an RA. • **retry period** *minutes*—Specifies the wait period between certificate request retries. The default is 1 minute between retries. • **retry count** *number*— Specifies the number of times a router will resend a certificate request when it does not receive a response from the previous request. (Specify from 1 to 100 retries.) • **url** *url*—URL of the file system where your router should send certificate requests. For enrollment method options, see the **enrollment** command in the *Cisco IOS Security Command Reference*. • **pem**—Adds privacy-enhanced mail (PEM) boundaries to the certificate request. **Note** An enrollment method other than TFTP or manual cut-and-paste must be configured to support autoenrollment.
Step 5	**subject-name** [*x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name cat`	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name*—If it is not specified, the fully qualified domain name (FQDN), which is the default subject name, will be used.
Step 6	**ip address** {*ip address* | *interface* | **none**} **Example:** `Router(ca-trustpoint)# ip address 192.168.1.66`	(Optional) Includes the IP address of the specified interface in the certificate request. Issue the **none** keyword if no IP address should be included. **Note** If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.

	Command or Action	Purpose
Step 7	**serial-number** [**none**] **Example:** `Router(ca-trustpoint)# serial-number`	(Optional) Specifies the router serial number in the certificate request, unless the **none** keyword is issued.
Step 8	**auto-enroll** [*percent*] [**regenerate**] **Example:** `Router(ca-trustpoint)# auto-enroll regenerate`	(Optional) Enables autoenrollment, allowing the client to automatically request a rollover certificate from the CA. If autoenrollment is not enabled, the client must be manually reenrolled in your PKI upon certificate expiration. • By default, only the Domain Name System (DNS) name of the router is included in the certificate. • Use the *percent* argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached. • Use the **regenerate** keyword to generate a new key for the certificate even if a named key already exists. **Note** If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: "! RSA key pair associated with trustpoint is exportable." **Note** It is recommended that a new key pair be generated for security reasons.
Step 9	**usage** *method1* [*method2* [*method3*]] **Example:** `Router(ca-trustpoint)# usage ssl-client`	(Optional) Specifies the intended use for the certificate. Available options are **ike**, **ssl-client**, and **ssl-server**; the default is **ike**.
Step 10	**password** *string* **Example:** `Router(ca-trustpoint)# password meow`	(Optional) Specifies the revocation password for the certificate. If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint. **Note** When SCEP is used, this password can be used to authorize the certificate request—often via a one-time password or similar mechanism.
Step 11	***rsakeypair*** *key-label* [*key-size* [*encryption-key-size*]] **Example:** `Router(ca-trustpoint)# rsakeypair cat`	(Optional) Specifies which key pair to associate with the certificate. • A key pair with *key-label* will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued. • Specify the *key-size* argument for generating the key, and specify the *encryption-key-size* argument to request separate encryption, signature keys, and certificates. **Note** If this command is not enabled, the FQDN key pair is used.

	Command or Action	Purpose
Step 12	`fingerprint ca-fingerprint` **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication. **Note** If the fingerprint is not provided and authentication of the CA certificate is interactive, the fingerprint will be displayed for verification.
Step 13	**on** *devicename:* **Example:** `Router(ca-trustpoint)# on usbtoken0:`	(Optional) Specifies that RSA keys will be created on the specified device upon autoenrollment initial key generation. Devices that may be specified include NVRAM, local disks, and USB tokens. USB tokens may be used as cryptographic devices in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication to be performed on the token.
Step 14	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 15	`crypto pki authenticate name` **Example:** `Router(config)# crypto pki authenticate mytp`	Retrieves the CA certificate and authenticates it. • Check the certificate fingerprint if prompted. **Note** This command is optional if the CA certificate is already loaded into the configuration.
Step 16	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 17	**copy system:running-config nvram:startup-config** **Example:** `Router# copy system:running-config nvram:startup-config`	(Optional) Copies the running configuration to the NVRAM startup configuration. **Note** Autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.
Step 18	`show crypto pki certificates` **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, including any rollover certificates.

Examples

The following example shows the configuration for the “mytp-A” certificate server and its associated trustpoint, where RSA keys generated by the initial autoenrollment for the trustpoint will be stored on a USB token, “usbtoken0”:

```
crypto pki server mytp-A
   database level complete
   issuer-name CN=company, L=city, C=country
   grant auto
! Specifies that certificate requests will be granted automatically.
```

```
!

crypto pki trustpoint mytp-A
   revocation-check none
   rsakeypair myTP-A
   storage usbtoken0:
! Specifies that keys will be stored on usbtoken0:.
   on usbtoken0:
```

! Specifies that keys generated on initial auto enroll will be generated on and stored on ! usbtoken0:

Configuring Manual Certificate Enrollment

Manual certificate enrollment can be set up via TFTP or the manual cut-and-paste method. Both options can be used if your CA does not support SCEP or if a network connection between the router and CA is not possible. Perform one of the following tasks to set up manual certificate enrollment:

- Configuring Cut-and-Paste Certificate Enrollment, page 11
- Configuring TFTP Certificate Enrollment, page 14

PEM-Formatted Files for Certificate Enrollment Request

Using PEM-formatted files for certificate requests can be helpful for customers who are using terminal or profile-based enrollment to request certificates from their CA server. Customers using PEM-formatted files can directly use existing certificates on their Cisco IOS routers.

Restrictions for Manual Certificate Enrollment

Switching Enrollment URLs When Using SCEP

We do not recommend switching URLs if SCEP is used; that is, if the enrollment URL is “http://myca,” do not change the enrollment URL after getting the CA certificate and before enrolling the certificate. A user can switch between TFTP and manual cut-and-paste

Key Regeneration Restriction

Do not regenerate the keys manually using the **crypto key generate** command; key regeneration will occur when the **crypto pki enroll** command is issued if the **regenerate** keyword is specified.

Configuring Cut-and-Paste Certificate Enrollment

Perform this task to configure manual certificate enrollment via the cut-and-paste method for peers participating in your PKI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment terminal** [**pem**]
5. **fingerprint** *ca-fingerprint*
6. **exit**

7. **crypto pki authenticate** *name*
8. **crypto pki enroll** *name*
9. **crypto pki import** *name* **certificate**
10. **exit**
11. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment terminal** [**pem**] **Example:** `Router(ca-trustpoint)# enrollment terminal`	Specifies manual cut-and-paste certificate enrollment method. The certificate request will be displayed on the console terminal so that you may manually copied (or cut). • **pem**—Configures the trustpoint to generate PEM-formatted certificate requests to the console terminal.
Step 5	`fingerprint` *ca-fingerprint* **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication. **Note** If the fingerprint is not provided, it will be displayed for verification.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 7	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate mytp`	Retrieves the CA certificate and authenticates it.
Step 8	**crypto pki enroll** *name* **Example:** `Router(config)# crypto pki enroll mytp`	Generates certificate request and displays the request for copying and pasting into the certificate server. You are prompted for enrollment information, such as whether to include the router FQDN and IP address in the certificate request. You are also given the choice about displaying the certificate request to the console terminal. The base-64 encoded certificate with or without PEM headers as requested is displayed.

	Command or Action	Purpose
Step 9	**crypto pki import** *name* **certificate** **Example:** `Router(config)# crypto pki import mytp certificate`	Imports a certificate manually at the console terminal (pasting). The base-64 encoded certificate is accepted from the console terminal and inserted into the internal certificate database. **Note** You must enter this command twice if usage keys, a signature key and an encryption key, are used. The first time the command is entered, one of the certificates is pasted into the router. The second time the command is entered, the other certificate is pasted into the router. It does not matter which certificate is pasted first. **Note** Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If this applies to the certificate authority you are using, import the general purpose certificate. The router will not use one of the two key pairs generated.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	`show crypto pki certificates` **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

Configuring TFTP Certificate Enrollment

Perform this task to configure manual certificate enrollment using a TFTP server.

Prerequisites for TFTP Certificate Enrollment

- You must know the correct URL to use if you are configuring certificate enrollment via TFTP.
- The router must be able to write a file to the TFTP server for the **crypto pki enroll** command.
- If using a file specification with the **enrollment** command, the file must contain the CA certificate either in binary format or be base-64 encoded.
- You must know if your CA ignores key usage information in a certificate request and issues only a general purpose usage certificate.

Caution Some TFTP servers require that the file must exist on the server before it can be written.

Most TFTP servers require that the file be “write-able” by the world. This requirement may pose a risk

because any router or other device may write or overwrite the certificate request; thus, the replacement certificate request will not used by the CA administrator, who must first check the enrollment request fingerprint before granting the certificate request.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]
5. **fingerprint** *ca-fingerprint*
6. **exit**
7. **crypto pki authenticate** *name*
8. **crypto pki enroll** *name*
9. **crypto pki import** *name* **certificate**
10. **exit**
11. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	`enrollment [mode] [retry period` *minutes*`] [retry count` *number*`] url` *url* `[pem]` **Example:** `Router(ca-trustpoint)# enrollment url tftp://certserver/file_specification`	Specifies TFTP as the enrollment method to send the enrollment request and to retrieve the CA certificate and router certificate and any optional parameters. **Note** For TFTP enrollment, the url must be configured as a TFTP url, tftp://example_tftp_url. An optional file specification filename may be included in the TFTP url. If the file specification is not included, the FQDN will be used. If the file specification is included, the router will append the extension ".ca" to the specified file name.
Step 5	`fingerprint` *ca-fingerprint* **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies the fingerprint of the CA certificate received via an out-of-band method from the CA administrator. **Note** If the fingerprint is not provided, it will be displayed for verification.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 7	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate mytp`	Retrieves the CA certificate and authenticates it from the specified TFTP server.

	Command or Action	Purpose
Step 8	**crypto pki enroll** *name* **Example:** `Router(config)# crypto pki enroll mytp`	Generates certificate request and writes the request out to the TFTP server. You are prompted for enrollment information, such as whether to include the router FQDN and IP address in the certificate request. You are queried about whether or not to display the certificate request to the console terminal. The filename to be written is appended with the extension ".req". For usage keys, a signature key and an encryption key, two requests are generated and sent. The usage key request filenames are appended with the extensions "-sign.req" and "-encr.req" respectively.
Step 9	**crypto pki import** *name* **certificate** **Example:** `Router(config)# crypto pki import mytp certificate`	Imports a certificate via TFTP at the console terminal, which retrieves the granted certificate. The router will attempt to retrieve the granted certificate via TFTP using the same filename used to send the request, except the extension is changed from ".req" to ".crt". For usage key certificates, the extensions "-sign.crt" and "-encr.crt" are used. The router will parse the received files, verify the certificates, and insert the certificates into the internal certificate database on the router. **Note** Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If your CA ignores the usage key information in the certificate request, only import the general purpose certificate. The router will not use one of the two keypairs generated.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	`show crypto pki certificates` **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

Configuring a Persistent Self-Signed Certificate for Enrollment via SSL

This section contains the following tasks:

- Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters, page 18
- Enabling the HTTPS Server, page 20

Note These tasks are optional because if you enable the HTTPS server, it generates a self-signed certificate automatically using default values.

Persistent Self-Signed Certificates Overview

The SSL protocol can be used to establish a secure connection between an HTTPS server and a client (web browser). During the SSL handshake, the client expects the SSL server's certificate to be verifiable using a certificate the client already possesses.

If Cisco IOS software does not have a certificate that the HTTPS server can use, the server generates a self-signed certificate by calling a PKI application programming interface (API). When the client receives this self-signed certificate and is unable to verify it, intervention is needed. The client asks you if the certificate should be accepted and saved for future use. If you accept the certificate, the SSL handshake continues.

Future SSL handshakes between the same client and the server use the same certificate. However, if the router is reloaded, the self-signed certificate is lost. The HTTPS server must then create a new self-signed certificate. This new self-signed certificate does not match the previous certificate so you are once again asked to accept it.

Requesting acceptance of the router's certificate each time that the router reloads may present an opportunity for an attacker to substitute an unauthorized certificate when you are being asked to accept the certificate. Persistent self-signed certificates overcome all these limitations by saving a certificate in the router's startup configuration.

Restrictions

You can configure only one trustpoint for a persistent self-signed certificate.

Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters

Perform the following task to configure a trustpoint and specify self-signed certificate parameters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment selfsigned**
5. subject-name [*x.500-name*]
6. rsakeypair *key-label* [*key-size* [*encryption-key-size*]]
7. **crypto pki enroll** *name*
8. end
9. show crypto pki certificates [*trustpoint-name* [verbose]]
10. **show crypto pki trustpoints** [**status** | *label* [**status**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint local`	Declares the CA that your router should use and enters ca-trustpoint configuration mode. **Note** Effective with Cisco IOS Release 12.3(8)T, the **crypto pki trustpoint** command replaced the **crypto ca trustpoint** command.
Step 4	**enrollment selfsigned** **Example:** `Router(ca-trustpoint)# enrollment selfsigned`	Specifies self-signed enrollment.
Step 5	`subject-name [`*x.500-name*`]` **Example:** `Router(ca-trustpoint)# subject-name`	(Optional) Specifies the requested subject name to be used in the certificate request. • If the *x-500-name* argument is not specified, the FQDN, which is the default subject name, is used.
Step 6	`rsakeypair` *key-label* `[`*key-size* `[`*encryption-key-size*`]]` **Example:** `Router(ca-trustpoint)# rsakeypair examplekeys 1024 1024`	(Optional) Specifies which key pair to associate with the certificate. • The *key-label* argument will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued. • Specify the *key-size* argument for generating the key, and specify the *encryption-key-size* argument to request separate encryption, signature keys, and certificates. **Note** If this command is not enabled, the FQDN key pair is used.
Step 7	**crypto pki enroll** *name* **Example:** `Router(ca-trustpoint)# crypto pki enroll local`	Tells the router to generate the persistent self-signed certificate.
Step 8	**end** **Example:** `Router(ca-trustpoint)# end` **Example:** `Router(config)# end`	(Optional) Exits ca-trustpoint configuration mode and global configuration mode.

	Command or Action	Purpose
Step 9	**show crypto pki certificates [***trustpoint-name* **[verbose]]** **Example:** `Router# show crypto pki certificates local verbose`	Displays information about your certificate, the certification authority certificate, and any registration authority certificates.
Step 10	**show crypto pki trustpoints [status |** *label* **[status]]** **Example:** `Router# show crypto pki trustpoints status`	Displays the trustpoints that are configured in the router.

Enabling the HTTPS Server

Perform the following task to enable the HTTPS server.

Prerequisites

To specify parameters, you must create a trustpoint and configure it. To use default values, delete any existing self-signed trustpoints. Deleting all self-signed trustpoints causes the HTTPS server to generate a persistent self-signed certificate using default values as soon as the server is enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. ip http secure-server
4. **end**
5. copy system:running-config nvram: startup-config

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`ip http secure-server` **Example:** `Router(config)# ip http secure-server`	Enables the secure HTTP web server. **Note** A key pair (modulus 1024) and a certificate are generated.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode.
Step 5	`copy system:running-config nvram: startup-config` **Example:** `Router# copy system:running-config nvram: startup-config`	Saves the self-signed certificate and the HTTPS server in enabled mode.

Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment

Perform this task to configure an enrollment profile for certificate enrollment or reenrollment of a router with a Cisco IOS CA that is already enrolled with a third-party vendor CA.

Enable a router that is enrolled with a third-party vendor CA to use its existing certificate to enroll with the Cisco IOS certificate server so the enrollment request is automatically granted. To enable this functionality, you must issue the **enrollment credential** command. Also, you cannot configure manual certificate enrollment.

Prerequisites

Before configuring a certificate enrollment profile for the client router that is already enrolled with a third party vendor CA so that the router can reenroll with a Cisco IOS certificate server, you should have already performed the following tasks at the client router:

- Defined a trustpoint that points to the third-party vendor CA.
- Authenticated and enrolled the client router with the third-party vendor CA.

Restrictions

- To use certificate profiles, your network must have an HTTP interface to the CA.

- If an enrollment profile is specified, an enrollment URL may not be specified in the trustpoint configuration. Although both commands are supported, only one command can be used at a time in a trustpoint.
- Because there is no standard for the HTTP commands used by various CAs, the user is required to enter the command that is appropriate to the CA that is being used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment profile** *label*
5. **exit**
6. **crypto pki profile enrollment** *label*
7. **authentication url** *url*

   or

   authentication terminal
8. authentication command
9. enrollment url *url*

   or

   enrollment terminal
10. **enrollment credential** *label*
11. **enrollment command**
12. **parameter** *number* {**value** *value* | **prompt** *string*}
13. **exit**
14. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint Entrust`	Declares the trustpoint and a given name and enter ca-trustpoint configuration mode.
Step 4	**enrollment profile** *label* **Example:** `Router(ca-trustpoint)# enrollment profile E`	Specifies that an enrollment profile is to be used for certificate authentication and enrollment.
Step 5	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 6	**crypto pki profile enrollment** *label* **Example:** `Router(config)# crypto pki profile enrollment E`	Defines an enrollment profile and enters ca-profile-enroll configuration mode. • *label*—Name for the enrollment profile; the enrollment profile name must match the name specified in the **enrollment profile** command.
Step 7	**authentication url** *url* **Example:** `Router(ca-profile-enroll)# authentication url http://entrust:81` or **authentication terminal** **Example:** `Router(ca-profile-enroll)# authentication terminal`	Specifies the URL of the CA server to which to send certificate authentication requests. • *url*—URL of the CA server to which your router should send authentication requests. If using HTTP, the URL should read "http://CA_name," where CA_name is the host DNS name or IP address of the CA. If using TFTP, the URL should read "tftp://certserver/file_specification." (If the URL does not include a file specification, the FQDN of the router will be used.) Specifies manual cut-and-paste certificate authentication.

	Command or Action	Purpose
Step 8	**authentication command** **Example:** `Router(ca-profile-enroll)# authentication command`	(Optional) Specifies the HTTP command that is sent to the CA for authentication. This command should be used after the **authentication url** command has been entered.
Step 9	**enrollment url** *url* **Example:** `Router(ca-profile-enroll)# enrollment url http://entrust:81/cda-cgi/clientcgi.exe` or **enrollment terminal** **Example:** `Router(ca-profile-enroll)# enrollment terminal`	Specifies the URL of the CA server to which to send certificate enrollment requests via HTTP or TFTP. Specifies manual cut-and-paste certificate enrollment.
Step 10	**enrollment credential** *label* **Example:** `Router(ca-profile-enroll)# enrollment credential Entrust`	(Optional) Specifies the third-party vendor CA trustpoint that is to be enrolled with the Cisco IOS CA. **Note** This command cannot be issued if manual certificate enrollment is being used.
Step 11	**enrollment command** **Example:** `Router(ca-profile-enroll)# enrollment command`	(Optional) Specifies the HTTP command that is sent to the CA for enrollment.
Step 12	**parameter** *number* {**value** *value* | **prompt** *string*} **Example:** `Router(ca-profile-enroll)# parameter 1 value aaaa-bbbb-cccc`	(Optional) Specifies parameters for an enrollment profile. This command can be used multiple times to specify multiple values.
Step 13	**exit** **Example:** `Router(ca-profile-enroll)# exit` `Router(config)# exit`	Enter this command two times—one time to exit ca-profile-enroll configuration mode and the second time to exit global configuration mode.
Step 14	`show crypto pki certificates` **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

What to Do Next

If you configured the router to reenroll with a Cisco IOS CA, you should configure the Cisco IOS certificate server to accept enrollment requests only from clients already enrolled with the specified third- party vendor CA trustpoint to take advantage of this functionality. For more information, see the module “Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment.”

Configuration Examples for PKI Certificate Enrollment Requests

This section contains the following configuration examples:

Configuring Autoenrollment: Example

The following example shows how to configure the router to automatically enroll with a CA on startup, enabling automatic rollover, and how to specify all necessary enrollment information in the configuration:

```
crypto pki trustpoint frog
 enrollment url http://frog.phoobin.com/
 subject-name OU=Spiral Dept., O=tiedye.com
 ip-address ethernet-0
 serial-number none
 usage ike
 auto-enroll regenerate
 password revokeme
 rsa-key frog 2048
!
crypto pki certificate chain frog
certificate pki 0B
30820293 3082023D A0030201 0202010B 300D0609 2A864886 F70D0101 04050030
79310B30 09060355 04061302 5553310B 30090603 55040813 02434131 15301306
0355040A 130C4369 73636F20 53797374 656D3120 301E0603 55040B13 17737562
6F726420 746F206B 6168756C 75692049 50495355 31243022 06035504 03131B79
6E692D75 31302043 65727469 66696361 7465204D 616E6167 6572301E 170D3030
30373134 32303536 32355A17 0D303130 37313430 31323834 335A3032 310E300C
06035504 0A130543 6973636F 3120301E 06092A86 4886F70D 01090216 11706B69
2D343562 2E636973 636F2E63 6F6D305C 300D0609 2A864886 F70D0101 01050003
4B003048 024100B3 0512A201 3B4243E1 378A9703 8AC5E3CE F77AF987 B5A422C4
15E947F6 70997393 70CF34D6 63A86B9C 4347A81A 0551FC02 ABA62360 01EF7DD2
6C136AEB 3C6C3902 03010001 A381F630 81F3300B 0603551D 0F040403 02052030
1C060355 1D110415 30138211 706B692D 3435622E 63697363 6F2E636F 6D301D06
03551D0E 04160414 247D9558 169B9A21 23D289CC 2DDA2A9A 4F77C616 301F0603
551D2304 18301680 14BD742C E892E819 1D551D91 683F6DB2 D8847A6C 73308185
0603551D 1F047E30 7C307AA0 3CA03AA4 38303631 0E300C06 0355040A 13054369
73636F31 24302206 03550403 131B796E 692D7531 30204365 72746966 69636174
65204D61 6E616765 72A23AA4 38303631 0E300C06 0355040A 13054369 73636F31
24302206 03550403 131B796E 692D7531 30204365 72746966 69636174 65204D61
6E616765 72300D06 092A8648 86F70D01 01040500 03410015 BC7CECF9 696697DF
E887007F 7A8DA24F 1ED5A785 C5C60452 47860061 0C18093D 08958A77 5737246B
0A25550A 25910E27 8B8B428E 32F8D948 3DD1784F 954C70
quit
```

Note In this example, keys are neither regenerated nor rolled over.

Configuring Certificate Autoenrollment with Key Regeneration: Example

The following example shows how to configure the router to automatically enroll with the CA named "trustme1" on startup and enable automatic rollover. The **regenerate** keyword is issued, so a new key will be generated for the certificate and reissued when the automatic rollover process is initiated. The renewal percentage is configured as 90 so if the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. The changes made to the running configuration are saved to the NVRAM startup configuration because autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.

```
crypto pki trustpoint trustme1
 enrollment url http://trustme1.company.com/
 subject-name OU=Spiral Dept., O=tiedye.com
 ip-address ethernet0
 serial-number none
 auto-enroll 90 regenerate
 password revokeme
 rsakeypair trustme1 2048
 exit
crypto pki authenticate trustme1
copy system:running-config nvram:startup-config
```

Configuring Cut-and-Paste Certificate Enrollment: Example

The following example shows how to configure certificate enrollment using the manual cut-and-paste enrollment method:

```
Router(config)# crypto pki trustpoint TP
Router(ca-trustpoint)# enrollment terminal
Router(ca-trustpoint)# crypto pki authenticate TP

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
MIICNDCCAd6gAwIBAgIQOsCmXpVHwodKryRoqULV7jANBgkqhkiG9w0BAQUFADA5
MQswCQYDVQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJ
bXNjYS1yb290MB4XDTAyMDIxNDAwNDYwMVoXDTA3MDIxNDAwNTQ0OFowOTELMAkG
A1UEBhMCVVMxFjAUBgNVBAoTDUNpc2NvIFN5c3RlbXMxEjAQBgNVBAMTCW1zY2Et
cm9vdDBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQCix8nIGFg+wvy3BjFbVi25wYoG
K2N0HWWHpqxFuFhqyBnIC0OshIn9CtrdN3JvUNHr0NIKocEwNKUGYmPwWGTfAgMB
AAGjgcEwgb4wCwYDVR0PBAQDAgHGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYE
FKIacsl6dKAfuNDVQymlSp7esf8jMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9t
c2NhLXJvb3QvQ2VydEVucm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8v
XFxtc2NhLXJvb3RcQ2VydEVucm9sbFxtc2NhLXJvb3QuY3JsMBAGCSsGAQQBgjcV
AQQDAgEAMA0GCSqGSIb3DQEBBQUAA0EAeuZkZMX9qkoLHfETYTpVWjZPQbBmwNRA
oJDSdYdtL3BcI/uLL5q7EmODyGfLyMGxuhQYx5r/40aSQgLCqBq+yg==
-----END CERTIFICATE-----

Certificate has the following attributes:
Fingerprint: D6C12961 CD78808A 4E02193C 0790082A
% Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.
% Certificate successfully imported

Router(config)# crypto pki enroll TP
% Start certificate enrollment..

% The subject name in the certificate will be: Router.company.com
% Include the router serial number in the subject name? [yes/no]: n
% Include an IP address in the subject name? [no]: n
Display Certificate Request to terminal? [yes/no]: y
Signature key certificate request -
Certificate Request follows:

MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTYW5kQmFnZ2VyLmNpc2NvLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAxdhXFDiWAn/hIZs9zfOtssKA
daoWYu0ms9Fe/Pew01dh14vXdxgacstOs2Pr5wk6jLOPxpvxOJPWyQM6ipLmyVxv
ojhyLTrVohrh6Dnqcvk+G/5ohss9o9RxvONwx042pQchFnx9EkMuZC7evwRxJEqR
mBHXBZ8GmP3jYQsjS8MCAwEAAaAhMB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/
BAQDAgeAMA0GCSqGSIb3DQEBBAUAA4GBAMT6WtyFw95POY7UtF+YIYHiVRUf4SCq
hRIAGrljUePLo9iTqyPU1Pnt8JnIZ5P5BHU3MfgP8sqodaWub6mubkzaohJ1qD06
O87fnLCNid5Tov5jKogFHIki2EGGZxBosUw9lJlenQdNdDPbJc5LIWdfDvciA6jO
Nl8rOtKnt8Q+
!
!
!
Redisplay enrollment request? [yes/no]:
Encryption key certificate request -
Certificate Request follows:

MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTYW5kQmFnZ2VyLmNpc2NvLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAwG60QojpDbzbKnyj8FyTiOcv
THkDP7XD4vLT1XaJ409z0gSIoGnIcdFtXhVlBWtpq3/O9zYFXr1tH+BMCRQi3Lts
0IpxYa3D9iFPqev7SPXpsAIsY8a6FMq7TiwLObqiQjLKL4cbuV0Frjl0Yuv5A/Z+
```

```
kqMOm7c+pWNWFdLe9lsCAwEAAaAhMB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/
BAQDAgUgMA0GCSqGSIb3DQEBBAUAA4GBACF7feURj/fJMojPBlR6fa9BrlMJx+2F
H91YM/CIiz2n4mHTeWTWKhLoT8wUfa9NGOk7yi+nF/F7035twLfq6n2bSCTW4aem
8jLMMaeFxwkrV/ceQKrucmNC1uVx+fBy9rhnKx8j60XE25tnp1U08r6om/pBQABU
eNPFhozcaQ/2
!
!
!
Redisplay enrollment request? [yes/no]: n
Router(config)# crypto pki import TP certificate

Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself

MIIDajCCAxSgAwIBAgIKFN7C6QAAAAAMRzANBgkqhkiG9w0BAQUFADA5MQswCQYD
VQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1y
b290MB4XDTAyMDYwODAxMTY0MloXDTAzMDYwODAxMjY0MlowJTEjMCEGCSqGSIb3
DQEJAhMUU2FuZEJhZ2dlci5jaXNjby5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAMXYVxQ4lgJ/4SGbPc3zrbLCgHWqFmLtJrPRXvz3sNNXYdeL13cYGnLL
TrNj6+cJOoyzj8ab8TiT1skDOoqS5slcb6I4ci061aIa4eg56nL5Phv+aIbLPaPU
cbzjcMdONqUHIRZ8fRJDLmQu3r8EcSRKkZgR1wWfBpj942ELI0vDAgMBAAGjggHM
MIIByDALBgNVHQ8EBAMCB4AwHQYDVR0OBBYEFL8Quz8dyz4EGIeKx9A8UMNHLE4s
MHAGA1UdIwRpMGeAFKIacsl6dKAfuNDVQymlSp7esf8joT2kOzA5MQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1yb290
ghA6wKZelUfCh0qvJGipQtXuMCIGA1UdEQEB/wQYMBaCFFNhbmRCYWdnZXIuY2lz
Y28uY29tMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9tc2NhLXJvb3QvQ2VydEVu
cm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8vXFxtc2NhLXJvb3RcQ2Vy
dEVucm9sbFxtc2NhLXJvb3QuY3JsMIGUBggrBgEFBQcBAQSBhzCBhDA/BggrBgEF
BQcwAoYzaHR0cDovL21zY2Etcm9vdC9DZXJ0RW5yb2xsL21zY2Etcm9vdF9tc2Nh
LXJvb3QuY3J0MEEGCCsGAQUFBzAChjVmaWxlOi8vXFxtc2NhLXJvb3RcQ2VydEVu
cm9sbFxtc2NhLXJvb3RfbXNjYS1yb290LmNydDANBgkqhkiG9w0BAQUFAANBAJo2
r6sHPGBdTQX2EDoJpR/A2UHXxRYqVSHkFKZw0z31r5JzUM0oPNUETV7mnZlYNVRZ
CSEX/G8boi3WOjz9wZo=

% Router Certificate successfully imported

Router(config)# crypto pki import TP cert

Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself

MIIDajCCAxSgAwIBAgIKFN7OBQAAAAAMSDANBgkqhkiG9w0BAQUFADA5MQswCQYD
VQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1y
b290MB4XDTAyMDYwODAxMTY0NVoXDTAzMDYwODAxMjY0NVowJTEjMCEGCSqGSIb3
DQEJAhMUU2FuZEJhZ2dlci5jaXNjby5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAMButEKI6Q282yp8o/Bck4jnL0x5Az+1w+Ly09V2ieNPc9IEiKBpyHHR
bV4VZQVraat/zvc2BV69bR/gTAkUIty7bNCKcWGtw/YhT6nr+0j16bACLGPGuhTK
u04sCzm6okIyyi+HG7ldBa45dGLr+QP2fpKjDpu3PqVjVhXS3vZbAgMBAAGjggHM
MIIByDALBgNVHQ8EBAMCBSAwHQYDVR0OBBYEFPDO29oRdlEUSgBMg6jZR+YFRWlj
MHAGA1UdIwRpMGeAFKIacsl6dKAfuNDVQymlSp7esf8joT2kOzA5MQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1yb290
ghA6wKZelUfCh0qvJGipQtXuMCIGA1UdEQEB/wQYMBaCFFNhbmRCYWdnZXIuY2lz
Y28uY29tMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9tc2NhLXJvb3QvQ2VydEVu
cm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8vXFxtc2NhLXJvb3RcQ2Vy
dEVucm9sbFxtc2NhLXJvb3QuY3JsMIGUBggrBgEFBQcBAQSBhzCBhDA/BggrBgEF
BQcwAoYzaHR0cDovL21zY2Etcm9vdC9DZXJ0RW5yb2xsL21zY2Etcm9vdF9tc2Nh
LXJvb3QuY3J0MEEGCCsGAQUFBzAChjVmaWxlOi8vXFxtc2NhLXJvb3RcQ2VydEVu
cm9sbFxtc2NhLXJvb3RfbXNjYS1yb290LmNydDANBgkqhkiG9w0BAQUFAANBAHaU
hyCwLirUghNxCmLzXRG7C3W1j0kSX7a4fX9OxKR/Z2SoMjdMNPPyApuh8SoT2zBP
ZKjZU2WjcZG/nZF4W5k=

% Router Certificate successfully imported
```

You can verify that the certificate was successfully imported by issuing the **show crypto pki certificate** command.

```
Router# show crypto pki certificate
Certificate
  Status: Available
  Certificate Serial Number: 14DECE05000000000C48
  Certificate Usage: Encryption
  Issuer:
    CN = TPCA-root
     O = Company
     C = US
  Subject:
    Name: Router.company.com
    OID.1.2.840.113549.1.9.2 = Router.company.com
  CRL Distribution Point:
    http://tpca-root/CertEnroll/tpca-root.crl
  Validity Date:
    start date: 18:16:45 PDT Jun 7 2002
    end   date: 18:26:45 PDT Jun 7 2003
    renew date: 16:00:00 PST Dec 31 1969
  Associated Trustpoints: TP

Certificate
  Status: Available
  Certificate Serial Number: 14DEC2E9000000000C47
  Certificate Usage: Signature
  Issuer:
    CN = tpca-root
     O = company
     C = US
  Subject:
    Name: Router.company.com
    OID.1.2.840.113549.1.9.2 = Router.company.com
  CRL Distribution Point:
    http://tpca-root/CertEnroll/tpca-root.crl
  Validity Date:
    start date: 18:16:42 PDT Jun 7 2002
    end   date: 18:26:42 PDT Jun 7 2003
    renew date: 16:00:00 PST Dec 31 1969
  Associated Trustpoints: TP

CA Certificate
  Status: Available
  Certificate Serial Number: 3AC0A65E9547C2874AAF2468A942D5EE
  Certificate Usage: Signature
  Issuer:
    CN = tpca-root
     O = Company
     C = US
  Subject:
    CN = tpca-root
     O = company
     C = US
  CRL Distribution Point:
    http://tpca-root/CertEnroll/tpca-root.crl
  Validity Date:
    start date: 16:46:01 PST Feb 13 2002
    end   date: 16:54:48 PST Feb 13 2007
  Associated Trustpoints: TP
```

Configuring Manual Certificate Enrollment with Key Regeneration: Example

The following example shows how to regenerate new keys with a manual certificate enrollment from the CA named “trustme2”:

```
crypto pki trustpoint trustme2
 enrollment url http://trustme2.company.com/
 subject-name OU=Spiral Dept., O=tiedye.com
 ip-address ethernet0
 serial-number none
 regenerate
 password revokeme
 rsakeypair trustme2 2048s
 exit
crypto pki authenticate trustme2
crypto pki enroll trustme2
```

Creating and Verifying a Persistent Self-Signed Certificate: Example

The following example shows how to declare and enroll a trustpoint named “local” and generate a self-signed certificate with an IP address:

```
crypto pki trustpoint local
 enrollment selfsigned
 end
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

crypto pki enroll local
Nov 29 20:51:13.067: %SSH-5-ENABLED: SSH 1.99 has been enabled
Nov 29 20:51:13.267: %CRYPTO-6-AUTOGEN: Generated new 512 bit key pair
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: yes
Enter Interface name or IP Address[]: ethernet 0
Generate Self Signed Router Certificate? [yes/no]: yes
Router Self Signed Certificate successfully created
```

Note A router can have only one self-signed certificate. If you attempt to enroll a trustpoint configured for a self-signed certificate and one already exists, you receive a notification and are asked if you want to replace it. If so, a new self-signed certificate is generated to replace the existing one.

Enabling the HTTPS Server: Example

The following example shows how to enable the HTTPS server and generate a default trustpoint because one was not previously configured:

```
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

ip http secure-server
% Generating 1024 bit RSA keys ...[OK]
*Dec 21 19:14:15.421:%PKI-4-NOAUTOSAVE:Configuration was modified.  Issue "write memory"
to save new certificate
Router(config)#
```

Note You need to save the configuration to NVRAM if you want to keep the self-signed certificate and have the HTTPS server enabled following router reloads.

The following message also appears:

```
*Dec 21 19:14:10.441:%SSH-5-ENABLED:SSH 1.99 has been enabled
```

Note Creation of the key pair used with the self-signed certificate causes the Secure Shell (SSH) server to start. This behavior cannot be suppressed. You may want to modify your access control lists (ACLs) to permit or deny SSH access to the router.

Verifying the Self-Signed Certificate Configuration: Example

The following example displays information about the self-signed certificate that you just created:

```
Router# show crypto pki certificates

Router Self-Signed Certificate
  Status: Available
  Certificate Serial Number: 01
  Certificate Usage: General Purpose
  Issuer:
    cn=IOS-Self-Signed-Certificate-3326000105
  Subject:
    Name: IOS-Self-Signed-Certificate-3326000105
    cn=IOS-Self-Signed-Certificate-3326000105
  Validity Date:
    start date: 19:14:14 GMT Dec 21 2004
    end   date: 00:00:00 GMT Jan 1 2020
  Associated Trustpoints: TP-self-signed-3326000105
```

Note The number 3326000105 above is the router's serial number and varies depending on the router's actual serial number.

The following example displays information about the key pair corresponding to the self-signed certificate:

```
Router# show crypto key mypubkey rsa

% Key pair was generated at: 19:14:10 GMT Dec 21 2004
Key name: TP-self-signed-3326000105
 Usage: General Purpose Key
 Key is not exportable.
 Key Data:
  30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00B88F70
  6BC78B6D 67D6CFF3 135C1D91 8F360292 CA44A032 5AC1A8FD 095E4865 F8C95A2B
  BFD1C2B7 E64A3804 9BBD7326 207BD456 19BAB78B D075E78E 00D2560C B09289AE
  6DECB8B0 6672FB3A 5CDAEE92 9D4C4F71 F3BCB269 214F6293 4BA8FABF 9486BCFC
  2B941BCA 550999A7 2EFE12A5 6B7B669A 2D88AB77 39B38E0E AA23CB8C B7020301 0001
% Key pair was generated at: 19:14:13 GMT Dec 21 2004
Key name: TP-self-signed-3326000105.server
 Usage: Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00C5680E 89777B42
  463E5783 FE96EA9E F446DC7B 70499AF3 EA266651 56EE29F4 5B003D93 2FC9F81D
  8A46E12F 3FBAC2F3 046ED9DD C5F27C20 1BBA6B9B 08F16E45 C34D6337 F863D605
  34E30F0E B4921BC5 DAC9EBBA 50C54AA0 BF551BDD 88453F50 61020301 0001
```

Note The second key pair with the name TP-self-signed-3326000105.server is the SSH key pair and is generated when any key pair is created on the router and SSH starts up.

The following example displays information about the trustpoint named “local”:

```
Router# show crypto pki trustpoints

Trustpoint local:
    Subject Name:
    serialNumber=C63EBBE9+ipaddress=10.3.0.18+hostname=test.company.com
          Serial Number: 01
    Persistent self-signed certificate trust point
```

Configuring Direct HTTP Enrollment: Example

The following example show how to configure an enrollment profile for direct HTTP enrollment with a CA server:

```
crypto pki trustpoint Entrust
 enrollment profile E
 serial

crypto pki profile enrollment E
 authentication url http://entrust:81
 authentication command GET /certs/cacert.der
 enrollment url http://entrust:81/cda-cgi/clientcgi.exe
 enrollment command POST reference_number=$P2&authcode=$P1
 &retrievedAs=rawDER&action=getServerCert&pkcs10Request=$REQ
 parameter 1 value aaaa-bbbb-cccc
 parameter 2 value 5001
```

Additional References

The following sections provide references related to certificate enrollment for a PKI.

Related Documents

Related Topic	Document Title
USB Token RSA Operations: Benefits of using USB tokens	"Storing PKI Credentials" module in the *Cisco IOS Security Configuration Guide*, Release 12.4T
USB Token RSA Operations: Certificate server configuration	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.4T. See the "Generating a Certificate Server RSA Key Pair" section, the "Configuring a Certificate Server Trustpoint" section, and related examples.
Overview of PKI, including RSA keys, certificate enrollment, and CAs.	*"Cisco IOS PKI Overview: Understanding and Planning a PKI" module* in the *Cisco IOS Security Configuration Guide*, Release 12.4
Secure Device Provisioning: functionality overview and configuration tasks	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI" module in the *Cisco IOS Security Configuration Guide*, Release 12.4T
RSA key generation and deployment	"Deploying RSA Keys Within a PKI" module in the *Cisco IOS Security Configuration Guide*, Release 12.4T
Cisco IOS certificate server overview information and configuration tasks	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" module in the *Cisco IOS Security Configuration Guide*, Release 12.4T
Setting up and using a USB token	"Storing PKI Credentials" module in the *Cisco IOS Security Configuration Guide*, Release 12.4T

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for PKI Certificate Enrollment

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Implementing and Managing PKI Features Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for PKI Certificate Enrollment

Feature Name	Releases	Feature Information
Cisco IOS USB Token PKI Enhancements—Phase 2	12.4(11)T	This feature enhances USB token functionality by using the USB token as a cryptographic device. USB tokens may be used for RSA operations such as key generation, signing, and authentication. The following sections in this document provide information about this feature: • Configuring Certificate Enrollment or Autoenrollment **Note** This document covers the use of utilizing USB tokens for RSA operations during initial autoenrollment for a trustpoint.For other documents on this topic, see the "Related Documents" section.
Certificate Authority (CA) Key Rollover	12.4(2)T	This feature introduces the ability for root CAs to roll over expiring CA certificates and keys and to have these changes propagate through the PKI network without manual intervention. The following sections provide information about this feature: • Automatic Certificate Enrollment • Configuring Certificate Enrollment or Autoenrollment The following commands were introduced or modified by this feature: **auto-rollover**, **crypto pki certificate chain, crypto pki export pem, crypto pki server, crypto pki server info request**, **show crypto pki certificates, show crypto pki server**, and **show crypto pki trustpoint**
Certificate Autoenrollment	12.2(8)T	This feature introduces certificate autoenrollment, which allows the router to automatically request a certificate from the CA that is using the parameters in the configuration. The following sections provide information about this feature: • Automatic Certificate Enrollment • Configuring Certificate Enrollment or Autoenrollment The following commands were introduced by this feature: **auto-enroll**, **rsakeypair**, **show crypto ca timers**

Table 1 Feature Information for PKI Certificate Enrollment (continued)

Feature Name	Releases	Feature Information
Certificate Enrollment Enhancements	12.2(8)T	This feature introduces five new **crypto ca trustpoint** subcommands that provide new options for certificate requests and allow users to specify fields in the configuration instead of having to go through prompts. The following section provides information about this feature: • Configuring Certificate Enrollment or Autoenrollment The following commands were introduced by this feature: **ip-address (ca-trustpoint)**, **password (ca-trustpoint)**, **serial-number**, **subject-name**, **usage**
Direct HTTP Enrollment with CA Servers	12.3(4)T	This feature allows users to configure an enrollment profile if their CA server does not support SCEP and they do not want to use an RA-mode CS. The enrollment profile allows users to send HTTP requests directly to the CA server instead of to an RA-mode CS. The following sections provide information about this feature: • Certificate Enrollment Profiles • Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment The following commands were introduced by this feature: **authentication command**, **authentication terminal**, **authentication url**, **crypto ca profile enrollment**, **enrollment command**, **enrollment profile**, **enrollment terminal**, **enrollment url**, **parameter**
Import of RSA Key Pair and Certificates in PEM Format	12.3(4)T	This feature allows customers to issue certificate requests and receive issued certificates in PEM-formatted files. The following section provides information about this feature: • Configuring Manual Certificate Enrollment The following commands were modified by this feature: **enrollment**, **enrollment terminal**

Table 1 Feature Information for PKI Certificate Enrollment (continued)

Feature Name	Releases	Feature Information
Key Rollover for Certificate Renewal	12.3(7)T	This feature allows the certificate renewal request to be made before the certificate expires and retains the old key and certificate until the new certificate is available. The following sections provide information about this feature: • Automatic Certificate Enrollment • Configuring Certificate Enrollment or Autoenrollment • Configuring Manual Certificate Enrollment The following commands were introduced or modified by this feature: **auto-enroll**, **regenerate**
Manual Certificate Enrollment (TFTP Cut-and-Paste)	12.2(13)T	This feature allows users to generate a certificate request and accept CA certificates as well as the router's certificates via a TFTP server or manual cut-and-paste operations. The following sections provide information about this feature: • Supported Certificate Enrollment Methods • Configuring Manual Certificate Enrollment The following commands were introduced or modified by this feature: **crypto ca import**, **enrollment**, **enrollment terminal**
Multiple-Tier CA Hierarchy[1]	12.2(15)T	This enhancement enables users to set up a PKI in a hierarchical framework to support multiple CAs. Within a hierarchical PKI, all enrolled peers can validate the certificate of one another as long as the peers share a trusted root CA certificate or a common subordinate CA. The following section provides information about this enhancement: • Hierarchical PKI: Multiple CAs
Persistent Self-Signed Certificates	12.2(33)SXH 12.2(33)SRA 12.3(14)T	This feature allows the HTTPS server to generate and save a self-signed certificate in the router startup configuration. Thus, future SSL handshakes between the client and the HTTPS server can use the same self-signed certificate without user intervention. The following sections provide information about this feature: • Supported Certificate Enrollment Methods • Configuring a Persistent Self-Signed Certificate for Enrollment via SSL The following commands were introduced or modified by this feature: **enrollment selfsigned**, **show crypto pki certificates**, **show crypto pki trustpoints**

Table 1 Feature Information for PKI Certificate Enrollment (continued)

Feature Name	Releases	Feature Information
PKI Status[1]	12.3(11)T	This enhancement added the **status** keyword to the **show crypto pki trustpoints** command, which allows you to view the current status of the trustpoint. Prior to this enhancement, you had to issue the **show crypto pki certificates** and the **show crypto pki timers** commands for the current status. The following section provides information about this enhancement: • How to Configure Certificate Enrollment for a PKI
Reenroll Using Existing Certificates	12.3(11)T	This feature allows users to reenroll a router with a Cisco IOS CA via existing certificates from a third-party vendor CA. The following section provides information about this enhancement: • Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment The following commands were introduced by this feature: **enrollment credential**, **grant auto trustpoint**
Trustpoint CLI	12.2(8)T	This feature introduces the **crypto pki trustpoint** command, which adds support for trustpoint CAs.
Certificate - Auto Enrollment	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Certificate - Enrollment Enhancements	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Direct http enroll with CA servers	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Key Rollover for Certificate Renewal	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Manual certificate enrollment (TFTP and cut-and-paste)	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Manual Certificate Enrollment via TFTP	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Persistent Self-Signed Certificates	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Re-Enroll Using Existing Certificate	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

1. This is a minor enhancement. Minor enhancements are not typically listed in Feature Navigator.

Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment

First Published: May 2, 2005
Last Updated: November 17, 2006

This module describes how to set up and manage a Cisco IOS certificate server for public key infrastructure (PKI) deployment. A certificate server embeds a simple certificate server, with limited certification authority (CA) functionality, into the Cisco IOS software. Thus, the following benefits are provided to the user:

- Easier PKI deployment by defining default behavior. The user interface is simpler because default behaviors are predefined. That is, you can leverage the scaling advantages of PKI without all of the certificate extensions that a CA provides, thereby allowing you to easily enable a basic PKI-secured network.
- Direct integration with Cisco IOS software.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for the Cisco IOS Certificate Server" section on page 50.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring a Cisco IOS Certificate Server

Planning Your PKI Before Configuring the Certificate Server

Before configuring a Cisco IOS certificate server, it is important that you have planned for and chosen appropriate values for the settings you intend to use within your PKI (such as certificate lifetimes and certificate revocation list (CRL) lifetimes). After the settings have been configured in the certificate server and certificates have been granted, settings cannot be changed without having to reconfigure the certificate server and reenrolling the peers. For information on certificate server default settings and recommended settings, see the section "Certificate Server Default Values and Recommended Values."

Enabling an HTTP Server

The certificate server supports Simple Certificate Enrollment Protocol (SCEP) over HTTP. The HTTP server must be enabled on the router for the certificate server to use SCEP. (To enable the HTTP server, use the **ip http server** command.) The certificate server will automatically enable or disable SCEP services after the HTTP server is enabled or disabled. If the HTTP server is not enabled, only manual PKCS10 enrollment is supported.

Note To take advantage of automatic CA certificate and key pair rollover functionality for all types of certificate servers, Cisco IOS Release 12.4(4)T or a later release must be used and SCEP must be used as the enrollment method.

Configuring Reliable Time Services

Time services must be running on the router because the certificate server must have reliable time knowledge. If a hardware clock is unavailable, the certificate server will depend on manually configured clock settings, such as Network Time Protocol (NTP). If there is not a hardware clock or the clock is invalid, the following message will be displayed at bootup:

```
% Time has not been set. Cannot start the Certificate server.
```

After the clock has been set, the certificate server will automatically switch to running status.

For information on manually configuring clock settings, see the section "Setting Time and Calendar Services" in the chapter "Performing Basic System Management" of the *Cisco IOS Network Management Configuration Guide*.

"crypto ca" to "crypto pki" CLI Change

As of Cisco IOS Release 12.3(7)T, all commands that begin as "crypto ca" have been changed to begin as "crypto pki." Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

Restrictions for Configuring a Cisco IOS Certificate Server

The certificate server does not provide a mechanism for modifying the certificate request that is received from the client; that is, the certificate that is issued from the certificate server matches the requested certificate without modifications. If a specific certificate policy, such as name constraints, must be issued, the policy must be reflected in the certificate request.

Information About Cisco IOS Certificate Servers

Before setting up and deploying a certificate server in your PKI, you should understand the following concepts:

RSA Key Pair and Certificate of the Certificate Server

The certificate server automatically generates a 1024-bit Rivest, Shamir, and Adelman (RSA) key pair. You must manually generate an RSA key pair if you prefer a different key pair modulus. For information on completing this task, see the section "Generating a Certificate Server RSA Key Pair."

Note The recommended modulus for a certificate server key pair is 2048 bits.

The certificate server will use a regular Cisco IOS RSA key pair as its CA key. This key pair must have the same name as the certificate server. If you do not generate the key pair before the certificate server is created on the router, a general-purpose key pair will be automatically generated during the configuration of the certificate server.

As of Cisco IOS Release 12.3(11)T and later releases, the CA certificate and CA key can be backed up automatically one time after they are generated by the certificate server. As a result, it is not necessary to generate an exportable CA key for backup purposes.

What to Do with Automatically Generated Key Pairs in Cisco IOS Software Prior to Release 12.3(11)T

If the key pair is automatically generated, it will not be marked as exportable. Thus, you must manually generate the key pair as exportable if you want to back up the CA key. For information on how to complete this task, see the section "Generating a Certificate Server RSA Key Pair."

How the CA Certificate and CA Key Are Automatically Archived

At initial certificate server setup, you can enable the CA certificate and the CA key to be automatically archived so that they may be restored later if either the original copy or the original configuration is lost.

When the certificate server is turned on the first time, the CA certificate and CA key will be generated. If automatic archive is also enabled, the CA certificate and the CA key will be exported (archived) to the server database. The archive can be in PKCS12 or privacy-enhanced mail (PEM) format.

Note

- This CA key backup file is extremely important and should be moved immediately to another secured place.
- This archiving action occurs only one time. Only the CA key that is (1) manually generated and marked exportable or (2) automatically generated by the certificate server will be archived (this key will be marked nonexportable).
- Autoarchiving will not occur if you generate the CA key manually and mark it "nonexportable."
- In addition to the CA certificate and CA key archive file, you should also regularly back up the serial number file (.ser) and the CRL file (.crl). The serial file and the CRL file are both critical for CA operation if you need to restore your certificate server.
- It is not possible to manually back up a server that uses nonexportable RSA keys or manually generated, nonexportable RSA keys. Although automatically generated RSA keys are marked as nonexportable, they are automatically archived once.

Certificate Server Database

The Cisco IOS certificate server stores files for its own use and may publish files for other processes to use. Critical files generated by the certificate server that are needed for its ongoing operation are stored to only one location per file type for its exclusive use. The certificate server reads from and writes to these files. The critical certificate server files are the serial number file (.ser) and the CRL storage location file (.crl). Files that the certificate server writes to, but does not read from again, may be published and available for use by other processes. An example of a file that may be published is the issued certificates file (.crt).

Performance of your certificate server may be affected by the following factors, which should be considered when you choose storage options and publication options for your certificate server files.

- The storage or publish locations you choose may affect your certificate server performance. Reading from a network location takes more time than reading directly from a router's local storage device.
- The number of files you choose to store or publish to a specific location may affect your certificate server performance. The local Cisco IOS file system may not always be suitable for a large number of files.
- The file types you choose to store or publish may affect your certificate server performance. Certain files, such as the .crl files, can become very large.

Note It is recommended that you store .ser and .crl files to your local Cisco IOS file system and publish your .crt files to a remote file system.

Certificate Server Database File Storage

The certificate server allows the flexibility to store different critical file types to different storage locations depending on the database level set (see the **database level** command for more information). When choosing storage locations, consider the file security needed and server performance. For instance, serial number files and archive files (.p12 or .pem) might have greater security restrictions than the issued certificates file storage location (.crt) or the name file storage location (.cnm).

Table 1 shows the critical certificate server file types by file extension that may be stored to a specific location.

Table 1 Certificate Server Storage Critical File Types

File Extension	File Type
.ser	The main certificate server database file.
.crl	The CRL storage location.
.crt	The issued certificates storage location.
.cnm	The certificate name and expiration file storage location.
.p12	The certificate server certificate archive file location in PKCS12 format.
.pem	The certificate server certificate archive file location in PEM format.

Cisco IOS certificate server files may be stored to three levels of specificity:

- Default location, NVRAM
- Specified primary storage location for all critical files
- Specified storage location for specific critical file(s).

A more specific storage location setting overrides a more general storage location setting. For instance, if you have not specified any certificate server file storage locations, all certificate server files will be stored to NVRAM. If you specify a storage location for the name file, only the name file will be stored there; all other files will still be stored to NVRAM. If you then specify a primary location, all files except the name file will now be stored to this location, instead of NVRAM.

Note You may specify either .p12 or .pem; you cannot specify both types of archive files.

Certificate Server Database File Publication

A publish file is a copy of the original file and is available for other processes to use or for your use. If the certificate server fails to publish a file, it does cause the server to shut down. You may specify one publish location for the issued certificates file and name file and multiple publish locations for the CRL file. See Table 2 for files types available for publication. You may publish files regardless of the database level that is set.

Table 2 Certificate Server Publish File Types

File Extension	File Type
.crl	The CRL publish location.
.crt	The issued certificates publish location.
.cnm	The certificate name and expiration file publish location.

Trustpoint of the Certificate Server

The certificate server will also have an automatically generated trustpoint of the same name; the trustpoint will store the certificate of the certificate server. After the router detects that a trustpoint is being used to store the certificate of the certificate server, the trustpoint will be locked so that it cannot be modified.

Before configuring the certificate server you can perform the following:

- Manually create and set up this trustpoint (using the **crypto pki trustpoint** command), which allows you to specify an alternative RSA key pair (using the **rsakeypair** command).
- Specify that the initial autoenrollment key pair will be generated on a specific device, such as a configured and available USB token, using the **on** command.

Note The automatically generated trustpoint and the certificate server certificate are not available for the certificate server device identity. Thus, any command-line interface (CLI) (such as the **ip http secure-trustpoint** command) that is used to specify the CA trustpoint to obtain certificates and authenticate the connecting client's certificate must point to an additional trustpoint configured on the certificate server device.

If the server is a root certificate server, it will use the RSA key pairs and several other attributes to generate a self-signed certificate. The associated CA certificate will have the following key usage extensions—Digital Signature, Certificate Sign, and CRL Sign.

After the CA certificate is generated, attributes can be changed only if the certificate server is destroyed.

Note A certificate server trustpoint must not be automatically enrolled using the **auto-enroll** command. Initial enrollment of the certificate server must be initiated manually and ongoing automatic rollover functionality may be configured with the **auto-rollover** command. For more information on automatic rollover functionality, see the section "Automatic CA Certificate and Key Rollover."

Certificate Revocation Lists (CRLs)

By default, CRLs are issued once every 168 hours (1 calendar week). To specify a value other than the default value for issuing the CRL, execute the **lifetime crl** command. After the CRL is issued, it is written to the specified database location as *ca-label*.crl, where *ca-label* is the name of the certificate server.

CRLs can be distributed via SCEP, which is the default method, or a CRL distribution point (CDP), if configured and available. If you set up a CDP, use the **cdp-url** command to specify the CDP location. If the **cdp-url** command is not specified, the CDP certificate extension will not be included in the

certificates that are issued by the certificate server. If the CDP location is not specified, Cisco IOS PKI clients will automatically request a CRL from the certificate server with a SCEP GetCRL message. The CA then returns the CRL in a SCEP CertRep message to the client. Because all SCEP messages are enveloped and signed PKCS#7 data, the SCEP retrieval of the CRL from the certificate server is costly and not highly scalable. In very large networks, an HTTP CDP provides better scalability and is recommended if you have many peer devices that will be checking CRLs. You may specify the CDP location by a simple HTTP URL string for example,

cdp-url http://my-cdp.company.com/filename.crl

The certificate server supports only one CDP; thus, all certificates that are issued include the same CDP.

If you have PKI clients that are not running Cisco IOS software and that do not support a SCEP GetCRL request and wish to use a CDP you may set up an external server to distribute CRLs and configure the CDP to point to that server. Or, you can specify a non-SCEP request for the retrieval of the CRL from the certificate server by specifying the **cdp-url** command with the URL in the following format where *cs-addr* is the location of the certificate server:

cdp-url http://*cs-addr*/cgi-bin/pkiclient.exe?operation=GetCRL

Note If your Cisco IOS CA is also configured as your HTTP CDP server, specify your CDP with the **cdp-url** http://*cs-addr*/cgi-bin/pkiclient.exe?operation=GetCRL command syntax.

It is the responsibility of the network administrator to ensure that the CRL is available from the location that is specified via the **cdp-url** command.

The CDP location may be changed after the certificate server is running via the **cdp-url** command. New certificates will contain the updated CDP location, but existing certificates will not be reissued with the newly specified CDP location. When a new CRL is issued, the certificate server uses its current cached CRL to generate a new CRL. (When the certificate server is rebooted, it reloads the current CRL from the database.) A new CRL cannot be issued unless the current CRL has expired. After the current CRL expires, a new CRL is issued only after a certificate is revoked from the CLI.

Certificate Server Error Conditions

At startup, the certificate server checks the current configuration before issuing any certificates. It reports the last known error conditions via the **show crypto pki server** command output. Example errors can include any of the following conditions:

- Storage inaccessible
- Waiting for HTTP server
- Waiting for time setting

If the certificate server experiences a critical failure at any time, such as failing to publish a CRL, the certificate server will automatically enter a disabled state. This state allows the network administrator to fix the condition; thereafter, the certificate server will return to the previous normal state.

Certificate Enrollment Using a Certificate Server

A certificate enrollment request functions as follows:

- The certificate server receives the enrollment request from an end user, and the following actions occur:
  - A request entry is created in the enrollment request database with the initial state. (See Table 3 for a complete list of certificate enrollment request states.)
  - The certificate server refers to the CLI configuration (or the default behavior any time a parameter is not specified) to determine the authorization of the request. Thereafter, the state of the enrollment request is updated in the enrollment request database.
- At each SCEP query for a response, the certificate server examines the current request and performs one of the following actions:
  - Responds to the end user with a “pending” or “denied” state.
  - Generates and signs the appropriate certificate and stores the certificate in the enrollment request database.

If the connection of the client has closed, the certificate server will wait for the client to request another certificate.

All enrollment requests transition through the certificate enrollment states that are defined in Table 3. To see current enrollment requests, use the **crypto pki server request pkcs10** command.

Table 3 Certificate Enrollment Request State Descriptions

Certificate Enrollment State	Description
authorized	The certificate server has authorized the request.
denied	The certificate server has denied the request for policy reasons.
granted	The CA core has generated the appropriate certificate for the certificate request.
initial	The request has been created by the SCEP server.
malformed	The certificate server has determined that the request is invalid for cryptographic reasons.
pending	The enrollment request must be manually accepted by the network administrator.

SCEP Enrollment

All SCEP requests are treated as new certificate enrollment requests, even if the request specifies a duplicate subject name or public key pair as a previous certificate request.

Types of CA Servers: Subordinate and Registration Authorities (RAs)

CA servers have the flexibility to be configured as a subordinate certificate server or an RA-mode certificate server.

Why Configure a Subordinate CA?

A subordinate certificate server provides all the same features as a root certificate server. The root RSA key pairs are extremely important in a PKI hierarchy, and it is often advantageous to keep them offline or archived. To support this requirement, PKI hierarchies allow for subordinate CAs that have been signed by the root authority. In this way, the root authority can be kept offline (except to issue occasional CRL updates), and the subordinate CA can be used during normal operation.

Why Configure an RA-Mode Certificate Server?

A Cisco IOS certificate server can be configured to run in RA mode. An RA offloads authentication and authorization responsibilities from a CA. When the RA receives a SCEP or manual enrollment request, the administrator can either reject or grant it on the basis of local policy. If the request is granted, it will be forwarded to the issuing CA, and the CA will automatically generate the certificate and return it to the RA. The client can later retrieve the granted certificate from the RA.

An RA is the authority charged with recording or verifying some or all of the data required for the CA to issue certificates. In many cases the CA will undertake all of the RA functions itself, but where a CA operates over a wide geographical area or when there is security concern over exposing the CA to direct network access, it may be administratively advisable to delegate some of the tasks to an RA and leave the CA to concentrate on its primary tasks of signing certificates and CRLs.

Automatic CA Certificate and Key Rollover

CAs—root CAs, subordinate CAs, and RA-mode CAs—like their clients, have certificates and key pairs with expiration dates that need to be reissued when the current certificate and key pair are about to expire. When a root CA's certificate and key pair are expiring it must generate a self-signed rollover certificate and key pair. If a subordinate CA or an RA-mode CA's certificate and key pair are expiring, it will request a rollover certificate and key pair from its superior CA, obtaining the superior CA's new self-signed rollover certificates at the same time. The CA must distribute the new CA rollover certificate and keys too all its peers. This process, called rollover, allows for continuous operation of the network while the CAs and their clients are switching from an expiring CA certificate and key pair to a new CA certificate and key pair.

Rollover relies on the PKI infrastructure requirements of trust relationships and synchronized clocks. The PKI trust relationships allow (1) the new CA certificate to be authenticated, and (2) the rollover to be accomplished automatically without the loss of security. Synchronized clocks allow the rollover to be coordinated throughout your network.

Automatic CA Certificate Rollover: How It Works

The CA server must have rollover configured. All levels of CAs must be automatically enrolled and have **auto-rollover** enabled. CA clients support rollover automatically when automatically enrolled. For more information about clients and automatic rollover, see the section "Automatic Certificate Enrollment" in the chapter "Configuring Certificate Enrollment for a PKI".

After CAs have rollover enabled and their clients are automatically enrolled, there are three stages to the automatic CA certificate rollover process.

Stage One: Active CA Certificate and Key Pair Only

In stage one, there is an active CA certificate and key pair only.

Stage Two: Rollover CA Certificate and Key Pair Generation and Distribution

In stage two, the rollover CA certificate and key pair are generated and distributed. The superior CA generates a rollover certificate and key pair. After the CA successfully saves its active configuration, the CA is ready to respond to client requests for the rollover certificate and key pair. When the superior CA receives a request for the new CA certificate and key pair from a client, the CA responds by sending the the new rollover CA certificate and key pair to the requesting client. The clients store the rollover CA certificate and key pair.

Note When a CA generates its rollover certificate and key pair, it must be able to save its active configuration. If the current configuration has been altered, saving of the rollover certificate and key pair will not happen automatically. In this case, the administrator must save the configuration manually or rollover information will be lost.

Stage Three: Rollover CA Certificate and Key Pair Become the Active CA Certificate and Key Pair

In stage three, the rollover CA certificate and key pair become the active CA certificate and key pair. All devices that have stored a valid rollover CA certificate rename the rollover certificate to the active certificate and the once-active certificate and key pair are deleted.

How to Set Up and Deploy a Cisco IOS Certificate Server

This section contains the following procedures:

Generating a Certificate Server RSA Key Pair

Perform this task to manually generate an RSA key pair for the certificate server. Manually generating a certificate server RSA key pair allows you to specify the type of key pair you want to generate, to create an exportable key pair for backup purposes, to specify the key pair storage location, or to specify the key generation location.

If you are running Cisco IOS Release 12.3(8)T or earlier releases, you may want to create an exportable certificate server key pair for backup, or archive purposes. If this task is not performed, the certificate server will automatically generate a key pair, which will not be marked as exportable. Automatic CA certificate archiving was introduced in Cisco IOS Release 12.3(11)T.

As of Cisco IOS Release 12.4(11)T and later releases, if your router has a USB token configured and available, the USB token can be used as cryptographic device in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication of credentials to be performed on a USB token. The private key never leaves the USB token and is not exportable. The public key is exportable. For titles of specific documents about configuring a USB token and making it available to use as a cryptographic device, see the "The following sections provide references related to Cisco IOS certificate server.Related Documents" section.

Note It is recommended that the private key be kept in a secure location and that you regularly archive the certificate server database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*]
4. **crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase*
5. **crypto key import rsa** *key-label* **pem** [**usage-keys** | **signature** | **encryption**] {**terminal** | **url** *url*} [**exportable**] [**on** *devicename:*] *passphrase*
6. **exit**
7. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*] **Example:** `Router (config)# crypto key generate rsa label mycs exportable modulus 2048`	Generates the RSA key pair for the certificate server. When specifying a label name, you must use the same name for the label that you plan to use for the certificate server (via the **crypto pki server** *cs-label* command). By default, the fully qualified domain name (FQDN) of the router is used for the key label. If you manually generate the exportable RSA key pair but wait until after the CA certificate has been generated before issuing the **no shutdown** command, you can use the **crypto ca export pkcs12** command to export a PKCS12 file that contains the certificate server certificate and the private key. By default, the modulus size of a CA key is 1024 bits. The recommended modulus for a CA key is 2048 bits. The range for a CA key is from 350 to 2048 bits.

	Command or Action	Purpose
Step 4	**crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase* **Example:** `Router (config)# crypto key export rsa mycs pem url nvram: 3des PASSWORD`	(Optional) Exports the generated RSA key pair. Allows you to export the generated keys.
Step 5	**crypto key import rsa** *key-label* **pem** [**usage-keys** | **signature** | **encryption**] {**terminal** | **url** *url*} [**exportable**] [**on** *devicename:*] *passphrase* **Example:** `Router (config)# crypto key import rsa mycs2 pem url nvram:mycs PASSWORD`	(Optional) Imports RSA key pair. To create the imported keys on a USB token, use the **on** keyword and specify the appropriate device location. If you exported the RSA keys using the **exportable** keyword and you want to change the RSA key pair to nonexportable, import the key back to the certificate server without the **exportable** keyword. The key cannot be exported again.
Step 6	**exit** **Example:** `Router (config)# exit`	Exits global configuration.
Step 7	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	Displays the RSA public keys of your router.

Examples

The following example generates a general usage 1024-bit RSA key pair on a USB token with the label "ms2" with crypto engine debugging messages shown:

```
Router(config)# crypto key generate rsa on usbtoken0 label ms2 modulus 1024
The name for the keys will be: ms2
% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be on-token, non-exportable...
Jan 7 02:41:40.895: crypto_engine: Generate public/private keypair [OK]
Jan 7 02:44:09.623: crypto_engine: Create signature
Jan 7 02:44:10.467: crypto_engine: Verify signature
Jan 7 02:44:10.467: CryptoEngine0: CRYPTO_ISA_RSA_CREATE_PUBKEY(hw)(ipsec)
Jan 7 02:44:10.467: CryptoEngine0: CRYPTO_ISA_RSA_PUB_DECRYPT(hw)(ipsec)
```

Now, the on-token keys labeled "ms2" may be used for enrollment.

The following example shows the successful import of an encryption key to a configured and available USB tokens:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# crypto key import rsa encryption on usbtoken0 url nvram:e password
% Importing public Encryption key or certificate PEM file...
filename [e-encr.pub]?
Reading file from nvram:e-encr.pub
% Importing private Encryption key PEM file...
Source filename [e-encr.prv]?
Reading file from nvram:e-encr.prv
% Key pair import succeeded.
```

Configuring Certificate Servers

The following tasks explain how to configure a certificate server, a subordinate certificate server, or an RA-mode certificate server, and how to enable automatic rollover.

- Configuring a Certificate Server, page 13
- Configuring a Subordinate Certificate Server, page 15
- Configuring a Certificate Server to Run in RA Mode, page 21
- Configuring the Root Certificate Server to Delegate Enrollment Tasks to the RA Mode Certificate Server, page 23

Prerequisites for Automatic CA Certificate Rollover

When configuring a certificate server, for automatic CA certificate rollover to run successfully, the following prerequisites are applicable for your CA servers:

- You must be running Cisco IOS Release 12.4(2)T or a later release on your CA servers.
- Your CA server must be enabled and fully configured with a reliable time of day, an available key pair, a self-signed, valid CA certificate associated with the key pair, a CRL, an accessible storage device, and an active HTTP/SCEP server.
- CA clients must have successfully completed automatic enrollment and have autoenrollment enabled with the same certificate server.

Note If you are running Cisco IOS 12.4(2)T or earlier releases, only your root CA will support automatic CA certificate rollover functionality. Cisco IOS 12.4(4)T or later releases support all CAs—root CAs, subordinate CAs, and RA-mode CAs.

Restrictions for Automatic CA Certificate Rollover

When configuring a certificate server, in order for automatic CA certificate rollover to run successfully, the following restrictions are applicable:

- SCEP must be used to support rollover. Any device that enrolls with the PKI using an alternative to SCEP as the certificate management protocol or mechanism (such as enrollment profiles, manual enrollment, or TFTP enrollment) will not be able to take advantage of the rollover functionality provided by SCEP.
- If you have automatic archive configured on your network and the archive fails, rollover will not occur because the certificate server will not enter the rollover state, and the rollover certificate and key pair will not be automatically saved.

Configuring a Certificate Server

Perform this task to configure a Cisco IOS certificate server and enable automatic rollover.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip http server**
4. **crypto pki server** *cs-label*
5. **no shutdown**
6. **auto-rollover** [*time-period*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip http server** **Example:** `Router(config)# ip http server`	Enables the HTTP server on your system.
Step 4	**crypto pki server** *cs-label* **Example:** `Router(config)# crypto pki server server-pki`	Defines a label for the certificate server and enters certificate server configuration mode. **Note** If you manually generated an RSA key pair, the *cs-label* argument must match the name of the key pair.
Step 5	**no shutdown** **Example:** `Router(cs-server)# no shutdown`	(Optional) Enables the certificate server. **Note** Only use this command at this point if you want to use the preconfigured default functionality. That is, do not issue this command just yet if you plan to change any of the default settings as shown in the task "Configuring Certificate Server Functionality."
Step 6	**auto-rollover** [*time-period*] **Example:** `Router(cs-server)# auto-rollover 90`	(Optional) Enables the automated CA certificate rollover functionality. • *time-period*—default is 30 days.

Examples

The following example shows how to configure the certificate server "ca":

```
Router(config)# crypto pki server ca
Router(cs-server)# no shutdown


% Once you start the server, you can no longer change some of
% the configuration.
Are you sure you want to do this? [yes/no]: yes
% Generating 1024 bit RSA keys ...[OK]
```

```
% Certificate Server enabled.
Router(cs-server)# end
!
Router# show crypto pki server

Certificate Server ca:
    Status: enabled, configured
    CA cert fingerprint: 5A856122 4051347F 55E8C246 866D0AC3
    Granting mode is: manual
    Last certificate issued serial number: 0x1
    CA certificate expiration timer: 19:44:57 GMT Oct 14 2006
    CRL NextUpdate timer: 19:45:25 GMT Oct 22 2003
    Current storage dir: nvram:
    Database Level: Complete - all issued certs written as <serialnum>.cer
```

The following example shows how to enable automated CA certificate rollover on the server mycs with the **auto-rollover** command. The **show crypto pki server** command shows that the automatic rollover has been configured on the server mycs with an overlap period of 25 days.

```
Router(config)# crypto pki server mycs
Router(cs-server)# auto-rollover 25
Router(cs-server)# no shut

%Some server settings cannot be changed after CA certificate generation.
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
% Exporting Certificate Server signing certificate and keys...

% Certificate Server enabled.
Router(cs-server)#

Router# show crypto pki server

Certificate Server mycs:
    Status:enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name:CN=mycs
    CA cert fingerprint:70AFECA9 211CDDCC 6AA9D7FF 3ADB03AE
    Granting mode is:manual
    Last certificate issued serial number:0x1
    CA certificate expiration timer:00:49:26 PDT Jun 20 2008
    CRL NextUpdate timer:00:49:29 PDT Jun 28 2005
    Current storage dir:nvram:
    Database Level:Minimum - no cert data written to storage
    Auto-Rollover configured, overlap period 25 days
    Autorollover timer:00:49:26 PDT May 26 2008
```

Configuring a Subordinate Certificate Server

Perform this task to configure a subordinate certificate server to grant all or certain SCEP or manual certificate requests and to enable automatic rollover.

Restrictions

- You must be running Cisco IOS Release 12.3(14)T or a later release. (Versions prior to Cisco IOS software Release 12.3(14)T support only one certificate server and no hierarchy; that is, subordinate certificate servers are not supported.)
- The root certificate server should be a Cisco IOS certificate server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*
5. **exit**
6. **crypto pki server** *cs-label*
7. **issuer name** *DN-string*
8. **mode sub-cs**
9. **auto-rollover** [*time-period*]
10. **grant auto rollover {ca-cert | ra-cert}**
11. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router (config)# crypto pki trustpoint sub`	Declares the trustpoint that your subordinate certificate server should use and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:** `Router (ca-trustpoint)# enrollment url http://192.0.2.6`	Specifies the enrollment URL of the issuing CA certificate server (root certificate server).
Step 5	**exit** **Example:** `Router (ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 6	**crypto pki server** *cs-label* **Example:** `Router(config)# crypto pki server sub`	Enables a Cisco IOS certificate server and enters cs-server configuration mode. **Note** The subordinate server must have the same name as the trustpoint that was created in Step 3 above.

	Command or Action	Purpose
Step 7	**issuer name** *DN-string* **Example:** `Router(cs-server)# issuer-name CN=sub CA, O=Cisco, C=us`	(Optional) Specifies the DN as the CA issuer name for the certificate server.
Step 8	**mode sub-cs** **Example:** `Router(cs-server)# mode sub-cs`	Places the PKI server into sub-certificate server mode.
Step 9	**auto-rollover** [*time-period*] **Example:** `Router(cs-server)# auto-rollover 90`	(Optional) Enables the automated CA certificate rollover functionality. • *time-period*—default is 30 days.
Step 10	**grant auto rollover** {**ca-cert** | **ra-cert**} **Example:** `Router(cs-server)# grant auto rollover ca-cert`	(Optional) Automatically grants reenrollment requests for subordinate CAs and RA-mode CAs without operator intervention. • **ca-cert** —Specifies that the subordinate CA rollover certificate will be automatically granted. • **ra-cert**—Specifies that the RA-mode CA rollover certificate will be automatically granted. **Note** If this is the first time that a subordinate certificate server is enabled and enrolled, the certificate request must be manually granted.
Step 11	**no shutdown** **Example:** `Router(cs-server)# no shutdown`	Enables or reenables the certificate server. If this is the first time that a subordinate certificate server is enabled, the certificate server will generate the key and obtain its signing certificate from the root certificate server.

Examples

If the certificate server fails to enable or if the certificate server has trouble handling the request that has been configured, you can use the **debug crypto pki server** command to troubleshoot your configuration as shown in the following examples (Clock Not Set and Trustpoint Not Configured):

```
Router# debug crypto pki server
```

Clock Not Set

```
Router (config)# crypto pki server sub
Router (cs-server)# mode sub-cs
Router (cs-server)# no shutdown
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key % or type Return to exit
Password:
*Jan  6 20:57:37.667: CRYPTO_CS: enter FSM: input state initial, input signal no shut

Re-enter password:
% Generating 1024 bit RSA keys ...
```

```
*Jan  6 20:57:45.303: CRYPTO_CS: starting enabling checks
*Jan  6 20:57:45.303: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]
% Time has not been set. Cannot start the Certificate server
```

Trustpoint Not Configured

```
Router (config)# crypto pki server sub
Router (cs-server)# mode sub-cs
Router (cs-server)# no shutdown

%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key or type Return to exit
Password:
Jan  6 21:00:15.961: CRYPTO_CS: enter FSM: input state initial, input signal no shut.
Jan  6 21:03:34.309: CRYPTO_CS: enter FSM: input state initial, input signal time set.
Jan  6 21:03:34.313: CRYPTO_CS: exit FSM: new state initial.
Jan  6 21:03:34.313: CRYPTO_CS: cs config has been unlocked
Re-enter password:
% Generating 1024 bit RSA keys ...
Jan  6 21:03:44.413: CRYPTO_CS: starting enabling checks
Jan  6 21:03:44.413: CRYPTO_CS: associated trust point 'sub' does not exist; generated
automatically
Jan  6 21:03:44.417: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]

Jan  6 21:04:03.993: CRYPTO_CS: nvram filesystem
Jan  6 21:04:04.077: CRYPTO_CS: serial number 0x1 written.
You must specify an enrollment URL for this CA before you can authenticate it.
% Failed to authenticate the Certificate Authority
```

If the certificate server fails to obtain its signing certificate from the root certificate server, you can use the **debug crypto pki transactions** command to troubleshoot your configuration as shown in the following example:

```
Router# debug crypto pki transactions

Jan  6 21:07:00.311: CRYPTO_CS: enter FSM: input state initial, input signal time set
Jan  6 21:07:00.311: CRYPTO_CS: exit FSM: new state initial
Jan  6 21:07:00.311: CRYPTO_CS: cs config has been unlocked no sh
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key % or type Return to exit
Password:
Jan  6 21:07:03.535: CRYPTO_CS: enter FSM: input state initial, input signal no shut
Re-enter password:
% Generating 1024 bit RSA keys ...
Jan  6 21:07:10.619: CRYPTO_CS: starting enabling checks
Jan  6 21:07:10.619: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]

Jan  6 21:07:20.535: %SSH-5-ENABLED: SSH 1.99 has been enabled
Jan  6 21:07:25.883: CRYPTO_CS: nvram filesystem
Jan  6 21:07:25.991: CRYPTO_CS: serial number 0x1 written.
Jan  6 21:07:27.863: CRYPTO_CS: created a new serial file.
Jan  6 21:07:27.863: CRYPTO_CS: authenticating the CA 'sub'
Jan  6 21:07:27.867: CRYPTO_PKI: Sending CA Certificate Request:
GET /cgi-bin/pkiclient.exe?operation=GetCACert&message=sub HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Cisco PKI)

Jan 6 21:07:27.867: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:07:27.871: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6 Certificate has the
following attributes:
     Fingerprint MD5: 328ACC02 52B25DB8 22F8F104 B6055B5B
     Fingerprint SHA1: 02FD799D DD40C7A8 61DC53AB 1E89A3EA 2A729EE2

% Do you accept this certificate? [yes/no]:
Jan  6 21:07:30.879: CRYPTO_PKI: http connection opened
```

```
Jan 6 21:07:30.903: CRYPTO_PKI: HTTP response header:
  HTTP/1.1 200 OK
Date: Thu, 06 Jan 2005 21:07:30 GMT
Server: server-IOS
Content-Type: application/x-x509-ca-cert
Expires: Thu, 06 Jan 2005 21:07:30 GMT
Last-Modified: Thu, 06 Jan 2005 21:07:30 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Accept-Ranges: none

Content-Type indicates we have received a CA certificate.

Jan  6 21:07:30.903: Received 507 bytes from server as CA certificate:
Jan  6 21:07:30.907: CRYPTO_PKI: transaction GetCACert completed
Jan  6 21:07:30.907: CRYPTO_PKI: CA certificate received.
Jan  6 21:07:30.907: CRYPTO_PKI: CA certificate received.
Jan  6 21:07:30.927: CRYPTO_PKI: crypto_pki_authenticate_tp_cert()

Jan  6 21:07:30.927: CRYPTO_PKI: trustpoint sub authentication status = 0 y Trustpoint CA
certificate accepted.%

% Certificate request sent to Certificate Authority

% Enrollment in progress...
Router (cs-server)#
Jan 6 21:07:51.772: CRYPTO_CA: certificate not found
Jan 6 21:07:51.772: CRYPTO_CA: certificate not found
Jan 6 21:07:52.460: CRYPTO_CS: Publishing 213 bytes to crl file nvram:sub.crl
Jan 6 21:07:54.348: CRYPTO_CS: enrolling the server's trustpoint 'sub'
Jan 6 21:07:54.352: CRYPTO_CS: exit FSM: new state check failed
Jan 6 21:07:54.352: CRYPTO_CS: cs config has been locked
Jan 6 21:07:54.356: CRYPTO_PKI: transaction PKCSReq completed
Jan 6 21:07:54.356: CRYPTO_PKI: status:
Jan 6 21:07:55.016: CRYPTO_PKI:  Certificate Request Fingerprint MD5: 1BA027DB 1C7860C7
EC188F65 64356C80
Jan 6 21:07:55.016: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 840DB52C E17614CB
0C7BE187 0DFC884D D32CAA75
Jan 6 21:07:56.508: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:07:56.508: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6
Jan 6 21:07:56.516: CRYPTO_PKI: http connection opened
Jan 6 21:07:59.136: CRYPTO_PKI: received msg of 776 bytes
Jan 6 21:07:59.136: CRYPTO_PKI: HTTP response header:
  HTTP/1.1 200 OK
  Date: Thu, 06 Jan 2005 21:07:57 GMT
  Server: server-IOS
  Content-Type: application/x-pki-message
  Expires: Thu, 06 Jan 2005 21:07:57 GMT
  Last-Modified: Thu, 06 Jan 2005 21:07:57 GMT
  Cache-Control: no-store, no-cache, must-revalidate
  Pragma: no-cache
  Accept-Ranges: none


Jan 6 21:07:59.324: The PKCS #7 message has 1 verified signers.
Jan 6 21:07:59.324: signing cert: issuer=cn=root1
Jan 6 21:07:59.324: Signed Attributes:

Jan 6 21:07:59.328: CRYPTO_PKI: status = 102: certificate request pending
Jan 6 21:08:00.788: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:08:00.788: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6
Jan 6 21:08:00.796: CRYPTO_PKI: http connection opened
Jan 6 21:08:11.804: CRYPTO_PKI: received msg of 776 bytes
Jan 6 21:08:11.804: CRYPTO_PKI: HTTP response header: HTTP/1.1 200 OK
```

```
  Date: Thu, 06 Jan 2005 21:08:01 GMT
  Server: server-IOS
  Content-Type: application/x-pki-message
  Expires: Thu, 06 Jan 2005 21:08:01 GMT
  Last-Modified: Thu, 06 Jan 2005 21:08:01 GMT
  Cache-Control: no-store, no-cache, must-revalidate
  Pragma: no-cache
  Accept-Ranges: none


Jan  6 21:08:11.992: The PKCS #7 message has 1 verified signers.
Jan  6 21:08:11.992: signing cert: issuer=cn=root1
Jan  6 21:08:11.996: Signed Attributes:

Jan  6 21:08:11.996: CRYPTO_PKI: status = 102: certificate request pending
Jan  6 21:08:21.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:08:31.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:08:41.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:08:51.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:09:01.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:09:11.996: CRYPTO_PKI: resend GetCertInitial, 1
Jan  6 21:09:11.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:09:11.996: CRYPTO_PKI: resend GetCertInitial for session: 0
Jan  6 21:09:11.996: CRYPTO_PKI: can not resolve server name/IP address
Jan  6 21:09:11.996: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6
Jan  6 21:09:12.024: CRYPTO_PKI: http connection opened% Exporting Certificate Server
signing certificate and keys...

Jan  6 21:09:14.784: CRYPTO_PKI: received msg of 1611 bytes
Jan  6 21:09:14.784: CRYPTO_PKI: HTTP response header:
  HTTP/1.1 200 OK
  Date: Thu, 06 Jan 2005 21:09:13 GMT
  Server: server-IOS
  Content-Type: application/x-pki-message
  Expires: Thu, 06 Jan 2005 21:09:13 GMT
  Last-Modified: Thu, 06 Jan 2005 21:09:13 GMT
  Cache-Control: no-store, no-cache, must-revalidate
  Pragma: no-cache
  Accept-Ranges: none


Jan  6 21:09:14.972: The PKCS #7 message has 1 verified signers.
Jan  6 21:09:14.972: signing cert: issuer=cn=root1
Jan  6 21:09:14.972: Signed Attributes:

Jan  6 21:09:14.976: CRYPTO_PKI: status = 100: certificate is granted
Jan  6 21:09:15.668: The PKCS #7 message contains 1 certs and 0 crls.
Jan  6 21:09:15.688: Newly-issued Router Cert: issuer=cn=root serial=2
Jan  6 21:09:15.688: start date: 21:08:03 GMT Jan 6 2005
Jan  6 21:09:15.688: end date: 21:08:03 GMT Jan 6 2006
Jan  6 21:09:15.688: Router date: 21:09:15 GMT Jan 6 2005
Jan  6 21:09:15.692: Received router cert from CA
Jan  6 21:09:15.740: CRYPTO_CA: certificate not found
Jan  6 21:09:15.744: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
Jan  6 21:09:15.744: %PKI-6-CERTRET: Certificate received from Certificate Authority
Jan  6 21:09:15.744: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
Jan  6 21:09:15.744: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
Jan  6 21:09:15.748: CRYPTO_CS: enter FSM: input state check failed, input signal cert
configured
Jan  6 21:09:15.748: CRYPTO_CS: starting enabling checks
Jan  6 21:09:15.748: CRYPTO_CS: nvram filesystem
Jan  6 21:09:15.796: CRYPTO_CS: found existing serial file.
Jan  6 21:09:15.820: CRYPTO_CS: old router cert flag 0x4
Jan  6 21:09:15.820: CRYPTO_CS: new router cert flag 0x44
```

```
Jan  6 21:09:18.432: CRYPTO_CS: DB version 1
Jan  6 21:09:18.432: CRYPTO_CS: last issued serial number is 0x1
Jan  6 21:09:18.480: CRYPTO_CS: CRL file sub.crl exists.
Jan  6 21:09:18.480: CRYPTO_CS: Read 213 bytes from crl file sub.crl.
Jan  6 21:09:18.532: CRYPTO_CS: SCEP server started
Jan  6 21:09:18.532: CRYPTO_CS: exit FSM: new state enabled
Jan  6 21:09:18.536: CRYPTO_CS: cs config has been locked
Jan  6 21:09:18.536: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
```

If the certificate server fails to enable or if the certificate server has trouble handling the request that has been configured, you can use the **debug crypto pki server** command to troubleshoot the progress of an enrollment. This command can also be used to debug the root CA (turn it on at the root CA).

Configuring a Certificate Server to Run in RA Mode

Restrictions for Configuring a Certificate Server for RA Mode

When the Cisco IOS certificate server is acting as an RA, the issuing CA should be a Cisco IOS certificate server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*
5. **subject-name** *x.500-name*
6. **exit**
7. **crypto pki server** *cs-label*
8. **mode ra**
9. **auto-rollover** [*time-period*]
10. **grant auto rollover {ca-cert | ra-cert}**
11. **no shutdown**
12. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto pki trustpoint** *name* **Example:** `Router (config)# crypto pki trustpoint ra-server`	Declares the trustpoint that your RA mode certificate server should use and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:** `Router (ca-trustpoint)# enrollment url http://ca-server.company.com`	Specifies the enrollment URL of the issuing CA certificate server (root certificate server).
Step 5	**subject-name** *x.500-name* **Example:** `Router (ca-trustpoint)# subject-name cn=ioscs RA`	(Optional) Specifies the subject name the RA will use. **Note** Include "cn=ioscs RA" or "ou=ioscs RA" in the subject name so that the issuing CA certificate server can recognize the RA (see Step 7 below).
Step 6	**exit** **Example:** `Router (ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 7	**crypto pki server** *cs-label* **Example:** `Router(config)# crypto pki server ra-server`	Enables a Cisco IOS certificate server and enters cs-server configuration mode. **Note** The certificate server must have the same name as the trustpoint that was created in Step 3 above.
Step 8	**mode ra** **Example:** `Router(cs-server)# mode ra`	Places the PKI server into RA certificate server mode.
Step 9	**auto-rollover** [*time-period*] **Example:** `Router(cs-server)# auto-rollover 90`	(Optional) Enables the automatic CA certificate rollover functionality. • *time-period*—default is 30 days.
Step 10	**grant auto rollover** {**ca-cert** | **ra-cert**} **Example:** `Router(cs-server)# grant auto rollover ra-cert`	(Optional) Automatically grants reenrollment requests for subordinate CAs and RA-mode CAs without operator intervention. • **ca-cert** —Specifies that the subordinate CA rollover certificate will be automatically granted. • **ra-cert**—Specifies that the RA-mode CA rollover certificate will be automatically granted. If this is the first time that a subordinate certificate server is enabled and enrolled, the certificate request must be manually granted.

	Command or Action	Purpose
Step 11	**no shutdown** **Example:** `Router(cs-server)# no shutdown`	Enables the certificate server. **Note** After this command is issued, the RA will automatically enroll with the root certificate server. After the RA certificate has been successfully received, you must issue the **no shutdown** command again, which reenables the certificate server.
Step 12	**no shutdown** **Example:** `Router(cs-server)# no shutdown`	Reenables the certificate server.

Configuring the Root Certificate Server to Delegate Enrollment Tasks to the RA Mode Certificate Server

Perform the following steps on the router that is running the issuing certificate server; that is, configure the root certificate server that is delegating enrollment tasks to the RA mode certificate server.

Note Granting enrollment requests for an RA is essentially the same process as granting enrollment requests for client devices—except that enrollment requests for an RA are displayed in the section "RA certificate requests" of the command output for the **crypto pki server info-requests** command.

SUMMARY STEPS

1. **enable**
2. **crypto pki server** *cs-label* **info requests**
3. **crypto pki server** *cs-label* **grant** *req-id*
4. **configure terminal**
5. **crypto pki server** *cs-label*
6. **grant ra-auto**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki server** *cs-label* **info requests** **Example:** `Router# crypto pki server root-server info requests`	Displays the outstanding RA certificate request. **Note** This command is issued on the router that is running the issuing certificate server.

	Command or Action	Purpose
Step 3	**crypto pki server** *cs-label* **grant** *req-id* **Example:** `Router# crypto pki server root-server grant 9`	Grants the pending RA certificate request. **Note** Because the issuing certificate server will delegate the enrollment request verification task to the RA, you must pay extra attention to the RA certificate request before granting it.
Step 4	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 5	**crypto pki server** *cs-label* **Example:** `Router (config)# crypto pki server root-server`	Enables a Cisco IOS certificate server and enters cs-server configuration mode.
Step 6	**grant ra-auto** **Example:** `Router(cs-server)# grant ra-auto`	(Optional) Specifies that all enrollment requests from an RA are to be granted automatically. **Note** For the **grant ra-auto** command to work, you have to include "cn=ioscs RA" or "ou=ioscs RA" in the subject name of the RA certificate. (See Step 2 above.)

What to Do Next

After you have configured a certificate server, you can use the preconfigured default values or specify values via the CLI for the functionality of the certificate server. If you choose to specify values other than the defaults, see the following section, "Configuring Certificate Server Functionality."

Configuring Certificate Server Functionality

After you have enabled a certificate server and are in certificate server configuration mode, use any of the steps in this task to configure basic certificate server functionality values other than the default values.

Certificate Server Default Values and Recommended Values

The default values for a certificate server are intended to address a relatively small network (of about ten devices). For example, the database settings are minimal (via the **database level minimal** command) and the certificate server handles all CRL requests via SCEP. For larger networks, it is recommended that you use either the database setting "names" or "complete" (as described in the **database level** command) for possible audit and revocation purposes. Depending on the CRL checking policy, you should also use an external CDP in a larger network.

Certificate Server File Storage and Publication Locations

You have the flexibility to store file types to different storage and publication locations.

SUMMARY STEPS

1. **database url** *root-url*
2. **database url** {**cnm** | **crl** | **crt** | **p12** | **pem** | **ser**} *root-url*
3. **database url** {**cnm** | **crl** | **crt**} **publish** *root-url*
4. **database level** {**minimal** | **names** | **complete**}
5. **database username** *username* [**password** [*encr-type*] *password*]
6. **database archive** {**pkcs12** | **pem}** [**password** [*encr-type*] *password*]
7. **issuer-name** *DN-string*
8. **lifetime** {**ca-certificate** | **certificate**} *time*
9. **lifetime crl** *time*
10. **lifetime enrollment-request** *time*
11. **cdp-url** *url*
12. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**database url** *root-url* **Example:** `Router (cs-server)# database url tftp://cert-svr-db.company.com`	Specifies the primary location where database entries for the certificate server will be written out. If this command is not specified, all database entries will be written to NVRAM.
Step 2	**database url** {**cnm** | **crl** | **crt** | **p12** | **pem** | **ser**} *root-url* **Example:** `Router (cs-server)# database url ser nvram:`	Specifies certificate server critical file storage location by file type. **Note** If this command is not specified, all critical files will be stored to the primary location if specified. If the primary location is not specified, all critical files will be stored to NVRAM.
Step 3	**database url** {**cnm** | **crl** | **crt**} **publish** *root-url* **Example:** `Router (cs-server)# database url crl publish tftp://csdb_specific_crl_files.company.com`	Specifies certificate server publish location by file type. **Note** If this command is not specified, all publish files will be stored to the primary location if specified. If the primary location is not specified, all publish files will be stored to NVRAM.

	Command or Action	Purpose
Step 4	**database level** {**minimal** | **names** | **complete**} **Example:** `Router (cs-server)# database level complete`	Controls what type of data is stored in the certificate enrollment database. • **minimal**—Enough information is stored only to continue issuing new certificates without conflict; the default value. • **names**—In addition to the information given in the minimal level, the serial number and subject name of each certificate. • **complete**—In addition to the information given in the minimal and names levels, each issued certificate is written to the database. **Note** The **complete** keyword produces a large amount of information; if it is issued, you should also specify an external TFTP server in which to store the data via the **database url** command.
Step 5	**database username** *username* [**password** [*encr-type*] *password*] **Example:** `Router (cs-server)# database username user password PASSWORD`	(Optional) Sets a username and password when a user is required to access a primary certificate enrollment database storage location.
Step 6	**database archive** {**pkcs12** | **pem}** [**password** [*encr-type*] *password*] **Example:** `Router (cs-server)# database archive pem`	(Optional) Sets the CA key and CA certificate archive format and password to encrypt the file. The default value is **pkcs12**, so if this subcommand is not configured, autoarchiving will still be done, and the PKCS12 format will be used. • The password is optional. If it is not configured, you will be prompted for the password when the server is turned on for the first time. **Note** It is recommended that you remove the password from the configuration after the archive is finished.
Step 7	**issuer-name** *DN-string* **Example:** `Router (cs-server)# issuer-name my-server`	(Optional) Sets the CA issuer name to the specified distinguished name (*DN-string*). The default value is as follows: **issuer-name cn**={*cs-label*}.
Step 8	**lifetime** {**ca-certificate** | **certificate**} *time* **Example:** `Router (cs-server)# lifetime certificate 888`	(Optional) Specifies the lifetime, in days, of a CA certificate or a certificate. Valid values range from 1 day to 1825 days. The default CA certificate lifetime is 3 years; the default certificate lifetime is 1 year. The maximum certificate lifetime is 1 month less than the lifetime of the CA certificate.
Step 9	**lifetime crl** *time* **Example:** `Router (cs-server)# lifetime crl 333`	(Optional) Defines the lifetime, in hours, of the CRL that is used by the certificate server. Maximum lifetime value is 336 hours (2 weeks). The default value is 168 hours (1 week).

	Command or Action	Purpose
Step 10	**lifetime enrollment-request** *time* **Example:** `Router (cs-server)# lifetime enrollment-request 888`	(Optional) Specifies how long an enrollment request should stay in the enrollment database before being removed. Maximum lifetime is 1000 hours.
Step 11	**cdp-url** *url* **Example:** `Router (cs-server)# cdp-url http://my-cdp.company.com`	(Optional) Defines the CDP location to be used in the certificates that are issued by the certificate server. • The URL must be an HTTP URL. If you have PKI clients that are not running Cisco IOS software and that do not support a SCEP GetCRL request, use the following URL format: http://server.company.com/certEnroll/filename.crl Or, if your Cisco IOS certificate server is also configured as your CDP, use the following URL format http://cs-addr/cgi-bin/pkiclient.exe?operation=GetCRL where *cs-addr* is the location of the certificate server. **Note** Although this command is optional, it is strongly recommended for any deployment scenario.
Step 12	**no shutdown** **Example:** `Router (cs-server)# no shutdown`	Enables the certificate server. You should issue this command only after you have completely configured your certificate server.

Examples

The following example shows how to configure a CDP location where the PKI clients do not support SCEP GetCRL requests:

```
Router(config)# crypto pki server aaa
Router(cs-server)# database level minimum
Router(cs-server)# database url tftp://10.1.1.1/username1/
Router(cs-server)# issuer-name CN=aaa
Router(cs-server)# cdp-url http://server.company.com/certEnroll/aaa.crl
```

After a certificate server has been enabled on a router, the **show crypto pki server** command displays the following output:

```
Router# show crypto pki server

    Certificate Server status:enabled, configured
    Granting mode is:manual
    Last certificate issued serial number:0x1
    CA certificate expiration timer:19:31:15 PST Nov 17 2006
    CRL NextUpdate timer:19:31:15 PST Nov 25 2003
    Current storage dir:nvram:
    Database Level:Minimum - no cert data written to storage
```

Working with Automatic CA Certificate Rollover

This section describes different methods of initiating automatic CA certificate rollover on the server and obtaining rollover certificates. Use the following tasks as appropriate:

- Starting Automated CA Certificate Rollover Immediately, page 28
- Requesting a Certificate Server Client's Rollover Certificate, page 28
- Exporting a CA Rollover Certificate, page 29

Starting Automated CA Certificate Rollover Immediately

Use this task to initiate the automated CA certificate rollover process immediately on your root CA server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki server** *cs-label* [**rollover** [**cancel**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`crypto pki server` *cs-label* [`rollover` [`cancel`]] **Example:** `Router(config)# crypto pki server mycs rollover`	Immediately starts the CA certificate rollover process by generating a shadow CA certificate. To delete the CA certificate rollover certificate and keys, use the **cancel** keyword.

Requesting a Certificate Server Client's Rollover Certificate

Use this task to request a certificate server client's rollover certificate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki server** *cs-label* [**rollover request pkcs10 terminal**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki server** *cs-label* [**rollover request pkcs10 terminal**] **Example:** `Router(config)# crypto pki server mycs rollover request pkcs10 terminal`	Requests a client rollover certificate from the server.

Examples

The following example shows a rollover certificate request being inputted into the server:

```
Router# crypto pki server mycs rollover request pkcs10 terminal

% Enter Base64 encoded or PEM formatted PKCS10 enrollment request.

% End with a blank line or "quit" on a line by itself.

-----BEGIN CERTIFICATE REQUEST-----

MIIBUTCBuwIBADASMRAwDgYDVQQDEwdOZXdSb290MIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDMHeev1ERSs320zbLQQk+3lhV/R2HpYQ/iM6uT1jkJf5iy0UPR
wF/X16yUNmG+ObiGiW9fsASF0nxZw+fO7d2X2yh1PakfvF2wbP27C/sgJNOw9uPf
sBxEc40Xe0d5FMh0YKOSAShfZYKOflnyQR2Drmm2x/33QGol5QyRvjkeWQIDAQAB
oAAwDQYJKoZIhvcNAQEEBQADgYEALM90r4d79X6vxhD0qjuYJXfBCOvv4FNyFsjr
aBS/y6CnNVYySF8UBUohXYIGTWf4I4+sj6i8gYfoFUW1/L82djS18TLrUr6wpCOs
RqfAfps7HW1e4cizOfjAUU+C7lNcobCAhwF1o6q2nIEjpQ/2yfK9O7sb3SCJZBfe
eW3tyCo=

-----END CERTIFICATE REQUEST-----
```

Exporting a CA Rollover Certificate

Use this task to export a CA rollover certificate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki export** *trustpoint* **pem** {**terminal** | **url** *url*} [**rollover**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki export** *trustpoint* **pem** {**terminal** | **url** *url*} [**rollover**] **Example:** `Router(config)# crypto pki export mycs pem terminal rollover`	Exports a CA shadow certificate.

Maintaining, Verifying, and Troubleshooting the Certificate Server, Certificates, and the CA

Use the tasks in this section to help maintain, verify, and troubleshoot the certificate server, certitudes and the CA as appropriate:

- Managing the Enrollment Request Database, page 30
- Removing Enrollment Requests from the Enrollment Request Database: Examples, page 38
- Deleting a Certificate Server, page 33
- Verifying and Troubleshooting Certificate Server and CA Status, page 34
- Verifying CA Certificate Information, page 34

Managing the Enrollment Request Database

SCEP supports two client authentication mechanisms—manual and preshared key. Manual enrollment requires the administrator at the CA server to specifically authorize the enrollment requests; enrollment using preshared keys allows the administrator to preauthorize enrollment requests by generating a one-time password (OTP).

Use any of the optional steps within this task to help manage the enrollment request database by performing functions such as specifying enrollment processing parameters that are to be used by SCEP and by controlling the run-time behavior or the certificate server.

SUMMARY STEPS

1. **enable**
2. **crypto pki server** *cs-label* **grant** {**all** | *req-id*}
3. **crypto pki server** *cs-label* **reject** {**all** | *req-id*}
4. **crypto pki server** *cs-label* **password generate** [*minutes*]

5. **crypto pki server** *cs-label* **revoke** *certificate-serial-number*
6. **crypto pki server** *cs-label* **request pkcs10** {*url* | **terminal**} [**base64** | **pem**]
7. **crypto pki server** *cs-label* **info crl**
8. **crypto pki server** *cs-label* **info requests**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki server** *cs-label* **grant** {**all** | *req-id*} **Example:** `Router# crypto pki server mycs grant all`	Grants all or specific SCEP requests.
Step 3	**crypto pki server** *cs-label* **reject** {**all** | *req-id*} `Router# crypto pki server mycs reject all`	Rejects all or specific SCEP requests.
Step 4	**crypto pki server** *cs-label* **password generate** [*minutes*] **Example:** `Router# crypto pki server mycs password generate 75`	Generates a OTP for SCEP requests. • *minutes*—Length of time, in minutes, that the password is valid. Valid values range from 1 to 1440 minutes. The default is 60 minutes. **Note** Only one OTP is valid at a time; if a second OTP is generated, the previous OTP is no longer valid.
Step 5	**crypto pki server** *cs-label* **revoke** *certificate-serial-number* **Example:** `Router# crypto pki server mycs revoke 3`	Revokes a certificate on the basis of its serial number. • *certificate-serial-number*—One of the following options: – A string with a leading 0x, which is treated as a hexadecimal value – A string with a leading 0 and no x, which is treated as octal – All other strings, which are treated as decimal

	Command or Action	Purpose
Step 6	**crypto pki server** *cs-label* **request pkcs10** {*url* | **terminal**} [**base64** | **pem**] **Example:** `Router# crypto pki server mycs request pkcs10 terminal pem`	Manually adds either a base64-encoded or PEM-formatted PKCS10 certificate enrollment request to the request database. After the certificate is granted, it will be displayed on the console terminal using base64 encoding. • **pem**—Specifies the certificate will be returned *with* PEM headers automatically added to the certificate after the certificate is granted, regardless of whether PEM headers were used in the request. • **base64**—Specifies the certificate will be returned *without* privacy-enhanced mail (PEM) headers, regardless of whether PEM headers were used in the request.
Step 7	**crypto pki server** *cs-label* **info crl** **Example:** `Router# crypto pki server mycs info crl`	Displays information regarding the status of the current CRL.
Step 8	**crypto pki server** *cs-label* **info requests** **Example:** `Router# crypto pki server mycs info requests`	Displays all outstanding certificate enrollment requests.

Removing Requests from the Enrollment Request Database

After the certificate server receives an enrollment request, the server can leave the request in a pending state, reject it, or grant it. The request stays in the enrollment request database for 1 week until the client polls the certificate server for the result of the request. If the client exits and never polls the certificate server, you can remove either individual requests or all requests from the database.

Use this task to remove requests from the database and allow the server to be returned to a clean slate with respect to the keys and transaction IDs. Also, you can use this task to help troubleshoot a SCEP client that may not be behaving properly.

SUMMARY STEPS

1. **enable**
2. **crypto pki server** *cs-label* **remove** {**all** | *req-id*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki server** *cs-label* **remove** {**all** | *req-id*} **Example:** `Router# crypto pki server mycs remove 15`	Removes enrollment requests from the enrollment request database.

Deleting a Certificate Server

Users can delete a certificate server from the PKI configuration if they no longer want it on the configuration. Typically, a subordinate certificate server or an RA is being deleted. However, users may delete a root certificate server if they are moving it to another device via the archived RSA keys.

Perform this task to delete a certificate server from your PKI configuration.

Note When a certificate server is deleted, the associated trustpoint and key are also deleted.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no crypto pki server** *cs-label*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no crypto pki server** *cs-label* **Example:** `Router (config)# no crypto pki server mycs`	Deletes a certificate server and associated trustpoint and key.

Verifying and Troubleshooting Certificate Server and CA Status

Use any of the following optional steps to verify the status of the certificate server or the CA.

SUMMARY STEPS

1. **enable**
2. **debug crypto pki server**
3. **dir** *filesystem*:

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto pki server** **Example:** `Router# debug crypto pki server`	Enables debugging for a crypto PKI certificate server. • This command can be used for monitoring the progress of an enrollment and for troubleshooting if the certificate server fails to respond or if the certificate server has trouble handling the request that has been configured.
Step 3	**dir** *filesystem*: **Example:** `Router# dir slot0:`	Displays a list of files on a file system. • This command can be used to verify the certificate server autoarchived file if the **database url** command was entered to point to a local file system. You should be able to at least see "*cs-label*.ser" and "*cs-label*.crl" files in the database.

Verifying CA Certificate Information

To obtain information relating to the CA certificates including the certificate server rollover process, rollover certificates, and timers, you may use any of the following commands.

Note These commands are not exclusive to shadow certificate information. If no shadow certificate exists, the following commands will simply display the active certificate information.

SUMMARY STEPS

1. **crypto pki certificate chain** *name*
2. **crypto pki server** *cs-label* **info requests**
3. **show crypto pki certificates**
4. **show crypto pki server**
5. **show crypto pki trustpoints**

DETAILED STEPS

Step 1 The **crypto pki certificate chain** command can be used to view the certificate chain details and to distinguish the current active certificate from the rollover certificate in the certificate chain. The following example shows a certificate chain with an active CA certificate and a shadow, or rollover, certificate:

```
Router(config)# crypto pki certificate chain mica

certificate 06
certificate ca 01
! This is the peer's shadow PKI certificate.
certificate rollover 0B
! This is the CA shadow PKI certificate
certificate rollover ca 0A
```

Step 2 The **crypto pki server info requests** command displays all outstanding certificate enrollment requests The following example shows the output for shadow PKI certificate information requests:

```
Router# crypto pki server myca info requests

Enrollment Request Database:
RA certificate requests:

  ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------

RA rollover certificate requests:

  ReqID  State      Fingerprint                      SubjectName
  --------------------------------------------------------------

Router certificates requests:

  ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------
1      pending    A426AF07FE3A4BB69062E0E47198E5BF hostname=client

  Router rollover certificates requests:

  ReqID  State      Fingerprint                      SubjectName
  --------------------------------------------------------------
  2      pending    B69062E0E47198E5BFA426AF07FE3A4B hostname=client
```

Step 3 The **show crypto pki certificates** command displays information about your certificate, the certification authority certificate, shadow certificates, and any registration authority certificates. The following example displays the certificate of the router and the certificate of the CA. There is no shadow certificate available. A single, general-purpose RSA key pair was previously generated, and a certificate was requested but not received for that key pair. Note that the certificate status of the router shows "Pending." After the router receives its certificate from the CA, the Status field changes to "Available" in the **show** output.

```
Router# show crypto pki certificates

Certificate
  Subject Name
    Name: myrouter.example.com
    IP Address: 192.0.2.1
    Serial Number: 04806682
  Status: Pending
  Key Usage: General Purpose
```

```
    Fingerprint: 428125BD A3419600 3F6C7831 6CD8FA95 00000000
CA Certificate
  Status: Available
  Certificate Serial Number: 3051DF7123BEE31B8341DFE4B3A338E5F
  Key Usage: Not Set
```

Step 4 The **show crypto pki server** command displays the current state and configuration of the certificate server. The following example shows that the certificate server "routercs" has rollover configured. The CA auto-rollover time has occurred and the rollover, or shadow, PKI certificate is available. The status shows the rollover certificate fingerprint and rollover CA certificate expiration timer information.

```
Router# show crypto pki server

Certificate Server routercs:
   Status: enabled, configured
   Issuer name: CN=walnutcs
   CA cert fingerprint: 800F5944 74337E5B C2DF6C52 9A7B1BDB
   Granting mode is: auto
   Last certificate issued serial number: 0x7
   CA certificate expiration timer: 22:10:29 GMT Jan 29 2007
   CRL NextUpdate timer: 21:50:56 GMT Mar 5 2004
   Current storage dir: nvram:
   Database Level: Minimum - no cert data written to storage
Rollover status: available for rollover
    Rollover CA cert fingerprint: 6AAF5944 74227A5B 23DF3E52 9A7F1FEF
    Rollover CA certificate expiration timer: 22:10:29 GMT Jan 29 2017
```

Step 5 The **show crypto pki trustpoints** command displays the trustpoints that are configured in the router. The following output shows that a shadow CA certificate is available and shows the SCEP capabilities reported during the last enrollment operation:

```
Router# show crypto pki trustpoints

Trustpoint vpn:
   Subject Name:
   cn=Cisco SSL CA
   o=Cisco Systems
   Serial Number: 0FFEBBDC1B6F6D9D0EA7875875E4C695
   Certificate configured.
   Rollover certificate configured.
   Enrollment Protocol:
   SCEPv1, PKI Rollover
```

Configuration Examples for Using a Certificate Server

This section contains the following configuration examples:

Configuring Specific Storage and Publication Locations: Examples

The following example shows the configuration of a minimal local file system, so that the certificate server can respond quickly to certificate requests. The .ser and .crl files are stored on the local Cisco IOS file system for fast access, and a copy of all of the .crt files are published to a remote location for long-term logging.

```
crypto pki server myserver
     !Pick your database level.
     database level minimum
     !Specify a location for the .crt files that is different than the default local
     !Cisco IOS file system.
     database url crt publish http://url username user1 password secret
```

Note Free space on the local file system should be monitored, in case the .crl file becomes too large.

The following example shows the configuration of a primary storage location for critical files, a specific storage location for the critical file serial number file, the main certificate server database file, and a password protected file publication location for the CRL file:

```
Router(config)# crypto pki server mycs
Router(cs-server)# database url ftp://cs-db.company.com
!
% Server database url was changed. You need to move the
% existing database to the new location.
!
Router(cs-server)# database url ser nvram:
Router(cs-server)# database url crl publish ftp://crl.company.com username myname password
mypassword
Router(cs-server)# end
```

The following output displays the specified primary storage location and critical file storage locations specified:

```
Router# show

Sep  3 20:19:34.216: %SYS-5-CONFIG_I: Configured from console by user on console

Router# show crypto pki server

Certificate Server mycs:
     Status: disabled
     Server's configuration is unlocked  (enter "no shut" to lock it)
     Issuer name: CN=mycs
     CA cert fingerprint: -Not found-
     Granting mode is: manual
     Last certificate issued serial number: 0x0
     CA certificate expiration timer: 00:00:00 GMT Jan 1 1970
     CRL not present.
     Current primary storage dir: ftp://cs-db.company.com
     Current storage dir for .ser files: nvram:
     Database Level: Minimum - no cert data written to storage The following output
displays all storage and publication locations. The serial number file (.ser) is stored in
NVRAM. The CRL file will be published to ftp://crl.company.com with a username and
password. All other critical files will be stored to the primary location,
ftp://cs-db.company.com.

Router# show running-config

   section crypto pki server
   crypto pki server mycs shutdown database url ftp://cs-db.company.com
```

```
   database url crl publish ftp://crl.company.com username myname password 7
12141C0713181F13253920
   database url ser nvram:
Router#
```

Removing Enrollment Requests from the Enrollment Request Database: Examples

The following examples show both the enrollment requests that are currently in the enrollment request database and the result after one of the enrollment requests has been removed from the database.

Enrollment Request Currently in the Enrollment Request Database

The following example shows that the **crypto pki server info requests** command has been used to display the enrollment requests that are currently in the Enrollment Request Database:

```
Router# crypto pki server myserver info requests

Enrollment Request Database:

RA certificate requests:
ReqID    State      Fingerprint                              SubjectName
--------------------------------------------------------------------------

Router certificates requests:
ReqID    State      Fingerprint                              SubjectName
--------------------------------------------------------------------------
2        pending    1B07F3021DAAB0F19F35DA25D01D8567      hostname=host1.company.com
1        denied     5322459D2DC70B3F8EF3D03A795CF636      hostname=host2.company.com
```

crypto pki server remove Command Used to Remove One Enrollment Request

The following example shows that the **crypto pki server remove** command has been used to remove Enrollment Request 1:

```
Router# crypto pki server myserver remove 1
```

Enrollment Request Database After the Removal of One Enrollment Request

The following example shows the result of the removal of Enrollment Request 1 from the Enrollment Request Database:

```
Router# crypto pki server mycs info requests

Enrollment Request Database:

RA certificate requests:
ReqID    State     Fingerprint                         SubjectName
--------------------------------------------------------------------

Router certificates requests:
ReqID    State     Fingerprint                         SubjectName
--------------------------------------------------------------------
2        pending   1B07F3021DAAB0F19F35DA25D01D8567   hostname=host1.company.com
```

Autoarchiving the Certificate Server Root Keys: Examples

The following output configurations and examples show what you might see if the **database archive** command has not been configured (that is, configured using the default value); if the **database archive** command has been configured to set the CA certificate and CA key archive format as PEM, without configuring a password; and if the **database archive** command has been configured to set the CA certificate and CA key archive format as PKCS12, with a password configured. The last example is sample content of a PEM-formatted archive file.

database archive Command Not Configured

Note The default is PKCS12, and the prompt for the password appears after the **no shutdown** command has been issued.

```
Router (config)# crypto pki server myserver
Router (cs-server)# no shutdown

% Generating 1024 bit RSA keys ...[OK]
% Ready to generate the CA certificate.
%Some server settings cannot be changed after CA certificate generation.

Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
! Note the next two lines, which are asking for a password.
% Please enter a passphrase to protect the private key.
Password:
% Certificate Server enabled.
Router (cs-server)# end

Router# dir nvram:

Directory of nvram:/

  125  -rw-        1693               <no date>  startup-config
  126  ----           5               <no date>  private-config
    1  -rw-          32               <no date>  myserver.ser
    2  -rw-         214               <no date>  myserver.crl
! Note the next line, which indicates PKCS12 format.
    3  -rw-        1499               <no date>  myserver.p12
```

database archive Command and pem Keyword Configured

Note The prompt for the password appears after the **no shutdown** command has been issued.

```
Router (config)# crypto pki server myserver
Router (cs-server)# database archive pem
Router (cs-server)# no shutdown

% Generating 1024 bit RSA keys ...[OK]
% Ready to generate the CA certificate.
%Some server settings cannot be changed after CA certificate generation.

Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
!Note the next two lines, which are asking for a password.
% Please enter a passphrase to protect the private key.
Password:
```

```
% Certificate Server enabled.
Router (cs-server)# end

Router# dir nvram

Directory of nvram:/

  125  -rw-        1693            <no date>  startup-config
  126  ----           5            <no date>  private-config
    1  -rw-          32            <no date>  myserver.ser
    2  -rw-         214            <no date>  myserver.crl
! Note the next line showing that the format is PEM.
    3  -rw-        1705            <no date>  myserver.pem
```

database archive Command and pkcs12 Keyword (and Password) Configured

Note When the password is entered, it will be encrypted. However, it is recommended that you remove the password from the configuration after the archive has finished.

```
Router (config)# crypto pki server myserver
Router (cs-server)# database archive pkcs12 password cisco123
Router (cs-server)# no shutdown

% Generating 1024 bit RSA keys ...[OK]
% Ready to generate the CA certificate.
% Some server settings cannot be changed after CA certificate generation.

Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
! Note that you are not being prompted for a password.
% Certificate Server enabled.
Router (cs-server)# end
Router# dir nvram:
Directory of nvram:/

   125   -rw-         1693             <no date>   startup-config
   126   ----            5             <no date>   private-config
     1   -rw-           32             <no date>   myserver.ser
     2   -rw-          214             <no date>   myserver.crl
! Note that the next line indicates that the format is PKCS12.
     3   -rw-         1499             <no date>   myserver.p12
```

PEM-Formatted Archive

The following sample output shows that autoarchiving has been configured in PEM file format. The archive consists of the CA certificate and the CA private key. To restore the certificate server using the backup, you would have to import the PEM-formatted CA certificate and CA key individually.

Note In addition to the CA certificate and CA key archive files, you should also back up the serial file (.ser) and the CRL file (.crl) regularly. The serial file and the CRL file are both critical for CA operation if you need to restore your certificate server.

```
Router# more nvram:mycs.pem

-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTA0MDgyNzAyMzI0NloXDTA3MDgyNzAyMzI0NlowDzENMAsGA1UEAxMEbXlj
czCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA1lZpKP4nGDJHgPkpYSkix7lD
nr23aMlZ9Kz5oo/qTBxeZ8mujpjYcZ0T8AZvoOiCuDnYMl796ZwpkMgjz1aZZbL+
```

```
BtuVvllsEOfhC+u/Ol/vxfGG5xpshoz/F5J3xdg5ZZuWWuIDAUYu9+QbI5feuG04
Z/BiPIb4AmGTP4B2MM0CAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8B
Af8EBAMCAYYwHwYDVR0jBBgwFoAUKi/cuK6wkz+ZswVtb06vUJboEeEwHQYDVR0O
BBYEFCov3LiusJM/mbMFbW9Or1CW6BHhMA0GCSqGSIb3DQEBBAUAA4GBAKLOmoE2
4+NeOKEXMCXG1jcohK7O2HrkFfl/vpK0+q92PTnMUFhxLOqI8pWIq5CCgC7heace
OrTv2zcUAoH4rzx3Rc2USIxkDokWWQMLujsMm/SLIeHit0G5uj//GCcbgK20MAW6
ymf7+TmblSFljWzstoUXC2hLnsJIMq/KffaD
-----END CERTIFICATE-----

!The private key is protected by the password that is
configured in “database archive pem password pwd” or that
is entered when you are prompted for the password.
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,106CE91FFD0A075E

zyiFC8rKv8Cs+IKsQG2QpsVpvDBHqZqBSM4D528bvZv7jzr6WuHj8E6zO+6G8R/A
zjsfTALo+e+ZDg7KMzbryHARvjskbqFdOMLlVIYBhCeSElKsskWB6chOuyPHJInW
JwC5YzZdZwOqcyLBP/xOYXcvjzzNfPAXZzN12VR8vWDNq/kHT+3Lplc8hY++ABMI
M+C9FB3dpNZzu5O1BZCJg46bqbkulaCCmScIDaVt0zDFZwWTSufiemmNxZBG4xS8
t5t+FEhmSfv8DAmwg4f/KVRFTm10phUArcLxQO38Al0W5YHHORdACnuzVUvHgco7
VT4XUTjO7qMhmJgFNWy1pu49fbdS2NnOn5IoiyAq5lk1KUPrz/WABWiCvLMylGnZ
kyMCWoaMtgS/vdx74BBCj09yRZJnLMlIi6SDofjCNTDHfmFEVg4LsSWCd4lP9OP8
0MqhP1D5VIx6PbMNwkWW12lpBbCCdesFRGHjZD2dOu96kHD7ItErx34CC8W04aG4
b7DLktUu6WNV6M8g3CAqJiC0V8ATlp+kvdHZVkXovgND5IU0OJpsj0HhGzKAGpOY
KTGTUekUboISjVVkI6efp1vO6temVL3Txg3KGhzWMJGrq1snghE0KnV8tkddv/9N
d/t1l+we9mrccTq50WNDnkEi/cwHI/0PKXg+NDNH3k3QGpAprsqGQmMPdqc5ut0P
86i4cF9078QwWg4Tpay3uqNH1Zz6UN0tcarVVNmDupFESUxYw10qJrrEYVRadu74
rKAU4Ey4xkAftB2kuqvr21Av/L+jne4kkGIoZYdB+p/M98pQRgkYyg==
-----END RSA PRIVATE KEY-----
```

Restoring a Certificate Server from Certificate Server Backup Files: Examples

The following example shows that restoration is from a PKCS12 archive and that the database URL is NVRAM (the default).

```
Router# copy tftp://192.0.2.71/backup.ser nvram:mycs.ser
Destination filename [mycs.ser]?

32 bytes copied in 1.320 secs (24 bytes/sec)
Router# copy tftp://192.0.2.71/backup.crl nvram:mycs.crl
Destination filename [mycs.crl]?

214 bytes copied in 1.324 secs (162 bytes/sec)
Router# configure terminal
Router (config)# crypto pki import mycs pkcs12 tftp://192.0.2.71/backup.p12 cisco123
Source filename [backup.p12]?
CRYPTO_PKI: Imported PKCS12 file successfully.
Router (config)# crypto pki server mycs
! fill in any certificate server configuration here
Router (cs-server)# no shutdown
% Certificate Server enabled.
Router (cs-server)# end
Router# show crypto pki server

Certificate Server mycs:
    Status: enabled
    Server's current state: enabled
    Issuer name: CN=mycs
    CA cert fingerprint: 34885330 B13EAD45 196DA461 B43E813F
    Granting mode is: manual
    Last certificate issued serial number: 0x1
```

```
    CA certificate expiration timer: 01:49:13 GMT Aug 28 2007
    CRL NextUpdate timer: 01:49:16 GMT Sep 4 2004
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

The following example shows that restoration is from a PEM archive and that the database URL is flash:

```
Router# copy tftp://192.0.2.71/backup.ser flash:mycs.ser
Destination filename [mycs.ser]?
32 bytes copied in 1.320 secs (24 bytes/sec)
Router# copy tftp://192.0.2.71/backup.crl flash:mycs.crl
Destination filename [mycs.crl]?
214 bytes copied in 1.324 secs (162 bytes/sec)
Router# configure terminal
! Because CA cert has Digital Signature usage, you need to import using the "usage-keys"
keyword
Router (config)# crypto ca import mycs pem usage-keys terminal cisco123
% Enter PEM-formatted CA certificate.
% End with a blank line or "quit" on a line by itself.
! Paste the CA cert from .pem archive.
-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTA0MDkwMjIxMDI1NloXDTA3MDkwMjIxMDI1NlowDzENMAsGA1UEAxMEbXlj
czCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAuGnnDXJbpDDQwCuKGs5Zg2rc
K7ZJauSUotTmWYQvNx+ZmWrUs5/j9Ee5FV2YonirGBQ9mc6u163kNlrIPFck062L
GpahBhNmKDgod1o2PHTnRlZpEZNDIqU2D3hACgByxPjrY4vUnccV36ewLnQnYpp8
szEu7PYTJr5dU5ltAekCAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8B
Af8EBAMCAYYwHwYDVR0jBBgwFoAUaEEQwYKCQ1dm9+wLYBKRTlzxaDIwHQYDVR0O
BBYEFGhBEMGCgkNXZvfsC2ASkU5c8WgyMA0GCSqGSIb3DQEBBAUAA4GBAHyhiv2C
mH+vswkBjRA1Fzzk8ttu9s5kwqG0dXp25QRUWsGlr9nsKPNdVKt3P7p0A/KochHe
eNiygiv+hDQ3FVnzsNv983le6O5jvAPxc17RO1BbfNhqvEWMsXdnjHOcUy7XerCo
+bdPcUf/eCiZueH/BEy/SZhD7yovzn2cdzBN
-----END CERTIFICATE-----

% Enter PEM-formatted encrypted private SIGNATURE key.
% End with "quit" on a line by itself.
! Paste the CA private key from .pem archive.
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,5053DC842B04612A

1CnlF5Pqvd0zp2NLZ7iosxzTy6nDeXPpNyJpxB5q+V29IuY8Apb6TlJCU7YrsEB/
nBTK7K76DCeGPlLpcuyEI171QmkQJ2gA0QhC0LrRo09WrINVH+b4So/y7nffZkVb
p2yDpZwqoJ8cmRH94Tie0YmzBtEh6ayOud11z53qbrsCnfSEwszt1xrW1MKrFZrk
/fTy6loHzGFzl3BDj4r5gBecExwcPp74ldHO+Ld4Nc9egG8BYkeBCsZZOQNVhXLN
I0tODOs6hP915zb6OrZFYv0NK6grTBO9D8hjNZ3U79jJzsSP7UNzIYHNTzRJiAyu
i56Oy/iHvkCSNUIK6zeIJQnW4bSoM1BqrbVPwHU6QaXUqlNzZ8SDtw7ZRZ/rHuiD
RTJMPbKquAzeuBss1132OaAUJRStjPXgyZTUbc+cWb6zATNws2yijPDTR6sRHoQL
47wHMr2Yj80VZGgkCSLAkL88ACz9TfUiVFhtfl6xMC2yuFl+WRk1XfF5VtWe5Zer
3Fn1DcBmlF7O86XUkiSHP4EV0cI6n5ZMzVLx0XAUtdAl1gD94y1V+6p9PcQHLyQA
pGRmj5IlSFw90aLafgCTbRbmC0ChIqHy91UFa1ub0130+yu7LsLGRlPmJ9NE61JR
bjRhlUXItRYWY7C4M3m/0wz6fmVQNSumJM08RHq6lUB3olzIgGIZlZkoaESrLG0p
qq2AENFemCPF0uhyVS2humMHjWuRr+jedfc/IMl7sLEgAdqCVCfV3RZVEaNXBud1
4QjkuTrwaTcRXVFbtrVioT/puyVUlpA7+k7w+F5TZwUV08mwvUEqDw==
-----END RSA PRIVATE KEY-----
quit
% Enter PEM-formatted SIGNATURE certificate.
% End with a blank line or "quit" on a line by itself.
! Paste the CA cert from .pem archive again.
-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTA0MDkwMjIxMDI1NloXDTA3MDkwMjIxMDI1NlowDzENMAsGA1UEAxMEbXlj
czCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAuGnnDXJbpDDQwCuKGs5Zg2rc
K7ZJauSUotTmWYQvNx+ZmWrUs5/j9Ee5FV2YonirGBQ9mc6u163kNlrIPFck062L
```

```
GpahBhNmKDgod1o2PHTnRlZpEZNDIqU2D3hACgByxPjrY4vUnccV36ewLnQnYpp8
szEu7PYTJr5dU5ltAekCAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8B
Af8EBAMCAYYwHwYDVR0jBBgwFoAUaEEQwYKCQ1dm9+wLYBKRTlzxaDIwHQYDVR0O
BBYEFGhBEMGCgkNXZvfsC2ASkU5c8WgyMA0GCSqGSIb3DQEBBAUAA4GBAHyhiv2C
mH+vswkBjRA1Fzzk8ttu9s5kwqG0dXp25QRUWsGlr9nsKPNdVKt3P7p0A/KochHe
eNiygiv+hDQ3FVnzsNv983le6O5jvAPxc17RO1BbfNhqvEWMsXdnjHOcUy7XerCo
+bdPcUf/eCiZueH/BEy/SZhD7yovzn2cdzBN
-----END CERTIFICATE-----

% Enter PEM-formatted encrypted private ENCRYPTION key.
% End with "quit" on a line by itself.
! Because the CA cert only has Digital Signature usage, skip the encryption part.
quit
% PEM files import succeeded.
Router (config)# crypto pki server mycs
Router (cs-server)# database url flash:
! Fill in any certificate server configuration here.
Router (cs-server)# no shutdown
% Certificate Server enabled.
Router (cs-server)# end

Router # show crypto pki server

Certificate Server mycs:
    Status: enabled
    Server's current state: enabled
    Issuer name: CN=mycs
    CA cert fingerprint: F04C2B75 E0243FBC 19806219 B1D77412
    Granting mode is: manual
    Last certificate issued serial number: 0x2
    CA certificate expiration timer: 21:02:55 GMT Sep 2 2007
    CRL NextUpdate timer: 21:02:58 GMT Sep 9 2004
    Current storage dir: flash:
```

Database Level: Minimum - no cert data written to storage

Subordinate Certificate Server: Example

The following configuration and output is typical of what you might see after configuring a subordinate certificate server:

```
Router (config)# crypto pki trustpoint sub
Router (ca-trustpoint)# enrollment url http://192.0.2.6
Router (ca-trustpoint)# exit

Router (config)# crypto pki server sub
Router (cs-server)# mode sub-cs
Router (ca-server)# no shutdown

%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key
% or type Return to exit
Password:
Jan  6 22:32:22.698: CRYPTO_CS: enter FSM: input state initial, input signal no shut

Re-enter password:
% Generating 1024 bit RSA keys ...
Jan  6 22:32:30.302: CRYPTO_CS: starting enabling checks
Jan  6 22:32:30.306: CRYPTO_CS: key 'sub' does not exist; generated automatically [OK]

Jan  6 22:32:39.810: %SSH-5-ENABLED: SSH 1.99 has been enabled
Certificate has the following attributes:
```

```
     Fingerprint MD5: 328ACC02 52B25DB8 22F8F104 B6055B5B
     Fingerprint SHA1: 02FD799D DD40C7A8 61DC53AB 1E89A3EA 2A729EE2

% Do you accept this certificate? [yes/no]:
Jan  6 22:32:44.830: CRYPTO_CS: nvram filesystem
Jan  6 22:32:44.922: CRYPTO_CS: serial number 0x1 written.
Jan  6 22:32:46.798: CRYPTO_CS: created a new serial file.
Jan  6 22:32:46.798: CRYPTO_CS: authenticating the CA 'sub'y
Trustpoint CA certificate accepted.%

% Certificate request sent to Certificate Authority

% Enrollment in progress...
Router (cs-server)#
Jan  6 22:33:30.562: CRYPTO_CS: Publishing 213 bytes to crl file nvram:sub.crl
Jan  6 22:33:32.450: CRYPTO_CS: enrolling the server's trustpoint 'sub'
Jan  6 22:33:32.454: CRYPTO_CS: exit FSM: new state check failed
Jan  6 22:33:32.454: CRYPTO_CS: cs config has been locked
Jan  6 22:33:33.118: CRYPTO_PKI:  Certificate Request Fingerprint MD5: CED89E5F 53B9C60E
> AA123413 CDDAD964
Jan  6 22:33:33.118: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 70787C76 ACD7E67F
7D2C8B23 98CB10E7 718E84B1
% Exporting Certificate Server signing certificate and keys...

Jan  6 22:34:53.839: %PKI-6-CERTRET: Certificate received from Certificate Authority
Jan  6 22:34:53.843: CRYPTO_CS: enter FSM: input state check failed, input signal cert
configured
Jan  6 22:34:53.843: CRYPTO_CS: starting enabling checks
Jan  6 22:34:53.843: CRYPTO_CS: nvram filesystem
Jan  6 22:34:53.883: CRYPTO_CS: found existing serial file.
Jan  6 22:34:53.907: CRYPTO_CS: old router cert flag 0x4
Jan  6 22:34:53.907: CRYPTO_CS: new router cert flag 0x44
Jan  6 22:34:56.511: CRYPTO_CS: DB version
Jan  6 22:34:56.511: CRYPTO_CS: last issued serial number is 0x1
Jan  6 22:34:56.551: CRYPTO_CS: CRL file sub.crl exists.
Jan  6 22:34:56.551: CRYPTO_CS: Read 213 bytes from crl file sub.crl.
Jan  6 22:34:56.603: CRYPTO_CS: SCEP server started
Jan  6 22:34:56.603: CRYPTO_CS: exit FSM: new state enabled
Jan  6 22:34:56.603: CRYPTO_CS: cs config has been locked
Jan  6 22:35:02.359: CRYPTO_CS: enter FSM: input state enabled, input signal time set
Jan  6 22:35:02.359: CRYPTO_CS: exit FSM: new state enabled
Jan  6 22:35:02.359: CRYPTO_CS: cs config has been locked
```

Root Certificate Server Differentiation: Example

When issuing certificates, the root certificate server (or parent subordinate certificate server) will differentiate the certificate request from “Sub CA,” “RA,” and peer requests, as shown in the following sample output:

```
Router# crypto pki server server1 info req

Enrollment Request Database:
RA certificate requests:
ReqID     State        Fingerprint                              SubjectName
--------------------------------------------------------------------------------
Subordinate CS certificate requests:
ReqID     State        Fingerprint                              SubjectName
--------------------------------------------------------------------------------
1     pending        CB9977AD8A73B146D3221749999B0F66 hostname=host-subcs.company.com
```

```
RA certificate requests:
ReqID     State        Fingerprint                                SubjectName
--------------------------------------------------------------------------------
Router certificate requests:
ReqID     State        Fingerprint                                SubjectName
--------------------------------------------------------------------------------
```

Show Output for a Subordinate Certificate Server: Example

The following **show crypto pki server command** output indicates that a subordinate certificate server has been configured:

```
Router# show crypto pki server

Certificate Server sub:
  Status: enabled
  Server's configuration is locked  (enter "shut" to unlock it)
  Issuer name: CN=sub
  CA cert fingerprint: 11B586EE 3B354F33 14A25DDD 7BD39187
  Server configured in subordinate server mode
  Upper CA cert fingerprint: 328ACC02 52B25DB8 22F8F104 B6055B5B
  Granting mode is: manual
  Last certificate issued serial number: 0x1
  CA certificate expiration timer: 22:33:44 GMT Jan 6 2006
  CRL NextUpdate timer: 22:33:29 GMT Jan 13 2005
  Current storage dir: nvram:
  Database Level: Minimum - no cert data written to storage
```

RA Mode Certificate Server: Example

The following output is typical of what you might see after having configured an RA mode certificate server:

```
Router-ra (config)# crypto pki trustpoint myra
Router-ra (ca-trustpoint)# enrollment url http://192.0.2.17
! Include "cn=ioscs RA" or "ou=ioscs RA" in the subject-name.
Router-ra (ca-trustpoint)# subject-name cn=myra, ou=ioscs RA, o=company, c=us
Router-ra (ca-trustpoint)# exit
Router-ra (config)# crypto pki server myra
Router-ra (cs-server)# mode ra
Router-ra (cs-server)# no shutdown
% Generating 1024 bit RSA keys ...[OK]

Certificate has the following attributes:
Fingerprint MD5: 32661452 0DDA3CE5 8723B469 09AB9E85
Fingerprint SHA1: 9785BBCD 6C67D27C C950E8D0 718C7A14 C0FE9C38
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
% Ready to request the CA certificate.
%Some server settings cannot be changed after the CA certificate has been requested.

Are you sure you want to do this? [yes/no]: yes
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.
```

```
Password:
Re-enter password:

% The subject name in the certificate will include: cn=myra, ou=ioscs RA, o=company, c=us
% The subject name in the certificate will include: Router-ra.company.com
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto pki certificate' command will also show the fingerprint.

% Enrollment in progress...
Router-ra (cs-server)#
Sep 15 22:32:40.197: CRYPTO_PKI:  Certificate Request Fingerprint MD5: 82B41A76 AF4EC87D
AAF093CD 07747D3A
Sep 15 22:32:40.201: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 897CDF40 C6563EAA
0FED05F7 0115FD3A 4FFC5231
Sep 15 22:34:00.366: %PKI-6-CERTRET: Certificate received from Certificate Authority
Router-ra (cs-server)#
Router-ra(cs-server)# end

Router-ra# show crypto pki server

Certificate Server myra:
    Status: enabled
    Issuer name: CN=myra
    CA cert fingerprint: 32661452 0DDA3CE5 8723B469 09AB9E85
    ! Note that the certificate server is running in RA mode
    Server configured in RA mode
    RA cert fingerprint: C65F5724 0E63B3CC BE7AE016 BE0D34FE
    Granting mode is: manual
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

The following output shows the enrollment request database of the issuing certificate server after the RA has been enabled:

Note The RA certificate request is recognized by the issuing certificate server because "ou=ioscs RA" is listed in the subject name.

```
Router-ca# crypto pki server mycs info request
Enrollment Request Database:

Subordinate CA certificate requests:
ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------

! The request is identified as RA certificate request.
RA certificate requests:
ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------
12     pending    88F547A407FA0C90F97CDE8900A30CB0
hostname=Router-ra.company.com,cn=myra,ou=ioscs RA,o=company,c=us
```

```
Router certificates requests:
ReqID   State     Fingerprint                       SubjectName
--------------------------------------------------------------

! Issue the RA certificate.
Router-ca# crypto pki server mycs grant 12
```

The following output shows that the issuing certificate server is configured to issue a certificate automatically if the request comes from an RA:

```
Router-ca(config)# crypto pki server mycs
Router-ca (cs-server)# grant ra-auto
% This will cause all certificate requests already authorized by known RAs to be
automatically granted.

Are you sure you want to do this? [yes/no]: yes
Router-ca (cs-server)# end
Router-ca# show crypto pki server

Certificate Server mycs:
    Status: enabled
    Server's current state: enabled
    Issuer name: CN=mycs
    CA cert fingerprint: 32661452 0DDA3CE5 8723B469 09AB9E85
    ! Note that the certificate server will issue certificate for requests from the RA.
    Granting mode is: auto for RA-authorized requests, manual otherwise
    Last certificate issued serial number: 0x2
    CA certificate expiration timer: 22:29:37 GMT Sep 15 2007
    CRL NextUpdate timer: 22:29:39 GMT Sep 22 2004
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

The following example shows the configuration of “myra”, an RA server, configured to support automatic rollover from “myca”, the CA. After the RA server is configured, automatic granting of certificate reenrollment requests is enabled:

```
crypto pki trustpoint myra
 enrollment url http://myca
 subject-name ou=iosca RA
 rsakeypair myra
crypto pki server myra
 mode ra
 auto-rollover

crypto pki server mycs
 grant auto rollover ra-cert
 auto-rollover 25
```

Enabling CA Certificate Rollover to Start Immediately: Example

The following example shows how to enable automated CA certificate rollover on the server mycs with the **crypto pki server** command. The **show crypto pki server** command then shows the current state of the mycs server and that the rollover certificate is currently available for rollover.

```
Router(config)# crypto pki server mycs rollover

Jun 20 23:51:21.211:%PKI-4-NOSHADOWAUTOSAVE:Configuration was
modified.  Issue "write memory" to save new IOS CA certificate

! The config has not been automatically saved because the config has been changed.
```

```
Router# show crypto pki server

Certificate Server mycs:
    Status:enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name:CN=mycs
    CA cert fingerprint:E7A5FABA 5D7AA26C F2A9F7B3 03CE229A
    Granting mode is:manual
    Last certificate issued serial number:0x2
    CA certificate expiration timer:00:49:26 PDT Jun 20 2008
    CRL NextUpdate timer:00:49:29 PDT Jun 28 2005
    Current storage dir:nvram:
    Database Level:Minimum - no cert data written to storage
    Rollover status:available for rollover
    ! Rollover certificate is available for rollover.
    Rollover CA certificate fingerprint:9BD7A443 00A6DD74 E4D9ED5F B7931BE0
    Rollover CA certificate expiration time:00:49:26 PDT Jun 20 2011
    Auto-Rollover configured, overlap period 25 days
```

Where to Go Next

After the certificate server is successfully running, you can either begin enrolling clients via manual mechanisms (as explained in the module "Configuring Certificate Enrollment for a PKI") or begin configuring SDP, which is a web-based enrollment interface, (as explained in the module "Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI.")

Additional References

The following sections provide references related to Cisco IOS certificate server.Related Documents

Related Topic	Document Title
USB Token RSA Operations: Using the RSA keys on a USB token for initial autoenrollment	"Configuring Certificate Enrollment for a PKI" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.4T. See the "Configuring Certificate Servers" section.
USB Token RSA Operations: Benefits of using USB tokens	"Storing PKI Credentials" module in the *Cisco IOS Security Configuration Guide*, Release 12.4T
Certificate server client certificate enrollment, autoenrollment, and automatic rollover	"Configuring Certificate Enrollment for a PKI" module in the *Cisco IOS Security Configuration Guide*, Release 12.4T
Setting up and logging into a USB token	"Storing PKI Credentials" module in the *Cisco IOS Security Configuration Guide*, Release 12.4T
Web-based certificate enrollment	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI" module in the *Cisco IOS Security Configuration Guide*, Release 12.4T
RSA keys in PEM formatted files	"Deploying RSA Keys Within a PKI" module in the *Cisco IOS Security Configuration Guide*, Release 12.4T
Choosing a certificate revocation mechanism	"Configuring Authorization and Revocation of Certificates in a PKI" module in the *Cisco IOS Security Configuration Guide*, Release 12.4T

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for the Cisco IOS Certificate Server

Table 4 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Implementing and Managing PKI Features Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 4 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 4 Feature Information for the Cisco IOS Certificate Server

Feature Name	Releases	Feature Information
Cisco IOS USB Token PKI Enhancements—Phase 2	12.4(11)T	This feature enhances USB token functionality by using the USB token as a cryptographic device. USB tokens may be used for RSA operations such as key generation, signing, and authentication. The following sections in this document provide information about this feature: • RSA Key Pair and Certificate of the Certificate Server • Trustpoint of the Certificate Server • Generating a Certificate Server RSA Key Pair **Note** This document covers the use of using USB tokens for RSA operations during certificate server configuration. For other documents on this topic, see the "The following sections provide references related to Cisco IOS certificate server.Related Documents" section.
IOS Certificate Server (CS) Split Database	12.4(4)T	This feature allows the user to set storage locations and publish locations for specific certificate server file types. The following sections provide information about this feature: • Certificate Server Database • Configuring Certificate Server Functionality • Configuring Specific Storage and Publication Locations: Examples The following command was modified by this feature: **database url**

Table 4 Feature Information for the Cisco IOS Certificate Server (continued)

Feature Name	Releases	Feature Information
Subordinate/RA Mode IOS Certificate Server (CS) Rollover	12.4(4)T	This feature expands on Certificate Authority (CA) Key Rollover introduced in 12.4(2)T to allow CA certificate rollover for subordinate CAs and RA-mode CAs. This functionality allows the rollover expiring CA certificates and keys and to have these changes propagate through the PKI network without manual intervention. The following sections provide information about this feature: • Automatic CA Certificate and Key Rollover • Configuring Certificate Servers • RA Mode Certificate Server: Example The following command was modified by this feature: **grant auto rollover**
Certificate Authority (CA) Key Rollover	12.4(2)T	This feature introduces the ability for root or subordinate CAs to roll over expiring CA certificates and keys and to have these changes propagate through the PKI network without manual intervention. The following sections provide information about this feature: • Automatic CA Certificate and Key Rollover • Configuring Certificate Servers • Working with Automatic CA Certificate Rollover • Enabling CA Certificate Rollover to Start Immediately: Example The following commands were introduced or modified by this feature: **auto-rollover**, **crypto pki certificate chain, crypto pki export pem, crypto pki server info request, crypto pki server**, **show crypto pki certificates, show crypto pki server**, and **show crypto pki trustpoint**
Cisco IOS Certificate Server	12.3(8)T	This feature introduces support for the Cisco IOS certificate server, which offers users a CA that is directly integrated with Cisco IOS software to more easily deploy basic PKI networks. The following sections provide information about this feature: • Information About Cisco IOS Certificate Servers • How to Set Up and Deploy a Cisco IOS Certificate Server

Table 4 Feature Information for the Cisco IOS Certificate Server (continued)

Feature Name	Releases	Feature Information
The Certificate Server Auto Archive Enhancement[1]	12.3(11)T	This enhancement enables the CA certificate and CA key to be backed up automatically just once after they are generated by the certificate server. As a result, it is not necessary to generate an exportable CA key if CA backup is desirable. The following sections provide information about this feature: • Certificate Enrollment Using a Certificate Server • Configuring Certificate Server Functionality The following commands were introduced by this feature: **crypto pki server remote**, **database archive**
The Certificate Server Registration Authority (RA) Mode enhancement	12.3(7)T	A certificate server can be configured to run in RA mode. The following section provides information about this feature: • Configuring a Certificate Server to Run in RA Mode The following commands were introduced by this feature: **grant ra-auto**, **lifetime enrollment-requests**
PKI Status[1]	12.3(11)T	This enhancement provides a quick snapshot of current trustpoint status. The following section provides information about this enhancement: • Maintaining, Verifying, and Troubleshooting the Certificate Server, Certificates, and the CA The following command was modified by this enhancement: **show crypto pki trustpoints**
Subordinate Certificate Server[1]	12.3(14)T	This enhancement allows you to configure a subordinate certificate server to grant all or certain SCEP or manual certificate requests. The following section provides information about this enhancement: • Configuring a Subordinate Certificate Server The following command was introduced by this enhancement: **mode sub-cs**
Cisco IOS Certificate Server	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Trustpoint CLI	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

1. This is a minor enhancement. Minor enhancements are not typically listed in Feature Navigator.

Storing PKI Credentials

First Published: May 2, 2005
Last Updated: August 21, 2007

This module explains how to store public key infrastructure (PKI) credentials, such as Rivest, Shamir, and Adelman (RSA) keys and certificates in a specific location.

An example of a certificate storage location includes NVRAM, which is the default location, and other local storage locations, such as flash, as supported by your platform.

An example of an RSA key and certificate storage location includes a USB token. Selected Cisco platforms support smart card technology in a USB key form factor (such as an Aladdin USB eToken key). USB tokens provide secure configuration distribution, provide RSA operations such as on-token key generation, signing, and authentication, and allow users to store Virtual Private Network (VPN) credentials for deployment.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Storing PKI Credentials" section on page 25.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Storing PKI Credentials

Prerequisites for Specifying a Local Certificate Storage Location

Before you can specify the local certificate storage location, your system should meet the following requirements:

- A Cisco IOS Release 12.4(2)T PKI-enabled image or a later image
- A platform that supports storing PKI credentials as separate files
- A configuration that contains at least one certificate
- An accessible local file system

Prerequisites for Specifying USB Token Storage for PKI Credentials

Before you can use a USB token, your system should meet the following requirements:

- A Cisco 871 router, Cisco 1800 series, Cisco 2800 series, a Cisco 3800 series router, or a Cisco 7200VXR NPE-G2 platform
- At least a Cisco IOS Release 12.3(14)T image running on any of the supported platforms
- A Cisco supported USB token
- A k9 image

Restrictions for Storing PKI Credentials

Restrictions for Specifying a Local Certificate Storage Location

When storing certificates to a local storage location, the following restrictions are applicable:

- Only local file systems may be used. An error message will be displayed if a remote file system is selected, and the command will not take effect.
- A subdirectory may be specified if supported by the local file system. NVRAM does not support subdirectories.

Restrictions for Specifying USB Token Storage

When using a USB token to store PKI data, the following restrictions are applicable:

- USB token support requires a 3DES (k9) Cisco IOS software image, which provides secure file storage.
- You cannot boot an image from a USB token. (However, you can boot a configuration from a USB token.)
- USB hubs are currently not supported. Thus, the number of supported devices is limited to the number of available USB ports.

Information About Storing PKI Credentials

To determine where to store PKI credentials, you should understand the following concepts:

Storing Certificates to a Local Storage Location

Certificates are stored to NVRAM by default, however some routers do not have the required amount of NVRAM to successfully store certificates. Introduced in Cisco IOS Release 12.4(2)T is the ability to specify where certificates are stored on a local file system.

All Cisco platforms support NVRAM and flash local storage. Depending on your platform, you may have other supported local storage options including bootflash, slot, disk, USB flash, or USB token.

During run time, you can specify what active local storage device you would like to use to store certificates.

PKI Credentials and USB Tokens

To use a secure USB token on your router, you should understand the following concepts:

How a USB Token Works

A smart card is a small plastic card, containing a microprocessor and memory that allows you to store and process data. A USB token is a smart card with a USB interface. The token can securely store any type of file within its available storage space (32 KB). Configuration files that are stored on the USB token can be encrypted and accessed only via a user PIN. The router will not load the configuration file unless the proper PIN has been configured for secure deployment of router configuration files.

After you plug the USB token into the router, you must log into the USB token; thereafter, you can change default settings, such as the user PIN (default: 1234567890) and the allowed number of failed login attempts (default: 15 attempts) before future logins are refused. For more information on accessing and configuring the USB token, see the section "Logging Into and Setting Up the USB Token."

After you have successfully logged into the USB token, you can copy files from the router on to the USB token via the **copy** command. USB token RSA keys and associated IPsec tunnels remain available until the router is reloaded. To specify the length of time before the keys are removed and the IPsec tunnels are torn down, issue the **crypto pki token removal timeout** command.

Table 1 highlights the capabilities of the USB token.

Table 1 Functionality Highlights for USB Tokens

Function	USB Token
Accessibility	Used to securely store and transfer digital certificates, preshared keys, and router configurations from the USB token to the router.
Storage Size	32 KB

Table 1 Functionality Highlights for USB Tokens (continued)

Function	USB Token
File Types	• Typically used to store digital certificates, preshared keys, and router configurations for IPsec VPNs. • USB tokens cannot store Cisco IOS images.
Security	• Files can be encrypted and accessed only with a user PIN. • Files can also be stored in a nonsecure format.
Boot Configurations	• The router can use the configuration stored in the USB token during boot time. • The router can use the secondary configuration stored in the USB token during boot time. (A secondary configuration allows users to load their IPsec configuration.)

Benefits of USB Tokens

USB token support on a Cisco router provides the following application benefits:

Removable Credentials: Provide or Store VPN Credentials on an External Device for Deployment

A USB token can use smart card technology to store a digital certificate and configuration for IPsec VPN deployment. This ability enhances the capability of the router to generate RSA public keys to authenticate at least one IPsec tunnel. (Because a router can initiate multiple IPsec tunnels, the USB token can contain several certificates, as appropriate.)

Storing VPN credentials on an external device reduces the threat of compromising secure data.

PIN Configuration for Secure File Deployment

A USB token can store a configuration file that can be used for enabling encryption on the router via a user-configured PIN. (That is, no digital certificates, preshared keys, or VPNs are used.)

Touchless or Low Touch Configuration

The USB token can provide remote software configuration and provisioning with little or no human interaction. Configuration is set up as an automated process. That is, the USB token can store a bootstrap configuration that the router can use to boot from after the USB token has been inserted into the router. The bootstrap configuration connects the router to a TFTP server, which contains a configuration that completely configures the router.

RSA Operations

As of Cisco IOS Release 12.4(11)T and later releases, a USB token may be used as a cryptographic device in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication to be performed on the token.

General-purpose, special-usage, encryption, or signature RSA key pairs with a modulus of 2048 bits or less may be generated from credentials located on your token storage device. Private keys are not distributed and remain on the token by default, however you may configure the private key storage location.

Keys that reside on a USB token are saved to persistent token storage when they are generated. Key deletion will remove the keys stored on the token from persistent storage immediately. (Keys that do not reside on a token are saved to or deleted from nontoken storage locations when the **write memory** or a similar command is issued.)

Remote Device Configuration and Provisioning in a Secure Device Provisioning (SDP) Environment

As of Cisco IOS Release 12.4(15)T and later releases, SDP may be used to configure a USB token. The configured USB token may be transported to provision a device at a remote location. That is, a USB token may be used to transfer cryptographic information from one network device to another remote network device providing a solution for a staged USB token deployment.

For information about using USB tokens with SDP, see document titles in the "Related Documents" section.

How to Configure PKI Storage

This section contains the following configuration tasks:

Specifying a Local Storage Location for Certificates

The following procedure allows you to specify the local storage location for certificates.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. crypto pki certificate storage *location-name*
4. exit
5. **copy** *source-url destination-url*
6. show crypto pki certificates storage

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki certificate storage** *location-name* **Example:** `Router(config)# crypto pki certificate storage flash:/certs`	Specifies the local storage location for certificates.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	`copy source-url destination-url` **Example:** `Router# copy system:running-config nvram:startup-config`	(Optional) Saves the running configuration to the startup configuration. **Note** Settings will only take effect when the running configuration is saved to the startup configuration.
Step 6	**show crypto pki certificates storage** **Example:** `Router# show crypto pki certificates storage`	(Optional) Displays the current setting for the PKI certificate storage location.

Examples

The following is sample output for the **show crypto pki certificates storage** command where the certificates are stored in the certs subdirectory of disk0:

```
Router# show crypto pki certificates storage

Certificates will be stored in disk0:/certs/
```

Setting Up and Using USB Tokens on Cisco Routers

This section contains the following procedures that allow you to configure a router to support USB tokens:

- Storing the Configuration on a USB Token, page 7
- Logging Into and Setting Up the USB Token, page 7
- Configuring the USB Token, page 10
- Setting Administrative Functions on the USB Token, page 13

Storing the Configuration on a USB Token

Perform this task to store the configuration file in a USB token.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. boot config usbtoken[0-9]:filename

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**boot config usbtoken**[**0**-**9**]:*filename* **Example:** `Router(config)# boot config usbtoken0:file`	Specifies that the startup configuration file is stored in a secure USB token.

Logging Into and Setting Up the USB Token

Perform this task to log into and to perform the initial set up of a USB token.

Use of RSA Keys with a USB Token

- RSA keys are loaded after the USB token is successfully logged into the router.
- By default, newly generated RSA keys are stored on the most recently inserted USB token. Regenerated keys should be stored in the same location where the original RSA key was generated.

Automatic Login

Automatic login allows the router to completely come back up without any user or operator intervention. The PIN is stored in the private NVRAM, so it is not visible in the startup or running configuration.

Note A hand-generated startup configuration can contain the automatic login command for deployment purposes, but the **copy system:running-config nvram: startup-config** command must be issued to put the hand-generated configuration in the private configuration.

Manual Login

Unlike automatic login, manual login requires that the user know the actual USB token PIN.

Manual login can be used when storing a PIN on the router is not desirable. Manual login may also be suitable for some initial deployment or hardware replacement scenarios for which the router is obtained from the local supplier or drop-shipped to the remote site. Manual login can be executed with or without privileges, and it will make files and RSA keys on the USB token available to the Cisco IOS software. If a secondary configuration file is configured, it will be executed only with the privileges of the user who is performing the login. Thus, if you want to use manual login and set up the secondary configuration on the USB token to perform anything useful, you need to enable privileges.

Manual login can also be used in recovery scenarios for which the router configuration has been lost. If the scenario contains a remote site that normally connects to the core network with a VPN, the loss of the configuration and RSA keys requires out-of-band services that the USB token can provide. The USB token can contain a boot configuration, a secondary configuration, or both, and RSA keys to authenticate the connection.

SUMMARY STEPS

1. **enable**
2. **crypto pki token** *token-name* [**admin**] **login** [*pin*]

   or

   configure terminal
3. **crypto pki token** *token-name* **user-pin** [*pin*]
4. exit
5. **show usbtoken**[**0-9**]**:***filename*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki token** *token-name* [**admin**] **login** [*pin*] **Example:** `Router# crypto pki token usbtoken0 admin login 5678` or **configure terminal** **Example:** `Router# configure terminal`	Manually logs into the USB token. You must specify the **admin** keyword if later you want to change the user PIN. or Puts the router in global configuration mode, which allows you to configure automatic USB token login.
Step 3	**crypto pki token** *token-name* **user-pin** [*pin*] **Example:** `Router(config)# crypto pki token usbtoken0 user-pin 1234`	(Optional) Configures the router to log into the token automatically, using the specified PIN at router startup or when the USB token is inserted into a USB slot. The PIN is encrypted and stored in NVRAM. **Note** You will be asked to enter your passphrase.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**show usbtoken**[**0-9**]:*filename* **Example:** `Router# show usbtoken0:usbfile`	(Optional) Verifies whether the USB token has been logged onto the router.

What to Do Next

After you have logged into the USB token, it is available for use.

- To further configure the USB token, see the "Configuring the USB Token" section.
- To perform USB token administrative tasks, such as changing the user PIN, copying files from the router to the USB token set key storage location, and changing USB tokens, see the "Setting Administrative Functions on the USB Token" section.
- To utilize the USB token as a cryptographic device to perform RSA operations, see the document titles in the "Related Documents" section.
- To specify that the USB token be used for RSA operations during initial autoenrollment, see the document titles in the "Related Documents" section.

Configuring the USB Token

After you have set up automatic login, you may perform this task to further configure the USB token.

PINs and Passphrases

For additional PIN security with automatic login, you may encrypt your PIN stored in NVRAM and set up a passphrase for your USB token. Establishing a passphrase allows you to keep your PIN secure; another user needs only to know the passphrase, not the PIN.

When the USB token is inserted into the router, the passphrase is needed to decrypt the PIN. Once the PIN is decrypted, the router can then use the PIN to login the USB token.

Note The user has only the access they would normally have and needs only privilege level 1 to log in.

Unlocking and Locking the USB Token

The USB token itself can be locked (encrypted) or unlocked (decrypted).

Unlocking the USB token allows it to be used. Once unlocked, Cisco IOS treats the token as if it were automatically logged in. Any keys on the USB token are loaded, and if a secondary configuration file is on the token, it is executed with full user privileges (privilege level 15) independent of the privilege level of the logged-in user.

Locking the token, unlike logging out of the token, deletes any RSA keys loaded from the token and runs the secondary unconfiguration file, if configured.

Secondary Configuration and Unconfiguration Files

Configuration files that exist on a USB token are called secondary configuration files. If you create and configure a secondary configuration file, it is executed after the token is logged in. The existence of a secondary configuration file is determined by the presence of a secondary configuration file option in the Cisco IOS configuration stored in NVRAM. When the token is removed or logged out and the removal timer expires, a separate secondary unconfiguration file is processed to remove all secondary configuration elements from the running configuration. Secondary configuration and secondary unconfiguration files are executed at privilege level 15 and are not dependent on the level of the user logged in.

SUMMARY STEPS

1. **enable**
2. **crypto pki token** *token-name* **unlock** [*pin*]
3. **configure terminal**
4. **crypto pki token** *token-name* **encrypted-user-pin** [**write**]
5. **crypto pki token** *token-name* **secondary unconfig** *file*
6. **exit**
7. **crypto pki token** *token-name* **lock** [*pin*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki token** *token-name* **unlock** [*pin*] **Example:** `Router# crypto pki token mytoken unlock mypin`	(Optional) Allows the token to be used if the USB token has been locked. Once unlocked, Cisco IOS treats the token as if it has been automatically logged in. Any keys on the token are loaded and if a secondary configuration file exists, it is executed.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**crypto pki token** *token-name* **encrypted-user-pin** [**write**] **Example:** `Router(config)# crypto pki token mytoken encrypted-user-pin write`	(Optional) Encrypts the stored PIN in NVRAM.
Step 5	**crypto pki token** *token-name* **secondary unconfig** *file* **Example:** `Router(config)# crypto pki token mytoken secondary unconfig configs/myunconfigfile.cfg`	(Optional) Specifies the secondary configuration file and its location.
Step 6	**exit** **Example:** `Router(config)# exit`	Enters privileged EXEC mode.
Step 7	**crypto pki token** *token-name* **lock** [*pin*] **Example:** `Router# crypto pki token mytoken lock mypin`	(Optional) Deletes any RSA keys loaded from the token and runs the secondary unconfiguration file, if it exists.

Examples

The following example shows both the configuration and encryption of a user PIN and then the router reloading and the user PIN being unlocked:

```
! Configuring the user PIN

Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto pki token usbtoken0: user-pin
     Enter password:

! Encrypt the user PIN
```

```
Router (config)# crypto pki token usbtoken0: encrypted-user-pin
     Enter passphrase:
Router(config)# exit
Router#
Sep 20 21:51:38.076: %SYS-5-CONFIG_I: Configured from console by console
!

Router# show running config
.
.
.
crypto pki token usbtoken0 user-pin *encrypted*
.
.
.

! Reloading the router.
!
Router> enable
 Password:
!
! Decrypting the user pin.
!
Router# crypto pki token usbtoken0: unlock
 Token eToken is usbtoken0
!
Enter passphrase:
Token login to usbtoken0(eToken) successful
Router#
Sep 20 22:31:13.128: %CRYPTO-6-TOKENLOGIN: Cryptographic Token eToken
Login Successful
```

The following example shows a how a secondary unconfiguration file might be used to remove secondary configuration elements from the running configuration. For example, a secondary configuration file might be used to set up a PKI trustpoint. A corresponding unconfiguration file, named mysecondaryunconfigfile.cfg, might contain this command line:

```
no crypto pki trustpoint token-tp
```

If the token were removed and the following commands executed, the trustpoint and associated certificates would be removed from the router's running configuration:

```
Router# configure terminal
Router(config)# no crypto pki token mytoken secondary unconfig mysecondaryunconfigfile.cfg
```

What to Do Next

After you have logged into and configured the USB token, it is available for use.

- To perform USB token administrative tasks, such as changing the user PIN, copying files from the router to the USB token set key storage location, and changing USB tokens, see the "Setting Administrative Functions on the USB Token" section.
- To utilize the USB token as a cryptographic device to perform RSA operations, see the document titles in the "Related Documents" section.

- To specify that the USB token be used for RSA operations during initial autoenrollment, see the document titles in the "Related Documents" section.

Setting Administrative Functions on the USB Token

Perform this task to change default settings, such as the user PIN, the maximum number of failed attempts on the USB token, or the credential storage location.

SUMMARY STEPS

1. **enable**
2. **crypto pki token** *token-name* [**admin**] **change-pin** [*pin*]
3. **crypto pki token** *token-name device:* **label** *token-label*
4. **configure terminal**
5. **crypto key storage** *device*:
6. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*]
7. **crypto key move rsa** *keylabel* [**non-exportable**] [**on** | **storage**] *location*
8. **crypto pki token** {*token-name* | **default**} **removal timeout** [*seconds*]
9. **crypto pki token** {*token-name* | **default**} **max-retries** [*number*]
10. **exit**
11. **copy usbflash**[**0-9**]**:***filename destination-url*
12. **show usbtoken**[**0-9**]**:***filename*
13. **crypto pki token** *token-name* **logout**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki token** *token-name* [**admin**] **change-pin** [*pin*] **Example:** `Router# crypto pki token usbtoken0 admin change-pin`	(Optional) Changes the user PIN number on the USB token. • If the PIN is not changed, the default PIN—1234567890—will be used. **Note** After the PIN has been changed, you must reset the login failure count to zero (via the **crypto pki token max-retries** command). The maximum number of allowable login failures is set (by default) to 15.
Step 3	**crypto pki token** *token-name device:* **label** *token-label* **Example:** `Router# crypto pki token my token usb0: label newlabel`	(Optional) Sets or changes the name of the USB token. • The value of the *token-label* argument may be up to 31 alphanumeric characters in length including dashes and underscores. **Tip** This command is useful when configuring multiple USB tokens for automatic login, secondary configuration files, or other token specific settings.
Step 4	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 5	**crypto key storage** *device*: **Example:** `Router(config)# crypto key storage usbtoken0:`	(Optional) Sets the default RSA key storage location for newly created keys. **Note** Regardless of configuration settings, existing keys are stored on the device from where they were originally loaded.
Step 6	**crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*] **Example:** `Router(config)# crypto key generate rsa label tokenkey1 storage usbtoken0:`	(Optional) Generates the RSA key pair. • The **storage** keyword specifies the key storage location. • The **on** keyword specifies that the keys will be generated on the designated device.

	Command or Action	Purpose
Step 7	**crypto key move rsa** *keylabel* [**non-exportable**] [**on** | **storage**] *location* **Example:** `Router(config)# crypto key move rsa keypairname non-exportable on token`	(Optional) Moves existing Cisco IOS credentials from the current storage location to the specified storage location. By default, the RSA key pair remains stored on the current device. Generating the key on the router and moving it to the token takes less than a minute. Generating a key on the token, using the **on** keyword could take five to ten minutes, and is dependent on hardware key generation routines available on the USB token. When an existing RSA key pair is generated in Cisco IOS, stored on a USB token, and used for an enrollment, it may be necessary to move those existing RSA key pairs to an alternate location for permanent storage. This command is useful when using SDP with USB tokens to deploy credentials.
Step 8	**crypto pki token** {*token-name* | **default**} **removal timeout** [*seconds*] **Example:** `Router(config)# crypto pki token usbtoken0 removal timeout 60`	(Optional) Sets the time interval, in seconds, that the router will wait before removing the RSA keys that are stored in the USB token after the USB token has been removed from the router. **Note** If this command is not issued, all RSA keys and IPsec tunnels associated with the USB token are torn down immediately after the USB token is removed from the router.
Step 9	**crypto pki token** {*token-name* | **default**} **max-retries** [*number*] **Example:** `Router(config)# crypto pki token usbtoken0 max-retries 20`	(Optional) Sets the maximum number of consecutive failed login attempts allowed before access to the USB token is denied. • By default, the value is set at 15.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**copy** usbflash[**0-9**]:*filename destination-url* **Example:** `Router# copy usbflash0:file1 nvram:`	Copies files from USB token to the router. • *destination-url*—See the **copy** command page documentation for a list of supported options.
Step 12	**show usbtoken**[**0-9**]:*filename* **Example:** `Router# show usbtoken:usbfile`	(Optional) Displays information about the USB token. You can use this command to verify whether the USB token has been logged onto the router.
Step 13	`crypto pki token token-name logout` **Example:** `Router# crypto pki token usbtoken0 logout`	Logs the router out of the USB token. **Note** If you want to save any data to the USB token, you must log back into the token.

Troubleshooting USB Tokens

This section contains descriptions of the following Cisco IOS commands that can be used to help troubleshoot possible problems that may arise while using a USB token:

The show file systems Command

Use the **show file systems** command to determine whether the router recognizes that there is a USB module plugged into a USB port. The USB module should appear on the list of file systems. If the module does not appear on the list, it can indicate any of the following problems:

- A connection problem with the USB module.
- The Cisco IOS image running on the router does not support a USB module.
- A hardware problem with the USB module itself.

SUMMARY STEPS

1. **show file systems**

DETAILED STEPS

Step 1 Sample output from the **show file systems** command showing a USB token appears below. The USB module listing appears in the last line of the examples.

```
Router# show file systems

File Systems:

     Size(b)     Free(b)      Type  Flags  Prefixes
           -           -    opaque     rw   archive:
           -           -    opaque     rw   system:
           -           -    opaque     rw   null:
           -           -   network     rw   tftp:
*  129880064    69414912      disk     rw   flash:#
      491512      486395     nvram     rw   nvram:
           -           -    opaque     wo   syslog:
           -           -    opaque     rw   xmodem:
           -           -    opaque     rw   ymodem:
           -           -   network     rw   rcp:
           -           -   network     rw   pram:
           -           -   network     rw   ftp:
           -           -   network     rw   http:
           -           -   network     rw   scp:
           -           -   network     rw   https:
           -           -    opaque     ro   cns:
    63158272    33037312  usbflash     rw   usbflash0:
       32768         858  usbtoken     rw   usbtoken1:
```

The show usb device Command

Use the **show usb device** command to determine if a USB token is supported by Cisco.

SUMMARY STEPS

1. **show usb device**

DETAILED STEPS

Step 1 The following sample output for the **show usb device** command indicates whether or not the module is supported is bold in the sample output below:

```
Router# show usb device

Host Controller:1
Address:0x11
Device Configured:YES
Device Supported:YES
Description:eToken Pro 4254
Manufacturer:AKS
Version:1.0
Serial Number:
Device Handle:0x1010000
USB Version Compliance:1.0
Class Code:0xFF
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x529
Product ID:0x514
Max. Packet Size of Endpoint Zero:8
Number of Configurations:1
Speed:Low
Selected Configuration:1
Selected Interface:0

Configuration:
    Number:1
    Number of Interfaces:1
    Description:
    Attributes:None
    Max Power:60 mA

    Interface:
        Number:0
        Description:
        Class Code:255
        Subclass:0
        Protocol:0
        Number of Endpoints:0
```

The show usb controllers Command

Use the **show usb controllers** command to determine if there is a hardware problem with a USB flash module. If the **show usb controllers** command displays an error, the error indicates a hardware problem in the USB module.

You can also use the **show usb controllers** command to verify that copy operations onto a USB flash module are occurring successfully. Issuing the **show usb controllers** command after performing a file copy should display successful data transfers.

SUMMARY STEPS

1. **show usb controllers**

DETAILED STEPS

Step 1 The following sample output for the **show usb controllers** command displays a working USB flash module:

```
Router# show usb controllers

Name:1362HCD
Controller ID:1
Controller Specific Information:
    Revision:0x11
    Control:0x80
    Command Status:0x0
    Hardware Interrupt Status:0x24
    Hardware Interrupt Enable:0x80000040
    Hardware Interrupt Disable:0x80000040
    Frame Interval:0x27782EDF
    Frame Remaining:0x13C1
    Frame Number:0xDA4C
    LSThreshold:0x628
    RhDescriptorA:0x19000202
    RhDescriptorB:0x0
    RhStatus:0x0
    RhPort1Status:0x100103
    RhPort2Status:0x100303
    Hardware Configuration:0x3029
    DMA Configuration:0x0
    Transfer Counter:0x1
    Interrupt:0x9
    Interrupt Enable:0x196
    Chip ID:0x3630
    Buffer Status:0x0
    Direct Address Length:0x80A00
    ATL Buffer Size:0x600
    ATL Buffer Port:0x0
    ATL Block Size:0x100
    ATL PTD Skip Map:0xFFFFFFFF
    ATL PTD Last:0x20
    ATL Current Active PTD:0x0
    ATL Threshold Count:0x1
    ATL Threshold Timeout:0xFF

Int Level:1
Transfer Completion Codes:
         Success              :920            CRC             :0
         Bit Stuff            :0              Stall           :0
         No Response          :0              Overrun         :0
         Underrun             :0              Other           :0
         Buffer Overrun       :0              Buffer Underrun :0
Transfer Errors:
         Canceled Transfers   :2              Control Timeout :0
Transfer Failures:
         Interrupt Transfer   :0              Bulk Transfer   :0
```

```
         Isochronous Transfer :0               Control Transfer:0
Transfer Successes:
         Interrupt Transfer   :0               Bulk Transfer   :26
         Isochronous Transfer :0               Control Transfer:894

USBD Failures:
         Enumeration Failures :0               No Class Driver Found:0
         Power Budget Exceeded:0

USB MSCD SCSI Class Driver Counters:
         Good Status Failures :3               Command Fail    :0
         Good Status Timed out:0               Device not Found:0
         Device Never Opened  :0               Drive Init Fail :0
         Illegal App Handle   :0               Bad API Command :0
         Invalid Unit Number  :0               Invalid Argument:0
         Application Overflow :0               Device in use   :0
         Control Pipe Stall   :0               Malloc Error    :0
         Device Stalled       :0               Bad Command Code:0
         Device Detached      :0               Unknown Error   :0
         Invalid Logic Unit Num:0

USB Aladdin Token Driver Counters:
         Token Inserted       :1               Token Removed   :0
         Send Insert Msg Fail :0               Response Txns   :434
         Dev Entry Add Fail   :0               Request Txns    :434
         Dev Entry Remove Fail:0               Request Txn Fail:0
         Response Txn Fail    :0               Command Txn Fail:0
         Txn Invalid Dev Handle:0

USB Flash File System Counters:
         Flash Disconnected   :0               Flash Connected :1
         Flash Device Fail    :0               Flash Ok        :1
         Flash startstop Fail :0               Flash FS Fail   :0

USB Secure Token File System Counters:
         Token Inserted       :1               Token Detached  :0
         Token FS success     :1               Token FS Fail   :0
         Token Max Inserted   :0               Create Talker Failures:0
         Token Event          :0               Destroy Talker Failures:0
         Watched Boolean Create Failures:0
```

The dir Command

Use the **dir** command with the **filesystem** keyword option **usbtoken**[**0**-**9**]**:** to display all files, directories, and their permission strings on the USB token.

SUMMARY STEPS

1. **dir** [**filesystem**:]

DETAILED STEPS

Step 1 The following sample output displays directory information for the USB token:

```
Router# dir usbtoken1:

Directory of usbtoken1:/

    2  d---          64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---        4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---           0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---         512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---           0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---           0  Dec 22 2032 05:23:42 +00:00  6000
   14  d---           0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----         940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----        1423  Jun 27 1992 12:51:14 +00:00  myrunning-config

32768 bytes total (858 bytes free)
```

The following sample output displays directory information for all devices the router is aware of:

```
Router# dir all-filesystems

Directory of archive:/

No files in directory

No space information available
Directory of system:/

    2  drwx           0                   <no date>  its
  115  dr-x           0                   <no date>  lib
  144  dr-x           0                   <no date>  memory
    1  -rw-        1906                   <no date>  running-config
  114  dr-x           0                   <no date>  vfiles

No space information available
Directory of flash:/

    1  -rw-    30125020  Dec 22 2032 03:06:04 +00:00  c3825-entservicesk9-mz.123-14.T

129880064 bytes total (99753984 bytes free)
Directory of nvram:/

  476  -rw-        1947                   <no date>  startup-config
  477  ----          46                   <no date>  private-config
  478  -rw-        1947                   <no date>  underlying-config
    1  -rw-           0                   <no date>  ifIndex-table
    2  ----           4                   <no date>  rf_cold_starts
    3  ----          14                   <no date>  persistent-data

491512 bytes total (486395 bytes free)
Directory of usbflash0:/

    1  -rw-    30125020  Dec 22 2032 05:31:32 +00:00  c3825-entservicesk9-mz.123-14.T

63158272 bytes total (33033216 bytes free)
Directory of usbtoken1:/

    2  d---          64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---        4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---           0  Dec 22 2032 05:23:40 +00:00  1002
```

```
  10  d---         512  Dec 22 2032 05:23:42 +00:00  1003
  12  d---           0  Dec 22 2032 05:23:42 +00:00  5000
  13  d---           0  Dec 22 2032 05:23:42 +00:00  6000
  14  d---           0  Dec 22 2032 05:23:42 +00:00  7000
  15  ----         940  Jun 27 1992 12:50:42 +00:00  mystartup-config
  16  ----        1423  Jun 27 1992 12:51:14 +00:00  myrunning-config

32768 bytes total (858 bytes free)
```

Configuration Examples for PKI Storage

This section contains the following configuration examples:

Storing Certificates to a Specific Local Storage Location: Example

The following configuration example shows how to store certificates to the certs subdirectory. The certs subdirectory does not exist and is automatically created.

```
Router# dir nvram:

 114  -rw-        4687                    <no date>  startup-config
 115  ----        5545                    <no date>  private-config
 116  -rw-        4687                    <no date>  underlying-config
   1  ----          34                    <no date>  persistent-data
   3  -rw-         707                    <no date>  ioscaroot#7401CA.cer
   9  -rw-         863                    <no date>  msca-root#826E.cer
  10  -rw-         759                    <no date>  msca-root#1BA8CA.cer
  11  -rw-         863                    <no date>  msca-root#75B8.cer
  24  -rw-        1149                    <no date>  storagename#6500CA.cer
  26  -rw-         863                    <no date>  msca-root#83EE.cer

129016 bytes total (92108 bytes free)

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# crypto pki certificate storage disk0:/certs
Requested directory does not exist -- created
Certificates will be stored in disk0:/certs/

Router(config)# end
Router# write
*May 27 02:09:00:%SYS-5-CONFIG_I:Configured from console by consolemem
Building configuration...
[OK]

Router# directory disk0:/certs

Directory of disk0:/certs/

  14  -rw-         707  May 27 2005 02:09:02 +00:00  ioscaroot#7401CA.cer
  15  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#826E.cer
  16  -rw-         759  May 27 2005 02:09:02 +00:00  msca-root#1BA8CA.cer
  17  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#75B8.cer
  18  -rw-        1149  May 27 2005 02:09:02 +00:00  storagename#6500CA.cer
```

```
  19  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#83EE.cer

47894528 bytes total (20934656 bytes free)


! The certificate files are now on disk0/certs:
```

Logging Into a USB Token and Saving RSA Keys to the USB Token: Example

The following configuration example shows to how log into the USB token, generate RSA keys, and store the RSA keys onto the USB token:

```
! Configure the router to automatically log into the eToken
configure terminal
 crypto pki token default user-pin 0 1234567890
! Generate RSA keys and enroll certificates with the CA.
crypto pki trustpoint IOSCA
 enrollment url http://10.23.2.2
 exit
crypto ca authenticate IOSCA
Certificate has the following attributes:
       Fingerprint MD5:23272BD4 37E3D9A4 236F7E1A F534444E
      Fingerprint SHA1:D1B4D9F8 D603249A 793B3CAF 8342E1FE 3934EB7A

% Do you accept this certificate? [yes/no]:yes
Trustpoint CA certificate accepted.
crypto pki enroll
crypto pki enroll IOSCA
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.

Password:
Re-enter password:

% The subject name in the certificate will include:c2851-27.cisco.com
% Include the router serial number in the subject name? [yes/no]:no
% Include an IP address in the subject name? [no]:no
Request certificate from CA? [yes/no]:yes
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate IOSCA verbose' command will show the fingerprint.


*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint MD5:E6DDAB1B
 0E30EFE6 54529D8A DA787DBA
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint SHA1:3B0F33B
7 57C02A10 3935042B C4B6CD3D 61039251
*Jan 13 06:47:21.021:%PKI-6-CERTRET:Certificate received from Certificate Authority
! Issue the write memory command, which will automatically save the RSA keys to the eToken
! instead of private NVRAM.
Router# write memory
Building configuration...
[OK]

*Jan 13 06:47:29.481:%CRYPTO-6-TOKENSTOREKEY:Key c2851-27.cisco.com stored on
Cryptographic Token eToken Successfully
```

The following sample output from the **show crypto key mypubkey rsa** command displays stored credentials after they are successfully loaded from the USB token. Credentials that are stored on the USB token are in the protected area. When storing the credentials on the USB token, the files are stored in a directory called /keystore. However, the key files are hidden from the command-line interface (CLI).

```
Router# show crypto key mypubkey rsa

% Key pair was generated at:06:37:26 UTC Jan 13 2005
Key name:c2851-27.cisco.com
 Usage:General Purpose Key
 Key is not exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E3C644 43AA7DDD
  732E0F4E 3CA0CDAB 387ABF05 EB8F22F2 2431F1AE 5D51FEE3 FCDEA934 7FBD3603
  7C977854 B8E999BF 7FC93021 7F46ABF8 A4BA2ED6 172D3D09 B5020301 0001
% Key pair was generated at:06:37:27 UTC Jan 13 2005
Key name:c2851-27.cisco.com.server
 Usage:Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00DD96AE 4BF912EB
  2C261922 4784EF98 2E70E837 774B3778 7F7AEB2D 87F5669B BF5DDFBC F0D521A5
  56AB8FDC 9911968E DE347FB0 A514A856 B30EAFF4 D1F453E1 003CFE65 0CCC6DC7
  21FBE3AC 2F8DEA16 126754BC 1433DEF9 53266D33 E7338C95 BB020301 0001
```

Additional References

The following sections provide references related to PKI storage support.

Related Documents

Related Topic	Document Title
Connecting the USB modules to the router	Cisco Access Router USB Flash Module and USB eToken Hardware Installation Guide
eToken and USB flash data sheet	USB eToken and USB Flash Features Support
RSA keys	*Deploying RSA Keys Within a PKI*
File management (loading, copying, and rebooting files)	*Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.4
USB Token RSA Operations: Certificate server configuration	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.4T. See the "Generating a Certificate Server RSA Key Pair" section, the "Configuring a Certificate Server Trustpoint" section, and related examples.
USB Token RSA Operations: Using USB tokens for RSA operations upon initial autoenrollment	"Configuring Certificate Enrollment for a PKI" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.4T. See the "Configuring Certificate Enrollment or Autoenrollment" section.
SDP setup, configuration and use with USB tokens	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.4T. See the feature information section for the feature names on using SDP and USB tokens to deploy PKI credentials.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Storing PKI Credentials

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Implementing and Managing PKI Features Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

.

Table 2 Feature Information for Storing PKI Credentials

Feature Name	Releases	Feature Information
USB Token and Secure Device Provisioning (SDP) Integration	12.4(15)T	This feature provides the ability to provision remote devices with USB tokens using SDP. The following sections in this document provide information about this feature: • Benefits of USB Tokens • Setting Administrative Functions on the USB Token The following commands were introduced by this feature: **binary file**, **crypto key move rsa**, **template file**. **Note** This document introduces the benefits of using USB tokens and SDP for a deployment solution. For other documentation on this topic, see the "Related Documents" section.

Table 2 Feature Information for Storing PKI Credentials (continued)

Feature Name	Releases	Feature Information
Cisco IOS USB Token PKI Enhancements — Phase 2	12.4(11)T	This feature enhances USB token functionality by using the USB token as a cryptographic device. USB tokens may be used for RSA operations such as key generation, signing, and authentication. The following sections in this document provide information about this feature: • Benefits of USB Tokens • Logging Into and Setting Up the USB Token • Setting Administrative Functions on the USB Token **Note** This document introduces the benefits of using USB tokens and the keys on the token for RSA operations. For other documentation on this topic, see the "Related Documents" section.
USB Storage PKI Enhancements	12.4(4)T 12.4(11)T	This feature enhances the USB token PIN security for automatic login and increases the flexibility of USB token configuration and the RSA key storage. Cisco IOS Release 12.4(11)T introduced support for USB Storage on NPE-G2. The following sections provide information about this feature: • Configuring the USB Token • Setting Administrative Functions on the USB Token The following commands were introduced or modified by this feature: **crypto key storage**, **crypto pki generate rsa, crypto pki token encrypted-user-pin**, **crypto pki token label**, **crypto pki token lock**, **crypto pki token secondary unconfig**, **crypto pki token unlock**
Certificate — Storage Location Specification	12.2(33)SXH 12.2(33)SRA 12.4(2)T	This feature allows you to specify the storage location of local certificates for platforms that support storing certificates as separate files. All Cisco platforms support NVRAM, which is the default location, and flash local storage. Depending on your platform, you may have other supported local storage options including bootflash, slot, disk, USB flash, or USB token. The following sections provide information about this feature: • Storing Certificates to a Local Storage Location • Specifying a Local Storage Location for Certificates • Storing Certificates to a Specific Local Storage Location: Example The following commands were introduced by this feature: **crypto pki certificate storage**, **show crypto pki certificates storage**

Table 2 Feature Information for Storing PKI Credentials (continued)

Feature Name	Releases	Feature Information
USB Storage	12.3(14)T 12.4(11)T	This feature enables certain models of Cisco routers to support USB tokens. USB tokens provide secure configuration distribution and allow users to VPN credentials for deployment. Cisco IOS Release 12.4(11)T introduced support for USB Storage on NPE-G2. The following sections provide information about this feature: • PKI Credentials and USB Tokens • Setting Up and Using USB Tokens on Cisco Routers • Troubleshooting USB Tokens • Logging Into a USB Token and Saving RSA Keys to the USB Token: Example The following commands were introduced or modified by this feature: **copy**, **crypto pki token change-pin**, **crypto pki token login**, **crypto pki token logout**, **crypto pki token max-retries**, **crypto pki token removal timeout**, **crypto pki token secondary config**, **crypto pki token user-pin**, **debug usb driver**, **dir**, **show usb controllers**, **show usb device**, **show usb driver**, **show usbtoken**
Certificate - Storage Location Specification	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Source Interface Selection for Outgoing Traffic with Certificate Authority

The Source Interface Selection for Outgoing Traffic with Certificate Authority feature allows you to specify that the address of an interface be used as the source address for all outgoing TCP connections associated with that trustpoint when a designated trustpoint has been configured.

Feature Specifications for Source Interface Selection for Outgoing Traffic with Certificate Authority

Feature History	
Release	**Modification**
12.2(15)T	This feature was introduced.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Supported Platforms	
Cisco 1600, Cisco 1600R, Cisco 1710, Cisco 1720, Cisco 1750, Cisco 1751, Cisco 1760, Cisco 2400, Cisco 2610–2613, Cisco 2610XM–2611XM, Cisco 2620–2621, Cisco 2620XM–2621XM, Cisco 2650–2651, Cisco 2650XM–2651XM, Cisco 2691, Cisco 3620, Cisco 3631, Cisco 3640, Cisco 3660, Cisco 3725, Cisco 3745, Cisco 7100, Cisco 7200, Cisco 7400, Cisco 7500, Cisco 801–Cisco 806, Cisco 811, Cisco 813, Cisco 828, Cisco 8850-RPM, Cisco AS5300, Cisco AS5350, Cisco AS5400, Cisco AS5800, Cisco MC3810, Cisco ubr7200, Cisco ubr905, Cisco ubr925	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About Source Interface Selection for Outgoing Traffic with Certificate Authority

To configure the Source Interface Selection for Outgoing Traffic with Cetificate Authority feature, you must understand the following concepts:

Certificates That Identify an Entity

Certificates can be used to identify an entity. A trusted server, known as the certification authority (CA), issues the certificate to the entity after determining the identity of the entity. A router that is running Cisco IOS software obtains its certificate by making a network connection to the CA. Using the Simple Certificate Enrollment Protocol (SCEP), the router transmits its certificate request to the CA and receives the granted certificate. The router obtains the certificate of the CA in the same manner using SCEP. When validating a certificate from a remote device, the router may again contact the CA or a Lightweight Directory Access Protocol (LDAP) or HTTP server to determine whether the certificate of the remote device has been revoked. (This process is known as checking the certificate revocation list [CRL].)

In some configurations, the router may make the outgoing TCP connection using an interface that does not have a valid or routable IP address. The user must specify that the address of a different interface be used as the source IP address for the outgoing connection. Cable modems are a specific example of this requirement because the outgoing cable interface (the RF interface) usually does not have a routable address. However, the user interface (usually Ethernet) does have a valid IP address.

Source Interface for Outgoing TCP Connections Associated with a Trustpoint

The **crypto ca trustpoint** command is used to specify a trustpoint. The **source interface** command is used along with the **crypto ca trustpoint** command to specify the address of the interface that is to be used as the source address for all outgoing TCP connections associated with that trustpoint.

Note If the interface address is not specified using the **source interface** command, the address of the outgoing interface is used.

How to Configure Source Interface Selection for Outgoing Traffic with Certificate Authority

This section includes the following procedure:

Configuring the Interface for All Outgoing TCP Connections Associated with a Trustpoint

Perform this task to configure the interface that you want to use as the source address for all outgoing TCP connections associated with a trustpoint.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ca trustpoint** *name*
4. **enrollment url** *url*
5. **source interface** *interface-address*
6. **interface** *type slot/port*
7. **description** *string*
8. **ip address** *ip-address mask*
9. **interface** *type slot/port*
10. **description** *string*
11. **ip address** *ip-address mask*
12. **crypto map** *map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto ca trustpoint** *name* **Example:** `Router (config)# crypto ca trustpoint ms-ca`	Declares the Certificate Authority (CA) that your router should use and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:** `Router (ca-trustpoint)# enrollment url http://yourname:80/certsrv/mscep/mscep.dll`	Specifies the enrollment parameters of your CA.
Step 5	**source interface** *interface-address* **Example:** `Router (ca-trustpoint)# interface ethernet 0`	Interface to be used as the source address for all outgoing TCP connections associated with that trustpoint.
Step 6	**interface** *type slot/port* **Example:** `Router (ca-trustpoint)# interface ethernet 1`	Configures an interface type and enters interface configuration mode.
Step 7	**description** *string* **Example:** `Router (config-if)# description inside interface`	Adds a description to an interface configuration.
Step 8	**ip address** *ip-address mask* **Example:** `Router (config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 9	**interface** *type slot/port* **Example:** `Router (config-if)# interface ethernet1/0`	Configures an interface type.
Step 10	**description** *string* **Example:** `Router (config-if)# description outside interface 10.1.1.205 255.255.255.0`	Adds a description to an interface configuration.
Step 11	**ip address** *ip-address mask* **Example:** `Router (config-if)# ip address 10.2.2.205 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 12	**crypto map** *map-name* **Example:** `Router (config-if)# crypto map mymap`	Applies a previously defined crypto map set to an interface.

Troubleshooting Tips

Ensure that the interface specified in the command has a valid address. Attempt to ping the router using the address of the specified interface from another device (possibly the HTTP or LDAP server that is serving the CRL). You can do the same thing by using a traceroute to the router from the external device.

You can also test connectivity between the router and the CA or LDAP server by using Cisco IOS command-line interface (CLI). Enter the **ping ip** command and respond to the prompts. If you answer "yes" to the "Extended commands [n]:" prompt, you will be able to specify the source address or interface.

In addition, you can use Cisco IOS CLI to input a traceroute command. If you enter the **traceroute ip** command (in EXEC mode), you will be prompted for the destination and source address. You should specify the CA or LDAP server as the destination and the address of the interface that you specified in the "source interface" as the source address.

Configuration Examples for Source Interface Selection for Outgoing Traffic with Certificate Authority

This section includes the following example:

- Source Interface Selection for Outgoing Traffic with Certificate Authority Example, page 5

Source Interface Selection for Outgoing Traffic with Certificate Authority Example

In the following example, the router is located in a branch office. The router uses IP Security (IPSec) to communicate with the main office. Ethernet 1 is the "outside" interface that connects to the Internet Service Provider (ISP). Ethernet 0 is the interface connected to the LAN of the branch office. To access the CA server located in the main office, the router must send its IP datagrams out interface Ethernet 1 (address 10.2.2.205) using the IPSec tunnel. Address 10.2.2.205 is assigned by the ISP. Address 10.2.2.205 is not a part of the branch office or main office.

The CA cannot access any address outside the company because of a firewall. The CA sees a message coming from 10.2.2.205 and cannot respond (that is, the CA does not know that the router is located in a branch office at address 10.1.1.1, which it is able to reach).

Adding the **source interface** command tells the router to use address 10.1.1.1 as the source address of the IP datagram that it sends to the CA. The CA is able to respond to 10.1.1.1.

This scenario is configured using the **source interface** command and the interface addresses as described above.

```
crypto ca trustpoint ms-ca
 enrollment url http://ms-ca:80/certsrv/mscep/mscep.dll
 source interface ethernet0
!
interface ethernet 0
 description inside interface
 ip address 10.1.1.1 255.255.255.0
!
interface ethernet 1
 description outside interface
 ip address 10.2.2.205 255.255.255.0
 crypto map main-office
```

Additional References

For additional information related to Source Interface Selection for Outgoing Traffic with Certificate Authority, refer to the following references:

Related Documents

Related Topic	Document Title
Configuring IPSec and certification authority	Cisco IOS Security Configuration Guide, *Release 12.2*
IPSec and certification authority commands	*Cisco IOS Security Command Reference*, *Release 12.2 T*

Standards

Standards	Title
No new or modified standards are supported by this feature.	–

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	–

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features

- **source interface**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Glossary

authenticate—To prove the identity of an entity using the certificate of an identity and a secret that the identity poses (usually the private key corresponding to the public key in the certificate).

CA—Certificate Authority. A CA is an entity that issues digital certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate.

CA authentication—The user manually approves a certificate from a root CA. Usually a fingerprint of the certificate is presented to the user, and the user is asked to accept the certificate based on the fingerprint. The certificate of a root CA is signed by itself (self-signed) so that it cannot be automatically authenticated using the normal certificate verification process.

CRL—certificate revocation list. A CRL is a data structure that enumerates digital certificates that have been invalidated by their issuer prior to when they were scheduled to expire.

enrollment—A router receives its certificate via the enrollment process. The router generates a request for a certificate in a specific format (known as PKCS #10). The request is transmitted to a CA, which grants the request and generates a certificate encoded in the same format as the request. The router receives the granted certificate and stores it in an internal database for use during normal operations.

certificate—A data structure defined in International Organization for Standardization (ISO) standard X.509 to associate an entity (machine or human) with the public key of that entity. The certificate contains specific fields, including the name of the entity. The certificate is normally issued by a CA on behalf of the entity. In this case the router will act as its own CA. Common fields within a certificate include the distinguished name (DN) of the entity, the DN of the authority issuing the certificate, and the public key of the entity.

LDAP—Lightweight Directory Access Protocol. A LDAP is a protocol that provides access for management and browser applications that provide read-and-write interactive access to the X.500 directory.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

Other Security Features

Unicast Reverse Path Forwarding

Configuring Unicast Reverse Path Forwarding

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes the Unicast Reverse Path Forwarding (Unicast RPF) feature. The Unicast RPF feature helps to mitigate problems that are caused by malformed or forged IP source addresses that are passing through a router.

For a complete description of the Unicast RPF commands in this chapter, refer to the chapter "Unicast Reverse Path Forwarding Commands" of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the chapter "Using Cisco IOS Software."

In This Chapter

This chapter has the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

About Unicast Reverse Path Forwarding

The Unicast RPF feature helps to mitigate problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf and Tribal Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers (ISPs) that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.

This section covers the following information:

How Unicast RPF Works

When Unicast RPF is enabled on an interface, the router examines all packets received as input on that interface to make sure that the source address and source interface appear in the routing table and match the interface on which the packet was received. This “look backwards” ability is available only when Cisco express forwarding (CEF) is enabled on the router, because the lookup relies on the presence of the Forwarding Information Base (FIB). CEF generates the FIB as part of its operation.

Note Unicast RPF is an input function and is applied only on the input interface of a router at the upstream end of a connection.

Unicast RPF checks to see if any packet received at a router interface arrives on the best return path (return route) to the source of the packet. Unicast RPF does this by doing a reverse lookup in the CEF table. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. If there is no reverse path route on the same interface from which the packet was received, it might mean that the source address was modified. If Unicast RPF does not find a reverse path for the packet, the packet is dropped or forwarded, depending on whether an access control list (ACL) is specified in the **ip verify unicast reverse-path** interface configuration command.

Note With Unicast RPF, all equal-cost “best” return paths are considered valid. This means that Unicast RPF works in cases where multiple return paths exist, provided that each path is equal to the others in terms of the routing cost (number of hops, weights, and so on) and as long as the route is in the FIB. Unicast RPF also functions where EIGRP variants are being used and unequal candidate paths back to the source IP address exist.

When a packet is received at the interface where Unicast RPF and ACLs have been configured, the following actions occur:

Step 1 Input ACLs configured on the inbound interface are checked.

Step 2 Unicast RPF checks to see if the packet has arrived on the best return path to the source, which it does by doing a reverse lookup in the FIB table.

Step 3 CEF table (FIB) lookup is carried out for packet forwarding.

Step 4 Output ACLs are checked on the outbound interface.

Step 5 The packet is forwarded.

This section provides information about Unicast RPF enhancements:

- Access Control Lists and Logging
- Per-Interface Statistics

Access Control Lists and Logging

If an ACL is specified in the command, then when (and only when) a packet fails the Unicast RPF check, the ACL is checked to see if the packet should be dropped (using a deny statement in the ACL) or forwarded (using a permit statement in the ACL). Whether a packet is dropped or forwarded, the packet is counted in the global IP traffic statistics for Unicast RPF drops and in the interface statistics for Unicast RPF.

If no ACL is specified in the Unicast RPF command, the router drops the forged or malformed packet immediately and no ACL logging occurs. The router and interface Unicast RPF counters are updated.

Unicast RPF events can be logged by specifying the logging option for the ACL entries used by the Unicast RPF command. Using the log information, administrators can see what source addresses are being used in the attack, the time the packets arrived at the interface, and so on.

Caution Logging requires CPU and memory resources. Logging Unicast RPF events for attacks having a high rate of forged packets can degrade the performance of the router.

Per-Interface Statistics

Each time a packet is dropped or forwarded at an interface, that information is counted two ways: globally on the router and at each interface where you have applied Unicast RPF. Global statistics on dropped packets provide information about potential attacks on the network; however, these global statistics do not help to specify which interface is the source of the attack.

Per-interface statistics allow network administrators to track two types of information about malformed packets: Unicast RPF drops and Unicast RPF suppressed drops. Statistics on the number of packets that Unicast RPF drops help to identify the interface that is the entry point of the attack. The Unicast RPF drop count tracks the number of drops at the interface. The Unicast RPF suppressed drop count tracks the number of packets that failed the Unicast RPF check but were forwarded because of the permit permission set up in the ACL. Using the drop count and suppressed drop count statistics, a network administrator can takes steps to isolate the attack at a specific interface.

Note Judicious use of ACL logging can further identify the address or addresses that are being dropped by Unicast RPF.

Figure 38 illustrates how Unicast RPF and CEF work together to validate IP source addresses by verifying packet return paths. In this example, a customer has sent a packet having a source address of 192.168.1.1 from interface FDDI 2/0/0. Unicast RPF checks the FIB to see if 192.168.1.1 has a path to FDDI 2/0/0. If there is a matching path, the packet is forwarded. If there is no matching path, the packet is dropped.

Figure 38 Unicast RPF Validating IP Source Addresses

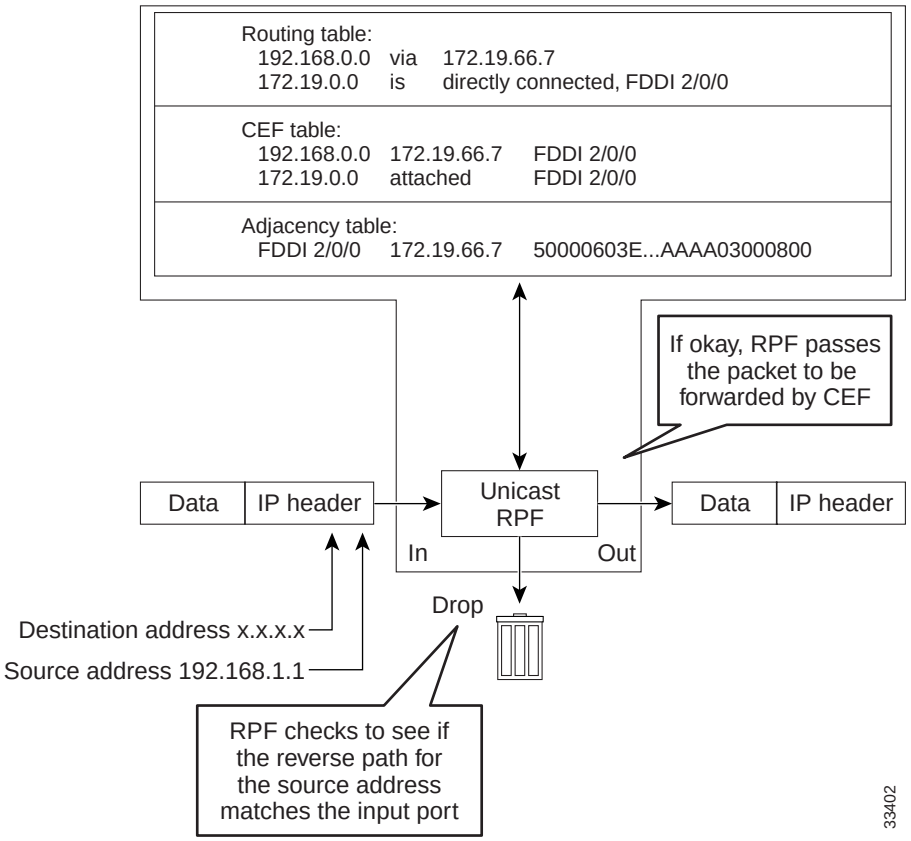
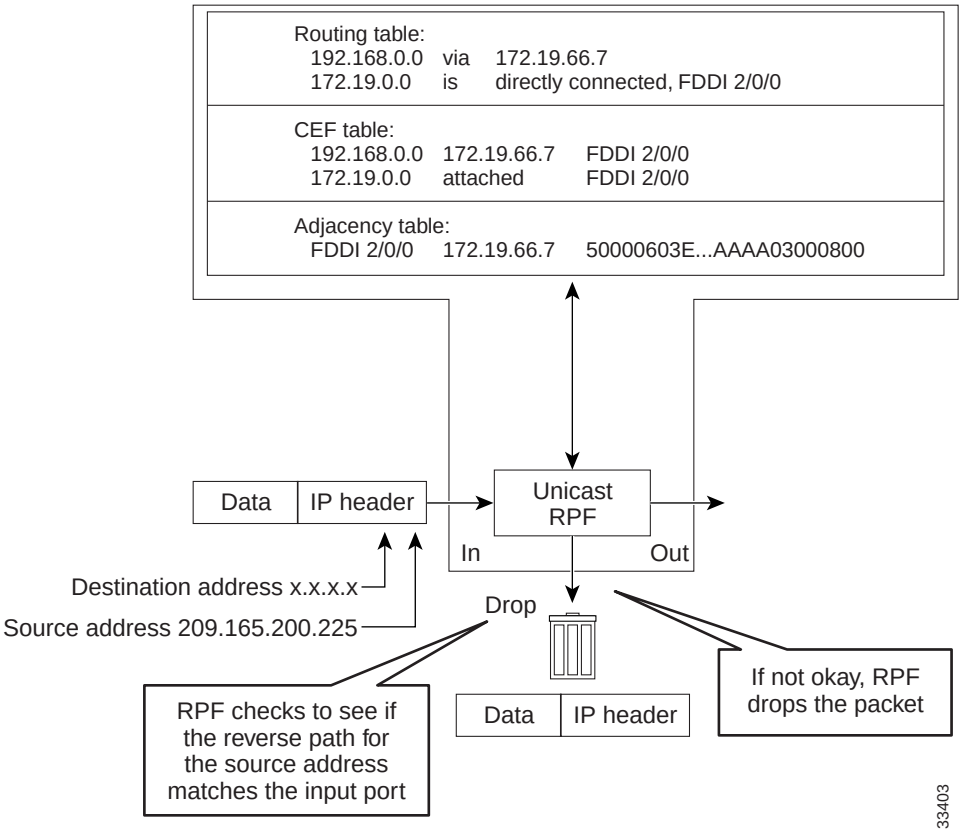


Figure 39 illustrates how Unicast RPF drops packets that fail validation. In this example, a customer has sent a packet having a source address of 209.165.200.225, which is received at interface FDDI 2/0/0. Unicast RPF checks the FIB to see if 209.165.200.225 has a return path to FDDI 2/0/0. If there is a matching path, the packet is forwarded. In this case, there is no reverse entry in the routing table that routes the customer packet back to source address 209.165.200.225 on interface FDDI 2/0/0, and so the packet is dropped.

Figure 39 Unicast RPF Dropping Packets That Fail Verification



Implementing Unicast RPF

Unicast RPF has several key implementation principles:

- The packet must be received at an interface that has the best return path (route) to the packet source (a process called *symmetric routing*). There must be a route in the FIB matching the route to the receiving interface. Adding a route in the FIB can be done via static route, network statement, or dynamic routing. (ACLs permit Unicast RPF to be used when packets are known to be arriving by specific, less optimal asymmetric input paths.)
- IP source addresses at the receiving interface must match the routing entry for the interface.
- Unicast RPF is an input function and is applied only on the input interface of a router at the upstream end of a connection.

Given these implementation principles, Unicast RPF becomes a tool that network administrators can use not only for their customers but also for their downstream network or ISP, even if the downstream network or ISP has other connections to the Internet.

Caution Using optional BGP attributes such as weight and local preference, the best path back to the source address can be modified. Modification would affect the operation of Unicast RPF.

This section provides information about the implementation of Unicast RPF:

Security Policy and Unicast RPF

Consider the following points in determining your policy for deploying Unicast RPF:

- Unicast RPF must be applied at the interface downstream from the larger portion of the network, preferably at the edges of your network.
- The further downstream you apply Unicast RPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying Unicast RPF on an aggregation router helps mitigate attacks from many downstream networks or clients and is simple to administer, but it does not help identify the source of the attack. Applying Unicast RPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying Unicast RPF across many sites does add to the administration cost of operating the network.
- The more entities that deploy Unicast RPF across Internet, intranet, and extranet resources, the better the chances of mitigating large-scale network disruptions throughout the Internet community, and the better the chances of tracing the source of an attack.
- Unicast RPF will not inspect IP packets encapsulated in tunnels, such as GRE, LT2P, or PPTP. Unicast RPF must be configured at a home gateway so that Unicast RPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.

Where to Use Unicast RPF

Unicast RPF can be used in any “single-homed” environment where there is essentially only one access point out of the network; that is, one upstream connection. Networks having one access point offer the best example of symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet. Unicast RPF is best used at the network perimeter for Internet, intranet, or extranet environments, or in ISP environments for customer network terminations.

The following sections provide a look at implementing Unicast RPF in two network environments:

Enterprise Networks with a Single Connection to an ISP

In enterprise networks, one objective of using Unicast RPF for filtering traffic at the input interface (a process called *ingress filtering*) is for protection from malformed packets arriving from the Internet. Traditionally, local networks that have one connection to the Internet would use ACLs at the receiving interface to prevent spoofed packets from the Internet from entering their local network.

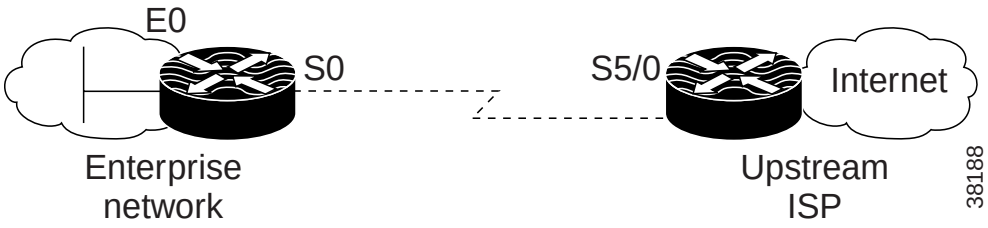
ACLs work well for many single-homed customers; however, there are trade-offs when ACLs are used as ingress filters, including two commonly referenced limitations:

- Packet per second (PPS) performance at very high packet rates
- Maintenance of the ACL (whenever there are new addresses added to the network)

Unicast RPF is one tool that addresses both of these limitations. With Unicast RPF, ingress filtering is done at CEF PPS rates. This processing speed makes a difference when the link is more than 1 Mbps. Additionally, since Unicast RPF uses the FIB, no ACL maintenance is necessary, and thus the administration overhead of traditional ACLs is reduced. The following figure and example demonstrate how Unicast RPF is configured for ingress filtering.

Figure 40 illustrates an enterprise network that has a single link to an upstream ISP. In this example, Unicast RPF is applied at interface S0 on the enterprise router for protection from malformed packets arriving from the Internet. Unicast RPF is also applied at interface S5/0 on the ISP router for protection from malformed packets arriving from the enterprise network.

Figure 40 Enterprise Network Using Unicast RPF for Ingress Filtering



Using the topography in Figure 40, a typical configuration (assuming that CEF is turned on) on the ISP router would be as follows:

```
ip cef
interface loopback 0
  description Loopback interface on Gateway Router 2
  ip address 192.168.3.1 255.255.255.255
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
interface Serial 5/0
  description 128K HDLC link to ExampleCorp WT50314E  R5-0
  bandwidth 128
  ip unnumbered loopback 0
  ip verify unicast reverse-path
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
ip route 192.168.10.0 255.255.252.0 Serial 5/0
```

The gateway router configuration of the enterprise network (assuming that CEF is turned on) would look similar to the following:

```
ip cef
interface Ethernet 0
 description ExampleCorp LAN
 ip address 192.168.10.1 255.255.252.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
interface Serial 0
```

```
 description 128K HDLC link to ExampleCorp Internet Inc WT50314E  C0
 bandwidth 128
 ip unnumbered ethernet 0
 ip verify unicast reverse-path
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
ip route 0.0.0.0 0.0.0.0 Serial 0
```

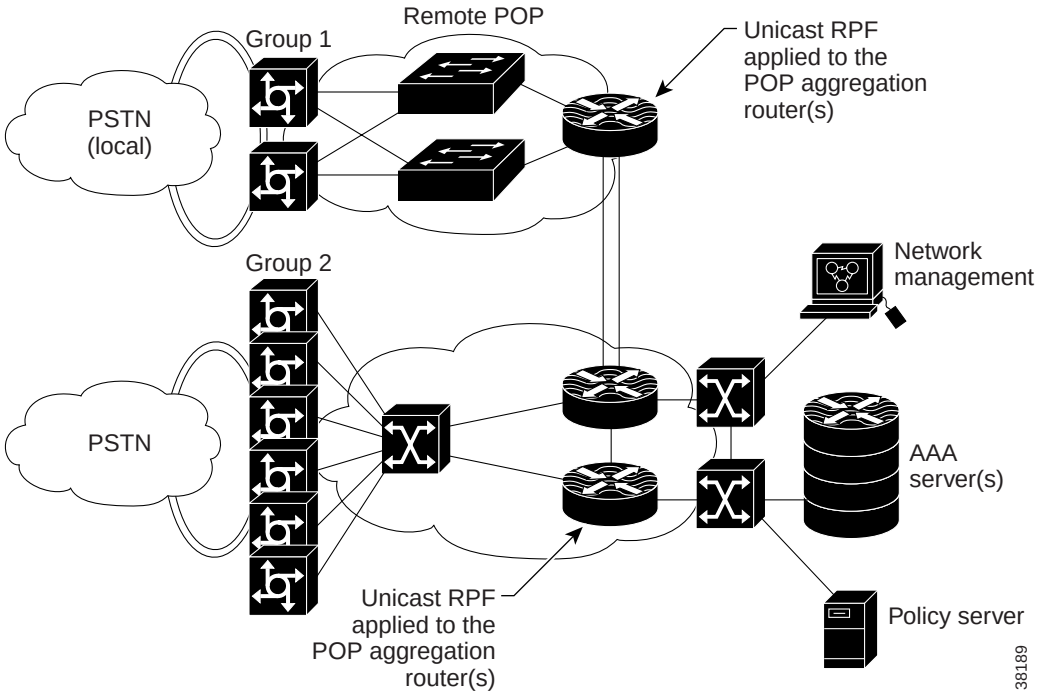
Notice that Unicast RPF works with a single default route. There are no additional routes or routing protocols. Network 192.168.10.0/22 is a connected network. Hence, packets coming from the Internet with a source address in the range 192.168.10.0/22 will be dropped by Unicast RPF.

Network Access Server Application (Applying Unicast RPF in PSTN/ISDN PoP Aggregation Routers)

Aggregation routers are ideal places to use Unicast RPF with single-homed clients. Unicast RPF works equally well on leased-line or PSTN/ISDN/xDSL customer connections into the Internet. In fact, dialup connections are reputed to be the greatest source of DoS attacks using forged IP addresses. As long as the network access server supports CEF, Unicast RPF will work. In this topology, the customer aggregation routers need not have the full Internet routing table. Aggregation routers need the routing prefixes information (IP address block); hence, information configured or redistributed in the Interior Gateway Protocol (IGP) or Internal Border Gateway Protocol (IBGP) (depending on the way that you add customer routes into your network) would be enough for Unicast RPF to do its job.

Figure 41 illustrates the application of Unicast RPF to the aggregation and access routers for an Internet service provider (ISP) point of presence (POP), with the ISP routers providing dialup customer connections. In this example, Unicast RPF is applied upstream from the customer dialup connection router on the receiving (input) interfaces of the ISP aggregation routers.

Figure 41 Unicast RPF Applied to PSTN/ISDN Customer Connections



Routing Table Requirements

To work correctly, Unicast RPF needs proper information in the CEF tables. This requirement does not mean that the router must have the entire Internet routing table. The amount of routing information needed in the CEF tables depends on where Unicast RPF is configured and what functions the router performs in the network. For example, in an ISP environment, a router that is a leased-line aggregation router for customers needs only the information based on the static routes redistributed into the IGP or IBGP (depending on which technique is used in the network). Unicast RPF would be configured on the customer interfaces—hence the requirement for minimal routing information. In another scenario, a single-homed ISP could place Unicast RPF on the gateway link to the Internet. The full Internet routing table would be required. Requiring the full routing table would help protect the ISP from external DoS attacks that use addresses that are not in the Internet routing table.

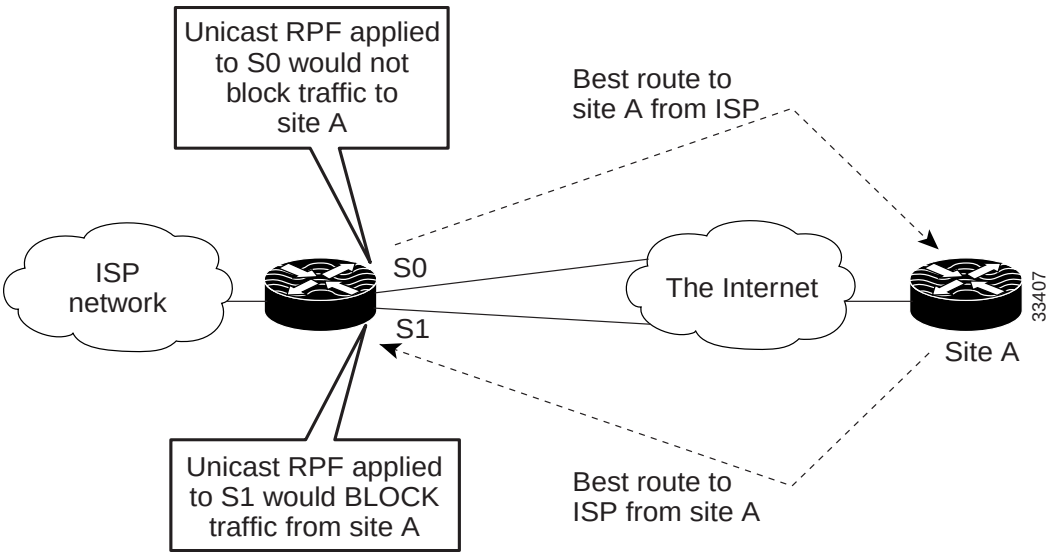
Where Not to Use Unicast RPF

Unicast RPF should not be used on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry (see Figure 42), meaning multiple routes to the source of a packet. Unicast RPF should be applied only where there is natural or configured symmetry. As long as administrators carefully plan which interfaces they activate Unicast RPF on, routing asymmetry is not a serious problem.

For example, routers at the edge of the network of an ISP are more likely to have symmetrical reverse paths than routers that are in the core of the ISP network. Routers that are in the core of the ISP network have no guarantee that the best forwarding path out of the router will be the path selected for packets returning to the router. Hence, it is not recommended that you apply Unicast RPF where there is a chance of asymmetric routing, unless you use ACLs to allow the router to accept incoming packets. ACLs permit Unicast RPF to be used when packets are known to be arriving by specific, less optimal asymmetric input paths. However, it is simplest to place Unicast RPF only at the edge of a network or, for an ISP, at the customer edge of the network.

Figure 42 illustrates how Unicast RPF can block legitimate traffic in an asymmetrical routing environment.

Figure 42 Unicast RPF Blocking Traffic in an Asymmetrical Routing Environment



Unicast RPF with BOOTP and DHCP

Unicast RPF will allow packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that Bootstrap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP) functions work properly. This enhancement was added in Cisco IOS Release 12.0 and later, but it is not in Cisco IOS Release 11.1CC.

Restrictions

There are some basic restrictions to applying Unicast RPF to multihomed clients:

- Clients should not be multihomed to the same router because multihoming defeats the purpose of building a redundant service for the client.
- Customers must ensure that the packets flowing up the link (out to the Internet) match the route advertised out the link. Otherwise, Unicast RPF filters those packets as malformed packets.
- Unicast RPF is available only for platform images that support CEF. Unicast RPF is supported in Cisco IOS Releases 11.1(17)CC and 12.0 and later. It is not available in Cisco IOS Release 11.2 or 11.3.

Related Features and Technologies

For more information about Unicast RPF-related features and technologies, review the following:

- Unicast RPF requires Cisco express forwarding (CEF) to function properly on the router. For more information about CEF, refer to the *Cisco IOS Switching Services Configuration Guide*.
- Unicast RPF can be more effective at mitigating spoofing attacks when combined with a policy of *ingress* and *egress* filtering using Cisco IOS access control lists (ACLs).
  - Ingress filtering applies filters to traffic received at a network interface from either internal or external networks. With ingress filtering, packets that arrive from other networks or the Internet and that have a source address that matches a local network, private, or broadcast address are dropped. In ISP environments, for example, ingress filtering can apply to traffic received at the router from either the client (customer) or the Internet.
  - Egress filtering applies filters to traffic exiting a network interface (the sending interface). By filtering packets on routers that connect your network to the Internet or to other networks, you can permit only packets with valid source IP addresses to leave your network.

  For more information on network filtering, refer to RFC 2267 and to the *Cisco IOS IP Configuration Guide*.
- Cisco IOS software provides additional features that can help mitigate DoS attacks:
  - Committed Access Rate (CAR). CAR allows you to enforce a bandwidth policy against network traffic that matches an access list. For example, CAR allows you to rate-limit what should be low-volume traffic, such as ICMP traffic. To find out more about CAR, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide*.
  - Context-based Access Control (CBAC). CBAC selectively blocks any network traffic not originated by a protected network. CBAC uses timeout and threshold values to manage session state information, helping to determine when to drop sessions that do not become fully established. Setting timeout values for network sessions helps mitigate DoS attacks by freeing up system resources, dropping sessions after a specified amount of time. For more information on CBAC, refer to the *Cisco IOS Security Configuration Guide*.

  - TCP Intercept. The TCP Intercept feature implements software to protect TCP servers from TCP SYN-flooding attacks, which are a type of DoS attack. A SYN-flooding attack occurs when a hacker floods a server with a barrage of requests for connection. Like CBAC, the TCP Intercept feature also uses timeout and threshold values to manage session state information, helping to determine when to drop sessions that do not become fully established. For more information on TCP Intercept, refer to the *Cisco IOS Security Configuration Guide*.

Prerequisites to Configuring Unicast RPF

Prior to configuring Unicast RPF, configure ACLs:

- Configure standard or extended ACLs to mitigate transmission of invalid IP addresses (perform egress filtering). Permit only valid source addresses to leave your network and get onto the Internet. Prevent all other source addresses from leaving your network for the Internet.
- Configure standard or extended ACLs entries to drop (deny) packets that have invalid source IP addresses (perform ingress filtering). Invalid source IP addresses include the following types:
  - Reserved addresses
  - Loopback addresses
  - Private addresses (RFC 1918, *Address Allocation for Private Internets*)
  - Broadcast addresses (including multicast addresses)
  - Source addresses that fall outside the range of valid addresses associated with the protected network
- Configure standard or extended ACL entries to forward (permit) packets that fail the Unicast RPF checks to allow specific traffic from known asymmetric routed sources.
- Configure ACLs to track Unicast RPF events by adding the logging option into the ACL command. During network attacks, judicious logging of dropped or forwarded packets (suppressed drops) can provide additional information about network attacks.

Unicast RPF Configuration Task List

The following sections describe the configuration tasks for Unicast RPF. Each task in the list is identified as either optional or required.

- Configuring Unicast RPF (Required)
- Verifying Unicast RPF (Optional)

See the section "Unicast RPF Configuration Examples" at the end of this chapter.

Configuring Unicast RPF

To use Unicast RPF, you must configure the router for CEF switching or CEF distributed switching. There is no need to configure the input interface for CEF switching because Unicast RPF has been implemented as a search through the FIB using the source IP address. As long as CEF is running on the router, individual interfaces can be configured with other switching modes. Unicast RPF is an input-side function that is enabled on an interface or subinterface that supports any type of encapsulation and operates on IP packets received by the router. It is very important that CEF be turned on globally in the router—Unicast RPF will not work without CEF.

To configure Unicast RPF, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip cef` or `Router(config)# ip cef distributed`	Enables CEF or distributed CEF on the router. Distributed CEF is required for routers that use a Route Switch Processor (RSP) and Versatile Interface Processor (VIP), which includes Unicast RPF. You might want to disable CEF or distributed CEF (dCEF) on a particular interface if that interface is configured with a feature that CEF or dCEF does not support. In this case, you would enable CEF globally, but disable CEF on a specific interface using the **no ip route-cache cef** interface command, which enables all but that specific interface to use express forwarding. If you have disabled CEF or dCEF operation on an interface and want to reenable it, you can do so by using the **ip route-cache cef** command in interface configuration mode.
Step 2	`Router(config-if)# interface` *type*	Selects the input interface on which you want to apply Unicast RPF. This is the receiving interface, which allows Unicast RPF to verify the best return path before forwarding the packet on to the next destination. The interface type is specific to your router and the types of interface cards installed on the router. To display a list of available interface types, enter the **interface ?** command.
Step 3	`Router(config-if)# ip verify unicast reverse-path` ***list***	Enables Unicast RPF on the interface. Use the *list* option to identify an access list. If the access list denies network access, spoofed packets are dropped at the interface. If the access list permits network access, spoofed packets are forwarded to the destination address. Forwarded packets are counted in the interface statistics. If the access list includes the logging option, information about the spoofed packets is logged to the log server. Repeat this step for each access list that you want specify.
Step 4	`Router(config-if)# exit`	Exits interface configuration mode. Repeat Steps 2 and 3 for each interface on which you want to apply Unicast RPF.

Verifying Unicast RPF

To verify that Unicast RPF is operational, use the **show cef interface** command. The following example shows that Unicast RPF is enabled at interface serial2/0/0.

```
Router-3# show cef interface serial 2/0/0
```

```
Serial2/0/0 is up (if_number 8)
 Internet address is 192.168.10.2/30
 ICMP redirects are never sent
 Per packet loadbalancing is disabled
!The next line displays Unicast RPF packet dropping information.
 IP unicast RPF check is enabled
 Inbound access list is not set
 Outbound access list is not set
 Interface is marked as point to point interface
 Packets switched to this interface on linecard are dropped to next slow path
 Hardware idb is Serial2/0/0
 Fast switching type 4, interface type 6
!The next line displays Unicast RPF packet dropping information.
 IP Distributed CEF switching enabled
 IP LES Feature Fast switching turbo vector
 IP Feature CEF switching turbo vector
 Input fast flags 0x40, Output fast flags 0x0, ifindex 7(7)
 Slot 2 Slot unit 0 VC -1
 Transmit limit accumulator 0x48001A02 (0x48001A02)
 IP MTU 1500
```

Troubleshooting Tips

If you experience problems while using Unicast RPF, check the following items.

HSRP Failure

Failure to disable Unicast RPF before disabling CEF can cause Hot Standby Router Protocol (HSRP) failure. If you want to disable CEF on the router, you must first disable Unicast RPF. To disable Unicast RPF, see the section "Monitoring and Maintaining Unicast RPF."

Dropped Boot Requests

In Cisco IOS Release 11.1(17)CC, Unicast RPF can drop BOOTP request packets that have a source address of 0.0.0.0 due to source address verification at the interface. To enable boot requests to work on the interface, you must use ACLs instead of Unicast RPF.

Monitoring and Maintaining Unicast RPF

This section describes commands used to monitor and maintain Unicast RPF.

Command	Purpose
`Router# show ip traffic`	Displays global router statistics about Unicast RPF drops and suppressed drops.
`Router# show ip interface type`	Displays per-interface statistics about Unicast RPF drops and suppressed drops.
`Router# show access-lists`	Displays the number of matches to a specific ACL.
`Router(config-if)# no ip verify unicast reverse-path list`	Disables Unicast RPF at the interface. Use the *list* option to disable Unicast RPF for a specific ACL at the interface.

Caution To disable CEF, you must first disable Unicast RPF. Failure to disable Unicast RPF before disabling CEF can cause HSRP failure. If you want to disable CEF on the router, you must first disable Unicast RPF.

Unicast RPF counts the number of packets dropped or suppressed because of malformed or forged source addresses. Unicast RPF counts dropped or forwarded packets that include the following global and per-interface information:

- Global Unicast RPF drops
- Per-interface Unicast RPF drops
- Per-interface Unicast RPF suppressed drops

The **show ip traffic** command shows the total number (global count) of dropped or suppressed packets for all interfaces on the router. The Unicast RPF drop count is included in the IP statistics section.

```
Router# show ip traffic

IP statistics:
  Rcvd:  1471590 total, 887368 local destination
         0 format errors, 0 checksum errors, 301274 bad hop count
         0 unknown protocol, 0 not a gateway
         0 security failures, 0 bad options, 0 with options
  Opts:  0 end, 0 nop, 0 basic security, 0 loose source route
         0 timestamp, 0 extended security, 0 record route
         0 stream ID, 0 strict source route, 0 alert, 0 other
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 couldn't fragment
  Bcast: 205233 received, 0 sent
  Mcast: 463292 received, 462118 sent
  Sent:  990158 generated, 282938 forwarded
  ! The second line below (“0 unicast RPF”) displays Unicast RPF packet dropping
  information.
  Drop:  3 encapsulation failed, 0 unresolved, 0 no adjacency
         0 no route, 0 unicast RPF, 0 forced drop
```

A nonzero value for the count of dropped or suppressed packets can mean one of two things:

- Unicast RPF is dropping or suppressing packets that have a bad source address (normal operation).

- Unicast RPF is dropping or suppressing legitimate packets because the route is misconfigured to use Unicast RPF in environments where asymmetric routing exists; that is, where multiple paths can exist as the best return path for a source address.

The **show ip interface** command shows the total of dropped or suppressed packets at a specific interface. If Unicast RPF is configured to use a specific ACL, that ACL information is displayed along with the drop statistics.

```
Router> show ip interface ethernet0/1/1

   Unicast RPF ACL 197
   1 unicast RPF drop
   1 unicast RPF suppressed drop
```

The **show access-lists** command displays the number of matches found for a specific entry in a specific access list.

```
Router> show access-lists

Extended IP access list 197
     deny ip 192.168.201.0 0.0.0.63 any log-input (1 match)
     permit ip 192.168.201.64 0.0.0.63 any log-input (1 match)
     deny ip 192.168.201.128 0.0.0.63 any log-input
     permit ip 192.168.201.192 0.0.0.63 any log-input
```

Unicast RPF Configuration Examples

This section provides the following configuration examples:

- Unicast RPF on a Leased-Line Aggregation Router Example
- Unicast RPF on the Cisco AS5800 Using Dialup Ports Example
- Unicast RPF with Inbound and Outbound Filters Example
- Unicast RPF with ACLs and Logging Example

Unicast RPF on a Leased-Line Aggregation Router Example

The following commands enable Unicast RPF on a serial interface:

```
ip cef
! or “ip cef distributed” for RSP+VIP based routers
!
interface serial 5/0/0
 ip verify unicast reverse-path
```

Unicast RPF on the Cisco AS5800 Using Dialup Ports Example

The following example enables Unicast RPF on a Cisco AS5800. The **interface Group-Async** command makes it easy to apply Unicast RPF on all the dialup ports.

```
ip cef
!
interface Group-Async1
 ip verify unicast reverse-path
```

Unicast RPF with Inbound and Outbound Filters Example

The following example uses a very simple single-homed ISP to demonstrate the concepts of ingress and egress filters used in conjunction with Unicast RPF. The example illustrates an ISP-allocated classless interdomain routing (CIDR) block 209.165.202.128/28 that has both inbound and outbound filters on the upstream interface. Be aware that ISPs are usually not single-homed. Hence, provisions for asymmetrical flows (when outbound traffic goes out one link and returns via a different link) need to be designed into the filters on the border routers of the ISP.

```
ip cef distributed
!
interface Serial 5/0/0
 description Connection to Upstream ISP
 ip address 209.165.200.225 255.255.255.252
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip verify unicast reverse-path
 ip access-group 111 in
 ip access-group 110 out
!
access-list 110 permit ip 209.165.202.128 0.0.0.31 any
access-list 110 deny ip any any log
access-list 111 deny ip host 0.0.0.0 any log
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 209.165.202.128 0.0.0.31 any log
access-list 111 permit ip any any
```

Unicast RPF with ACLs and Logging Example

The following example demonstrates the use of ACLs and logging with Unicast RPF. In this example, extended ACL 197 provides entries that deny or permit network traffic for specific address ranges. Unicast RPF is configured on interface Ethernet0 to check packets arriving at that interface.

For example, packets with a source address of 192.168.201.10 arriving at interface Ethernet0 are dropped because of the deny statement in ACL 197. In this case, the ACL information is logged (the logging option is turned on for the ACL entry) and dropped packets are counted per interface and globally. Packets with a source address of 192.168.201.100 arriving at interface Ethernet0 are forwarded because of the permit statement in ACL 197. ACL information about dropped or suppressed packets is logged (logging option turned on for the ACL entry) to the log server.

```
ip cef distributed
!
int eth0/1/1
 ip address 192.168.200.1 255.255.255.0
 ip verify unicast reverse-path 197
!
int eth0/1/2
 ip address 192.168.201.1 255.255.255.0
!
access-list 197 deny   ip 192.168.201.0 0.0.0.63 any log-input
access-list 197 permit ip 192.168.201.64 0.0.0.63 any log-input
access-list 197 deny   ip 192.168.201.128 0.0.0.63 any log-input
access-list 197 permit ip 192.168.201.192 0.0.0.63 any log-input
access-list 197 deny ip host 0.0.0.0 any log
```

Unicast Reverse Path Forwarding Loose Mode

The Unicast Reverse Path Forwarding Loose Mode feature creates a new option for Unicast Reverse Path Forwarding (Unicast RPF), providing a scalable anti-spoofing mechanism suitable for use in multihome network scenarios. This mechanism is especially relevant for Internet Service Providers (ISPs), specifically on routers that have multiple links to multiple ISPs. In addition, Unicast RPF (strict or loose mode), when used in conjunction with a Border Gateway Protocol (BGP) "trigger," provides an excellent quick reaction mechanism that allows network traffic to be dropped on the basis of either the source or destination IP address, giving network administrators an efficient tool for mitigating denial of service (DoS) and distributed denial of service (DDoS) attacks.

History for the Unicast Reverse Path Forwarding Loose Mode Feature

Release	Modification
12.0(15)S	This feature was introduced.
12.1(8a)E	This feature was integrated into Cisco IOS Release 12.1(8a)E.
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2(13)T.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Unicast Reverse Path Forwarding Loose Mode

- To use Unicast RPF, you must enable Cisco Express Forwarding (CEF) switching or distributed CEF (dCEF) switching in the router. There is no need to configure the input interface for CEF switching. As long as CEF is running on the router, individual interfaces can be configured for other switching modes.

Information About Unicast Reverse Path Forwarding Loose Mode

Before configuring Unicast Reverse Path Forwarding Loose Check, you should understand the following concepts:

- Unicast Reverse Path Forwarding: Background, page 2
- Loose Mode, page 3

Unicast Reverse Path Forwarding: Background

A number of common types of DoS attacks take advantage of forged or rapidly changing source IP addresses, allowing attackers to thwart efforts by ISPs to locate or filter these attacks. Unicast RPF was originally created to help mitigate such attacks by providing an automated, scalable mechanism to implement the Internet Engineering Task Force (IETF) Best Common Practices 38/Request for Comments 2827 (BCP 38/RFC 2827) anti-spoofing filtering on the customer-to-ISP network edge. By taking advantage of the information stored in the Forwarding Information Base (FIB) that is created by the CEF switching process, Unicast RPF can determine whether IP packets are spoofed or malformed by matching the IP source address and ingress interface against the FIB entry that reaches “back” to this source (a so-called “reverse lookup”). Packets that are received from one of the best reverse path routes back out of the same interface are forwarded as normal. If there is no reverse path route on the same interface from which the packet was received, it might mean that the source address was modified, and the packet is dropped (by default).

This original implementation of Unicast RPF, known as “strict mode,” required a match between the ingress interface and the reverse path FIB entry. With Unicast RPF, all equal-cost “best” return paths are considered valid, meaning that it works for cases in which multiple return paths exist, provided that each path is equal in routing cost to the others (number of hops, weights, and so on), and as long as the route is in the FIB. Unicast RPF also functions when Enhanced Interior Gateway Routing Protocol (EIGRP) variants are being used and unequal candidate paths back to the source IP address exist. The strict mode works well for customer-to-ISP network edge configurations that have symmetrical flows (including some multihomed configurations in which symmetrical flows can be enforced).

However, some customer-to-ISP network edges and nearly all ISP-to-ISP network edges use multihomed configurations in which routing asymmetry is typical. When traffic flows are asymmetrical, that is, those in which traffic from Network A to Network B would normally take a different path from traffic flowing from Network B to Network A, the Unicast RPF check will always fail the strict mode test. Because this type of asymmetric routing is common among ISPs and in the Internet core, the original implementation of Unicast RPF was not available for use by ISPs on their core routers and ISP-to-ISP links.

Over time and with an increase in DDoS attacks on the Internet, the functionality of Unicast RPF was reviewed as a tool that ISPs can use on the ISP-to-ISP network edge (an ISP router "peered" with another ISP router) to enable dynamic BGP, triggered black-hole filtering. To provide this functionality, however, the mechanisms used with Unicast RPF had to be modified to permit its deployment on the ISP-to-ISP network edge so that asymmetrical routing is not an issue.

Loose Mode

To provide ISPs with a DDoS resistance tool on the ISP-to-ISP edge of a network, Unicast RPF was modified from its original strict mode implementation to check the source addresses of each ingress packet without regard for the specific interface on which it was received. This modification is known as "loose mode." Loose mode allows Unicast RPF to automatically detect and drop packets such as the following:

- IETF RFC 1918 source addresses
- Other Documenting Special Use Addresses (DUSA) that should not appear in the source
- Unallocated addresses that have not been allocated by the Regional Internet Registries (RIRs)
- Source addresses that are routed to a null interface on the router

Loose mode removes the match requirement on the specific ingress interface, allowing Unicast RPF to loose-check packets. This packet checking allows the "peering" router of an ISP having multiple links to multiple ISPs to check the source IP address of ingress packets to determine whether they exist in the FIB. If they exist, the packets are forwarded. If they do not exist in the FIB, the packets fail and are dropped. This checking increases resistance against DoS and DDoS attacks that use spoofed source addresses and unallocated IP addresses.

How to Configure Unicast Reverse Path Forwarding Loose Mode

This section contains the following procedure:

- Configuring Unicast Reverse Path Forwarding Loose Mode, page 3

Configuring Unicast Reverse Path Forwarding Loose Mode

To configure Unicast RPF loose mode, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef**
4. **interface** *type slot/port-adapter/port*
5. **ip verify unicast source reachable-via any**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** **Example:** `Router (config)# ip cef`	Enables CEF on the route processor card.
Step 4	**interface** *type slot/port-adapter/port* **Example:** `Router (config)# interface serial5/0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**ip verify unicast source reachable-via any** `Router (config-if)# ip verify unicast source reachable-via any`	Enables Unicast RPF using loose mode.

Troubleshooting Tips

CEF Not Enabled

If CEF is not enabled on your device and an attempt is made to deploy Unicast RPF, the following error message is generated:

```
Router(config-if)# ip verify unicast source reachable-via any
% CEF not enabled. Enable first.
```

Dropped Packets

If it is believed that Unicast RPF is dropping packets that are deemed valid, it may be necessary to configure an access list within Unicast RPF to pass these specific packets.

- Check to see if Unicast RPF is dropping packets using the following **show** commands.

  ```
  Router# show ip traffic | include unicast RPF
  ```

  The above command output displays the global counter for packets dropped by Unicast RPF. If the packet drop counter is increasing, Unicast RPF is dropping packets.

  ```
  Router# show ip interface {type/slot/port} | include verif
  ```

  The above command output displays drop counters on a per-interface basis. If the packet drop counter is increasing, Unicast RPF is dropping packets on the referenced interface.

- Configure a “classification” access list (one that is used to identify traffic types) and add it to the Unicast RPF configuration on the interface or interfaces that are in question.

In this case, the most prudent classification access list would be one that includes a series of “deny” statements covering the traffic types in question (instead of the more traditional “permit” statements that would be used, for example, in a typical classification access list that would be applied directly to an interface). The **logging** keyword may be useful for this access list as well.

- Apply the above access list to Unicast RPF on the interface in question using the following command:

```
Router (config-if)# ip verify unicast source reachable-via any 199
```

- Periodically check the counters on the above access list using the following **show** command:

```
Router# show ip access-list 199
```

If the access list hit counters are increasing for the packet type in question, Unicast RPF is dropping the packets in question. To permit them, configure an access list using a “permit” statement for the packet type in question and apply it to Unicast RPF.

Configuration Examples for Unicast Reverse Path Forwarding Loose Mode

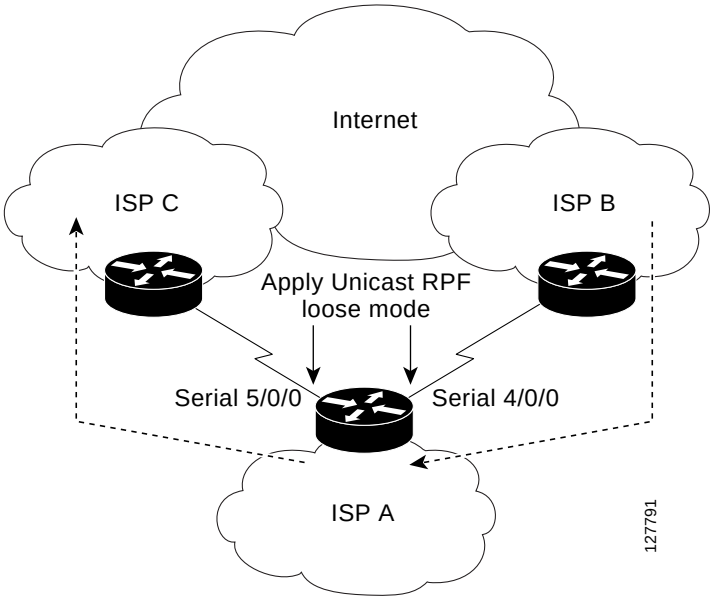
This section includes the following configuration example:

- Configuring Unicast RPF Using Loose Mode: Example, page 5

Configuring Unicast RPF Using Loose Mode: Example

The following example (see Figure 1) uses a simple dual-homed ISP to demonstrate the concept of Unicast RPF loose mode. The example illustrates an ISP (A) peering router that is connected to two different upstream ISPs (B and C) and shows that traffic flows into and out of ISP A may be asymmetric given this dual-homed configuration. Hence, provisions for asymmetrical flows (when outbound traffic goes out one link and returns via a different link) must be accounted for by the Unicast RPF deployment. In this case, it is appropriate to use the loose-mode configuration of Unicast RPF because this configuration alleviates the interface dependency of strict mode.

Figure 1 Unicast RPF Loose Mode




```
interface Serial4/0/0
 description – link to ISP B
 ip address 192.168.200.225 255.255.255.252
 no ip redirects
 no ip directed-broadcasts
 no ip proxy-arp
 ip verify unicast source reachable-via any
!
interface Serial5/0/0
 description – link to ISP C
 ip address 172.16.100.9 255.255.255.252
 no ip redirects
 no ip directed-broadcasts
 no ip proxy-arp
 ip verify unicast source reachable-via any
!
```

Additional References

The following sections provide references related to Unicast Reverse Path Forwarding Loose Check.

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Command Reference*, *Release 12.3T*
Best practices using Unicast RPF	*Internet Service Provider (ISP) Security Bootcamp/Best Practices—CPN–Summit–2004/Paris–Sept–04*

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable content, including links to products, technologies, solutions, technical tips, and tools.Registered Cisco.com users can log on from this page to access even more content.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information

about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **ip verify unicast reverse-path**
- **ip verify unicast source reachable-via**

Secure Shell

Configuring Secure Shell

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes the Secure Shell (SSH) feature. The SSH feature consists of an application and a protocol.

For a complete description of the SSH commands in this chapter, refer to the chapter "Secure Shell Commands" of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the chapter "Using Cisco IOS Software."

In This Chapter

This chapter has the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

About Secure Shell

Secure Shell (SSH) is an application and a protocol that provide a secure replacement to the Berkeley r-tools. The protocol secures the sessions using standard cryptographic mechanisms, and the application can be used similarly to the Berkeley **rexec** and **rsh** tools. There are currently two versions of SSH available: SSH Version 1 and SSH Version 2. This document describes SSH Version 1. For information about SSH Version 2, see the document Secure Shell Version 2 Support.

Note Hereafter, unless otherwise noted, the term “SSH” will denote “SSH Version 1” only.

This rest of this section covers the following information:

- How SSH Works
- Restrictions
- Related Features and Technologies
- Prerequisites to Configuring SSH

How SSH Works

This section provides the following information about how SSH works:

- SSH Server
- SSH Integrated Client

SSH Server

The SSH Server feature enables a SSH client to make a secure, encrypted connection to a Cisco router. This connection provides functionality that is similar to that of an inbound Telnet connection. Before SSH, security was limited to Telnet security. SSH allows a strong encryption to be used with the Cisco IOS software authentication. The SSH server in Cisco IOS software will work with publicly and commercially available SSH clients.

SSH Integrated Client

The SSH Integrated Client feature is an application running over the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco router to make a secure, encrypted connection to another Cisco router or to any other device running the SSH server. This connection provides functionality that is similar to that of an outbound Telnet connection except that the connection is encrypted. With authentication and encryption, the SSH client allows for a secure communication over an insecure network.

The SSH client in the Cisco IOS software works with publicly and commercially available SSH servers. The SSH client supports the ciphers of Data Encryption Standard (DES), Triple DES (3DES), and password authentication. User authentication is performed like that in the Telnet session to the router. The user authentication mechanisms supported for SSH are RADIUS, TACACS+ and the use of locally stored user names and passwords.

Note The SSH client functionality is available only when the SSH server is enabled.

Restrictions

There following are some basic SSH restrictions:

- RSA authentication available in SSH clients is not supported in the SSH server for Cisco IOS software.
- SSH server and SSH client are supported on DES (56-bit) and 3DES (168-bit) data encryption software images only. In DES software images, DES is the only encryption algorithm available. In 3DES software images, both DES and 3DES encryption algorithms are available.
- Execution shell is the only application supported.
- The login banner is not supported in Secure Shell Version 1. It is supported in Secure Shell Version 2.

Related Features and Technologies

For more information about SSH-related features and technologies, review the following:

- Authentication, Authorization, and Accounting (AAA) feature. AAA is a suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server. For more information on AAA, refer to the Authentication, Authorization, and Accounting chapters earlier in this book and the *Cisco IOS Security Command Reference*.
- IP Security (IPSec) feature. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses Internet Key Exchange (IKE) to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host. For more information on IPSec, refer to the chapter "Configuring IPSec Network Security" and the *Cisco IOS Security Command Reference*.

Prerequisites to Configuring SSH

Prior to configuring SSH, perform the following tasks:

- Download the required image on your router. (The SSH server requires you to have an IPSec (DES or 3DES) encryption software image from Cisco IOS Release 12.1(1)T downloaded on your router; the SSH client requires you to have an IPSec (DES or 3DES) encryption software image from Cisco IOS Release 12.1(3)T downloaded on your router.) For more information on downloading a software image, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*.
- Configure a host name and host domain for your router.

  To configure a host name and host domain, enter the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# hostname hostname`	Configures a host name for your router.
`Router(config)# ip domain-name domainname`	Configures a host domain for your router.

- Generate an RSA key pair for your router, which automatically enables SSH.

  To generate an RSA key pair, enter the following global configuration command:

Command	Purpose
`Router(config)# crypto key generate rsa`	Enables the SSH server for local and remote authentication on the router. The recommended minimum modulus size is 1024 bits. **Note** To delete the RSA key-pair, use the **crypto key zeroize rsa** global configuration command. Once you delete the RSA key-pair, you automatically disable the SSH server.

- Configure user authentication for local or remote access. You can configure authentication with or without AAA. For more information, refer to the "Authentication, Authorization, and Accounting (AAA)" chapters earlier in the book.

SSH Configuration Task List

The following sections describe the configuration tasks for SSH. Each task in the list is identified as either optional or required.

- Configuring SSH Server (Required)
- Verifying SSH (Optional)

See the section "SSH Configuration Examples" at the end of this chapter.

Configuring SSH Server

Note The SSH client feature runs in user EXEC mode and has no specific configuration on the router.

Note The SSH commands are optional and are disabled when the SSH server is disabled.

To enable and configure a Cisco Router for SSH, you can configure SSH parameters. If you do not configure SSH parameters, the the default values will be used.

To configure SSH server, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip ssh {[timeout seconds] | [authentication-retries integer]}`	(Required) Configures SSH control variables on your router. • You can specify the timeout in seconds, not to exceed 120 seconds. The default is 120. This setting applies to the SSH negotiation phase. Once the EXEC session starts, the standard timeouts configured for the vty apply. By default, there are 5 vtys defined (0–4), therefore 5 terminal sessions are possible. After the SSH executes a shell, the vty timeout starts. The vty timeout defaults to 10 minutes. • You can also specify the number of authentication retries, not to exceed 5 authentication retries. The default is 3.

Verifying SSH

To verify that the SSH server is enabled and view the version and configuration data for your SSH connection, use the **show ip ssh** command. The following example shows that SSH is enabled:

```
Router# show ip ssh

SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
```

The following example shows that SSH is disabled:

```
Router# show ip ssh

%SSH has not been enabled
```

To verify the status of your SSH server connections, use the **show ssh** command. The following example shows the SSH server connections on the router when SSH is enabled:

```
Router# show ssh
Connection      Version     EncryptionStateUsername
    0   1.5 3DESSession Startedguest
```

The following example shows that SSH is disabled:

```
Router# show ssh

%No SSH server connections running.
```

Troubleshooting Tips

- If your SSH configuration commands are rejected as illegal commands, you have not successfully generated a RSA key pair for your router. Make sure you have specified a host name and domain. Then use the **crypto key generate rsa** command to generate a RSA key pair and enable the SSH server.
- When configuring the RSA key pair, you might encounter the following error messages:

  - No hostname specified

    You must configure a host name for the router using the **hostname** global configuration command. For more information, see "Prerequisites to Configuring SSH."

  - No domain specified

    You must configure a host domain for the router using the **ip domain-name** global configuration command. For more information, see "Prerequisites to Configuring SSH."

- The number of allowable SSH connections is limited to the maximum number of vtys configured for the router. Each SSH connection will use a vty resource.
- SSH uses either local security or the security protocol that is configured through AAA on your router for user authentication. When configuring AAA, you must ensure that the console is not running under AAA by applying a keyword in the global configuration mode to disable AAA on the console.

Monitoring and Maintaining SSH

To monitor and maintain your SSH connections, use the following commands in user EXEC mode:

Command	Purpose
`Router# show ip ssh`	Displays the version and configuration data for SSH.
`Router# show ssh`	Displays the status of SSH server connections.

SSH Configuration Examples

This section provides the following configuration examples, which are output from the **show running configuration** EXEC command on a Cisco 7200, Cisco 7500, and Cisco 12000.

- SSH on a Cisco 7200 Series Router Example
- SSH on a Cisco 7500 Series Router Example
- SSH on a Cisco 1200 Gigabit Switch Router Example

Note The **crypto key generate rsa** command is not displayed in the **show running configuration** output.

SSH on a Cisco 7200 Series Router Example

In the following example, SSH is configured on a Cisco 7200 with a timeout that is not to exceed 60 seconds, and no more than 2 authentication retries. Also, before configuring the SSH server feature on the router, TACACS+ is specified as the method of authentication.

```
hostname Router72K
aaa new-model
aaa authentication login default tacacs+
aaa authentication login aaa7200kw none
enable password enable7200pw

username mcisco password 0 maryspw
username jcisco password 0 johnspw
```

```
ip subnet-zero
no ip domain-lookup
ip domain-name cisco.com
! Enter the ssh commands.
ip ssh time-out 60
ip ssh authentication-retries 2

controller E1 2/0

controller E1 2/1

interface Ethernet1/0
ip address 192.168.110.2 255.255.255.0 secondary
ip address 192.168.109.2 255.255.255.0
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
no keepalive
no cdp enable

interface Ethernet1/1
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
no cdp enable

interface Ethernet1/2
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
no cdp enable

no ip classless
ip route 192.168.1.0 255.255.255.0 10.1.10.1
ip route 192.168.9.0 255.255.255.0 10.1.1.1
ip route 192.168.10.0 255.255.255.0 10.1.1.1

map-list atm
ip 10.1.10.1 atm-vc 7 broadcast
no cdp run

tacacs-server host 192.168.109.216 port 9000
tacacs-server key cisco
radius-server host 192.168.109.216 auth-port 1650 acct-port 1651
radius-server key cisco

line con 0
exec-timeout 0 0
login authentication aaa7200kw
transport input none
line aux 0
line vty 0 4
password enable7200pw

end
```

SSH on a Cisco 7500 Series Router Example

In the following example, SSH is configured on a Cisco 7500 with a timeout that is not to exceed 60 seconds and no more than 5 authentication retries. Before the SSH Server feature is configured on the router, RADIUS is specified as the method of authentication.

```
hostname Router75K
aaa new-model
aaa authentication login default radius
aaa authentication login aaa7500kw none
enable password enable7500pw

username mcisco password 0 maryspw
username jcisco password 0 johnspw
ip subnet-zero
no ip cef
no ip domain-lookup
ip domain-name cisco.com
! Enter ssh commands.
ip ssh time-out 60
ip ssh authentication-retries 5

controller E1 3/0
channel-group 0 timeslots 1

controller E1 3/1
channel-group 0 timeslots 1
channel-group 1 timeslots 2

interface Ethernet0/0/0
no ip address
no ip directed-broadcast
no ip route-cache distributed
shutdown

interface Ethernet0/0/1
no ip address
no ip directed-broadcast
no ip route-cache distributed
shutdown
interface Ethernet0/0/2
no ip address
no ip directed-broadcast
no ip route-cache distributed
shutdown

interface Ethernet0/0/3
no ip address
no ip directed-broadcast
no ip route-cache distributed
shutdown

interface Ethernet1/0
ip address 192.168.110.2 255.255.255.0 secondary
ip address 192.168.109.2 255.255.255.0
no ip directed-broadcast
no ip route-cache
no ip mroute-cache

interface Ethernet1/1
ip address 192.168.109.2 255.255.255.0
no ip directed-broadcast
no ip route-cache
```

```
no ip mroute-cache
shutdown

interface Ethernet1/2
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache

interface Ethernet1/3
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown

interface Ethernet1/4
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
interface Ethernet1/5
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown

interface Serial2/0
ip address 10.1.1.2 255.0.0.0
no ip directed-broadcast
encapsulation ppp
no ip route-cache
no ip mroute-cache

ip classless
ip route 192.168.9.0 255.255.255.0 10.1.1.1
ip route 192.168.10.0 255.255.255.0 10.1.1.1

tacacs-server host 192.168.109.216 port 9000
tacacs-server key cisco
radius-server host 192.168.109.216 auth-port 1650 acct-port 1651
radius-server key cisco

line con 0
exec-timeout 0 0
login authentication aaa7500kw
transport input none
line aux 0
transport input all
line vty 0 4

end
```

SSH on a Cisco 1200 Gigabit Switch Router Example

In the following example, SSH is configured on a Cisco 12000 with a timeout that is not to exceed 60 seconds and no more than 2 authentication retries. Before the SSH Server feature is configured on the router, TACACS+ is specified as the method of authentication.

```
hostname Router12K
aaa new-model
aaa authentication login default tacacs+ local
aaa authentication login aaa12000kw local
enable password enable12000pw

username mcisco password 0 maryspw
username jcisco password 0 johnspw
redundancy
main-cpu
    auto-sync startup-config
ip subnet-zero
no ip domain-lookup
ip domain-name cisco.com
! Enter ssh commands.
ip ssh time-out 60
ip ssh authentication-retries 2

interface ATM0/0
no ip address
no ip directed-broadcast
no ip route-cache cef
shutdown

interface POS1/0
ip address 10.100.100.2 255.255.255.0
no ip directed-broadcast
encapsulation ppp
no ip route-cache cef
no keepalive
crc 16
no cdp enable

interface POS1/1
no ip address
no ip directed-broadcast
no ip route-cache cef
shutdown
crc 32

interface POS1/2
no ip address
no ip directed-broadcast
no ip route-cache cef
shutdown
crc 32

interface POS1/3
no ip address
no ip directed-broadcast
no ip route-cache cef
shutdown
crc 32

interface POS2/0
ip address 10.1.1.1 255.255.255.0
no ip directed-broadcast
encapsulation ppp
no ip route-cache cef
crc 16

interface Ethernet0
ip address 172.17.110.91 255.255.255.224
no ip directed-broadcast
```

```
router ospf 1
network 0.0.0.0 255.255.255.255 area 0.0.0.0

ip classless
ip route 0.0.0.0 0.0.0.0 172.17.110.65

logging trap debugging
tacacs-server host 172.17.116.138
tacacs-server key cisco

radius-server host 172.17.116.138 auth-port 1650 acct-port 1651
radius-server key cisco

line con 0
exec-timeout 0 0
login authentication aaa12000kw
transport input none
line aux 0
line vty 0 4

no scheduler max-task-time
no exception linecard slot 0 sqe-registers
no exception linecard slot 1 sqe-registers
no exception linecard slot 2 sqe-registers
no exception linecard slot 3 sqe-registers
no exception linecard slot 4 sqe-registers
no exception linecard slot 5 sqe-registers
no exception linecard slot 6 sqe-registers
end
```

Secure Copy

The Secure Copy (SCP) feature provides a secure and authenticated method for copying router configuration or router image files. SCP relies on Secure Shell (SSH), an application and a protocol that provide a secure replacement for the Berkeley r-tools.

Feature History for Secure Copy

Release	Modification
12.2(2)T	This feature was introduced.
12.0(21)S	This feature was integrated into Cisco IOS 12.0(21)S.
12.2(25)S	This feature was integrated into Cisco IOS 12.2(25)S.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Secure Copy

- Before enabling SCP, you must correctly configure SSH, authentication, and authorization on the router.
- Because SCP relies on SSH for its secure transport, the router must have an Rivest, Shamir, and Adelman (RSA) key pair.

Information About Secure Copy

To configure Secure Copy feature, you should understand the following concepts.

- How SCP Works, page 2

How SCP Works

The behavior of SCP is similar to that of remote copy (rcp), which comes from the Berkeley r-tools suite, except that SCP relies on SSH for security. In addition, SCP requires that authentication, authorization, and accounting (AAA) authorization be configured so the router can determine whether the user has the correct privilege level.

SCP allows a user who has appropriate authorization to copy any file that exists in the Cisco IOS File System (IFS) to and from a router by using the **copy** command. An authorized administrator may also perform this action from a workstation.

How to Configure SCP

This section contains the following procedures:

- Configuring SCP, page 2
- Verifying SCP, page 3
- Troubleshooting SCP, page 4

Configuring SCP

To enable and configure a Cisco router for SCP server-side functionality, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** {**default** | *list-name*} *method1* [*method2...*]
5. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [*method1* [*method2...*]]
6. **username** *name* [**privilege** *level]* {**password** *encryption-type encrypted-password}*

7. **ip scp server enable**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Sets AAA authentication at login.
Step 4	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router (config)# aaa authentication login default group tacacs+`	Enables the AAA access control system.
Step 5	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access | configuration**} {**default** | *list-name*} [*method1* [*method2...*]] **Example:** `Router (config)# aaa authorization exec default group tacacs+`	Sets parameters that restrict user access to a network. **Note** **The exec** keyword runs authorization to determine if the user is allowed to run an EXEC shell; therefore, you must use it when you configure SCP.
Step 6	**username** *name* [**privilege** *level] {***password** *encryption-type encrypted-password}* **Example:** `Router (config)# username superuser privilege 2 password 0 superpassword`	Establishes a username-based authentication system. **Note** You may skip this step if a network-based authentication mechanism—such as TACACS+ or RADIUS—has been configured.
Step 7	`ip scp server enable` **Example:** `Router (config)# ip scp server enable`	Enables SCP server-side functionality.

Verifying SCP

To verify SCP server-side functionality, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`show running-config` **Example:** `Router# show running-config`	Verifies the SCP server-side functionality.

Troubleshooting SCP

To troubleshoot SCP authentication problems, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug ip scp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`debug ip scp` **Example:** `Router# debug ip scp`	Troubleshoots SCP authentication problems.

Configuration Examples for Secure Copy

This section provides the following configuration examples:

- SCP Server-Side Configuration Using Local Authentication: Example, page 5
- SCP Server-Side Configuration Using Network-Based Authentication: Example, page 5

SCP Server-Side Configuration Using Local Authentication: Example

The following example shows how to configure the server-side functionality of SCP. This example uses a locally defined username and password.

```
! AAA authentication and authorization must be configured properly for SCP to work.
aaa new-model
aaa authentication login default local
aaa authorization exec default local
username tiger privilege 15 password 0 lab
! SSH must be configured and functioning properly.
ip ssh time-out 120
ip ssh authentication-retries 3
ip scp server enable
```

SCP Server-Side Configuration Using Network-Based Authentication: Example

The following example shows how to configure the server-side functionality of SCP using a network-based authentication mechanism:

```
! AAA authentication and authorization must be configured properly for SCP to work.
aaa new-model
aaa authentication login default group tacacs+
aaa authorization exec default group tacacs+
! SSH must be configured and functioning properly.
ip ssh time-out 120
ip ssh authentication-retries 3
ip scp server enable
```

Additional References

The following sections provide references related to Secure Copy.

Related Documents

Related Topic	Document Title
Secure Shell	• *Secure Shell Version 1 Support* • *Secure Shell Version 2 Support*
Authentication and authorization commands	*Cisco IOS Security Command Reference*, Release 12.3 T
Configuring authentication and authorization	"Authentication, Authorization, and Accounting (AAA)" section of Cisco IOS Security Configuration Guide, Release 12.3

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **debug ip scp**
- **ip scp server enable**

Glossary

AAA—authentication, authorization, and accounting. Framework of security services that provide the method for identifying users (authentication), for remote access control (authorization), and for collecting and sending security server information used for billing, auditing, and reporting (accounting).

rcp—remote copy. Relying on Remote Shell (Berkeley r-tools suite) for security, rcp copies files, such as router images and startup configurations, to and from routers.

SCP—secure copy. Relying on SSH for security, SCP support allows the secure and authenticated copying of anything that exists in the Cisco IOS File Systems. SCP is derived from rcp.

SSH—Secure Shell. Application and a protocol that provide a secure replacement for the Berkeley r-tools. The protocol secures the sessions using standard cryptographic mechanisms, and the application can be used similarly to the Berkeley rexec and rsh tools. SSH Version 1 is implemented in the Cisco IOS software.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Secure Shell Version 2 Support

First Published: November 3, 2003
Last Updated: July 11, 2008

The Secure Shell Version 2 Support feature allows you to configure Secure Shell (SSH) Version 2 (SSH Version 1 support was implemented in an earlier Cisco IOS software release). SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. Currently, the only reliable transport that is defined for SSH is TCP. SSH provides a means to securely access and securely execute commands on another computer over a network. The Secure Copy Protocol (SCP) feature that is provided with SSH also allows for the secure transfer of files.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Secure Shell Version 2 Support" section on page 20.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Secure Shell Version 2 Support

Prior to configuring SSH, perform the following task:

- Download the required image on your router. The SSH server requires you to have a k9 (Triple Data Encryption Standard [3DES]) software image from Cisco IOS Release 12.3(4)T, 12.2(25)S, or 12.3(7)JA downloaded on your router.

Note The SSH Version 2 server is supported in Cisco IOS Release 12.3(4)T, 12.3(2)XE, 12.2(25)S, and 12.3(7)JA; the SSH Version 2 client is supported beginning with Cisco IOS Release 12.3(7)T and is supported in Cisco IOS Release12.3(7)JA. (The SSH client runs both the SSH Version 1 and Version 2 protocol and is supported in both k8 and k9 images in Cisco IOS Release 12.3(4)T.)

For more information on downloading a software image, refer to *Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*.

Restrictions for Secure Shell Version 2 Support

- Rivest, Shamir, and Adelman (RSA) user authentication is not supported in the SSH server or SSH client for Cisco IOS software.
- SSH servers and SSH clients are supported in 3DES software images.
- Execution Shell, remote command execution, and Secure Copy Protocol (SCP) are the only applications supported.
- Compression is not supported.
- The RSA key-pair size must be greater than or equal to 768.

Information About Secure Shell Version 2 Support

To configure SSH Version 2, you should understand the following concept:

- Secure Shell Version 2, page 2
- SNMP Trap Generation, page 3
- SSH Keyboard Interactive Authentication, page 4

Secure Shell Version 2

The Secure Shell Version 2 Support feature allows you to configure SSH Version 2.

The configuration for the SSH Version 2 server is similar to the configuration for SSH Version 1. The command **ip ssh version** has been introduced so that you may define which version of SSH that you want to configure. If you do not configure this command, SSH by default runs in compatibility mode; that is, both SSH Version 1 and SSH Version 2 connections are honored.

Note SSH Version 1 is a protocol that has never been defined in a standard. If you do not want your router to fall back to the undefined protocol (Version 1), you should use the **ip ssh version** command and specify Version 2.

The **ip ssh rsa keypair-name** command was also introduced in Cisco IOS Release 12.3(4)T so that you can enable a SSH connection using RSA keys that you have configured. Previously, SSH was linked to the first RSA keys that were generated (that is, SSH was enabled when the first RSA key pair was generated). The behavior still exists, but by using the **ip ssh rsa keypair-name** command, you can overcome that behavior. If you configure the **ip ssh rsa keypair-name** command with a key-pair name, SSH is enabled if the key pair exists, or SSH will be enabled if the key pair is generated later. If you use this command to enable SSH, you are not forced to configure a host name and a domain name, which was required in SSH Version 1 of the Cisco IOS software.

Note The login banner is supported in Secure Shell Version 2, but it is not supported in Secure Shell Version 1.

Secure Shell Version 2 Enhancements

The Secure Shell Version 2 Enhancements include a number of additional capabilities such as supporting VRF aware SSH, SSH debug enhancements, and Diffie-Hellman group exchange support.

The Cisco IOS SSH implementation has traditionally used 768 bit modulus but with an increasing need for higher key sizes to accommodate Diffie-Hellman (DH) Group 14 (2048 bits) and Group 16 (4096 bits) cryptographic applications a message exchange between the client and server to establish the favored DH group becomes necessary. The **ip ssh dh min size** command was introduced in Cisco IOS Release 12.4(20)T so you can configure modulus size on the SSH server. In addition to this the **ssh** command was extended to add VRF awareness to SSH client side functionality through which the VRF instance name in the client is provided with the IP address to look up the correct routing table and establish a connection.

Debugging has been enhanced by modifying SSH debug commands. The **debug ip ssh** command has been extended to allow you to simplify the debugging process. Previously this command printed all debug messages related to SSH regardless of what was specifically required. The behavior still exists, but if you configure the **debug ip ssh** command with a keyword messages are limited to information specified by the keyword.

SNMP Trap Generation

Effective with Cisco IOS Release 12.4(17), Simple Network Management Protocol (SNMP) traps will be generated automatically when an SSH session terminates if the traps have been enabled and SNMP debugging has been turned on. For information about enabling SNMP traps, see the chapter "Configuring SNMP Support" in the *Cisco IOS Network Management Configuration Guide*.

Note When configuring the **snmp-server host** command, the IP address must be the address of the PC that has the SSH (telnet) client and that has IP connectivity to the SSH server. For an example of an SNMP trap generation configuration, see the section "Setting an SNMP Trap: Example."

You must also turn on SNMP debugging using the **debug snmp packet** command to display the traps. The trap information includes information such as the number of bytes sent and the protocol that was used for the SSH session. For an example of SNMP debugging, see the section "SNMP Debugging: Example."

SSH Keyboard Interactive Authentication

The SSH Keyboard Interactive Authentication feature, also known as Generic Message Authentication for SSH, is a method that can be used to implement different types of authentication mechanisms. Basically, any currently supported authentication method that requires only user input can be performed with this feature. The feature is automatically deployed.

The following methods are currently supported:

- Password
- SecurID and hardware tokens printing a number or a string in response to a challenge sent by the server
- Pluggable Authentication Module (PAM)
- S/KEY (and other One-Time-Pads)

For examples of various scenarios in which the SSH Keyboard Interactive Authentication feature has been automatically deployed, see the chapter "SSH Keyboard Interactive Authentication: Examples."

How to Configure Secure Shell Version 2 Support

This section contains the following procedures:

Configuring a Router for SSH Version 2 Using a Host Name and Domain Name

To configure your router for SSH Version 2 using a host name and domain name, perform the following steps. You may also configure SSH Version 2 by using the RSA key pair configuration (See the section "Configuring a Router for SSH Version 2 Using RSA Key Pairs").

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *hostname*
4. **ip domain-name** *name*

5. **crypto key generate rsa**
6. **ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*]
7. **ip ssh version** [**1** | *2*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**hostname** *hostname* **Example:** `Router (config)# hostname cisco 7200`	Configures a host name for your router.
Step 4	**ip domain-name** *name* **Example:** `Router (config)# ip domain-name cisco.com`	Configures a domain name for your router.
Step 5	**crypto key generate rsa** **Example:** `Router (config)# crypto key generate rsa`	Enables the SSH server for local and remote authentication.
Step 6	**ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*] **Example:** `Router (config)# ip ssh timeout 120`	(Optional) Configures SSH control variables on your router.
Step 7	**ip ssh version** [**1** | **2**] **Example:** `Router (config)# ip ssh version 1`	(Optional) Specifies the version of SSH to be run on your router.

Configuring a Router for SSH Version 2 Using RSA Key Pairs

To enable SSH Version 2 without configuring a host name or domain name, perform the following steps. SSH Version 2 will be enabled if the key pair that you configure already exists or if it is generated later. You may also configure SSH Version 2 by using the host name and domain name configuration (See the section "Configuring a Router for SSH Version 2 Using a Host Name and Domain Name").

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh rsa keypair-name** *keypair-name*
4. **crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size*
5. **ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*]
6. **ip ssh version 2**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ssh rsa keypair-name** *keypair-name* **Example:** `Router (config)# ip ssh rsa keypair-name sshkeys`	Specifies which RSA keypair to use for SSH usage. **Note** A Cisco IOS router can have many RSA key pairs.
Step 4	**crypto key generate rsa usage-keys label** ***key-label*** **modulus** *modulus-size* **Example:** `Router (config)# crypto key generate rsa usage-keys label sshkeys modulus 768`	Enables the SSH server for local and remote authentication on the router. For SSH Version 2, the modulus size must be at least 768 bits. **Note** To delete the RSA key-pair, use the **crypto key zeroize rsa** command. After you have deleted the RSA key-pair, you automatically disable the SSH server.
Step 5	**ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*] **Example:** `Router (config)# ip ssh timeout 120`	Configures SSH control variables on your router.
Step 6	**ip ssh version 2** **Example:** `Router (config)# ip ssh version 2`	Specifies the version of SSH to be run on a router.

Starting an Encrypted Session with a Remote Device

To start an encrypted session with a remote networking device, perform the following step. (You do not have to enable your router. SSH can be run in disabled mode.)

Note The device you wish to connect with must support a SSH server that has an encryption algorithm that is supported in Cisco IOS software.

SUMMARY STEPS

1. **ssh** [**-v** {**1** | **2**}] [**-c** {**3des** | **aes128-cbc** | **aes192-cbc** | **aes256-cbc**}] [**-m** {**hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96**}] [**l** *userid*] [**-o numberofpasswordprompts** *n*] [**-p** *port-num*] {*ip-addr* | *hostname*} [*command*]

DETAILED STEPS

Step 1	**ssh** [**-v** {**1 | 2**}] [**-c** {**3des** | **aes128-cbc** | **aes192-cbc** | **aes256-cbc**}] [**-m** {**hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96**}] [**l** *userid*] [**-o numberofpasswordprompts** *n*] [**-p** *port-num*] {*ip-addr* | *hostname*} [*command*] **Example:** `Router# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 -l user2 10.76.82.24` Or The above example adheres to the SSH Version 2 conventions. A more natural and common way to start a session is by linking the username with the hostname. For example, the following configuration example provides an end result that is identical to that of the above example: `Router# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 user2@10.76.82.24`	Starts an encrypted session with a remote networking device.

Troubleshooting Tips

The **ip ssh version** command can be used for troubleshooting your SSH configuration. By changing versions, you can determine which SSH version has a problem.

Verifying the Status of the Secure Shell Connection Using the show ssh Command

To display the status of the SSH connection on your router, use the **show ssh** command.

SUMMARY STEPS

1. **enable**
2. **show ssh**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ssh** **Example:** `Router# show ssh`	Displays the status of SSH server connections.

Examples

The following output examples from the **show ssh** command display status about various SSH Version 1 and Version 2 connections.

Version 1 and Version 2 Connections

```
------------------------------------------------------------------------
Router# show ssh

Connection      Version Encryption      State                   Username
 0              1.5     3DES            Session started         lab
Connection Version Mode Encryption  Hmac               State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
------------------------------------------------------------------------
```

Version 2 Connection with No Version 1

```
------------------------------------------------------------------------
Router# show ssh

Connection Version Mode Encryption  Hmac               State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
%No SSHv1 server connections running.
------------------------------------------------------------------------
```

Version 1 Connection with No Version 2

```
------------------------------------------------------------------------
Router# show ssh

Connection      Version Encryption      State                   Username
 0              1.5     3DES            Session started         lab
%No SSHv2 server connections running.
------------------------------------------------------------------------
```

Verifying the Secure Shell Status Using the show ip ssh Command

To verify your SSH configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show ip ssh**

DETAILED STEPS

Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`show ip ssh` **Example:** `Router# show ip ssh`	Displays the version and configuration data for SSH.

Examples

The following examples from the **show ip ssh** command display the version of SSH that is enabled, the authentication timeout values, and the number of authentication retries.

Version 1 and Version 2 Connections

```
---------------------------------------------------------------------
router# show ip ssh

3d06h: %SYS-5-CONFIG_I: Configured from console by consoleh
SSH Enabled - version 1.99
Authentication timeout: 120 secs; Authentication retries: 3
---------------------------------------------------------------------
```

Version 2 Connection with No Version 1

```
---------------------------------------------------------------------
Router# show ip ssh

SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
---------------------------------------------------------------------
```

Version 1 Connection with No Version 2

```
---------------------------------------------------------------------
Router# show ip ssh

3d06h: %SYS-5-CONFIG_I: Configured from console by console
SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
---------------------------------------------------------------------
```

Monitoring and Maintaining Secure Shell Version 2

To display debug messages about the SSH connections, use the **debug ip ssh** command.

SUMMARY STEPS

1. **enable**
2. **debug ip ssh**
3. **debug snmp packet**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip ssh** **Example:** `Router# debug ip ssh`	Displays debugging messages for SSH.
Step 3	**debug snmp packet** **Example:** `Router# debug snmp packet`	Displays information about every SNMP packet sent or received by the router.

Example

The following output from the **debug ip ssh** command shows that the digit 2 keyword has been assigned, signifying that it is an SSH Version 2 connection.

```
Router# debug ip ssh

00:33:55: SSH1: starting SSH control process
00:33:55: SSH1: sent protocol version id SSH-1.99-Cisco-1.25
00:33:55: SSH1: protocol version id is - SSH-2.0-OpenSSH_2.5.2p2
00:33:55: SSH2 1: send: len 280 (includes padlen 4)
00:33:55: SSH2 1: SSH2_MSG_KEXINIT sent
00:33:55: SSH2 1: ssh_receive: 536 bytes received
00:33:55: SSH2 1: input: packet len 632
00:33:55: SSH2 1: partial packet 8, need 624, maclen 0
00:33:55: SSH2 1: ssh_receive: 96 bytes received
00:33:55: SSH2 1: partial packet 8, need 624, maclen 0
00:33:55: SSH2 1: input: padlen 11
00:33:55: SSH2 1: received packet type 20
00:33:55: SSH2 1: SSH2_MSG_KEXINIT received
00:33:55: SSH2: kex: client->server aes128-cbc hmac-md5 none
00:33:55: SSH2: kex: server->client aes128-cbc hmac-md5 none
00:33:55: SSH2 1: expecting SSH2_MSG_KEXDH_INIT
00:33:55: SSH2 1: ssh_receive: 144 bytes received
00:33:55: SSH2 1: input: packet len 144
00:33:55: SSH2 1: partial packet 8, need 136, maclen 0
00:33:55: SSH2 1: input: padlen 5
00:33:55: SSH2 1: received packet type 30
```

```
00:33:55: SSH2 1: SSH2_MSG_KEXDH_INIT received
00:33:55: SSH2 1: signature length 111
00:33:55: SSH2 1: send: len 384 (includes padlen 7)
00:33:55: SSH2: kex_derive_keys complete
00:33:55: SSH2 1: send: len 16 (includes padlen 10)
00:33:55: SSH2 1: newkeys: mode 1
00:33:55: SSH2 1: SSH2_MSG_NEWKEYS sent
00:33:55: SSH2 1: waiting for SSH2_MSG_NEWKEYS
00:33:55: SSH2 1: ssh_receive: 16 bytes received
00:33:55: SSH2 1: input: packet len 16
00:33:55: SSH2 1: partial packet 8, need 8, maclen 0
00:33:55: SSH2 1: input: padlen 10
00:33:55: SSH2 1: newkeys: mode 0
00:33:55: SSH2 1: received packet type 2100:33:55: SSH2 1: SSH2_MSG_NEWKEYS received
00:33:56: SSH2 1: ssh_receive: 48 bytes received
00:33:56: SSH2 1: input: packet len 32
00:33:56: SSH2 1: partial packet 16, need 16, maclen 16
00:33:56: SSH2 1: MAC #3 ok
00:33:56: SSH2 1: input: padlen 10
00:33:56: SSH2 1: received packet type 5
00:33:56: SSH2 1: send: len 32 (includes padlen 10)
00:33:56: SSH2 1: done calc MAC out #3
00:33:56: SSH2 1: ssh_receive: 64 bytes received
00:33:56: SSH2 1: input: packet len 48
00:33:56: SSH2 1: partial packet 16, need 32, maclen 16
00:33:56: SSH2 1: MAC #4 ok
00:33:56: SSH2 1: input: padlen 9
00:33:56: SSH2 1: received packet type 50
00:33:56: SSH2 1: send: len 32 (includes padlen 13)
00:33:56: SSH2 1: done calc MAC out #4
00:34:04: SSH2 1: ssh_receive: 160 bytes received
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #5 ok
00:34:04: SSH2 1: input: padlen 13
00:34:04: SSH2 1: received packet type 50
00:34:04: SSH2 1: send: len 16 (includes padlen 10)
00:34:04: SSH2 1: done calc MAC out #5
00:34:04: SSH2 1: authentication successful for lab
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #6 ok
00:34:04: SSH2 1: input: padlen 6
00:34:04: SSH2 1: received packet type 2
00:34:04: SSH2 1: ssh_receive: 64 bytes received
00:34:04: SSH2 1: input: packet len 48
00:34:04: SSH2 1: partial packet 16, need 32, maclen 16
00:34:04: SSH2 1: MAC #7 ok
00:34:04: SSH2 1: input: padlen 19
00:34:04: SSH2 1: received packet type 90
00:34:04: SSH2 1: channel open request
00:34:04: SSH2 1: send: len 32 (includes padlen 10)
00:34:04: SSH2 1: done calc MAC out #6
00:34:04: SSH2 1: ssh_receive: 192 bytes received
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #8 ok
00:34:04: SSH2 1: input: padlen 13
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: pty-req request
00:34:04: SSH2 1: setting TTY - requested: height 24, width 80; set: height 24,
width 80
00:34:04: SSH2 1: input: packet len 96
00:34:04: SSH2 1: partial packet 16, need 80, maclen 16
```

```
00:34:04: SSH2 1: MAC #9 ok
00:34:04: SSH2 1: input: padlen 11
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: x11-req request
00:34:04: SSH2 1: ssh_receive: 48 bytes received
00:34:04: SSH2 1: input: packet len 32
00:34:04: SSH2 1: partial packet 16, need 16, maclen 16
00:34:04: SSH2 1: MAC #10 ok
00:34:04: SSH2 1: input: padlen 12
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: shell request
00:34:04: SSH2 1: shell message received
00:34:04: SSH2 1: starting shell for vty
00:34:04: SSH2 1: send: len 48 (includes padlen 18)
00:34:04: SSH2 1: done calc MAC out #7
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #11 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #8
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #12 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #9
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #13 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #10
00:34:08: SSH2 1: ssh_receive: 48 bytes received
00:34:08: SSH2 1: input: packet len 32
00:34:08: SSH2 1: partial packet 16, need 16, maclen 16
00:34:08: SSH2 1: MAC #14 ok
00:34:08: SSH2 1: input: padlen 17
00:34:08: SSH2 1: received packet type 94
00:34:08: SSH2 1: send: len 32 (includes padlen 17)
00:34:08: SSH2 1: done calc MAC out #11
00:34:08: SSH2 1: ssh_receive: 48 bytes received
00:34:08: SSH2 1: input: packet len 32
00:34:08: SSH2 1: partial packet 16, need 16, maclen 16
00:34:08: SSH2 1: MAC #15 ok
00:34:08: SSH2 1: input: padlen 17
00:34:08: SSH2 1: received packet type 94
00:34:08: SSH2 1: send: len 32 (includes padlen 16)
00:34:08: SSH2 1: done calc MAC out #12
00:34:08: SSH2 1: send: len 48 (includes padlen 18)
00:34:08: SSH2 1: done calc MAC out #13
00:34:08: SSH2 1: send: len 16 (includes padlen 6)
00:34:08: SSH2 1: done calc MAC out #14
00:34:08: SSH2 1: send: len 16 (includes padlen 6)
00:34:08: SSH2 1: done calc MAC out #15
00:34:08: SSH1: Session terminated normally
```

Configuration Examples for Secure Shell Version 2 Support

This section provides the following configuration examples:

Configuring Secure Shell Version 1: Example

```
Router# configure terminal
Router (config)# ip ssh version 1
c7200-25-2013(config)# end
```

Configuring Secure Shell Version 2: Example

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip ssh version 2
Router(config)# end
```

Configuring Secure Shell Versions 1 and 2: Example

```
Router# configure terminal
Router (config)# no ip ssh version
Router (config)# end
```

Starting an Encrypted Session with a Remote Device: Example

```
Router# ssh -v 2 -c aes256-cbc -m hmac-sha1-160 -l shaship 10.76.82.24
```

Setting an SNMP Trap: Example

The following shows that an SNMP trap has been set. The trap notification is generated automatically when the SSH session terminates. For an example of SNMP trap debug output, see the section "SNMP Debugging: Example."

```
snmp-server
snmp-server host a.b.c.d public tty
```

Where a.b.c.d is the IP address of the SSH client.

SSH Keyboard Interactive Authentication: Examples

The following are examples of various scenarios in which the SSH Keyboard Interactive Authentication feature has been automatically deployed:

Client-Side Debugs

In the following example, client-side debugs are turned on and the maximum number of prompts = six, (three each for the SSH Keyboard Interactive Authentication method and for the password method of authentication).

```
Password:
Password:
Password:
Password:
Password:
Password: cisco123
Last login: Tue Dec 6 13:15:21 2005 from 10.76.248.213
user1@courier:~> exit
logout
[Connection to 10.76.248.200 closed by foreign host]

Router1# debug ip ssh client

SSH Client debugging is on

Router1# ssh -l lab 10.1.1.3
Password:
*Nov 17 12:50:53.199: SSH0: sent protocol version id SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.199: SSH CLIENT0: protocol version id is - SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.199: SSH CLIENT0: sent protocol version id SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.199: SSH CLIENT0: protocol version exchange successful
*Nov 17 12:50:53.203: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.335: SSH CLIENT0: key exchange successful and encryption on
*Nov 17 12:50:53.335: SSH2 CLIENT 0: using method keyboard-interactive
Password:
Password:
Password:
*Nov 17 12:51:01.887: SSH2 CLIENT 0: using method password authentication
Password:
Password: lab

Router2>
*Nov 17 12:51:11.407: SSH2 CLIENT 0: SSH2_MSG_USERAUTH_SUCCESS message received
*Nov 17 12:51:11.407: SSH CLIENT0: user authenticated
*Nov 17 12:51:11.407: SSH2 CLIENT 0: pty-req request sent
*Nov 17 12:51:11.411: SSH2 CLIENT 0: shell request sent
*Nov 17 12:51:11.411: SSH CLIENT0: session open
```

TACACS+ ACS Is the Backend AAA Server, ChPass Is Enabled, and a Blank Password Change Is Made

In the following example, a TACACS+ access control server (ACS) is the backend Accounting, Authentication, and Authorization (AAA) server; the ChPass feature is enabled; and a blank password change is accomplished using the SSH Keyboard Interactive Authentication method:

```
Router1# ssh -l cisco 10.1.1.3
Password:
Old Password: cisco
New Password: cisco123
Re-enter New password: cisco123

Router2> exit
[Connection to 10.1.1.3 closed by foreign host]
```

TACACS+ ACS Is the Backend AAA Server, ChPass Is Enabled, and the Password Is Changed on First Login

In the following example, a TACACS+ ACS is the backend server, and the ChPass feature is enabled. The password is changed on the first login using the SSH Keyboard Interactive Authentication method:

```
Router1# ssh -l cisco 10.1.1.3
Password: cisco
Your password has expired.
Enter a new one now.
New Password: cisco123
Re-enter New password: cisco123

Router2> exit
[Connection to 10.1.1.3 closed by foreign host]

Router1# ssh -l cisco 10.1.1.3
Password:cisco1
Your password has expired.
Enter a new one now.
New Password: cisco
Re-enter New password: cisco12
The New and Re-entered passwords have to be the same.
Try again.
New Password: cisco
Re-enter New password: cisco

Router2>
```

TACACS+ ACS Is the Backend AAA Server, ChPass Is Enabled, and the Password Expires After Three Logins

In the following example, a TACACS+ ACS is the backend AAA server, and the ChPass feature is enabled. The password expires after three logins using the SSH Keyboard Interactive Authentication method:

```
Router# ssh -l cisco. 10.1.1.3
Password: cisco

Router2> exit
[Connection to 10.1.1.3 closed by foreign host]

Router1# ssh -l cisco 10.1.1.3
Password: cisco

Router2> exit

Router1# ssh -l cisco 10.1.1.3
Password: cisco

Router2> exit
[Connection to 10.1.1.3 closed by foreign host]

Router1# ssh -l cisco 10.1.1.3
Password: cisco
Your password has expired.
Enter a new one now.
New Password: cisco123
Re-enter New password: cisco123

Router2>
```

SNMP Debugging: Example

The following is sample output using the **debug snmp packet** command. The output provides SNMP trap information for an SSH session.

```
Router1# debug snmp packet

SNMP packet debugging is on

Router1# ssh -l lab 10.0.0.2

Password:

Router2# exit

[Connection to 10.0.0.2 closed by foreign host]
Router1#
*Jul 18 10:18:42.619: SNMP: Queuing packet to 10.0.0.2
*Jul 18 10:18:42.619: SNMP: V1 Trap, ent cisco, addr 10.0.0.1, gentrap 6, spectrap 1
local.9.3.1.1.2.1 = 6
tcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 4
ltcpConnEntry.5.10.0.0.1.22.10.0.0.2.55246 = 1015
ltcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 1056
ltcpConnEntry.2.10.0.0.1.22.10.0.0.2.55246 = 1392
local.9.2.1.18.2 = lab
*Jul 18 10:18:42.879: SNMP: Packet sent via UDP to 10.0.0.2
Router1#
```

SSH Debugging Enhancements: Examples

The following is sample output from the **debug ip ssh detail** command. The output provides debugging information regarding the SSH protocol and channel requests.

```
Router# debug ip ssh detail

00:04:22: SSH0: starting SSH control process
00:04:22: SSH0: sent protocol version id SSH-1.99-Cisco-1.25
00:04:22: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
00:04:22: SSH2 0: SSH2_MSG_KEXINIT sent
00:04:22: SSH2 0: SSH2_MSG_KEXINIT received
00:04:22: SSH2:kex: client->server enc:aes128-cbc mac:hmac-sha1
00:04:22: SSH2:kex: server->client enc:aes128-cbc mac:hmac-sha1
00:04:22: SSH2 0: expecting SSH2_MSG_KEXDH_INIT
00:04:22: SSH2 0: SSH2_MSG_KEXDH_INIT received
00:04:22: SSH2: kex_derive_keys complete
00:04:22: SSH2 0: SSH2_MSG_NEWKEYS sent
00:04:22: SSH2 0: waiting for SSH2_MSG_NEWKEYS
00:04:22: SSH2 0: SSH2_MSG_NEWKEYS received
00:04:24: SSH2 0: authentication successful for lab
00:04:24: SSH2 0: channel open request
00:04:24: SSH2 0: pty-req request
00:04:24: SSH2 0: setting TTY - requested: height 24, width 80; set: height 24, width 80
00:04:24: SSH2 0: shell request
00:04:24: SSH2 0: shell message received
00:04:24: SSH2 0: starting shell for vty
00:04:38: SSH0: Session terminated normally
```

The following is sample output from the **debug ip ssh packet** command.The output provides debugging information regarding the ssh packet.

```
Router# debug ip ssh packet

00:05:43: SSH2 0: send:packet of  length 280 (length also includes padlen of 4)
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: input: total packet length of 280 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 24 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 4 bytes
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: input: total packet length of 144 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 16 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 6 bytes
00:05:43: SSH2 0: signature length 143
00:05:43: SSH2 0: send:packet of  length 448 (length also includes padlen of 7)
00:05:43: SSH2 0: send:packet of  length 16 (length also includes padlen of 10)
00:05:43: SSH2 0: newkeys: mode 1
00:05:43: SSH2 0: ssh_receive: 16 bytes received
00:05:43: SSH2 0: input: total packet length of 16 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 8 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 10 bytes
00:05:43: SSH2 0: newkeys: mode 0
00:05:43: SSH2 0: ssh_receive: 52 bytes received
00:05:43: SSH2 0: input: total packet length of 32 bytes
00:05:43: SSH2 0: partial packet length(block size)16 bytes,needed 16 bytes, maclen 20
00:05:43: SSH2 0: MAC compared for #3 :ok
```

Where to Go Next

You have to use a SSH remote device that supports SSH Version 2, and you have to connect to a Cisco IOS router.

Additional References

The following sections provide references related to Secure Shell Version 2.

Related Documents

Related Topic	Document Title
AAA	"Authentication, Authorization, and Accounting (AAA)" section of the *Cisco IOS Security Configuration Guide*
Configuring a host name and host domain	"Configuring Secure Shell" chapter in the *Cisco IOS Security Configuration Guide*

Related Topic	Document Title
Configuring Secure Shell	"Configuring Secure Shell" chapter of the *Cisco IOS Security Configuration Guide*
Debugging commands	*Cisco IOS Debug Command Reference*, Release 12.4T
Downloading a Cisco software image	*Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*
IOS configuration fundamentals	*Cisco IOS Configuration Fundamentals and Network Management Configuration Guide* and *Cisco IOS Configuration Fundamentals and Network Management Command Reference*
IPSec	"IP Security and Encryption" section of the *Cisco IOS Security Configuration Guide*
Security commands	*Cisco IOS Security Command Reference*, Release 12.4 T
SNMP, configuring traps	"Configuring SNMP Support" chapter in *Cisco IOS Network Management Configuration Guide*

Standards

Standards	Title
Internet Engineering Task Force (IETF) Secure Shell Version 2 Draft Standards	Internet Engineering Task Force website

MIBs

MIBs	MIBs Link
•	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **debug ip ssh**
- **ip ssh min dh size**
- **ip ssh rsa keypair-name**
- **ip ssh version**
- **ssh**

Feature Information for Secure Shell Version 2 Support

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Secure Shell Version 2 Support

Feature Name	Releases	Feature Information
Secure Shell Version 2 Support	12.3(4)T 12.2(25)S	The Secure Shell Version 2 Support feature allows you to configure Secure Shell (SSH) Version 2 (SSH Version 1 support was implemented in an earlier Cisco IOS software release). SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities.
Secure Shell Version 2 Client and Server Support	12.3(7)JA 12.0(32)SY	This feature was integrated into Cisco IOS Release 12.3(7)JA.
Secure Shell Version 2 Client and Server Support	12.4(17)	The Cisco IOS image was updated to provide for the automatic generation of SNMP traps when an SSH session terminates. For information about this feature, see the following section: • "SNMP Trap Generation" section on page 3 • "SNMP Debugging: Example" section on page 16
SSH Keyboard Interactive Authentication	12.4(18) 12.2(33)SXH3	This feature, also known as Generic Message Authentication for SSH, is a method that can be used to implement different types of authentication mechanisms. Basically, any currently supported authentication method that requires only user input can be performed with this feature. For information about this feature see the following sections: • "SSH Keyboard Interactive Authentication" section on page 4 • "SSH Keyboard Interactive Authentication: Examples" section on page 14

Table 1 Feature Information for Secure Shell Version 2 Support (continued)

Feature Name	Releases	Feature Information
Secure Shell SSH Version 2 Client Support	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Secure Shell Version 2 Enhancements	12.4(20)T	The Secure Shell Version 2 Enhancements include a number of additional capabilities such as support for VRF aware SSH, SSH debug enhancements and Diffie-Hellman group 14 and group 16 exchange support. For information about this feature see the following sections: • "Secure Shell Version 2 Enhancements" section on page 3

802.1X Authentication Services

Standalone MAB Support

First Published: November 11, 2008
Last Updated: November 11, 2008

Standalone MAC Authentication Bypass (MAB) is an authentication method that grants network access to specific MAC addresses—regardless of 802.1x capability or credentials. As a result, devices such as cash registers, fax machines, and printers can be readily authenticated, and network features that are based on authorization policies can be made available.

Before standalone MAB support was available, MAB could be configured only as a failover method for 802.1x authentication. Standalone MAB is independent of 802.1x authentication.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Standalone MAB Support" section on page 11.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Standalone MAB Support

IEEE 802.1x—Port-Based Network Access Control

You should understand the concepts of port-based network access control and have an understanding of how to configure port-based network access control on your Cisco platform. For more information, see the documentation for your Cisco platform and the *Cisco IOS Security Configuration Guide*.

RADIUS and ACLs

You should understand the concepts of the RADIUS protocol and have an understanding of how to create and apply access control lists (ACLs). For more information, see the documentation for your Cisco platform and the *Cisco IOS Security Configuration Guide*.

The switch must have a RADIUS configuration and be connected to the Cisco secure access control server (ACS). For more information, see the *Configuration Guide for CISCO Secure ACS.*

Information About Standalone MAB Support

To set up standalone MAB, you should understand the following concepts:

- Overview of the Cisco IOS Auth Manager, page 2
- Standalone MAB, page 2

Overview of the Cisco IOS Auth Manager

The capabilities of devices connecting to a given network can be different, thus requiring that the network support different authentication methods and authorization policies. The Cisco IOS Auth Manager handles network authentication requests and enforces authorization policies, regardless of authentication method. The Auth Manager maintains operational data for all port-based network connection attempts, authentications, authorizations, and disconnections and as such, serves as a session manager.

The possible states for Auth Manager sessions are:

- Idle—In the idle state, the authentication session has been initialized, but no methods have yet been run. This is an intermediate state.
- Running—A method is currently running. This is an intermediate state.
- Authc Success— The authentication method has run successfully. This is an intermediate state.
- Authc Failed—The authentication method has failed. This is an intermediate state.
- Authz Success—All features have been successfully applied for this session. This is a terminal state.
- Authz Failed—At least one feature has failed to be applied for this session. This is a terminal state.
- No methods—No method provided a result for this session. This is a terminal state.

Standalone MAB

MAB uses the MAC address of the connecting device to grant or deny network access. To support MAB, the RADIUS authentication server maintains a database of MAC addresses for devices that require access to the network. MAB generates a RADIUS request with a MAC address in the Calling-Station-Id

(attribute 31) and Service-Type (attribute 6) with value 10. After a successful authentication, the Auth Manager enables various authorization features specified by the authorization policy, such as ACL assignment and VLAN assignment.

How to Configure Standalone MAB

This section contains the following tasks:

Enabling Standalone MAB

Ports enabled with the Standalone MAB feature can use the MAC address of connecting devices to grant or deny network access. Perform the steps described in this section to enable standalone MAB on individual ports.

Prerequisites

Before you can configure standalone MAB, the switch must be connected to a Cisco Secure ACS server and RADIUS authentication, authorization, and accounting (AAA) must be configured.

Restrictions

Standalone MAB can be configured on switched ports only—it cannot be configured on routed ports.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **switchport**
5. **switchport mode access**
6. **authentication port-control auto**
7. **mab** [**eap**]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Switch(config)# interface FastEthernet2/1`	Enters interface configuration mode.
Step 4	**switchport** **Example:** `Switch(config-if)# switchport`	Places interface in Layer2-switched mode.
Step 5	**switchport mode access** **Example:** `Switch(config-if)# switchport mode access`	Sets a nontrunking, nontagged single VLAN Layer 2 interface.
Step 6	**authentication port-control auto** **Example:** `Switch(config-if)# authentication port-control auto`	Configures the authorization state of the port.
Step 7	**mab** [**eap**] **Example:** `Switch(config-if)# mab`	Enables MAB.
Step 8	**end** **Example:** `Switch(config-if)# end`	Returns to global configuration mode.

Troubleshooting Tips

The following commands can help troubleshoot standalone MAB:

- **debug authentication**
- **debug mab all**
- **show authentication registrations**
- **show authentication sessions**
- **show mab**

Enabling Reauthentication on a Port

By default, ports are not automatically reauthenticated. You can enable automatic reauthentication and specify how often reauthentication attempts are made.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **switchport**
5. **switchport mode access**
6. **authentication port-control auto**
7. **mab** [**eap**]
8. **authentication periodic**
9. **authentication timer reauthenticate** {*seconds* | **server**}
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Switch(config)# interface FastEthernet2/1`	Enters interface configuration mode.
Step 4	**switchport** **Example:** `Switch(config-if)# switchport`	Places interface in Layer2-switched mode.
Step 5	**switchport mode access** **Example:** `Switch(config-if)# switchport mode access`	Sets a nontrunking, nontagged single VLAN Layer 2 interface.

	Command or Action	Purpose
Step 6	**authentication port-control auto** **Example:** `Switch(config-if)# authentication port-control auto`	Configures the authorization state of the port.
Step 7	**mab** [**eap**] **Example:** `Switch(config-if)# mab`	Enables MAB.
Step 8	**authentication periodic** **Example:** `Switch(config-if)# authentication periodic`	Enables reauthentication.
Step 9	**authentication timer reauthenticate** {*seconds* | **server**} **Example:** `Switch(config-if)# authentication timer reauthenticate 900`	Configures the time, in seconds, between reauthentication attempts.
Step 10	**end** **Example:** `Switch(config-if)# end`	Returns to global configuration mode.

Specifying the Security Violation Mode

When there is a security violation on a port, the port can be shut down or traffic can be restricted. By default, the port is shut down. You can configure the period of time for which the port is shut down.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **switchport**
5. **switchport mode access**
6. **authentication port-control auto**
7. **mab** [**eap**]
8. **authentication violation** {**restrict** | **shutdown**}
9. **authentication timer restart** *seconds*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Switch(config)# interface FastEthernet2/1`	Enters interface configuration mode.
Step 4	**switchport** **Example:** `Switch(config-if)# switchport`	Places interface in Layer2-switched mode.
Step 5	**switchport mode access** **Example:** `Switch(config-if)# switchport mode access`	Sets a nontrunking, nontagged single VLAN Layer 2 interface.
Step 6	**authentication port-control auto** **Example:** `Switch(config-if)# authentication port-control auto`	Configures the authorization state of the port.
Step 7	**mab** [**eap**] **Example:** `Switch(config-if)# mab`	Enables MAB.
Step 8	**authentication violation** {**restrict** | **shutdown**} **Example:** `Switch(config-if)# authentication violation shutdown`	Configures the action to be taken when a security violation occurs on the port.
Step 9	**authentication timer restart** *seconds* **Example:** `Switch(config-if)# authentication timer restart 30`	Configures the period of time, in seconds, after which an attempt is made to authenticate an unauthorized port.
Step 10	**end** **Example:** `Switch(config-if)# end`	Returns to global configuration mode.

Configuration Examples for Standalone MAB

This section contains the following example:

Standalone MAB Configuration: Example

The following example shows how to configure standalone MAB on a port. In this example, the client is reauthenticated every 1200 seconds and the connection is dropped after 600 seconds of inactivity.

```
enable
 configure terminal
  interface GigabitEthernet2/1
   switchport
   switchport mode access
   switchport access vlan 2
   authentication port-control auto
   mab
   authentication violation shutdown
   authentication timer restart 30
   authentication periodic
   authentication timer reauthenticate 1200
   authentication timer inactivity 600
```

Additional References

The following sections provide references related to the standalone MAB feature.

Related Documents

Related Topic	Document Title
Authentication commands	*Cisco IOS Security Command Reference*
IEEE 802.1x - Flexible Authentication	*Cisco IOS Security Configuration Guide*

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
• CISCO-AUTH-FRAMEWORK-MIB • CISCO-MAC-AUTH-BYPASS-MIB • CISCO-PAE-MIB • IEEE8021-PAE-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3580	*IEEE 802.1x Remote Authentication Dial In User Service (RADIUS)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **authentication periodic**
- **authentication port-control**
- **authentication timer inactivity**
- **authentication timer reauthenticate**
- **authentication timer restart**
- **authentication violation**

- **debug authentication**
- **mab**
- **show authentication interface**
- **show authentication registrations**
- **show authentication sessions**
- **show mab**

Removed and Obsolete Commands

- **dot1x critical recovery delay**
- **dot1x host-mode**
- **dot1x port control**
- **dot1x reauthentication**
- **dot1x timer reauthentication**

Feature Information for Standalone MAB Support

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS, Catalyst OS, and Cisco IOS XE software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Standalone MAB Support

Feature Name	Releases	Feature Information
Standalone MAB Support	12.2(33)SXI	This feature grants network access to devices based on MAC address regardless of 802.1x capability or credentials. The following commands were introduced or modified: **authentication periodic**, **authentication port-control**, **authentication timer inactivity**, **authentication timer reauthenticate**, **authentication timer restart**, **authentication violation**, **debug authentication**, **mab**, **show authentication interface**, **show mab**, **show authentication registrations**, **show authentication sessions**.

IEEE 802.1x—Flexible Authentication

First Published: November 11, 2008
Last Updated: November 11, 2008

The IEEE 802.1x—Flexible Authentication feature provides a means of assigning authentication methods to ports and specifying the order in which the methods are executed when an authentication attempt fails. Using this feature, you can control which ports use which authentication methods, and you can control the failover sequencing of methods on those ports.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for IEEE 802.1x—Flexible Authentication" section on page 10.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for IEEE 802.1x—Flexible Authentication

IEEE 802.1x—Port-Based Network Access Control

You should understand the concepts of port-based network access control and have an understanding of how to configure port-based network access control on your Cisco platform. For more information, see the documentation for your Cisco platform and the *Cisco IOS Security Configuration Guide*.

RADIUS and ACLs

You should understand the concepts of the RADIUS protocol and have an understanding of how to create and apply access control lists (ACLs). For more information, see the documentation for your Cisco platform and the *Cisco IOS Security Configuration Guide*.

The switch must have a RADIUS configuration and be connected to the Cisco secure access control server (ACS). For more information, see the *Configuration Guide for CISCO Secure ACS.*

Restrictions for IEEE 802.1x—Flexible Authentication

In Cisco IOS Release 12.2(33)SXI, the web authentication method cannot fail over to the 802.1x or the MAB authentication method. So, when you configure authentication order, no other authentication method can follow web authentication.

Information About IEEE 802.1x—Flexible Authentication

To set up IEEE 802.1x—Flexible Authentication, you should understand the following concepts:

- Overview of the Cisco IOS Auth Manager, page 2
- Authentication Methods, page 3
- Host Mode Authentication, page 3
- Authentication Order and Authentication Priority, page 3

Overview of the Cisco IOS Auth Manager

The capabilities of devices connecting to a given network can be different, thus requiring that the network support different authentication methods and authorization policies. The Cisco IOS Auth Manager handles network authentication requests and enforces authorization policies, regardless of authentication method. The Auth Manager maintains operational data for all port-based network connection attempts, authentications, authorizations, and disconnections and as such, serves as a session manager.

The possible states for Auth Manager sessions are:

- Idle—In the idle state, the authentication session has been initialized, but no methods have yet been run. This is an intermediate state.
- Running—A method is currently running. This is an intermediate state.
- Authc Success— The authentication method has run successfully. This is an intermediate state.
- Authc Failed—The authentication method has failed. This is an intermediate state.

- Authz Success—All features have been successfully applied for this session. This is a terminal state.
- Authz Failed—At least one feature has failed to be applied for this session. This is a terminal state.
- No methods—No method provided a result for this session. This is a terminal state.

Authentication Methods

The IEEE 802.1x—Flexible Authentication feature supports three authentication methods:

- **dot1x**—IEEE 802.1x authentication is a Layer 2 authentication method.
- **mab**—MAC-Authentication Bypass is a Layer 2 authentication method.
- **webauth**—Web authentication is a Layer 3 authentication method.

Host Mode Authentication

The IEEE 802.1x—Flexible Authentication feature supports two new host modes:

- **multi-auth—**Multiauthentication allows one authentication on a voice VLAN and multiple authentications on the data VLAN.
- **multi-domain**—Multidomain authentication allows two authentications: one on the voice VLAN and one of the data VLAN.

The IEEE 802.1x—Flexible Authentication feature also supports single-host and multiple-host authentications.

Authentication Order and Authentication Priority

The IEEE 802.1x—Flexible Authentication feature enables authentication order and authentication priority. The **authentication order** command sets the default authentication priority. You can use the **authentication priority** command to override the default authentication priority. For example, you might specify an authentication order of MAB and 802.1x. However, after authorization, you might not want to ignore subsequent 802.1x handshakes. In this case, you can give the 802.1x authentication method a higher priority than the MAB method.

How to Configure IEEE 802.1x—Flexible Authentication

This section contains the following tasks:

- Configuring Authentication Order, page 3
- Configuring Authentication Priority, page 6

Configuring Authentication Order

Authentication order is configured on individual ports to control which ports use which authentication methods. Perform the steps described in this section to configure authentication order.

Prerequisites

Before you can use the IEEE 802.1x—Flexible Authentication feature, the switch must be connected to a Cisco secure ACS and RADIUS authentication, authorization, and accounting (AAA) must be configured for Web authentication. If appropriate, you must enable ACL download.

If the authentication order includes the 802.1x port authentication method, you must enable IEEE 802.1x authentication on the switch.

If the authentication order includes web authentication, configure a fallback profile that enables web authentication on the switch and the interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x system-auth-control**
4. **interface** *type slot/port*
5. **switchport**
6. **switchport mode access**
7. **switchport access vlan** *vlan-id*
8. **mab** [**eap**]
9. **authentication port-control** {**auto** | **force-authorized** | **fort unauthorized**}
10. **authentication fallback** *profile*
11. **authentication order** {**dot1x** [**mab** | **webauth**] [**webauth**] | **mab** [**dot1x** | **webauth**] [**webauth**] | **webauth**}
12. **dot1x pae authenticator**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	`dot1x system-auth-control` **Example:** `Switch(config)# dot1x system-auth-control`	(Optional) Enables IEEE 802.1x authentication globally on the switch. Enable IEEE 802.1x authentication if the authentication order includes the **dot1x** authentication method.

	Command or Action	Purpose
Step 4	**interface type** *slot/port* **Example:** `Switch(config)# interface FastEthernet2/1`	Enters interface configuration mode.
Step 5	**switchport** **Example:** `Switch(config-if)# switchport`	Places interface in Layer2-switched mode.
Step 6	**switchport mode access** **Example:** `Switch(config-if)# switchport mode access`	Sets a nontrunking, nontagged single VLAN Layer 2 interface.
Step 7	**switchport access vlan** *vlan-id* **Example:** `Switch(config-if)# switchport access vlan 2`	Sets the VLAN for the port.
Step 8	**mab** [**eap**] **Example:** `Switch(config-if)# mab`	(Optional) Enables MAB. Enable MAB if the authentication order includes the **mab** keyword (Step 11).
Step 9	**authentication port-control** {**auto** | **force-authorized** | **fort unauthorized**} **Example:** `Switch(config-if)# authentication port-control auto`	Configures the authorization state of the port.
Step 10	**authentication fallback** *profile* **Example:** `Switch(config-if)# authentication fallback web-profile`	(Optional) Enables web authentication. Enable web authentication if the authentication order includes the **webauth** keyword (Step 11).
Step 11	**authentication order** {**dot1x** [**mab** | **webauth**] [**webauth**] | **mab** [**dot1x** | **webauth**] [**webauth**] | **webauth**} **Example:** `Switch(config-if)# authentication order mab dot1x webauth`	Configures the authentication order.
Step 12	**dot1x pae authenticator** **Example:** `Switch(config)# dot1x pae authenticator`	Enables the port to respond to messages meant for an IEEE 802.1x authenticator.
Step 13	**end** **Example:** `Switch(config-if)# end`	Returns to global configuration mode.

Troubleshooting Tips

The following commands can help troubleshoot the Flexible Authentication feature:

- **debug authentication**
- **show authentication registrations**
- **show authentication sessions**
- **show dot1x**
- **show mab**

Configuring Authentication Priority

Authentication priority is configured to control the fail over sequencing of methods on individual ports. Perform the steps described in this section to configure authentication priority.

Prerequisites

Before you configure authentication priority, you should configure authentication order as described in the "Configuring Authentication Order" section on page 3.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **authentication priority** {**dot1x** [**mab** | **webauth**] [**webauth**] | **mab** [**dot1x** | **webauth**] [**webauth**] | **webauth**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	`interface` *`type slot/port`* **Example:** `Switch(config)# interface FastEthernet2/1`	Enters interface configuration mode.

	Command or Action	Purpose
Step 4	**authentication priority** {**dot1x** [**mab** | **webauth**] [**webauth**] | **mab** [**dot1x** | **webauth**] [**webauth**] | **webauth**} **Example:** `Switch(config-if)# authentication priotiry dot1x mab webauth`	Configures authentication priority.
Step 5	**end** **Example:** `Switch(config-if)# end`	Returns to global configuration mode.

Configuration Examples for IEEE 802.1x—Flexible Authentication

This section provides the following configuration example:

Flexible Authentication: Example, page 7

Flexible Authentication: Example

The following example configures the port in multiple authentication host mode with the order of authentication to be 802.11x first, then MAB, and finally, web authentication.

```
enable
configure terminal
dot1x system-auth-control

aaa new-model
aaa authentication login default group radius
aaa authentication dot1x default group radius
aaa authorization network default group radius
aaa authorization auth-proxy default group radius
aaa session-id common
ip http server

ip admission name webauth-rule proxy http
fallback profile webauth-profile
 ip access-group webauthlist in
 ip admission webauth-rule


interface GigabitEthernet2/1
 switchport
 switchport mode access
 switchport access vlan 125
 switchport voice vlan 127
 mab
 authentication port-control auto
 authentication fallback webauth-profile
 authentication host-mode multi-auth
 authentication order dot1x mab webauth
 dot1x pae authenticator
```

Additional References

The following sections provide references related to the IEEE 802.1x—Flexible Authentication feature.

Related Documents

Related Topic	Document Title
Authentication commands	*Cisco IOS Security Command Reference*
Standalone MAB Support	*Cisco IOS Security Configuration Guide*

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
• CISCO-AUTH-FRAMEWORK-MIB • CISCO-MAC-AUTH-BYPASS-MIB • CISCO-PAE-MIB • IEEE8021-PAE-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3580	*IEEE 802.1x Remote Authentication Dial In User Service (RADIUS)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **authentication fallback**
- **authentication host-mode**
- **authentication order**
- **authentication port-control**
- **authentication priority**
- **authentication timer restart**
- **debug authentication**
- **mab**
- **show authentication interface**
- **show authentication registrations**
- **show authentication sessions**
- **show mab**

Removed and Obsolete Commands

- **dot1x fallback**
- **dot1x host-mode**
- **dot1x port control**

Feature Information for IEEE 802.1x—Flexible Authentication

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS, Catalyst OS, and Cisco IOS XE software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for IEEE 802.1x—Flexible Authentication

Feature Name	Releases	Feature Information
IEEE 802.1x—Flexible Authentication	12.2(33)SXI	This feature provides a means of configuring ports with one or more authentication methods and specifying the order in which those authentication methods are attempted. The following commands were introduced or modified: **authentication fallback**, **authentication host-mode**, **authentication order**, **authentication port-control**, **authentication priority**, **authentication timer restart**, **debug authentication**, **mab**, **show authentication interface**, **show authentication registrations**, **show authentication sessions**, **show mab**.

Secure Infrastructure

AutoSecure

By using a single command-line interface (CLI), the AutoSecure feature allows a user to perform the following functions:

- Disable common IP services that can be exploited for network attacks
- Enable IP services and features that can aid in the defense of a network when under attack.

This feature also simplifies the security configuration of a router and hardens the router configuration.

Feature History for AutoSecure

Release	Modification
12.3(1)	This feature was introduced.
12.2(18)S	This feature was integrated into Cisco IOS Release 12.2(18)S.
12.3(8)T	Support for the roll-back functionality and system logging messages were added to Cisco IOS Release 12.3(8)T.
12.2(27)SBC	This feature was integrated into Cisco IOS Release 12.(27)SBC.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About AutoSecure

To configure the AutoSecure feature, you should understand the following concepts:

Benefits of AutoSecure

Simplified Router Security Configuration

AutoSecure is valuable to customers without special Security Operations Applications because it allows them to quickly secure their network without thorough knowledge of all the Cisco IOS features.

This feature eliminates the complexity of securing a router by creating a new CLI that automates the configuration of security features and disables certain features enabled by default that could be exploited for security holes.

Enhanced Password Security

AutoSecure provides the following mechanisms to enhance security access to the router:

- The ability to configure a required minimum password length, which can eliminate common passwords that are prevalent on most networks, such as “lab” and “cisco.”

  To configure a minimum password length, use the **security passwords min-length** command.

- Syslog messages are generated after the number of unsuccessful attempts exceeds the configured threshold.

  To configure the number of allowable unsuccessful login attempts (the threshold rate), use the **security passwords min-length** command.

Roll-Back and System Logging Message Support

In Cisco IOS Release 12.3(8)T, support for roll-back of the AutoSecure configuration is introduced. Roll-back enables a router to revert back to its preautosecure configuration state if the AutoSecure configuration fails.

Note Prior to Cisco IOS Release 12.3(8)T, roll-back of the AutoSecure configuration is unavailable; thus, you should always save the running configuration before configuring AutoSecure.

System Logging Messages capture any changes or tampering of the AutoSecure configuration that were applied on the running configuration. That is, more detailed audit trail information is provided when autosecure is executed.

Secure Management Plane

Securing the management plane is one of two focus areas for the AutoSecure feature. (The other focus area is described in the following section, "Secure Forwarding Plane.") Securing the management plane is done by turning off certain global and interface services that can be potentially exploited for security attacks and turning on global services that help mitigate the threat of attacks. Secure access and secure logging are also configured for the router.

Caution If your device is managed by a network management (NM) application, securing the management plane could turn off some services like HTTP server and disrupt the NM application support.

The following subsections define how AutoSecure helps to secure the management plane:

- Disable Global Services
- Disable Per Interface Services
- Enable Global Services
- Secure Access to the Router
- Log for Security

Disable Global Services

After enabling this feature (via the **auto secure** command), the following global services will be disabled on the router without prompting the user:

- Finger—Collects information about the system (reconnaissance) before an attack. If enabled, the information can leave your device vulnerable to attacks.
- PAD—Enables all packet assembler and disassembler (PAD) commands and connections between PAD devices and access servers. If enabled, it can leave your device vulnerable to attacks.
- Small Servers—Causes TCP and User Datagram Protocol (UDP) diagnostic port attacks: a sender transmits a volume of fake requests for UDP diagnostic services on the router, consuming all CPU resources.
- Bootp Server—Bootp is an insecure protocol that can be exploited for an attack.
- HTTP Server—Without secure-http or authentication embedded in the HTTP server with an associated ACL, the HTTP server is insecure and can be exploited for an attack. (If you must enable the HTTP server, you will be prompted for the proper authentication or access list.)

Note If you are using Security Device Manager (SDM), you must manually enable the HTTP server via the **ip http server** command.

- Identification Service—An unsecure protocol, defined in RFC 1413, that allows one to query a TCP port for identification. An attacker can access private information about the user from the ID server.
- CDP—If a large number of Cisco Discovery Protocol (CDP) packets are sent to the router, the available memory of the router can be consumed, causing the router to crash.

Caution NM applications that use CDP to discover network topology will not be able to perform discovery.

- NTP—Without authentication or access-control, Network Time Protocol (NTP) is insecure and can be used by an attacker to send NTP packets to crash or overload the router. (If you want to turn on NTP, you must configure NTP authentication using Message Digest 5 (MD5) and the **ntp access-group** command. If NTP is enabled globally, disable it on all interfaces on which it is not needed.)
- Source Routing—Provided only for debugging purposes, so source routing should be disabled in all other cases. Otherwise, packets may slip away from some of the access control mechanisms that they should have gone through.

Disable Per Interface Services

After enabling this feature, the following per interface services will be disabled on the router without prompting the user:

- ICMP redirects—Disabled on all interfaces. Does not add a useful functionality to a correctly configured to network, but it could be used by attackers to exploit security holes.
- ICMP unreachables—Disabled on all interfaces. Internet Control Management Protocol (ICMP) unreachables are a known cause for some ICMP-based denial of service (DoS) attacks.
- ICMP mask reply messages—Disabled on all interfaces. ICMP mask reply messages can give an attacker the subnet mask for a particular subnetwork in the internetwork.
- Proxy-Arp—Disabled on all interfaces. Proxy-Arp requests are a known cause for DoS attacks because the available bandwith and resources of the router can be consumed in an attempt to respond to the repeated requests that are sent by an attacker.
- Directed Broadcast—Disabled on all interfaces. Potential cause of SMURF attacks for DoS.
- Maintenance Operations Protocol (MOP) service—Disabled on all interfaces.

Enable Global Services

After enabling this feature, the following global services will be enabled on the router without prompting the user:

- The **service password-encryption** command—Prevents passwords from being visible in the configuration.
- The **service tcp-keepalives-in** and **service tcp-keepalives-out** commands—Ensures that abnormally terminated TCP sessions are removed.

Secure Access to the Router

Caution If your device is managed by an NM application, securing access to the router could turn off vital services and may disrupt the NM application support.

After enabling this feature, the following options in which to secure access to the router are available to the user:

- If a text banner does not exist, users will be prompted to add a banner. This feature provides the following sample banner:

```
Authorized access only
    This system is the property of ABC Enterprise
    Disconnect IMMEDIATELY if you are not an authorized user!
    Contact abc@xyz.com +99 876 543210 for help.
```

- The login and password (preferably a secret password, if supported) are configured on the console, AUX, vty, and tty lines. The **transport input** and **transport output** commands are also configured on all of these lines. (Telnet and secure shell (SSH) are the only valid transport methods.) The **exec-timeout** command is configured on the console and AUX as 10.
- When the image on the device is a crypto image, AutoSecure enables SSH and secure copy (SCP) for access and file transfer to and from the router. The **timeout** *seconds* and **authentication-retries** *integer* options for the **ip ssh** command are configured to a minimum number. (Telnet and FTP are not affected by this operation and remain operational.)
- If the AutoSecure user specifies that their device does not use Simple Network Management Protocol (SNMP), one of the following functionalities will occur:
  - In interactive mode, the user is asked whether to disable SNMP regardless of the values of the community strings, which act like passwords to regulate access to the agent on the router.
  - In non-interact mode, SNMP will be disabled if the community string is “public” or “private.”

Note After AutoSecure has been enabled, tools that use SNMP to monitor or configure a device will be unable to communicate with the device via SNMP.

- If authentication, authorization, and accounting (AAA) is not configured, configure local AAA. Autosecure will prompt users to configure a local username and password on the router.

Log for Security

After this feature is enabled, the following logging options, which allow you to identify and respond to security incidents, are available:

- Sequence numbers and time stamps for all debug and log messages. This option is useful when auditing logging messages.
- Logging messages can be generated for login-related events; for example, the message “Blocking Period when Login Attack Detected” will be displayed when a login attack is detected and the router enters “quiet mode.” (Quiet mode means that the router will not allows any login attempts via Telnet, HTTP, or SSH.)

  For more information on login system messages, see the Cisco IOS Release 12.3(4)T feature module *Cisco IOS Login Enhancements*.
- The **logging console critical** command, which sends system logging (syslog) messages to all available TTY lines and limits messages based on severity.
- The **logging buffered** command, which copies logging messages to an internal buffer and limits messages logged to the buffer based on severity.
- The **logging trap debugging** command, which allows all commands with a severity higher than debugging to be sent to the logging server.

Secure Forwarding Plane

To minimize the risk of attacks on the router forward plane, AutoSecure provides the following functions:

- Cisco Express Forwarding (CEF)—AutoSecure enables CEF or distributed CEF (dCEF) on the router whenever possible. Because there is no need to build cache entries when traffic starts arriving for new destinations, CEF behaves more predictably than other modes when presented with large volumes of traffic addressed to many destinations. Thus, routers configured for CEF perform better under SYN attacks than routers using the traditional cache.

Note CEF consumes more memory than a traditional cache.

- If the TCP intercept feature is available, it can be configured on the router for connection timeout.
- If strict Unicast Reverse Path Forwarding (uRPF) is available, it can be configured on the router to help mitigate problems that are caused by the introduction of forged (spoofed) IP source addresses. uRPF discards IP packets that lack a verifiable IP source address.
- If the router is being used as a firewall, it can be configured for context-based access control (CBAC) on public interfaces that are facing the Internet.

Note At the beginning of the AutoSecure dialogue, you will be prompted for a list of public interfaces.

How to Configure AutoSecure

This section contains the following procedures:

- Configuring AutoSecure, page 6 (required)
- Configuring Additional Security, page 7 (required)
- Verifying AutoSecure, page 8 (optional)

Configuring AutoSecure

To configure AutoSecure, you must perform the following tasks.

The auto secure Command

The **auto secure** command takes you through a semi-interactive session (also known as the AutoSecure dialogue) to secure the management and forwarding planes. This command gives you the option to secure just the management or the forwarding plane; if neither option is selected, the dialogue will ask you to configure both planes.

This command also allows you to go through all noninteractive configuration portions of the dialogue before the interactive portions. The noninteractive portions of the dialogue can be enabled by selecting the optional **no-interact** keyword.

Caution Although the **auto secure** command helps to secure a router, it does not guarantee the complete security of the router.

Restrictions

The AutoSecure configuration can be configured at run time or setup time. If any related configuration is modified after AutoSecure has been enabled, the AutoSecure configuration may not be fully effective.

SUMMARY STEPS

1. **enable**
2. **auto secure** [**management** | **forwarding**] [**no-interact** | **full**] [**ntp** | **login** | **ssh** | **firewall** | **tcp-intercept**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	`auto secure [management | forwarding] [no-interact | full] [ntp | login | ssh | firewall | tcp-intercept]` **Example:** `Router# auto secure`	Secures the management and forwarding planes of the router. • **management**—Only the management plane will be secured. • **forwarding**—Only the forwarding plane will be secured. • **no-interact**—The user will not be prompted for any interactive configurations. • **full**—The user will be prompted for all interactive questions. This is the default.

Configuring Additional Security

To enable enhanced security access to your router, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **security passwords min-length** *length*
4. **enable password** {*password* | [*encryption-type*] *encrypted-password*}
5. **security authentication failure rate** *threshold-rate* **log**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**security passwords min-length** *length* **Example:** `Router(config)# security passwords min-length 6`	Ensures that all configured passwords are at least a specified length. • *length*—Minimum length of a configured password.
Step 4	**enable password** {*password* | [*encryption-type*] *encrypted-password*} **Example:** Router(config)# enable password elephant	Sets a local password to control access to various privilege levels.
Step 5	**security authentication failure rate** *threshold-rate* **log** **Example:** Router(config)# security authentication failure rate 10 log	Configures the number of allowable unsuccessful login attempts. • *threshold-rate*—Number of allowable unsuccessful login attempts. • **log**—Syslog authentication failures if the rate exceeds the threshold.

Verifying AutoSecure

To verify that the AutoSecure feature is working successfully, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show auto secure config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	`show auto secure config` **Example:** `Router# show auto secure config`	(Optional) Displays all configuration commands that have been added as part of the AutoSecure configuration.

Configuration Examples for AutoSecure

This section provides the following configuration example:

- AutoSecure Configuration Dialogue: Example, page 9

AutoSecure Configuration Dialogue: Example

The following example is a sample AutoSecure dialogue. After you enable the **auto secure** command, the feature will automatically prompt you with a similar dialogue unless you enable the **no-interact** keyword. (For information on which features are disabled and which features are enabled, see the sections, "Secure Management Plane" and "Secure Forwarding Plane" earlier in this document.)

```
Router# auto secure
                --- AutoSecure Configuration ---

*** AutoSecure configuration enhances the security of the router but it will not make
router absolutely secure from all security attacks ***

All the configuration done as part of AutoSecure will be shown here. For more details of
why and how this configuration is useful, and any possible side effects, please refer to
Cisco documentation of AutoSecure.
At any prompt you may enter '?' for help.
Use ctrl-c to abort this session at any prompt.

Gathering information about the router for AutoSecure

Is this router connected to internet? [no]:y
Enter the number of interfaces facing internet [1]:
Interface                  IP-Address OK? Method Status
Protocol
FastEthernet0/1            10.1.1.1  YES NVRAM  up down

FastEthernet1/0            10.2.2.2  YES NVRAM  up down

FastEthernet1/1            10.0.0.1  YES NVRAM  up up

Loopback0                  unassigned YES NVRAM  up up

FastEthernet0/0            10.0.0.2  YES NVRAM  up down

Enter the interface name that is facing internet:FastEthernet0/0
```

```
Securing Management plane services..

Disabling service finger
Disabling service pad
Disabling udp & tcp small servers
Enabling service password encryption
Enabling service tcp-keepalives-in
Enabling service tcp-keepalives-out
Disabling the cdp protocol

Disabling the bootp server
Disabling the http server
Disabling the finger service
Disabling source routing
Disabling gratuitous arp
Enable secret is either not configured or is same as enable password
Enter the new enable secret:abc123
Configuring aaa local authentication
Configuring console, Aux and vty lines for
local authentication, exec-timeout, transport

Configure SSH server? [yes]:
Enter the domain-name:cisco.com

Configuring interface specific AutoSecure services
Disabling the following ip services on all interfaces:

 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
Disabling mop on Ethernet interfaces

Securing Forwarding plane services..

Enabling CEF (it might have more memory requirements on some low end
platforms)

Enabling unicast rpf on all interfaces connected to internet

Configure CBAC Firewall feature? [yes/no]:yes

This is the configuration generated:

no service finger
no service pad
no service udp-small-servers
no service tcp-small-servers
service password-encryption
service tcp-keepalives-in
service tcp-keepalives-out
no cdp run
no ip bootp server
no ip http server
no ip finger
no ip source-route
no ip gratuitous-arps
no ip identd
security passwords min-length 6
security authentication failure rate 10 log
enable secret 5 $1$CZ6G$GkGOnHdNJCO3CjNHHyTUA.
aaa new-model
```

```
aaa authentication login local_auth local
line console 0
 login authentication local_auth
 exec-timeout 5 0
 transport output telnet
line aux 0
 login authentication local_auth
 exec-timeout 10 0
 transport output telnet
line vty 0 4
 login authentication local_auth
 transport input telnet
ip domain-name cisco.com
crypto key generate rsa general-keys modulus 1024
ip ssh time-out 60
ip ssh authentication-retries 2
line vty 0 4
 transport input ssh telnet
service timestamps debug datetime localtime show-timezone msec
service timestamps log datetime localtime show-timezone msec
logging facility local2
logging trap debugging
service sequence-numbers
logging console critical
logging buffered
int FastEthernet0/1
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
int FastEthernet1/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
int FastEthernet1/1
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
int FastEthernet0/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
ip cef

interface FastEthernet0/0
 ip verify unicast reverse-path
ip inspect audit-trail
ip inspect dns-timeout 7
ip inspect tcp idle-time 14400
ip inspect udp idle-time 1800
ip inspect name autosec_inspect cuseeme timeout 3600
ip inspect name autosec_inspect ftp timeout 3600
ip inspect name autosec_inspect http timeout 3600
```

```
ip inspect name autosec_inspect rcmd timeout 3600
ip inspect name autosec_inspect realaudio timeout 3600
ip inspect name autosec_inspect smtp timeout 3600
ip inspect name autosec_inspect tftp timeout 30
ip inspect name autosec_inspect udp timeout 15
ip inspect name autosec_inspect tcp timeout 3600
access-list 100 deny ip any any
interface FastEthernet0/0
 ip inspect autosec_inspect out
 ip access-group 100 in
!
end


Apply this configuration to running-config? [yes]:yes

Applying the config generated to running-config
The name for the keys will be:ios210.cisco.com

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys ...[OK]

Router#
```

Additional References

The following sections provide references related to AutoSecure.

Related Documents

Related Topic	Document Title
Login functionality (such as login delays and login blocking periods)	*Cisco IOS Login Enhancements*, Cisco IOS Release 12.3(4)T feature module
Additional information regarding router configuration	*Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.3T*
Additional router configuration commands	*Cisco IOS Configuration Fundamentals Command Reference, Release 12.3T*

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1918	Address Allocation for Private Internets
RFC 2267	*Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **auto secure**
- **security passwords min-length**
- **show auto secure config**

Cisco IOS Login Enhancements (Login Block)

Document First Published: August 2005
Last Updated: October 2007

The Cisco IOS Login Enhancements (Login Block) feature allows users to enhance the security of a router by configuring options to automatically block further login attempts when a possible denial-of-service (DoS) attack is detected.

The login block and login delay options introduced by this feature can be configured for Telnet or SSH virtual connections. By enabling this feature, you can slow down “dictionary attacks” by enforcing a “quiet period” if multiple failed connection attempts are detected, thereby protecting the routing device from a type of denial-of-service attack.

Feature History for Cisco IOS Login Enhancements

Release	Modification
12.3(4)T	This feature was introduced.
12.2(25)S	This feature was integrated into Cisco IOS Release 12.2 S.
12.2(33)SRA	This feature was integrated into Cisco IOS Release 12.2 SR.
12.2(33)SRB, 12.2(33)SXH, 12.4(15)T1	Support for HTTP login blocking was added.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About Cisco IOS Login Enhancements

To use login enhancements, you should understand the following concepts:

Protecting Against Denial of Service and Dictionary Login Attacks

Connecting to a routing device for the purposes of administering (managing) the device, at either the User or Executive level, is most frequently performed using Telnet or SSH (secure shell) from a remote console (such as a PC). SSH provides a more secure connection option because communication traffic between the user's device and the managed device are encrypted. The Login Block capability, when enabled, applies to both Telent connections and SSH connections. Beginning in Release versions 12.3(33)SRB2, 12.2(33)SXH2, and 12.4(15)T1, the Login Block capability also applies to HTTP connections."

The automated activation and logging of the Login Block and Quiet Period capabilities introduced by this feature are designed to further enhance the security of your devices by specifically addressing two well known methods that individuals use to attempt to disrupt or compromise networked devices.

If the connection address of a device is discovered and is reachable, a malicious user may attempt to interfere with the normal operations of the device by flooding it with connection requests. This type of attack is referred to as an attempted Denial-of-Service, because it is possible that the device may become too busy trying to process the repeated login connection attempts to properly handle normal routing services or will not be able to provide the normal login service to legitimate system administrators.

The primary intention of a dictionary attack, unlike a typical DoS attack, is to actually gain administrative access to the device. A dictionary attack is an automated process to attempt to login by attempting thousands, or even millions, of username/password combinations. (This type of attack is called a "dictionary attack" because it typically uses, as a start, every word found in a typical dictionary as a possible password.) As scripts or progams are used to attempt this access, the profile for such attempts is typically the same as for DoS attempts; multiple login attempts in a short period of time.

By enabling a detection profile, the routing device can be configured to react to repeated failed login attempts by refusing further connection request (login blocking). This block can be configured for a period of time, called a "quiet period". Legitimate connection attempts can still be permitted during a quiet period by configuring an access-list (ACL) with the addresses that you know to be associated with system adminstrators.

Login Enhancements Functionality Overview

To better configure security for virtual login connections, the following requirements have been added to the login process:

- Delays Between Successive Login Attempts
- Login Shutdown If DoS Attacks Are Suspected
- Generation of System Logging Messages for Login Detection

Delays Between Successive Login Attempts

A Cisco IOS device can accept virtual connections as fast as they can be processed. Introducing a delay between login attempts helps to protect the Cisco IOS software-based device against malicious login connections such as dictionary attacks and DoS attacks. Delays can be enabled in one of the following ways:

- Via the **auto secure** command. If you enable the AutoSecure feature, the default login delay time of one second is automatically enforced.
- Via the **login block-for** command. You must enter this command before issuing the **login delay** command. If you enter only the **login block-for** command, the default login delay time of one second is automatically enforced.
- Via the new global configuration mode command, **login delay**, which allows you to specify a the login delay time to be enforced, in seconds.

Login Shutdown If DoS Attacks Are Suspected

If the configured number of connection attempts fail within a specified time period, the Cisco IOS device will not accept any additional connections for a “quiet period.” (Hosts that are permitted by a predefined access-control list [ACL] are excluded from the quiet period.)

The number of failed connection attempts that trigger the quiet period can be specified via the new global configuration mode command **login block-for**. The predefined ACL that is excluded from the quiet period can be specified via the new global configuration mode command **login quiet-mode access-class**.

This functionality is disabled by default, and it is not enabled if autosecure if enabled.

Generation of System Logging Messages for Login Detection

After the router switches to and from quiet mode, logging messages are generated. Also, if configured, logging messages are generated upon every successful or failed login request.

Logging messages can be generated for successful login requests via the new global configuration command **login on-success**; the **login on-failure** command generates logs for failed login requests.

Logging messages for failed login attempts are automatically enabled when the **auto secure** command is issued; they are not automatically enabled for successful login attempts via autosecure.

Note Currently, only system logging (syslog) messages can be generated for login-related events. Support for SNMP notifications (traps) will be added in a later release.

System Logging Messages for a Quiet Period

The following logging message is generated after the router switches to quiet-mode:

```
00:04:07:%SEC_LOGIN-1-QUIET_MODE_ON:Still timeleft for watching failures is 158 seconds,
[user:sfd] [Source:10.4.2.11] [localport:23] [Reason:Invalid login], [ACL:22] at 16:17:23
UTC Wed Feb 26 2003
```

The following logging message is generated after the router switches from quiet mode back to normal mode:

```
00:09:07:%SEC_LOGIN-5-QUIET_MODE_OFF:Quiet Mode is OFF, because block period timed out at
16:22:23 UTC Wed Feb 26 2003
```

System Logging Messages for Successful and Failed Login Requests

The following logging message is generated upon a successful login request:

```
00:04:32:%SEC_LOGIN-5-LOGIN_SUCCESS:Login Success [user:test] [Source:10.4.2.11]
[localport:23] at 20:55:40 UTC Fri Feb 28 2003
```

The following logging message is generated upon a failed login request:

```
00:03:34:%SEC_LOGIN-4-LOGIN_FAILED:Login failed [user:sdfs] [Source:10.4.2.11]
[localport:23] [Reason:Invalid login] at 20:54:42 UTC Fri Feb 28 2003
```

How to Configure Cisco IOS Login Enhancements

This section contains the following procedures:

- Configuring Login Parameters, page 4 (Required)
- Verifying Login Parameters, page 6 (Optional)

Configuring Login Parameters

Use this task to configure your Cisco IOS device for login parameters that help detect suspected DoS attacks and slow down dictionary attacks.

Login Parameter Defaults

All login parameters are disabled by default. You must issue the **login block-for** command, which enables default login functionality, before using any other login commands. After the **login block-for** command is enabled, the following defaults are enforced:

- A default login delay of one second
- All login attempts made via Telnet or SSH are denied during the quiet period; that is, no ACLs are exempt from the login period until the **login quiet-mode access-class** command is issued.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **login block-for** *seconds* **attempts** *tries* **within** *seconds*
4. **login quiet-mode access-class** {*acl-name* | *acl-number*}

5. **login delay** *seconds*
6. **login on-failure log** [**every** *login*]
7. **login on-success log** [**every** *login*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**login block-for** *seconds* **attempts** *tries* **within** *seconds* **Example:** `Router(config)# login block-for 100 attempts 2 within 100`	Configures your Cisco IOS device for login parameters that help provide DoS detection. **Note** This command must be issued before any other login command can be used.
Step 4	**login quiet-mode access-class** {*acl-name* | *acl-number*} **Example:** `Router(config)# login quiet-mode access-class myacl`	(Optional) Specifies an ACL that is to be applied to the router when it switches to quiet mode. If this command is not enabled, all login requests will be denied during quiet mode.
Step 5	**login delay** *seconds* **Example:** `Router(config)# login delay 10`	(Optional) Configures a delay between successive login attempts.
Step 6	**login on-failure log** [**every** *login*] **Example:** `Router(config)# login on-failure log`	(Optional) Generates logging messages for failed login attempts.
Step 7	**login on-success log** [**every** *login*] **Example:** `Router(config)# login on-success log every 5`	(Optional) Generates logging messages for successful login attempts.

What to Do Next

After you have configured login parameters on your router, you may wish to verify the settings. To complete this task, see the following section "Verifying Login Parameters."

Verifying Login Parameters

Use this task to verify the applied login configuration and present login status on your router.

SUMMARY STEPS

1. **enable**
2. **show login** [**failures**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show login** [**failures**] **Example:** `Router# show login`	Displays login parameters. • **failures**—Displays information related only to failed login attempts.

Examples

The following sample output from the **show login** command verifies that no login parameters have been specified:

```
Router# show login

No login delay has been applied.
No Quiet-Mode access list has been configured.
All successful login is logged and generate SNMP traps.
All failed login is logged and generate SNMP traps

Router NOT enabled to watch for login Attacks
```

The following sample output from the **show login** command verifies that the **login block-for** command is issued. In this example, the command is configured to block login hosts for 100 seconds if 16 or more login requests fail within 100 seconds; five login requests have already failed.

```
Router# show login

A default login delay of 1 seconds is applied.
No Quiet-Mode access list has been configured.
All successful login is logged and generate SNMP traps.
All failed login is logged and generate SNMP traps.

Router enabled to watch for login Attacks.
If more than 15 login failures occur in 100 seconds or less, logins will be disabled for
100 seconds.

Router presently in Watch-Mode, will remain in Watch-Mode for 95 seconds.
Present login failure count 5.
```

The following sample output from the **show login** command verifies that the router is in quiet mode. In this example, the **login block-for** command was configured to block login hosts for 100 seconds if 3 or more login requests fail within 100 seconds.

```
Router# show login

A default login delay of 1 seconds is applied.
No Quiet-Mode access list has been configured.
All successful login is logged and generate SNMP traps.
All failed login is logged and generate SNMP traps.

Router enabled to watch for login Attacks.
If more than 2 login failures occur in 100 seconds or less, logins will be disabled for
100 seconds.

Router presently in Quiet-Mode, will remain in Quiet-Mode for 93 seconds.
Denying logins from all sources.
```

The following sample output from **show login failures** command shows all failed login attempts on the router:

```
Router# show login failures

Information about login failure's with the device

Username      Source IPAddr  lPort Count  TimeStamp
try1          10.1.1.1        23    1     21:52:49 UTC Sun Mar 9 2003
try2          10.1.1.2        23    1     21:52:52 UTC Sun Mar 9 2003
```

The following sample output from **show login failures** command verifies that no information is presently logged:

```
Router# show login failures

*** No logged failed login attempts with the device.***
```

Configuration Examples for Login Parameters

This section includes the following example:

- Setting Login Parameters: Example, page 7

Setting Login Parameters: Example

The following example shows how to shows how to configure your router to enter a 100 second quiet period if 15 failed login attempts is exceeded within 100 seconds; all login requests will be denied during the quiet period except hosts from the ACL “myacl.” Also, logging messages will be generated for every 10th failed login and every 15th successful login.

```
Router(config)# login block-for 100 attempts 15 within 100
Router(config)# login quiet-mode access-class myacl
Router(config)# login on-failure log every 10
Router(config)# login on-success log every 15
```

Additional References

The following sections provide references related to Cisco IOS Login Enhancements.

Related Documents

Related Topic	Document Title
AutoSecure	• *AutoSecure* (Cisco IOS Release 12.3(1) feature module) • Cisco IOS Security Configuration Guides, Release 12.4.
Secure Management/Administrative Access	Role-Based CLI Access

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information

about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **login block-for**
- **login delay**
- **login on-failure**
- **login on-success**
- **login quiet-mode access-class**
- **show login**

Image Verification

The Image Verification feature allows users to automatically verify the integrity of Cisco IOS images. Thus, users can be sure that the image is protected from accidental corruption, which can occur at any time during transit, starting from the moment the files are generated by Cisco until they reach the user.

Feature History for Image Verification

Release	Modification
12.2(18)S	This feature was introduced.
12.0(26)S	This feature was integrated into Cisco IOS Release 12.0(26)S.
12.3(4)T	This feature was integrated in Cisco IOS Release 12.3(4)T.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Image Verification

Cisco IOS Release 12.2(18)S and 12.0(26)S Only

Image Verification is applied to and attempted on any file; however, if the file is not an image file, image verification will not occur and you will see the following error, “SIGNATURE-NOT-FOUND.”

Cisco IOS Release 12.3(4)T Only

Image Verification is applied only to image files. If any other file type is copied or verified, you will not receive a warning that image verification did occur, and the command (copy or verify) will silently succeed.

Information About Image Verification

To use image authentication for your Cisco IOS images, you should understand the following concepts:

- Benefit of Image Verification, page 2
- How Image Verification Works, page 2

Benefit of Image Verification

The efficiency of Cisco IOS routers is improved because the routers can now automatically detect when the integrity of an image is accidentally corrupted as a result of transmission errors or disk corruption.

How Image Verification Works

Because a production image undergoes a sequence of transfers before it is copied into the memory of a router, the integrity of the image is at risk of accidental corruption every time a transfer occurs. When downloading an image from Cisco.com, a user can run a message-digest5 (MD5) hash on the downloaded image and verify that the MD5 digest posted on Cisco.com is the same as the MD5 digest that is computed on the user’s server. However, many users choose not to run an MD5 digest because it is 128-bits long and the verification is manual. Image verification allows the user to automatically validate the integrity of all downloaded images, thereby, significantly reducing user interaction.

How to Use Image Verification

This section contains the following procedures:

- Globally Verifying the Integrity of an Image, page 3
- Verifying the Integrity of an Image That Is About to Be Copied, page 4
- Verifying the Integrity of an Image That Is About to Be Reloaded, page 4

Globally Verifying the Integrity of an Image

The **file verify auto** command enables image verification globally; that is, all images that are to be copied (via the **copy** command) or reloaded (via the **reload** command) are automatically verified. Although both the **copy** and **reload** commands have a **/verify** keyword that enables image verification, you must issue the keyword each time you want to copy or reload an image. The **file verify auto** command enables image verification by default, so you no longer have to specify image verification multiple times.

If you have enabled image verification by default but prefer to disable verification for a specific image copy or reload, the **/noverify** keyword, along with either the **copy** or the **reload** command, will override the **file verify auto** command.

Use this task to enable automatic image verification.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **file verify auto**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**file verify auto** **Example:** `Router(config)# file verify auto`	Enables automatic image verification.
Step 4	`exit` **Example:** `Router(config)# exit`	Exits global configuration mode. You must exit global configuration mode if you are going to copy or reload an image.

What to Do Next

After issuing the **file verify auto** command, you do not have to issue the **/verify** keyword with the **copy** or the **reload** command because each image that is copied or reloaded will be automatically verified.

Verifying the Integrity of an Image That Is About to Be Copied

When issuing the **copy** command, you can verify the integrity of the copied file by entering the **/verify** keyword. If the integrity check fails, the copied file will be deleted. If the file that is about to be copied does not have an embedded hash (an old image), you will be prompted whether or not to continue with the copying process. If you choose to continue, the file will be successfully copied; if you choose not to continue, the copied file will be deleted.

Without the **/verify** keyword, the **copy** command could copy a file that is not valid. Thus, after the **copy** command has been successfully executed, you can issue the **verify** command at any time to check the integrity of the files that are in the storage of the router.

Use this task to verify the integrity of an image before it is copied onto a router.

SUMMARY STEPS

1. **enable**
2. **copy** [**/erase**] [**/verify** | **/noverify**] *source-url destination-url*
3. **verify** [**/md5** [*md5-value*]] *filesystem*:[*file-url*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**copy** [**/erase**] [**/verify** | **/noverify**] *source-url destination-url* **Example:** `Router# copy /verify tftp://10.1.1.1/jdoe/c7200-js-mz disk0:`	Copies any file from a source to a destination. • **/verify**—Verifies the signature of the destination file. If verification fails, the file will be deleted. • **/noverify**—Does not verify the signature of the destination file before the image is copied. **Note** **/noverify** is often issued if the **file verify auto** command is enabled, which automatically verifies the signature of all images that are copied.
Step 3	**verify** [**/md5** [*md5-value*]] *filesystem:*[*file-url*] **Example:** `Router# verify bootflash://c7200-kboot-mz.121-8a.E`	(Optional) Verifies the integrity of the images in the router's storage.

Verifying the Integrity of an Image That Is About to Be Reloaded

By issuing the **reload** command with the /**verify** keyword, the image that is about to be loaded onto your system will be checked for integrity. If the /**verify** keyword is specified, image verification will occur before the system initiates the reboot. Thus, if verification fails, the image will not be loaded.

Note Because different platforms obtain the file that is to be loaded in various ways, the file specified in BOOTVAR will be verified. If a file is not specified, the first file on each subsystem will be verified.

On certain platforms, because of variables such as the configuration register, the file that is verified may not be the file that is loaded.

Use this task to verify the integrity of an image before it is reloaded onto a router.

SUMMARY STEPS

1. **enable**
2. **reload** [
   [**warm**] [**/verify** | **/noverify**] *text* |
   [**warm**] [**/verify** | **/noverify**] **in** [*hh***:**]*mm* [*text*] |
   [**warm**] [**/verify** | **/noverify**] **at** *hh***:***mm* [*month day* | *day month*] [*text*] |
   [**warm**] [**/verify** | **/noverify**] **cancel**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**reload** [[**warm**] [**/verify** | **/noverify**] *text* | [**warm**] [**/verify** | **/noverify**] **in** [*hh*:]*mm* [*text*] | [**warm**] [**/verify** | **/noverify**] **at** *hh:mm* [*month day* | *day month*] [*text*] | [**warm**] [**/verify** | **/noverify**] **cancel**] **Example:** `Router# reload /verify`	Reloads the operating system. • **/verify**—Verifies the signature of the destination file. If verification fails, the file will be deleted. • **/noverify**—Does not verify the signature of the destination file before the image is reloaded. **Note** **/noverify** is often issued if the **file verify auto** command is enabled, which automatically verifies the signature of all images that are copied.

Configuration Examples for Image Verification

This section contains the following configuration examples:

Global Image Verification: Example

The following example shows how to enable automatic image verification. After enabling this command, image verification will automatically occur for all images that are either copied (via the **copy** command) or reloaded (via the **reload** command).

Router(config)# **file verify auto**

Image Verification via the copy Command: Example

The following example shows how to specify image verification before copying an image:

```
Router# copy /verify tftp://10.1.1.1/jdoe/c7200-js-mz disk0:

Destination filename [c7200-js-mz]?
Accessing tftp://10.1.1.1/jdoe/c7200-js-mz...
Loading jdoe/c7200-js-mz from 10.1.1.1 (via FastEthernet0/0):!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 19879944 bytes]

19879944 bytes copied in 108.632 secs (183003 bytes/sec)
Verifying file integrity of disk0:/c7200-js-mz
.....................................................................................
.....................................................................................
.....................................................................................
.........................Done!
Embedded Hash MD5 :CFA258948C4ECE52085DCF428A426DCD
Computed Hash    MD5 :CFA258948C4ECE52085DCF428A426DCD
CCO Hash            MD5 :44A7B9BDDD9638128C35528466318183

Signature Verified
```

Image Verification via the reload Command: Example

The following example shows how to specify image verification before reloading an image onto the router:

```
Router# reload /verify

Verifying file integrity of bootflash:c7200-kboot-mz.121-8a.E
%ERROR:Signature not found in file bootflash:c7200-kboot-mz.121-8a.E.
Signature not present. Proceed with verify? [confirm]
Verifying file disk0:c7200-js-mz
.....................................................................
.....................................................Done!
Embedded Hash   MD5 :CFA258948C4ECE52085DCF428A426DCD
Computed Hash   MD5 :CFA258948C4ECE52085DCF428A426DCD
CCO Hash        MD5 :44A7B9BDDD9638128C35528466318183
Signature Verified

Proceed with reload? [confirm]n
```

verify Command Sample Output: Example

The following example shows how to specify image verification via the **verify** command:

```
Router# verify disk0:c7200-js-mz

%Filesystem does not support verify operations
Verifying file integrity of disk0:c7200-js-mz.....................................
.....................................................................................Done!
Embedded Hash   MD5 :CFA258948C4ECE52085DCF428A426DCD
Computed Hash   MD5 :CFA258948C4ECE52085DCF428A426DCD
CCO Hash        MD5 :44A7B9BDDD9638128C35528466318183

Signature Verified
```

Additional References

The following sections provide references related to Image Verification.

Related Documents

Related Topic	Document Title
Configuration tasks and information for loading, maintaining, and rebooting system images	*The section "File Management" in the Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*
Additional commands for loading, maintaining, and rebooting system images	*Cisco IOS Configuration Fundamentals and Network Management Command Reference*, Release 12.3 T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information

about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

New Command

- **file verify auto**

Modified Commands

- **copy**
- **reload**
- **verify**

Control Plane Policing

Part Number OL-8695-01 (Rev A0), January 19, 2006

The Control Plane Policing feature allows users to configure a quality of service (QoS) filter that manages the traffic flow of control plane packets to protect the control plane of Cisco IOS routers and switches against reconnaissance and denial-of-service (DoS) attacks. In this way, the control plane (CP) can help maintain packet forwarding and protocol states despite an attack or heavy traffic load on the router or switch.

History for the Control Plane Policing Feature

Release	Modification
12.2(18)S	This feature was introduced.
12.3(4)T	Control Plane Policing was integrated into Cisco IOS Release 12.3(4)T, and the output rate-limiting (silent mode operation) feature was added.
12.3(7)T	CISCO-CLASS-BASED-QOS-MIB was extended to manage control plane QoS policies, and the **police rate** command was introduced to support traffic policing on the basis of packets per second for control plane traffic.
12.0(29)S	The Control Plane Policing feature was integrated into Cisco IOS Release 12.0(29)S.
12.2(18)SXD1	The Control Plane Policing feature was integrated into Cisco IOS Release 12.2(18)SXD1.
12.0(30)S	Support for distributed control plane services on the Cisco 12000 series Internet router was added.
12.2(27)SBC	This feature was integrated into Cisco IOS Release 12.2(27)SBC.
12.0(32)S	Support for aggregate control plane services on the Cisco 10720 Internet router was added.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Prerequisites for Control Plane Policing

- Understanding the concepts and general configuration procedure (class map and policy map) for applying quality-of-service (QoS) policies on a router

  For information about Cisco IOS QoS and the procedure for configuring QoS in your network using the modular QoS command-line interface (MQC), refer to *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.3.

Restrictions for Control Plane Policing

Aggregate and Distributed Control Plane Policing

Aggregate policing is supported in Cisco IOS Release 12.0(29)S, Cisco IOS Release 12.2(18)S, and Cisco IOS Release 12.3(4)T and later releases.

Distributed policing is supported only in Cisco IOS Release 12.0(30)S and later Cisco IOS 12.0S releases.

Output Rate-Limiting Support

Output rate-limiting is performed in silent (packet discard) mode. Silent mode enables a router to silently discard packets using policy maps applied to output control plane traffic with the **service-policy output** command. For more information, see Output Rate-Limiting and Silent Mode Operation, page 10.

Output rate-limiting (policing) in silent mode is supported only in:

- Cisco IOS Release 12.2(25)S and later Cisco IOS 12.2S releases
- Cisco IOS Release 12.3(4)T and later Cisco IOS 12.3T releases

Output rate-limiting is not supported for distributed control plane services in Cisco IOS 12.0S releases or in Cisco IOS 12.2SX releases.

Output rate-limiting is not supported on the Cisco 7500 series and Cisco 10720 Internet router.

Modular QoS Restrictions

The Control Plane Policing feature requires the modular QoS command-line interface (CLI) (MQC) to configure packet classification and policing. All restrictions that apply when you use the MQC to configure policing also apply when you configure control plane policing. Only two MQC actions are supported in policy maps—**police** and **drop**.

Note On the Cisco 10720 Internet router, only the **police** command, not the **drop** command, is supported in policy maps. In addition, in a QoS service policy attached to the 10720 control plane, the **police** command does not support **set** actions as arguments in **conform-action**, **exceed-action**, and **violate-action** parameters.

Features that require network-based application recognition (NBAR) classification may not work well at the control plane level. The following classification (match) criteria are supported on all platforms:

- Standard and extended IP access lists (ACLs)
- In class-map configuration mode: **match ip dscp**, **match ip precedence**, and **match protocol arp** commands.

Note In the Cisco IOS 12.2SX release, the **match protocol arp** command is not supported.

On the Cisco 10720 Internet router, the following MQC commands are also supported in class-map configuration mode: **match input-interface**, **match mpls experimental**, **match protocol ipv6**, and **match qos-group**. When using these commands for control plane policing on the Cisco 10720 Internet router, note the following restrictions:

- Packet classification using match criteria is not supported for packets that cannot be classified in the 10720 data path, such as unknown Layer 2 encapsulation and IP options.
- The following IPv6 fields are not be supported in packet classification for IPv6 QoS on the Cisco 10720 Internet router and are, therefore, not supported for control plane policing:
  - IPv6 source and destination addresses
  - Layer 2 class of service (CoS)
  - IPv6 routing header flag
  - IPv6 undetermined transport flag
  - IPv6 flow label
  - IP Real-Time transport Protocol (RTP)

Note Packets that are not supported for QoS packet classification on the Cisco 10720 Internet router are not policed in the default traffic class for control plane policing.

CISCO-CLASS-BASED-QOS-MIB Control Plane Support

In Cisco IOS Release 12.3(7)T and later Cisco IOS 12.3T releases, the CISCO-CLASS-BASED-QOS-MIB is extended to manage control plane QoS policies and provide information about the control plane.

Cisco IOS Release 12.2(18)SXD1

In Cisco IOS Release 12.2(18)SXD1 and later releases, Hardware Control Plane Interface for Control Plane Policing has the following restrictions:

- Supported only with Supervisor Engine 720. Not supported with Supervisor Engine 2.
- Does not support CoPP output rate limiting (policing).
- Does not support the CoPP silent operation mode.
- Cisco IOS Release 12.2(18)SXD1 and later releases automatically install the CoPP service policy on all DFC-equipped switching modules.

For more information about Control Plane Policing in Cisco IOS Release 12.2(18)SXD1 and later releases, see either of these publications:

- For Control Plane Policing on Catalyst 6500 series switches:

  http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/dos.htm
- For Control Plane Policing on Cisco 7600 series routers:

  http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/swcg/dos.htm

Information About Control Plane Policing

To configure the Control Plane Policing feature, you should understand the following concepts:

Benefits of Control Plane Policing

Configuring the Control Plane Policing feature on your Cisco router or switch provides the following benefits:

- Protection against DoS attacks at infrastructure routers and switches
- QoS control for packets that are destined to the control plane of Cisco routers or switches
- Ease of configuration for control plane policies
- Better platform reliability and availability

Terms to Understand

Because different platforms can have different architectures, the following set of terms is defined. Figure 1 illustrates how control plane policing works.

Figure 1 Layout of Control Plane, Central Switch Engine, Distributed Switch Engines, and Line Cards on a Router

- Control plane (CP)—A collection of processes that run at the process level on the route processor (RP). These processes collectively provide high-level control for most Cisco IOS functions.
- Central switch engine—A device that is responsible for high-speed routing of IP packets. It also typically performs high-speed input and output services for nondistributed interfaces. (See nondistributed line cards.) The central switch engine is used to implement aggregate CP protection for all interfaces on the router.

Note All IP packets that are destined for the CP should pass through the central switch engine before they are forwarded to the process level.

On the Cisco 10720 Internet router, control plane policing is implemented on Cisco Parallel eXpress Forwarding (PXF) in a Toaster-based architecture. PXF is a hardware-based central switch engine that can filter traffic at a higher rate than the route processor. PXF switches all data traffic separately from the route processor. PXF packet processing occurs at an intermediate step between the nondistributed line cards and route processor shown in Figure 1. In addition to the regular punting, PXF also punts certain types of packets (such as unknown Layer 2 encapsulation and packets with IP options) to the RP for further processing at interrupt level. For more information, refer to *Queueing Architecture and Modular Quality of Service (QoS) on the Cisco 10720 Internet Router*.

Note On the Cisco 10720 Internet router, you can configure enhanced RP protection by using the **ip option drop** command to drop IPv4 packets with IP options that are punted to the RP by PXF. Tunneled IPv4 packets and IPv4 packets with an unsupported encapsulation method are not dropped. For more information, refer to *ACL IP Options Selective Drop*.

- Distributed switch engine—A device that is responsible for high-speed switching of IP packets on distributed line cards without using resources from the central switch engine. It also typically performs input and output services for the line card. Each distributed switch engine is used to

implement distributed CP services for all ports on a line card. Input CP services distribute the processing load across multiple line cards and conserve vital central switch engine resources. Distributed CP services are optional; however, they provide a more refined level of service than aggregate services.

- Nondistributed line cards—Line cards that are responsible for receiving packets and occasionally performing input and output services. All packets must be forwarded to the central switch engine for a routing or switching decision. Aggregate CP services provide coverage for nondistributed line cards.

Note Distributed CP services are supported only in 12.0(30)S and later 12.0S releases.

Control Plane Security and Packet QoS Overview

To protect the CP on a router from DoS attacks and to provide packet QoS, the Control Plane Policing feature treats the CP as a separate entity with its own ingress (input) and egress (output) ports, which are like ports on a router and switch. Because the Control Plane Policing feature treats the CP as a separate entity, a set of rules can be established and associated with the ingress and egress port of the CP.

These rules are applied only after the packet has been determined to have the CP as its destination or when a packet exits from the CP. Thereafter, you can configure a service policy to prevent unwanted packets from progressing after a specified rate limit has been reached; for example, a system administrator can limit all TCP/SYN packets that are destined for the CP to a maximum rate of 1 megabit per second.

Input CP services are executed after router input port services and a routing decision on the input path have been made. As shown in Figure 2, CP security and packet QoS are applied on:

- An aggregate level by the central switch engine and applied to all CP packets received from all line cards on the router (see Aggregate Control Plane Services, page 7)
- A distributed level by the distributed switch engine of a line card and applied to all CP packets received from all interfaces on the line card (see Distributed Control Plane Services, page 8)

Figure 2 Input Control Plane Services: Aggregate and Distributed Services

The following types of Layer 3 packets are forwarded to the control plane and processed by aggregate and distributed control plane policing:

- Routing protocol control packets
- Packets destined for the local IP address of the router
- Packets from management protocols (such as Simple Network Management Protocol [SNMP], Telnet, and secure shell [SSH])

Note Ensure that Layer 3 control packets have priority over other packet types that are destined for the control plane.

Aggregate Control Plane Services

Aggregate control plane services provide control plane policing for all CP packets received from all line-card interfaces on the router.

The central switch engine executes normal input port services and makes routing decisions for an incoming packet: if the packet is destined for the CP, aggregate services are performed. Because CP traffic from all line cards must pass through aggregate CP services, these services manage the cumulative amount of CP traffic that reaches the CP.

Aggregate CP service steps are as follows:

1. The line card receives a packet and delivers it to the central switch engine.

Note Before the packet is sent to the central switch engine, additional processing may be necessary for platforms that support hardware-level policing or platform-specific aggregate policing. It is possible that the packet may undergo multiple checks before it undergoes the generic Cisco IOS check.

2. The interfaces perform normal (interface-level) input port services and QoS.
3. The central switch engine performs Layer 3 switching or makes a routing decision, determining whether or not the packet is destined for the CP.
4. The central switch engine performs aggregate CP services for all CP packets.
5. On the basis of the results of the aggregate CP services, the central switch engine either drops the packet or delivers the packet to the CP for final processing.

Functionality Highlights of Aggregate CP Services

The following list highlights the functionality of aggregate CP services:

- Defined for a single input interface, such as the CP, and represents an aggregate for all ports on a router.
- Modular QoS is used to define CP services. Class maps and policy maps for both DoS protection and packet QoS are defined for a single aggregate CP service policy.
- Modular QoS does not prevent a single bad port from consuming all allocated bandwidth. Class maps that match an interface or subinterface may be able to constrain the contribution of each interface through an interface-specific policy map.

Distributed Control Plane Services

Distributed control plane services provide control plane policing for all CP packets received from the interfaces on a line card.

A distributed switch engine executes normal input port services and makes routing decisions for a packet: if the packet is destined for the CP, distributed CP services are performed. Afterwards, CP traffic from each line card is forwarded to the central switch engine where aggregate CP services are applied.

Note Distributed CP services may also forward conditioned packets to the central switch engine. In this case, aggregate CP services are also performed on the conditioned CP traffic.

Distributed CP service steps are as follows:

1. A line card receives a packet and delivers it to the distributed switch engine.
2. The distributed switch engine performs normal (interface-level) input port services and QoS.
3. The distributed switch engine performs Layer 2 or Layer 3 switching, or makes a routing decision, determining whether or not the packet is destined for the CP.
4. The distributed switch engine performs distributed CP services for all CP packets.
5. On the basis of the results of the distributed CP services, the distributed switch engine either drops the packet or marks the packet and delivers it to the central switch engine for further processing.
6. The central switch engine performs aggregate CP services and delivers the packet to the CP for final processing.

Functionality Highlights of Distributed CP Services

The following list highlights the functionality of distributed CP services:

- Distributed CP services are defined for a single input interface, such as the distributed CP, and represent an aggregate for all ports on a line card.
- Modular QoS is used to define CP services. Class maps and policy maps for both DoS protection and packet QoS are defined for a single distributed CP service policy. Each line card may have a unique CP service policy that applies traffic classifications, QoS policies and DoS services to packets received from all ports on the line card in an aggregate way.
- Modular QoS does not prevent one bad port from consuming all allocated bandwidth on a line card. Class maps that match an interface or subinterface may be able to constrain the contribution of each interface through an interface-specific policy map.

  Distributed CP services allow you to limit the number of CP packets forwarded from a line card to the central switch engine. The total amount of CP packets received from all line cards on a router may exceed aggregate CP levels.

Using Distributed CP Services

The purpose of CP protection and packet QoS is to apply sufficient control to the packets that reach the control plane. To successfully configure this level of CP protection, you must:

- Apply traditional QoS services using the modular QoS command-line interface to CP packets.
- Protect the path to the control plane against indiscriminate packet dropping due to resource exhaustion. If packets are not dropped according to user-defined QoS policies, but are dropped due to a resource limitation, the QoS policy is not maintained.

Distributed CP services allow you to configure specific CP services that are enforced at the line card level and required for the following reasons:

- While under a DoS attack, line card resources may be consumed. In this case, you must configure a drop policy to identify important packets. The drop policy ensures that all important packets arrive to the central switch engine for aggregate CP protection and later to the CP. Distributed CP services allow routers to apply the appropriate drop policy when resources are consumed, and therefore maintain the desired QoS priorities. If a line card indiscriminately drops packets, the aggregate CP filter becomes ineffective and the QoS priorities are no longer maintained.
- It is not possible to prevent one interface from consuming all aggregate CP resources. A DoS attack on one port may negatively impact CP processing of traffic from other ports. Distributed CP services allow you to limit the amount of important traffic forwarded by a line card to the CP. For example, you can configure a layered approach in which the combined rates of all line cards is over-subscribed compared to the aggregate rate. The rate of each individual line card would be below the aggregate rate, but combined together, the rates of all line cards exceed it. This over-subscription model is commonly used for other resource-related functions and helps limit the contribution of CP packets from any one line card.
- Distributed CP services provide for slot-level (line card) filtering. Customer-facing interfaces may have greater security requirements (with more restrictions or for billing reasons) than network-facing interfaces to backbone devices.
- Because distributed CP protection allows you to configure packet filters on a per-line card basis, processing cycles on line cards may offload aggregate level processing. You can configure Border Gateway Protocol (BGP) filtering at the distributed level for interfaces that use BGP, allowing the aggregate level to filter packets with the remaining filter requirements. Or you can configure identical filters for distributed and aggregate CP services with a distributed packet marking scheme

that informs the aggregate filter that a packet has already been checked. Distributed CP service processing further reduces aggregate processing and can significantly reduce the load on aggregate CP services.

Output Rate-Limiting and Silent Mode Operation

A router is automatically enabled to silently discard packets when you configure output policing on control plane traffic, using the **service-policy output** *policy-map-name* command.

Rate-limiting (policing) of output traffic from the CP is performed in silent mode. In silent mode, a router that is running Cisco IOS software operates without sending any system messages. If a packet that is exiting from the control plane is discarded for output policing, you do not receive an error message.

When control plane policing is configured for output traffic, error messages are not generated in the following cases:

- Traffic that is being transmitted to a port to which the router is not listening
- A connection to a legitimate address and port that is rejected because of a malformed request

Note The silent mode functionality and output policing on CP traffic are supported only in:
- Cisco IOS Release 12.2(25)S and later Cisco IOS 12.2S releases.
- Cisco IOS Release 12.3(4)T and later Cisco IOS 12.3T releases.
Silent mode and output policing on CP traffic are not supported for distributed control plane services.

How to Use the Control Plane Policing Feature

This section documents the following procedures:

Defining Aggregate Control Plane Services

Perform this task to configure aggregate CP services, such as packet rate control and silent packet discard, for the active route processor.

Prerequisites

Before you enter control-plane configuration mode to attach an existing QoS policy to the control plane, you must first create the policy using MQC to define a class map and policy map for control plane traffic.

For information about how to classify traffic and create a QoS policy, refer to the "Modular Quality of Service Command-Line Interface" chapter in the *Cisco IOS Quality of Service Solutions Configuration Guide*.

Restrictions

- Platform-specific restrictions, if any, are checked when the service policy is applied to the control plane interface.
- Support for output policing is available only in Cisco IOS Release 12.3(4)T and later T-train releases. (Note that output policing does not provide any performance benefits. It simply controls the information that is leaving the device.)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **control-plane**
4. **service-policy** {**input** | **output**} *policy-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**control-plane** **Example:** `Router(config)# control-plane`	Enters control-plane configuration mode to attach a QoS policy that manages CP traffic.
Step 4	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-cp)# service-policy input control-plane-policy`	Attaches a QoS service policy to the control plane. • **input**—Applies the specified service policy to packets received on the control plane. • **output**—Applies the specified service policy to packets transmitted from the control plane and enables the router to silently discard packets. • *policy-map-name*—Name of a service policy map (created using the **policy-map** command) to be attached. The name can be a maximum of 40 alphanumeric characters.

Defining Distributed Control Plane Services

Perform this task to configure distributed CP services, such as packet rate control, for packets that are destined for the CP and sent from the interfaces on a line card.

Prerequisites

Before you enter control-plane configuration mode to attach an existing QoS policy for performing distributed control-plane services, you must first create the policy using MQC to define a class map and policy map for control-plane traffic.

For information about how to classify traffic and create a QoS policy, refer to the "Modular Quality of Service Command-Line Interface" chapter in the *Cisco IOS Quality of Service Solutions Configuration Guide*.

Restrictions

- Platform-specific restrictions, if any, are checked when the service policy is applied to the control plane interface.
- Support for output policing is available only in Cisco IOS Release 12.3(4)T and later T-train releases. (Note that output policing does not provide any performance benefits. It simply controls the information that is leaving the device.)
- With Cisco IOS 12.2SX releases, Supervisor Engine 720 automatically installs the service policy on all DFC-equipped switching modules.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **control-plane** [**slot** *slot-number*]
4. **service-policy input** *policy-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**control-plane** [**slot** *slot-number*] **Example:** `Router(config)# control-plane slot 3`	Enters control-plane configuration mode to attach a QoS policy to manage CP traffic on the line card in the specified slot.
Step 4	**service-policy input** *policy-map-name* **Example:** `Router(config-cp)# service-policy input control-plane-policy`	Attaches a QoS service policy to filter and manage CP traffic on a specified line card before the aggregate CP policy is applied. • **input**—Applies the specified service policy using the distributed switch engine to CP packets received from all interfaces on the line card. • *policy-map-name*—Name of a service policy map (created using the **policy-map** command) to be attached. The name can be a maximum of 40 alphanumeric characters. **Note** The **service-policy output** *policy-map-name* command is not supported for applying a QoS policy for distributed control plane services.

Verifying Aggregate CP Services

To display information about the service policy attached to the control plane for aggregate CP services, perform the following optional steps.

SUMMARY STEPS

1. **enable**
2. **show policy-map control-plane** [**all**] [**input** [**class** *class-name*] | **output** [**class** *class-name*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show policy-map control-plane** [**all**] [**input** [**class** *class-name*]	**output** [**class** *class-name*]] **Example:** `Router# show policy-map control-plane all`	Displays information about the control plane. • **all**—Service policy information about all QoS policies used in aggregate and distributed CP services. • **input**—Statistics for the attached input policy. • **output**—Statistics for the attached output policy. • **class** *class-name*—Name of the traffic class whose configuration and statistics are displayed.

Examples

The following example shows that the policy map TEST is associated with the control plane. This policy map polices traffic that matches the class map TEST, while allowing all other traffic (that matches the class map "class-default") to go through as is. (Table 1 describes the significant fields shown in the display.)

```
Router# show policy-map control-plane

Control Plane

Service-policy input:TEST

Class-map:TEST (match-all)
      20 packets, 11280 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:access-group 101
      police:
        8000 bps, 1500 limit, 1500 extended limit
        conformed 15 packets, 6210 bytes; action:transmit
        exceeded 5 packets, 5070 bytes; action:drop
        violated 0 packets, 0 bytes; action:drop
        conformed 0 bps, exceed 0 bps, violate 0 bps

Class-map:class-default (match-any)
      105325 packets, 11415151 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:any
```

Verifying Distributed CP Services

To display information about the service policy attached to the control plane to perform distributed CP services, perform the following optional steps.

SUMMARY STEPS

1. **enable**
2. **show policy-map control-plane** [**all** | **slot** *slot-number*] [**input** [**class** *class-name*] | **output** [**class** *class-name*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show policy-map control-plane** [**all**] [**slot** *slot-number*] [**input** [**class** *class-name*] | **output** [**class** *class-name*]] **Example:** `Router# show policy-map control-plane slot 2`	Displays information about the service policy used to apply distributed CP services on the router. • **all**—Service policy information about all QoS policies used in aggregate and distributed CP services. • **slot** *slot-number*—Service policy information about the QoS policy used to perform distributed CP services on the specified line card. • **input**—Statistics for the attached input policy. • **output**—Statistics for the attached output policy. • **class** *class-name*—Name of the traffic class whose configuration and statistics are displayed.

Examples

The following example shows how to display information about the classes of CP traffic received from all interfaces on the line card in slot 1 to which the policy map TESTII is applied for distributed CP services. This policy map polices traffic that matches the traffic class TESTII, while allowing all other traffic (that matches the class map “class-default”) to go through as is. (Table 1 describes the significant fields shown in the display.)

```
Router# show policy-map control-plane slot 1

Control Plane - slot 1

Service-policy input: TESTII (1048)

Class-map: TESTII (match-all) (1049/4)
        0 packets, 0 bytes
        5 minute offered rate 0 bps, drop rate 0 bps
        Match: protocol arp (1050)
        police:
            cir 8000 bps, bc 4470 bytes, be 4470 bytes
          conformed 0 packets, 0 bytes; actions:
            transmit
          exceeded 0 packets, 0 bytes; actions:
            drop
          violated 0 packets, 0 bytes; actions:
            drop
          conformed 0 bps, exceed 0 bps, violate 0 bps

Class-map: class-default (match-any) (1052/0)
        0 packets, 0 bytes
        5 minute offered rate 0 bps, drop rate 0 bps
        Match: any  (1053)
```

Configuration Examples for Control Plane Policing

This section contains examples that shows how to configure aggregate control plane services on both an input and an output interface:

Configuring Control Plane Policing on Input Telnet Traffic: Example

The following example shows how to apply a QoS policy for aggregate CP services to Telnet traffic received on the control plane. Trusted hosts with source addresses 10.1.1.1 and 10.1.1.2 forward Telnet packets to the control plane without constraint, while allowing all remaining Telnet packets to be policed at the specified rate:

```
! Allow 10.1.1.1 trusted host traffic.
Router(config)# access-list 140 deny tcp host 10.1.1.1 any eq telnet
! Allow 10.1.1.2 trusted host traffic.
Router(config)# access-list 140 deny tcp host 10.1.1.2 any eq telnet
! Rate limit all other Telnet traffic.
Router(config)# access-list 140 permit tcp any any eq telnet
! Define class-map “telnet-class.”
Router(config)# class-map telnet-class
Router(config-cmap)# match access-group 140
Router(config-cmap)# exit
Router(config)# policy-map control-plane-in
Router(config-pmap)# class telnet-class
Router(config-pmap-c)# police 80000 conform transmit exceed drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
! Define aggregate control plane service for the active Route Processor.
Router(config)# control-plane
Router(config-cp)# service-policy input control-plane-in
Router(config-cp)# exit
```

Configuring Control Plane Policing on Output ICMP Traffic: Example

The following example shows how to apply a QoS policy for aggregate CP services to Telnet traffic transmitted from the control plane. Trusted networks with source addresses 3.3.3.0 and 4.4.4.0 receive Internet Control Management Protocol (ICMP) port-unreachable responses without constraint, while allowing all remaining ICMP port-unreachable responses to be dropped:

```
! Allow 3.3.3.0 trusted network traffic.
Router(config)# access-list 141 deny icmp 3.3.3.0 0.0.0.255 any port-unreachable
! Allow 4.4.4.0 trusted network traffic.
Router(config)# access-list 141 deny icmp 4.4.4.0 0.0.0.255 any port-unreachable
! Rate limit all other ICMP traffic.
Router(config)# access-list 141 permit icmp any any port-unreachable
Router(config)# class-map icmp-class
Router(config-cmap)# match access-group 141
Router(config-cmap)# exit
Router(config)# policy-map control-plane-out
! Drop all traffic that matches the class "icmp-class."
Router(config-pmap)# class icmp-class
Router(config-pmap-c)# drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
```

```
Router(config)# control-plane
! Define aggregate control plane service for the active route processor.
Router(config-cp)# service-policy output control-plane-out
Router(config-cp)# exit
```

Additional References

The following sections provide references related to Control Plane Policing.

Related Documents

Related Topic	Document Title
QoS information and configuration tasks	*Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.3
Additional QoS commands	*Cisco IOS Quality of Service Solutions Command Reference*, Release 12.3T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
• CISCO-CLASS-BASED-QOS-MIB **Note** Supported only in Cisco IOS Release 12.3(7)T.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator, found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
The Cisco Technical Support website, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.cisco.com/ techsupport

Command Reference

The following commands are introduced or modified in the feature or features

- **control-plane**

- **service-policy (control-plane)**
- **show policy-map control-plane**

For information about these commands, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html.

For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Role-Based CLI Access

The Role-Based CLI Access feature allows the network administrator to define "views," which are a set of operational commands and configuration capabilities that provide selective or partial access to Cisco IOS EXEC and configuration (Config) mode commands. Views restrict user access to Cisco IOS command-line interface (CLI) and configuration information; that is, a view can define what commands are accepted and what configuration information is visible. Thus, network administrators can exercise better control over access to Cisco networking devices.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Role-Based CLI Access" section on page 14.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Role-Based CLI Access

Your image must support CLI views.

Restrictions for Role-Based CLI Access

Lawful Intercept Images Limitation

Because CLI views are a part of the Cisco IOS parser, CLI views are a part of all platforms and Cisco IOS images. However, the lawful intercept view is available only in images that contain the lawful intercept subsystem.

Maximum Number of Allowed Views

The maximum number of CLI views and superviews, including one lawful intercept view, that can be configured is 15. (This does not include the root view.)

Information About Role-Based CLI Access

To create and use views, you should understand the following concepts:

- Benefits of Using CLI Views, page 2
- Root View, page 2
- View Authentication via a New AAA Attribute, page 3

Benefits of Using CLI Views

Views: Detailed Access Control

Although users can control CLI access via both privilege levels and enable mode passwords, these functions do not provide network administrators with the necessary level of detail needed when working with Cisco IOS routers and switches. CLI views provide a more detailed access control capability for network administrators, thereby, improving the overall security and accountability of Cisco IOS software.

As of Cisco IOS Release 12.3(11)T, network administrators can also specify an interface or a group of interfaces to a view; thereby, allowing access on the basis of specified interfaces.

Root View

When a system is in “root view,” it has all of the access privileges as a user who has level 15 privileges. If the administrator wishes to configure any view to the system (such as a CLI view, a superview, or a lawful intercept view), the system must be in root view.

The difference between a user who has level 15 privileges and a root view user is that a root view user can configure a new view and add or remove commands from the view. Also, when you are in a CLI view, you have access only to the commands that have been added to that view by the root view user.

View Authentication via a New AAA Attribute

View authentication is performed by an external authentication, authorization, and accounting (AAA) server via the new attribute “cli-view-name.”

AAA authentication associates only one view name to a particular user; that is, only one view name can be configured for a user in an authentication server.

How to Use Role-Based CLI Access

This section contains the following procedures:

- Configuring a CLI View, page 3 (required)
- Configuring a Lawful Intercept View, page 5 (optional)
- Configuring a Superview, page 7 (optional)
- Monitoring Views and View Users, page 9 (optional)

Configuring a CLI View

Use this task to create a CLI view and add commands or interfaces to the view, as appropriate.

Prerequisites

Before you create a view, you must perform the following tasks:

- Enable AAA via the **aaa new-model** command.
- Ensure that your system is in root view—not privilege level 15.

SUMMARY STEPS

1. **enable view**
2. **configure terminal**
3. **parser view** *view-name*
4. **secret 5** *encrypted-password*
5. **commands** *parser-mode* {**include** | **include-exclusive** | **exclude**} [**all**] [**interface** *interface-name* | *command*]
6. **exit**
7. **exit**
8. **enable** [*privilege-level*] [**view** *view-name*]
9. **show parser view** [**all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable view** **Example:** `Router> enable view`	Enables root view. • Enter your privilege level 15 password (for example, root password) if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parser view** *view-name* **Example:** `Router(config)# parser view first`	Creates a view and enters view configuration mode.
Step 4	**secret 5** *encrypted-password* **Example:** `Router(config-view)# secret 5 secret`	Associates a command-line interface (CLI) view or superview with a password. **Note** You must issue this command before you can configure additional attributes for the view.
Step 5	**commands** *parser-mode* {**include** | **include-exclusive** | **exclude**} [**all**] [**interface** *interface-name* | *command*] **Example:** `Router(config-view)# commands exec include show version`	Adds commands or interfaces to a view. • *parser-mode*—The mode in which the specified command exists. • **include**—Adds a command or an interface to the view and allows the same command or interface to be added to an additional view. • **include-exclusive**—Adds a command or an interface to the view and excludes the same command or interface from being added to all other views. • **exclude**—Excludes a command or an interface from the view; that is, customers cannot access a command or an interface. • **all**—A “wildcard” that allows every command in a specified configuration mode that begins with the same keyword or every subinterface for a specified interface to be part of the view. • **interface** *interface-name*— Interface that is added to the view. • *command*—Command that is added to the view.
Step 6	**exit** **Example:** `Router(config-view)# exit`	Exits view configuration mode.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 8	**enable** [*privilege-level*] [**view** *view-name*] **Example:** `Router# enable view first`	Prompts the user for a password, which allows the user to access a configured CLI view, and is used to switch from one view to another view. After the correct password is given, the user can access the view.
Step 9	**show parser view** [**all**] **Example:** `Router# show parser view`	(Optional) Displays information about the view that the user is currently in. • **all**—Displays information for all views that are configured on the router. **Note** Although this command is available for both root and lawful intercept users, the **all** keyword is available only to root users. However, the **all** keyword can be configured by a user in root view to be available for users in lawful intercept view and CLI view.

Troubleshooting Tips

After you have successfully created a view, a system message such as the following will be displayed:

```
%PARSER-6-VIEW_CREATED: view ‘first’ successfully created.
```

After you have successfully deleted a view, a system message such as the following will be displayed:

%PARSER-6-VIEW_DELETED: view ‘first’ successfully deleted.

You must associate a password with a view. If you do not associate a password, and you attempt to add commands to the view via the **commands** command, a system message such as the following will be displayed:

```
%Password not set for view <viewname>.
```

Configuring a Lawful Intercept View

Use this task to initialize and configure a view for lawful-intercept-specific commands and configuration information. (Only an administrator or a user who has level 15 privileges can initialize a lawful intercept view.)

About Lawful Intercept Views

Like a CLI view, a lawful intercept view restricts access to specified commands and configuration information. Specifically, a lawful intercept view allows a user to secure access to lawful intercept commands that are held within the TAP-MIB, which is a special set of simple network management protocol (SNMP) commands that store information about calls and users.

Commands available in lawful intercept view belong to one of the following categories:

- Lawful intercept commands that should not be made available to any other view or privilege level
- CLI views that are useful for lawful intercept users but do not have to be excluded from other views or privilege levels

Prerequisites

Before you initialize a lawful intercept view, ensure that the privilege level is set to 15 via the **privilege** command.

SUMMARY STEPS

1. **enable view**
2. **configure terminal**
3. **li-view** *li-password* **user** *username* **password** *password*
4. **username** [**lawful-intercept**] *name* [**privilege** *privilege-level* | **view** *view-name*] **password** *password*
5. **parser view** *view-name*
6. **secret 5** *encrypted-password*
7. **name** *new-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable view** **Example:** `Router> enable view`	Enables root view. • Enter your privilege level 15 password (for example, root password) if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**li-view** *li-password* **user** *username* **password** *password* **Example:** `Router(config)# li-view lipass user li_admin password li_adminpass`	Initializes a lawful intercept view. After the li-view is initialized, you must specify at least one user via **user** *username* **password** *password* options.
Step 4	**username** [**lawful-intercept** [*name*] [**privilege** *privilege-level*	**view** *view-name*] **password** *password* **Example:** `Router(config)# username lawful-intercept li-user1 password li-user1pass`	Configures lawful intercept users on a Cisco device.

	Command or Action	Purpose
Step 5	**parser view** *view-name* **Example:** `Router(config)# parser view li view name`	(Optional) Enters view configuration mode, which allows you to change the lawful intercept view password or the lawful intercept view name.
Step 6	**secret 5** *encrypted-password* **Example:** `Router(config-view)# secret 5 secret`	(Optional) Changes an existing password for a lawful intercept view.
Step 7	**name** *new-name* **Example:** `Router(config-view)# name second`	(Optional) Changes the name of a lawful intercept view. If this command is not issued, the default name of the lawful intercept view is "li-view."

Troubleshooting Tips

To display information for all users who have access to a lawful intercept view, issue the **show users lawful-intercept** command. (This command is available only to authorized lawful intercept view users.)

Configuring a Superview

Use this task to create a superview and add at least one CLI view to the superview.

About Superviews

A superview consists of one or more CLI views, which allow users to define what commands are accepted and what configuration information is visible. Superviews allow a network administrator to easily assign all users within configured CLI views to a superview instead of having to assign multiple CLI views to a group of users.

Superviews contain the following characteristics:

- A CLI view can be shared among multiple superviews.
- Commands cannot be configured for a superview; that is, you must add commands to the CLI view and add that CLI view to the superview.
- Users who are logged into a superview can access all of the commands that are configured for any of the CLI views that are part of the superview.
- Each superview has a password that is used to switch between superviews or from a CLI view to a superview.
- If a superview is deleted, all CLI views associated with that superview will not be deleted too.

Adding CLI Views to a Superview

You can add a view to a superview only after a password has been configured for the superview (via the **secret 5** command). Thereafter, issue the **view** command in view configuration mode to add at least one CLI view to the superview.

Note Before adding a CLI view to a superview, ensure that the CLI views that are added to the superview are valid views in the system; that is, the views have been successfully created via the **parser view** command.

SUMMARY STEPS

1. **enable view**
2. **configure terminal**
3. **parser view** *superview-name* **superview**
4. **secret 5** *encrypted-password*
5. **view** *view-name*
6. **exit**
7. **exit**
8. **show parser view** [**all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable view** **Example:** `Router> enable view`	Enables root view. • Enter your privilege level 15 password (for example, root password) if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parser view** *superview-name* **superview** **Example:** `Router(config)# parser view su_view1 superview`	Creates a superview and enters view configuration mode.
Step 4	**secret 5** *encrypted-password* **Example:** `Router(config-view)# secret 5 secret`	Associates a CLI view or superview with a password. **Note** You must issue this command before you can configure additional attributes for the view.
Step 5	**view** *view-name* **Example:** `Router(config-view)# view view_three`	Adds a normal CLI view to a superview. Issue this command for each CLI view that is to be added to a given superview.
Step 6	**exit** **Example:** `Router(config-view)# exit`	Exits view configuration mode.

	Command or Action	Purpose
Step 7	`exit` **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 8	`show parser view [all]` **Example:** `Router# show parser view`	(Optional) Displays information about the view that the user is currently in. • **all**—Displays information for all views that are configured on the router. **Note** Although this command is available for both root and lawful intercept users, the **all** keyword is available only to root users. However, the **all** keyword can be configured by a user in root view to be available for users in lawful intercept view and CLI view.

Monitoring Views and View Users

To display debug messages for all views—root, CLI, lawful intercept, and super, use the **debug parser view** command in privileged EXEC mode.

Configuration Examples for Role-Based CLI Access

This section contains the following configuration examples:

Configuring a CLI View: Example

The following example shows how to configure two CLI views, “first” and “second.” Thereafter, you can verify the CLI view in the running configuration.

```
Router(config)# parser view first
00:11:40:%PARSER-6-VIEW_CREATED:view 'first' successfully created.
Router(config-view)# secret 5 firstpass
Router(config-view)# command exec include show version
Router(config-view)# command exec include configure terminal
Router(config-view)# command exec include all show ip
Router(config-view)# exit
Router(config)# parser view second
00:13:42:%PARSER-6-VIEW_CREATED:view 'second' successfully created.
Router(config-view)# secret 5 secondpass
Router(config-view)# command exec include-exclusive show ip interface
Router(config-view)# command exec include logout
Router(config-view)# exit
!
```

```
!
Router(config-view)# do show run | beg view
parser view first
 secret 5 $1$MCmh$QuZaU8PIMPlff9sFCZvgW/
 commands exec include configure terminal
 commands exec include configure
 commands exec include all show ip
 commands exec include show version
 commands exec include show
!
parser view second
 secret 5 $1$iP2M$R16BXKecMEiQesxLyqygW.
 commands exec include-exclusive show ip interface
 commands exec include show ip
 commands exec include show
 commands exec include logout
!
```

Verifying a CLI View: Example

After you have configured the CLI views “first” and “second,” you can issue the **enable view** command to verify which commands are available in each view. The following example shows which commands are available inside the CLI view “first” after the user has logged into this view. (Because the **show ip** command is configured with the all option, a complete set of suboptions is shown, except the **show ip interface** command, which is using the include-exclusive keyword in the second view.)

```
Router# enable view first
Password:

00:28:23:%PARSER-6-VIEW_SWITCH:successfully set to view 'first'.
Router# ?
Exec commands:
  configure  Enter configuration mode
  enable     Turn on privileged commands
  exit       Exit from the EXEC
  show       Show running system information

Router# show ?

  ip       IP information
  parser   Display parser information
  version  System hardware and software status

Router# show ip ?

  access-lists           List IP access lists
  accounting             The active IP accounting database
  aliases                IP alias table
  arp                    IP ARP table
  as-path-access-list    List AS path access lists
  bgp                    BGP information
  cache                  IP fast-switching route cache
  casa                   display casa information
  cef                    Cisco Express Forwarding
  community-list         List community-list
  dfp                    DFP information
  dhcp                   Show items in the DHCP database
  drp                    Director response protocol
  dvmrp                  DVMRP information
  eigrp                  IP-EIGRP show commands
  extcommunity-list      List extended-community list
```

```
  flow                    NetFlow switching
  helper-address          helper-address table
  http                    HTTP information
  igmp                    IGMP information
  irdp                    ICMP Router Discovery Protocol
.
.
.
```

Configuring a Lawful Intercept View: Example

The following example shows how to configure a lawful intercept view, add users to the view, and verify the users that were added:

```
!Initialize the LI-View.
Router(config-view)# li-view lipass user li_admin password li_adminpass
00:19:25:%PARSER-6-LI_VIEW_INIT:LI-View initialized.
Router(config-view)# end

! Enter the LI-View; that is, check to see what commands are available within the view.
Router# enable view li-view
Password:

Router#
00:22:57:%PARSER-6-VIEW_SWITCH:successfully set to view 'li-view'.
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# parser view li-view
Router(config-view)# ?
View commands:
  commands  Configure commands for a view
  default   Set a command to its defaults
  exit      Exit from view configuration mode
  name      New LI-View name      ===This option only resides in LI View.
  no        Negate a command or set its defaults
  password  Set a password associated with CLI views

Router(config-view)#

! NOTE:LI View configurations are never shown as part of ‘running-configuration’.

! Configure LI Users.
Router(config)# username lawful-intercept li-user1 password li-user1pass
Router(config)# username lawful-intercept li-user2 password li-user2pass

! Displaying LI User information.
Router# show users lawful-intercept

li_admin
li-user1
li-user2
Router#
```

Configuring a Superview: Example

The following sample output from the **show running-config** command shows that “view_one” and “view_two” have been added to superview “su_view1,” and “view_three” and “view_four” have been added to superview “su_view2”:

```
!
parser view su_view1 superview
 secret 5 <encoded password>
 view view_one
 view view_two
!
parser view su_view2 superview
 secret 5 <encoded password>
 view view_three
 view view_four
!
```

Additional References

The following sections provide references related to Role-Based CLI Access.

Related Documents

Related Topic	Document Title
SNMP, MIBs, CLI configuration	The chapter “Configuring SNMP” in the *Cisco IOS Network Management Configuration Guide.*
Privilege levels	The chapter “Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices” in the *Cisco IOS Security Configuration Guide*.

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **commands (view)**
- **enable**
- **li-view**
- **name (view)**
- **parser view**
- **parser view superview**
- **secret**
- **show parser view**
- **show users**
- **username**
- **view**

Feature Information for Role-Based CLI Access

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS, Catalyst OS, and Cisco IOS XE software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Role-Based CLI Access

Feature Name	Releases	Feature Information
Role-Based CLI Access	12.3(7)T 12.3(11)T 12.2(33)SRB 12.2(33)SB Cisco IOS XE Release 2.1 12.2(33)SXI	This feature enables network administrators to restrict user access to CLI and configuration information. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Routers In 12.3(11)T, the CLI view capability was extended to restrict user access on a per-interface level, and additional CLI views were introduced to support the extended view capability. Also, support to group configured CLI views into a superview was introduced.

Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices

Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices

Cisco IOS based networking devices provide several features that can be used to implement basic security for CLI sessions using only the operating system running on the device. These features include the following:

- Different levels of authorization for CLI sessions to control access to commands that can modify the status of the networking device versus commands that are used to monitor the device
- Assigning passwords to CLI sessions
- Requiring users log in to a networking device with a username
- Changing the privilege levels of commands to create new authorization levels for CLI sessions.

This module is a guide to implementing a baseline level of security for your networking devices. It focuses on the least complex options available for implementing a baseline level of security. If you have networking devices installed in your network with no security options configured, or you are about to install a networking device and you need help understanding the how to implement a baseline of security, this document will help you.

Module History

This module was first published on May 2nd, 2005, and last updated on May 2nd, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices" section on page 42.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

Your networking device must not be configured to use any local or remote authentication, authorization, and accounting (AAA) security features. This document describes only the non-AAA security features that can be configured locally on the networking device.

For information how to configure AAA security features that can be run locally on a networking device, or for information on how to configure remote AAA security using TACACS+ or RADIUS servers, see the *Cisco IOS Security Configuration Guide*, Release 12.4.

Information About Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

To configure router security with passwords, CLI privilege levels and usernames, you should understand the following concepts:

Benefits of Creating a Security Scheme for Your Networking Device

The foundation of a good security scheme in the network is the protection of the user interfaces of the networking devices from unauthorized access. Protecting access to the user interfaces on your networking devices prevents unauthorized users from making configuration changes that can disrupt the stability of your network or compromise your network security.

The Cisco IOS features described in this document can be combined in many different ways to create a unique security scheme for each of your networking devices. Here are some possible examples that you can configure:

- You can enable non administrative users to run a subset of the administrative commands available on the networking device by lowering the entitlement level for the commands to the non administrative privilege level. This can be useful for the following scenarios:
  - ISPs that want their first-line technical support staff to perform tasks such as enabling new interfaces for new customers or resetting the connection for a customer whose connection has stopped passing traffic. See the "Configuring and Verifying a Networking Device to Allow Non Administrative Users to Shutdown and Enable Interfaces: Example" section on page 38 section for an example of how to do this.
  - When you want your first-line technical support staff to have the ability to clear console port sessions that were disconnected improperly from a terminal server. See the "Configuring and Verifying a Networking Device to Allow Non Administrative Users to Clear Remote CLI Sessions: Example" section on page 37 section for an example of how to do this.
  - When you want your first-line technical support staff to have the ability to view, but not change, the configuration of a networking device to facilitate troubleshooting a networking problem. See the "Configuring and Verifying a Networking Device to Allow Non Administrative Users to View the Running Configuration Automatically: Example" section on page 38 section for an example of how to do this.

Cisco IOS CLI Modes

To aid in the configuration of Cisco devices, the Cisco IOS command-line interface is divided into different command modes. Each command mode has its own set of commands available for the configuration, maintenance, and monitoring of router and network operations. The commands available to you at any given time depend on the mode you are in. Entering a question mark (**?**) at the system prompt (router prompt) allows you to obtain a list of commands available for each command mode.

The use of specific commands allows you to navigate from one command mode to another. The standard order in which a user would access the modes is as follows: user EXEC mode; privileged EXEC mode; global configuration mode; specific configuration modes; configuration submodes; and configuration subsubmodes.

Note The default configuration of a Cisco IOS software based networking device only allows you to configure passwords to protect access to user EXEC mode (for local, and remote CLI sessions) and privileged EXEC mode. This document describes how you can provide additional levels of security by protecting access to other modes, and commands, using a combination of usernames, passwords and the **privilege** command.

Most EXEC mode commands are one-time commands, such as **show** or **more** commands, which show the current configuration status, and **clear** commands, which clear counters or interfaces. EXEC mode commands are not saved across reboots of the router.

From privileged EXEC mode, you can enter *global configuration mode*. In this mode, you can enter commands that configure general system characteristics. You also can use global configuration mode to enter specific configuration modes. Configuration modes, including global configuration mode, allow you to make changes to the running configuration. If you later save the configuration, these commands are stored across router reboots.

From global configuration mode you can enter a variety of protocol-specific or feature-specific configuration modes. The CLI hierarchy requires that you enter these specific configuration modes only through global configuration mode. For example, *interface configuration mode*, is a commonly used configuration mode.

From configuration modes, you can enter configuration submodes. Configuration submodes are used for the configuration of specific features within the scope of a given configuration mode. As an example, this chapter describes the *subinterface configuration mode*, a submode of the interface configuration mode.

ROM monitor mode is a separate mode used when the router cannot boot properly. If your system (router, switch, or access server) does not find a valid system image to load when it is booting, the system will enter ROM monitor mode. ROM monitor (ROMMON) mode can also be accessed by interrupting the boot sequence during startup. ROMMON is not covered in this document because it does not have any security features available in it.

The following sections contain detailed information on these command modes:

- User EXEC Mode
- Privileged EXEC Mode
- Global Configuration Mode
- Interface Configuration Mode
- Subinterface Configuration Mode

User EXEC Mode

When you start a session on a router, you generally begin in *user EXEC mode*, which is one of two access levels of the EXEC mode. For security purposes, only a limited subset of EXEC commands are available in user EXEC mode. This level of access is reserved for tasks that do not change the configuration of the router, such as determining the router status.

If your device is configured to require users to log-in the log-in process will require a username and a password. You may try three times to enter a password before the connection attempt is refused.

User EXEC mode is set by default to privilege level 1. Privileged EXEC mode is set by default to privilege level 15. For more information see the "Privileged EXEC Mode" section on page 6. When you are logged into a networking device in user EXEC mode your session is running at privilege level 1. By default the EXEC commands at privilege level 1 are a subset of those available at privilege level 15. When you are logged into a networking device in privileged EXEC mode your session is running at privilege level 15. You can move commands to any privilege level between 1 and 15 using the **privilege** command. See the "Cisco IOS Privilege Levels" section on page 13 for more information on privilege levels and the **privilege** command.

In general, the user EXEC commands allow you to connect to remote devices, change terminal line settings on a temporary basis, perform basic tests, and list system information.

To list the available user EXEC commands, use the following command:

Command	Purpose
`Router(config)# ?`	Lists the user EXEC mode commands

The user EXEC mode prompt consists of the host name of the device followed by an angle bracket (>), as shown in the following example:

```
Router>
```

The default host name is generally `Router,` unless it has been changed during initial configuration using the **setup** EXEC command. You also change the host name using the **hostname** global configuration command.

Note Examples in Cisco IOS documentation assume the use of the default name of "Router." Different devices (for example, access servers) may use a different default name. If the routing device (router, access server, or switch) has been named with the **hostname** command, that name will appear as the prompt instead of the default name.

To list the commands available in user EXEC mode, enter a question mark (**?**) as shown in the following example:

```
Router> ?
Exec commands:
 <1-99>           Session number to resume
 connect          Open a terminal connection
 disconnect       Disconnect an existing telnet session
 enable           Turn on privileged commands
 exit             Exit from Exec mode
 help             Description of the interactive help system
 lat              Open a lat connection
 lock             Lock the terminal
 login            Log in as a particular user
 logout           Exit from Exec mode and log out
 menu             Start a menu-based user interface
 mbranch          Trace multicast route for branch of tree
 mrbranch         Trace reverse multicast route to branch of tree
 mtrace           Trace multicast route to group
 name-connection  Name an existing telnet connection
 pad              Open a X.29 PAD connection
 ping             Send echo messages
 resume           Resume an active telnet connection
 show             Show running system information
 systat           Display information about terminal lines
 telnet           Open a telnet connection
 terminal         Set terminal line parameters
 tn3270           Open a tn3270 connection
 trace            Trace route to destination
 where            List active telnet connections
 x3               Set X.3 parameters on PAD
```

The list of commands will vary depending on the software feature set and router platform you are using.

Note You can enter commands in uppercase, lowercase, or mixed case. Only passwords are case sensitive. However, Cisco IOS documentation convention is to always present commands in lowercase.

Privileged EXEC Mode

In order to have access to all commands, you must enter *privileged EXEC mode*, which is the second level of access for the EXEC mode. Normally, you must enter a password to enter privileged EXEC mode. In privileged EXEC mode, you can enter any EXEC command, because privileged EXEC mode is a superset of the user EXEC mode commands.

Because many privileged EXEC mode commands set operating parameters, privileged EXEC level access should be password protected to prevent unauthorized use. The privileged EXEC command set includes those commands contained in user EXEC mode. Privileged EXEC mode also provides access to configuration modes through the **configure** command, and includes advanced testing commands, such as **debug**.

Privileged EXEC mode is set by default to privilege level 15. User EXEC mode is set by default to privilege level 1. For more information see the "User EXEC Mode" section on page 4. When you are logged into a networking device in privileged EXEC mode your session is running at privilege level 15. When you are logged into a networking device in user EXEC mode your session is running at privilege level 1. By default the EXEC commands at privilege level 15 are a superset of those available at privilege level 1. You can move commands to any privilege level between 1 and 15 using the **privilege** command. See the "Cisco IOS Privilege Levels" section on page 13 for more information on privilege levels and the **privilege** command.

The privileged EXEC mode prompt consists of the host name of the device followed by a pound sign(#), as shown in the following example:

```
Router#
```

To access privileged EXEC mode, use the following command:

Command	Purpose
`Router> enable` `Password` `Router# exit` `Router>`	Enables privileged EXEC mode. • If a privileged EXEC mode password has been configured the system will prompt you for a password after you issue the enable command. • Use the exit command to leave privileged EXEC mode.

Note Privileged EXEC mode is sometimes referred to as "enable mode," because the **enable** command is used to enter the mode.

If a password has been configured on the system, you will be prompted to enter it before being allowed access to privileged EXEC mode. The password is not displayed on the screen and is case sensitive. If an enable password has not been set, privileged EXEC mode can be accessed only by a local CLI session (terminal connected to the console port).

If you attempt to access privileged EXEC mode on a router over a remote connection, such as a telnet connection, and you have not configured a password for privileged EXEC mode you will see the **% No password set** error message. For more information on remote connections see the "Remote CLI Sessions" section on page 10. The system administrator uses the **enable secret** or **enable password** global configuration commands to set the password that restricts access to privileged EXEC mode. For information on configuring a password for privileged EXEC mode, see the "Protecting Access to Privileged Exec Mode" section on page 20.

To return to user EXEC mode, use the following command:

Command	Purpose
`Router# disable`	Exits from privileged EXEC mode to user EXEC mode.

The following example shows the process of accessing privileged EXEC mode:

```
Router> enable
Password:<letmein>
Router#
```

Note that the password will not be displayed as you type, but is shown here for illustrational purposes.To list the commands available in privileged EXEC mode, issue the **?** command at the prompt. From privileged EXEC mode you can access global configuration mode, which is described in the following section.

Note Because the privileged EXEC command set contains all of the commands available in user EXEC mode, some commands can be entered in either mode. In Cisco IOS documentation, commands that can be entered in either user EXEC mode or privileged EXEC mode are referred to as EXEC mode commands. If user or privileged is not specified in the documentation, assume that you can enter the referenced commands in either mode.

Global Configuration Mode

The term “global” is used to indicate characteristics or features that affect the system as a whole. Global configuration mode is used to configure your system globally, or to enter specific configuration modes to configure specific elements such as interfaces or protocols. Use the **configure terminal** privileged EXEC command to enter global configuration mode.

To access global configuration mode, use the following command in privileged EXEC mode:

Command	Purpose
`Router# configure terminal`	From privileged EXEC mode, enters global configuration mode.

The following example shows the process of entering global configuration mode from privileged EXEC mode:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Note that the system prompt changes to indicate that you are now in global configuration mode. The prompt for global configuration mode consists of the host-name of the device followed by `(config)` and the pound sign (#). To list the commands available in privileged EXEC mode, issue the **?** command at the prompt.

Commands entered in global configuration mode update the running configuration file as soon as they are entered. In other words, changes to the configuration take effect each time you press the Enter or Return key at the end of a valid command. However, these changes are not saved into the startup configuration file until you issue the **copy running-config startup-config** EXEC mode command. This behavior is explained in more detail later in this document.

As shown in the example above, the system dialogue prompts you to end your configuration session (exit configuration mode) by pressing the Control (Ctrl) and "z" keys simultaneously; when you press these keys, **^Z** is printed to the screen. You can actually end your configuration session by entering the Ctrl-Z key combination, using the **end** command, using the Ctrl-C key combination. The **end** command is the recommended way to indicate to the system that you are done with the current configuration session.

Caution If you use Ctrl-Z at the end of a command line in which a valid command has been typed, that command will be added to the running configuration file. In other words, using Ctrl-Z is equivalent to hitting the Enter (Carriage Return) key before exiting. For this reason, it is safer to end your configuration session using the **end** command. Alternatively, you can use the Ctrl-C key combination to end your configuration session without sending a Carriage Return signal.

You can also use the **exit** command to return from global configuration mode to EXEC mode, but this only works in global configuration mode. Pressing Ctrl-Z or entering the **end** command will always take you back to EXEC mode regardless of which configuration mode or configuration submode you are in.

To exit global configuration command mode and return to privileged EXEC mode, use one of the following commands:

Command	Purpose
`Router(config)# end` or `Router(config)# ^Z`	Ends the current configuration session and returns to privileged EXEC mode.
`Router(config)# exit`	Exits the current command mode and returns to the preceding mode. For example, exits from global configuration mode to privileged EXEC mode.

From global configuration mode, you can enter a number of protocol-specific, platform-specific, and feature-specific configuration modes.

Interface configuration mode, described in the following section, is an example of a configuration mode you can enter from global configuration mode.

Interface Configuration Mode

One example of a specific configuration mode you enter from global configuration mode is interface configuration mode.

Many features are enabled on a per-interface basis. Interface configuration commands modify the operation of an interface such as an Ethernet, FDDI, or serial port. Interface configuration commands always follow an **interface** global configuration command, which defines the interface type.

For details on interface configuration commands that affect general interface parameters, such as bandwidth or clock rate, refer to the Release 12.2 *Cisco IOS Interface Configuration Guide*. For protocol-specific commands, refer to the appropriate Cisco IOS software command reference.

To access and list the interface configuration commands, use the following command:

Command	Purpose
`Router(config)# interface` *type number*	Specifies the interface to be configured, and enters interface configuration mode.

In the following example, the user enters interface configuration mode for serial interface 0. The new prompt, *hostname*(config-if)#, indicates interface configuration mode.

```
Router(config)# interface serial 0
Router(config-if)#
```

To exit interface configuration mode and return to global configuration mode, enter the **exit** command.

Configuration submodes are configuration modes entered from other configuration modes (besides global configuration mode). Configuration submodes are for the configuration of specific elements within the configuration mode. One example of a configuration submode is subinterface configuration mode, described in the following section.

Subinterface Configuration Mode

From interface configuration mode, you can enter subinterface configuration mode. Subinterface configuration mode is a submode of interface configuration mode. In subinterface configuration mode you can configure multiple virtual interfaces (called subinterfaces) on a single physical interface. Subinterfaces appear to be distinct physical interfaces to the various protocols.

For detailed information on how to configure subinterfaces, refer to the appropriate documentation module for a specific protocol in the Cisco IOS software documentation set.

To access subinterface configuration mode, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# **interface** *type number*	Specifies the virtual interface to be configured and enters subinterface configuration mode.

In the following example, a subinterface is configured for serial line 2, which is configured for Frame Relay encapsulation. The subinterface is identified as "2.1" to indicate that it is subinterface 1 of serial interface 2. The new prompt *hostname*(config-subif)# indicates subinterface configuration mode. The subinterface can be configured to support one or more Frame Relay PVCs.

```
Router(config)# interface serial 2
Router(config-if)# encapsulation frame-relay
Router(config-if)# interface serial 2.1
Router(config-subif)#
```

To exit subinterface configuration mode and return to interface configuration mode, use the **exit** command. To end your configuration session and return to privileged EXEC mode, press Ctrl-Z or enter the **end** command.

Cisco IOS CLI Sessions

This section describes the following concepts:

Local CLI Sessions

Local CLI sessions require direct access to the the console port of the networking device. Local CLI sessions start in user EXEC mode. See the "Cisco IOS CLI Modes" section on page 3 for more information on the different modes that are supported on your networking device. All of the tasks required to configure and manage a networking device can be done using a local CLI session. The most common method for establishing a local CLI session is to connect the serial port on a PC to the console port of the networking device and then to launch a terminal emulation application on the PC. The type of cable and connectors required and the settings for the terminal emulation application on the PC are dependant on the type of networking device that you are configuring. See to the documentation for your networking device for more information on setting it up for a local CLI session.

Remote CLI Sessions

Remote CLI sessions are created between a host such as a PC and a networking device such as a router over a network using a remote terminal access application such as Telnet and Secure Shell (SSH). Local CLI sessions start in user EXEC mode. See the "Cisco IOS CLI Modes" section on page 3 for more information on the different modes that are supported on your networking device. Most of the tasks required to configure and manage a networking device can be done using a remote CLI session. The exceptions are tasks that interact directly with the console port (such as recovering from a corrupted operating system (OS) by uploading a new OS image over the console port) and interacting with the networking device when it is in ROM Monitor Mode.

This document explains how to configure security for remote Telnet sessions. Telnet is the most common method for accessing a remote CLI session on a networking device.

Note SSH is a more secure alternative to Telnet. SSH provides encryption for the session traffic between your local management device such as a PC and the networking device that you are managing. Encrypting the session traffic with SSH prevents hackers that might intercept the traffic from being able to decode it. See *Secure Shell Version 2 Support* (http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00802045dc.html) for more information on using SSH.

Terminal Lines are Used for Local and Remote CLI Sessions

Cisco networking devices use the word lines to refer to the software components that manage local and remote CLI sessions. You use the **line console 0** global configuration command to enter line configuration mode to configure options, such as a password, for the console port.

```
Router# configure terminal
Router(config)# line console 0
Router(config-line)# password password-string
```

Remote CLI sessions use lines that are referred to virtual teletypewriter (VTY) lines. You use the **line vty** ***line-number*** [***ending-line-number***] global configuration command to enter line configuration mode to configure options, such as a password, for remote CLI sessions.

```
Router# configure terminal
Router(config)# line vty 0 4
Router(config-line)# password password-string
```

Protect Access to Cisco IOS EXEC Modes

Cisco IOS provides the ability to configure passwords that protect access to the following:

- Protecting Access to User EXEC Mode, page 11
- Protecting Access to Privileged EXEC mode, page 11

Protecting Access to User EXEC Mode

The first step in creating a secure environment for your networking device is protecting access to user EXEC mode by configuring passwords for local and remote CLI sessions.

You protect access to user EXEC mode for local CLI sessions by configuring a password on the console port. See the "Configuring and Verifying a Password for Local CLI Sessions" section on page 18.

You protect access to user EXEC mode for remote CLI sessions by configuring a password on the virtual terminal lines (VTYs). See the "Configuring and Verifying a Password for Remote CLI Sessions" section on page 15 for instructions on how to configure passwords for remote CLI sessions.

Protecting Access to Privileged EXEC mode

The second step in creating a secure environment for your networking device is protecting access to privileged EXEC mode with a password. The method for protecting access to privileged EXEC mode is the same for local and remote CLI sessions.

You protect access to privileged EXEC mode by configuring a password for it. This is sometimes referred to as the enable password because the command to enter privileged EXEC mode is **enable**.

Command	Purpose
enable **Example:** `Router> enable` `Password` `Router#`	Enables privileged EXEC mode. • Enter your password if prompted. The password will not be shown in the terminal window. • The ">" at the end of the prompt string is changed to a "#" to indicate that you are in privileged EXEC mode.

Cisco IOS Password Encryption Levels

Some of the passwords that you configure on your networking device are saved in the configuration in plain text. This means that if you store a copy of the configuration file on a disk, anybody with access to the disk can discover the passwords by reading the configuration file. The following password types are stored as plain text in the configuration by default:

- Console passwords for local CLI sessions
- Virtual terminal line passwords for remote CLI sessions
- Username passwords using the default method for configuring the password
- Privileged EXEC mode password when it is configured with the **enable password** *password* command
- Authentication key chain passwords used by RIPv2 and EIGRP
- BGP passwords for authenticating BGP neighbors
- OSPF authentication keys for authenticating OSPF neighbors
- ISIS passwords for authenticating ISIS neighbors

This excerpt from a router configuration file shows examples of passwords and authentication keys that are stored as clear text.

```
!
enable password O9Jb6D
!
username gjones password 0 kV9sIj3
!
key chain trees
 key 1
  key-string willow
!
interface Ethernet1/0.1
 ip address 172.16.6.1 255.255.255.0
 ip router isis
 ip rip authentication key-chain trees
 ip authentication key-chain eigrp 1 trees
 ip ospf authentication-key j7876
 no snmp trap link-status
 isis password u7865k
!
line vty 0 4
 password V9jA5M
!
```

You can encrypt these clear text passwords in the configuration file by using the **service password-encryption** command. This should be considered only a minimal level of security because the encryption algorithm used by the **service password-encryption** command to encrypt passwords creates text strings that be decrypted using tools that are publicly available. You should still protect access to any electronic or paper copies of your configuration files after you use the **service password-encryption** command.

The **service password-encryption** command does not encrypt the passwords when they are sent to the remote device. Anybody with a network traffic analyzer who has access to you network can capture these passwords from the packets as they are transmitted between the devices. See the "Configuring Password Encryption for Clear Text Passwords" section on page 22 for more information on encrypting clear text passwords in configuration files.

Many of the Cisco IOS features that use clear text passwords can also be configured to use the more secure MD5 algorithm. The MD5 algorithm creates a text string in the configuration file that is much more difficult to decrypt. The MD5 algorithm does not send the password to the remote device. This prevents people using a traffic analyzer to capture traffic on your network from being able to discover your passwords.

You can determine the type of password encryption that has been used by the number that is stored with the password string in the configuration file of the networking device. The number 5 in the configuration excerpt below indicates that the enable secret password has been encrypted using the MD5 algorithm.

```
!
enable secret 5 $1$fGCS$rkYbR6.Z8xo4qCl3vghWQ0
!
```

The number 7 in the excerpt below indicates that the enable password has been encrypted using the less secure algorithm used by the **service password-encryption** command.

```
!
enable password 7 00081204
!
```

Cisco IOS CLI Session Usernames

After you have protected access to user EXEC mode and privileged EXEC mode by configuring passwords for them you can further increase the level of security on your networking device by configuring usernames to limit access to CLI sessions to your networking device to specific users.

Usernames that are intended to be used for managing a networking device can be modified with additional options such as:

- Automatically starting a CLI session at a specific privilege level. See "Configuring and Verifying the Networking Device to Require a Username for the First-Line Technical Support Staff" section on page 30.
- Running a CLI command automatically. See "Configuring and Verifying a Networking Device to Allow Non Administrative Users to View the Running Configuration Automatically: Example" section on page 38.

See the *Cisco IOS Security Command Reference*, Release,12.4 (http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124cr/hsec_r/index.htm) for more information on how to configure the **username** command.

Cisco IOS Privilege Levels

The default configuration for Cisco IOS based networking devices uses privilege level 1 for user EXEC mode and privilege level 15 for privileged EXEC. The commands that can be run in user EXEC mode at privilege level 1 are a subset of the commands that can be run in privileged EXEC mode at privilege 15.

The **privilege** command is used to move commands from one privilege level to another. For example, some ISPs allow their first level technical support staff to enable and disable interfaces to activate new customer connections or to restart a connection that has stopped transmitting traffic. See the "Configuring and Verifying a Networking Device to Allow Non Administrative Users to Shutdown and Enable Interfaces: Example" section on page 38 for an example of how to configure this option.

The **privilege** command can also be used to assign a privilege level to a username so that when a user logs in with the username, the session will run at the privilege level specified by the **privilege** command. For example if you want your technical support staff to view the configuration on a networking device to help them troubleshoot network problems without being able to modify the configuration, you can create a username, configure it with privilege level 15, and configure it to run the **show running-config** command automatically. When a user logs in with the username the running configuration will be displayed automatically. The user's session will be logged out automatically after the user has viewed the last line of the configuration. See the "Configuring and Verifying a Networking Device to Allow Non Administrative Users to View the Running Configuration Automatically: Example" section on page 38 for an example of how to configure this option.

These command privileges can also be implemented when using AAA with TACACS+ and RADIUS. For example, TACACS+ provides two ways to control the authorization of router commands on a per-user or per-group basis. The first way is to assign privilege levels to commands and have the router verify with the TACACS+ server whether or not the user is authorized at the specified privilege level. The second way is to explicitly specify in the TACACS+ server, on a per-user or per-group basis, the commands that are allowed. For more information about implementing AAA with TACACS+ and RADIUS, see the technical note *How to Assign Privilege Levels with TACACS+ and RADIUS*.

Cisco IOS Password Configuration

Cisco IOS software does not prompt you to repeat any passwords that you configure to verify that you have entered the passwords exactly as you intended. New passwords, and changes to existing passwords, go into effect immediately after you press the Enter key at the end of a password configuration command string. If you make a mistake when you enter a new password and have saved the configuration on the networking device to its startup configuration file and exited privileged EXEC mode before you realize that you made a mistake, you may find that you are no longer able to manage the device.

The following are common situations that can happen:

- You make a mistake configuring a password for local CLI sessions on the console port.
  - If you have properly configured access to your networking device for remote CLI sessions, you can Telnet to it and reconfigure the password on the console port.
- You make a mistake configuring a password for remote Telnet or SSH sessions.
  - If you have properly configured access to your networking device for local CLI sessions, you can connect a terminal to it and reconfigure the password for the remote CLI sessions.
- You make a mistake configuring a password for privileged EXEC mode (enable password or enable secret password).
  - You will have to perform a lost password recovery procedure.
- You make a mistake configuring your username password, and the networking device requires that you log into it with your username.
  - If you do not have access to another account name, you will have to perform a lost password recovery procedure.

To protect yourself from having to perform a lost password recovery procedure open two CLI sessions to the networking device and keep one of them in privilege EXEC mode while you reset the passwords using the other session. You can use the same device (PC or terminal) to run the two CLI sessions or two different devices. You can use a local CLI session and a remote CLI session or two remote CLI sessions for this procedure. The CLI session that you use to configure the password can also be used to verify that the password was changed properly. The other CLI session that you keep in privileged EXEC mode can be used to change the password again if you made a mistake the first time you configured it.

You should not save password changes that you have made in the running configuration to the startup configuration until you have verified that your password was changed successfully. If you discover that you made a mistake configuring a password, and you were not able to correct the problem using the second CLI session technique described above, you can power cycle the networking device so that it returns to the previous passwords that are stored in the startup configuration.

How To Configure Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

This section contains the following procedures:

Protecting Access to User Exec Mode

This section contains the following procedures:

Configuring and Verifying a Password for Remote CLI Sessions

This task will assign a password for remote CLI sessions. After you have completed this task the networking device will prompt you for a password the next time that you start a remote CLI session with it.

Cisco IOS based networking devices require that you have a password configured for remote CLI sessions. If you attempt to start a remote CLI session with a device that doesn’t have a password configured for remote CLI sessions you will see a message that a password is required and has not been set. The remote CLI session will be terminated by the remote host.

Prerequisites

If you have not previously configured a password for remote CLI sessions, you must perform this task over a local CLI session using a terminal or a PC running a terminal emulation application, attached to the console port.

Your terminal, or terminal emulation application, must be configured with the settings that are used by the console port on the networking device. The console ports on most Cisco networking devices require the following settings: 9600 baud, 8 data bits, 1 stop bit, no parity, and flow control is set to "none." See the documentation for your networking device if these settings do not work for your terminal.

To perform the verification step (Step 6) for this task, your networking device must have an interface that is in an operational state. The interface must have a valid IP address.

Restrictions

If you have not previously configured a password for remote CLI sessions, you must perform this task over a local CLI session using a terminal attached to the console port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line vty** *line-number* [*ending-line-number*]
4. **password** *password*
5. **end**
6. telnet ip-address
7. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**line vty** *line-number* [*ending-line-number*] **Example:** `Router(config)# line vty 0 4`	Enters line configuration mode.
Step 4	**password** *password* **Example:** `Router(config-line)# password H7x3U8`	The argument *password* is a character string that specifies the line password. The following rules apply to the *password* argument: • The first character cannot be a number. • The string can contain any alphanumeric characters, including spaces, up to 80 characters. You cannot specify the password in the format number-space-anything. • Passwords are case sensitive.
Step 5	**end** **Example:** `Router(config-line)# end`	Exits the current configuration mode and returns to privileged EXEC mode.
Step 6	**telnet** *ip-address* **Example:** Router# telnet 172.16.1.1	Start a remote CLI session with the networking device from your current CLI session using the IP address of an interface in the networking device that is in an operational state (interface up, line protocol up). • Enter the password that you configured is step 4 when prompted. **Note** This procedure is often referred to as a starting a recursive Telnet session because you are initiating a remote Telnet session with the networking device from the networking device itself.
Step 7	**exit**	Terminates the remote CLI session (recursive Telnet session) with the networking device.

Troubleshooting Tips

Repeat this task if you made a mistake configuring the remote CLI session password.

What to Do Next

Proceed to the "Configuring and Verifying a Password for Local CLI Sessions" section on page 18 .

Configuring and Verifying a Password for Local CLI Sessions

This task will assign a password for local CLI sessions over the console port. After you have completed this task, the networking device will prompt you for a password the next time that you start a local CLI session on the console port.

This task can be performed over a local CLI session using the console port or a remote CLI session. If you want to perform the optional step of verifying that you configured the password correctly you should perform this task using a local CLI session using the console port.

Prerequisites

If you want to perform the optional step of verifying the local CLI session password, you must perform this task using a local CLI session. You must have a terminal or a PC running a terminal emulation program, connected to the console port of the networking device. Your terminal must be configured with the settings that are used by the console port on the networking device. The console ports on most Cisco networking devices require the following settings: 9600 baud, 8 data bits, 1 stop bit, no parity, and flow control is set to "none." See the documentation for your networking device if these settings do not work for your terminal.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line console 0**
4. **password** *password*
5. **end**
6. **exit**
7. Press the Enter key, and enter the password from Step 4 when prompted.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**line console 0** **Example:** `Router(config)# line console 0`	Enters line configuration mode and selects the console port as the line that you are configuring.
Step 4	**password** *password* **Example:** `Router(config-line)# password Ji8F5Z`	The argument *password* is a character string that specifies the line password. The following rules apply to the *password* argument: • The first character cannot be a number. • The string can contain any alphanumeric characters, including spaces, up to 80 characters. You cannot specify the password in the format number-space-anything. • Passwords are case sensitive.
Step 5	**end** **Example:** `Router(config-line)# end`	Exits the current configuration mode and returns to privileged EXEC mode.
Step 6	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.
Step 7	Press the Enter key.	(Optional) Initiates the local CLI session on the console port. • Enter the password that you configured is step 4 when prompted to verify that it was configured correctly. **Note** This step can be performed only if you are using a local CLI session to perform this task.

Troubleshooting Tips

If your new password is not accepted proceed to the "Recovering from a Lost or Misconfigured Password for Local CLI Sessions" section on page 33 for instructions on what to do next.

What to Do Next

Proceed to the "Protecting Access to Privileged Exec Mode" section on page 20.

Protecting Access to Privileged Exec Mode

This section contains the following procedures:

- Configuring and Verifying the Enable Password, page 20 (optional)
- Configuring Password Encryption for Clear Text Passwords, page 22 (optional)
- Configuring and Verifying the Enable Secret Password, page 23 (recommended)

Configuring and Verifying the Enable Password

Cisco no longer recommends that you use the **enable password** command to configure a password for privileged EXEC mode. The password that you enter with the **enable password** command is stored as plain text in the configuration file of the networking device. You can encrypt the password for the **enable password** command in the configuration file of the networking device using the **service password-encryption** command. However the encryption level used by the **service password-encryption** command can be decrypted using tools available on the Internet.

Instead of using the **enable password** command, Cisco recommends using the **enable secret** command because it encrypts the password that you configure with it with strong encryption . For more information on password encryption issues see the "Cisco IOS Password Encryption Levels" section on page 11. For information on configuring the **enable secret** command see the "Configuring and Verifying the Enable Secret Password" section on page 23.

Restrictions

The networking device must not have a password configured by the **enable secret** command in order to perform this task successfully. If you have already configured a password for privileged EXEC mode using the **enable secret** command, the password configured takes precedences over the password that you configure in this task using the **enable password** command.

You cannot use the same password for the **enable secret** command and the **enable** *password* command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **enable password** *password*
4. **end**
5. exit
6. enable

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**enable password** *password* **Example:** `Router(config)# enable password t6D77CdKq`	The argument *password* is a character string that specifies the enable password. The following rules apply to the *password* argument: • Must contain from 1 to 25 uppercase and lowercase alphanumeric characters. • Must not have a number as the first character. • Can have leading spaces, but they are ignored. However, intermediate and trailing spaces are recognized. • Can contain the question mark (?) character if you precede the question mark with the key combination Crtl-v when you create the password; for example, to create the password abc?123, do the following: – Enter abc – Type Crtl-v – Enter ?123
Step 4	**end** **Example:** `Router(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.
Step 5	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.
Step 6	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter the password you configured in step 3.

Troubleshooting Tips

If your new password is not accepted, proceed to the "Recovering from a Lost or Misconfigured Passwords for Privileged EXEC Mode" section on page 35 for instructions on what to do next.

What to Do Next

Encrypt the clear text enable password in the configuration file of the networking device using the procedure described in "Configuring Password Encryption for Clear Text Passwords" section on page 22.

Configuring Password Encryption for Clear Text Passwords

Cisco IOS stores passwords in clear text in network device configuration files for several features such as passwords for local and remote CLI sessions, and passwords for neighbor authentication for routing protocols. Clear text passwords are a security risk because anybody with access to archived copies of the configuration files can discover the passwords that are stored as clear text. The **service password-encryption** command can be used to encrypt clear text commands in the configuration files of networking devices. See the "Cisco IOS Password Encryption Levels" section on page 11 for more information.

Perform the following steps to configure password encryption for passwords that are stored as clear text in the configuration files of your networking device.

Prerequisites

You must have at least one feature that uses clear text passwords configured on your networking device for this command to have any immediate effect.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service password-encryption**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**service password-encryption** **Example:** `Router(config)# service password-encryption`	Enables Password encryption for all passwords clear text passwords, including username passwords, authentication key passwords, the privileged command password, console and virtual terminal line access passwords, and Border Gateway Protocol neighbor passwords.
Step 4	**end** **Example:** `Router(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Configuring and Verifying the Enable Secret Password

Cisco recommends that you use the **enable secret** command, instead of the **enable** *password* command to configure a password for privileged EXEC mode. The password created by the **enable secret** command is encrypted with the more secure MD5 algorithm.

Restrictions

You cannot use the same password for the **enable secret** command and the **enable** *password* command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **enable secret** *password*
   or
   enable secret *5 previously-encrypted-password*
4. **end**
5. exit
6. enable

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**enable secret** *password* or **enable secret** *5 previously-encrypted-password* **Example:** `Router(config)# enable secret t6D77CdKq` or **Example:** `Router(config)# enable secret 5 $1$/x6H$RhnDI3yLC4GA01aJnHLQ4/`	The argument *password* is a character string that specifies the **enable secret** password. The following rules apply to the *password* argument: • Must contain from 1 to 25 uppercase and lowercase alphanumeric characters. • Must not have a number as the first character. • Can have leading spaces, but they are ignored. However, intermediate and trailing spaces are recognized. • Can contain the question mark (?) character if you precede the question mark with the key combination Crtl-v when you create the password; for example, to create the password abc?123, do the following: – Enter abc – Type Crtl-v – Enter ?123 or Sets a previously encrypted password for privileged EXEC mode by entering the number 5 before the previously encrypted string. You must enter an exact copy of a password from a configuration file that was previously encrypted by the **enable secret** command to use this method.
Step 4	**end** **Example:** `Router(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	`exit` **Example:** `Router# exit`	Exits privileged EXEC mode.
Step 6	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter the password that you configured in Step 3.

Troubleshooting Tips

If your new password is not accepted proceed to the "Recovering from a Lost or Misconfigured Passwords for Privileged EXEC Mode" section on page 35 for instructions on what to do next.

What to Do Next

If you have finished configuring passwords for local and remote CLI sessions and you want to configure additional security features, such as usernames, and privilege levels proceed to the "Configuring Security Options with Passwords, Privilege Levels and usernames to Manage Access to CLI Sessions and CLI Commands" section on page 25.

Configuring Security Options with Passwords, Privilege Levels and usernames to Manage Access to CLI Sessions and CLI Commands

The tasks in this section describe how to configure your networking device to permit the use of a subset of privileged EXEC mode commands by users who should not have access to all of of the commands available in privileged EXEC mode.

These tasks are beneficial for companies that have multiple levels of network support staff and the company wants the staff at each level to have access to a different subset of the privileged EXEC mode commands.

In this task the users who should not have access to all of of the commands available in privileged EXEC mode are referred to as the first-line technical support staff.

This section contains the following procedures:

Configuring the Networking Device for the First-Line Technical Support Staff

This task describes how to configure the networking device for first-line technical support users. First-line technical support staff are usually not allowed to run all of the commands available in privileged EXEC mode (privilege level 15) on a networking device. They are prevented from running commands that they are not authorized for by not being granted access to the password assigned to privileged EXEC mode or to other roles that have been configured on the networking device.

The **privilege** command is used to move commands from one privilege level to another in order to create the additional levels of administration of a networking device that is required by companies that have different levels of network support staff with different skill levels.

The default configuration of a Cisco IOS device permits two types of users to access the CLI. The first type of user is a person who is only allowed to access user EXEC mode. The second type of user is a person who is allowed access to privileged EXEC mode. A user who is only allowed to access user EXEC mode is not allowed to view or change the configuration of the networking device, or to make any changes to the operational status of the networking device. On the other hand, a user who is allowed access to privileged EXEC mode can make any change to a networking device that is allowed by the CLI.

In this task the two commands that normally run at privilege level 15 are reset to privilege level 7 using the privilege command in order that first-line technical support users will be allowed to run the two commands. The two commands for which the privilege levels will be reset are the **clear counters** command and **reload** command.

- The **clear counters** command is used to reset the counter fields on interfaces for statistics such as packets received, packets transmitted, and errors. When a first-line technical support user is troubleshooting an interface related connectivity issue between networking devices, or with remote users connecting to the network, it is useful to reset the interface statistics to zero and them monitor the interfaces for a period of time to see if the values in the interface statistics counters change.
- The **reload** command is used initiate a reboot sequence for the networking device. One common use of the reload command by first-line technical support staff is to cause the networking device to reboot during a maintenance window so that it loads a new operating system that was previously copied onto the networking device's file system by a user with a higher level of authority.

Any user that is permitted to know the **enable secret** password that is assigned to the first-line technical support user role privilege level can access the networking device as a first-line technical support user. You can add an additional level of security by configuring a username on the networking device and requiring that the users know the username and the password. Configuring a username as an additional level of security is described in the "Configuring and Verifying the Networking Device to Require a Username for the First-Line Technical Support Staff" section on page 30.

Privilege Command Enhancement

Before Cisco IOS Releases 12.0(22)S and 12.2(13)T, each command in a privilege level had to be specified with a separate **privilege** command. In Cisco IOS Releases 12.0(22)S, 12.2(13)T, and later releases, a "wildcard" option specified by the new keyword **all** was introduced that allows you to configure access to multiple commands with only one **privilege** command. By using the new **all** keyword, you can specify a privilege level for all commands which begin with the string you enter. In other words, the **all** keyword allows you to grant access to all command-line options and suboptions for a specified command.

For example, if you wanted to create a privilege level to allow users to configure all commands which begin with **service-module t1** (such as **service-module t1 linecode** or **service-module t1 clock source**) you can use the **privilege interface all level** *2* ***service-module t1*** command instead of having to specify each **service-module t1** command separately.

If the command specified in the privilege command (used with the **all** keyword) enables a configuration submode, all commands in the submode of that command will also be set to the specified privilege level.

Restrictions

The **all** "wildcard" keyword option for the **privilege** command is not supported in versions of Cisco IOS software prior to Cisco IOS Releases 12.0(22)S and, 12.2(13)T.

You must not have the **aaa new-model** command enabled on the networking device. You must not have the **login local** command configured for the local CLI sessions over the console port or the remote CLI sessions.

Note For clarity, only the arguments and keywords that are relevant for each step are shown in the syntax for the steps in this task. See the Cisco IOS command reference book for your Cisco IOS release for further information on the additional arguments and keywords that can be used with these commands.

Caution Do not use the no form of the **privilege** command to reset the privilege level of a command to its default because it might not return the configuration to the correct default state. Use the **reset** keyword for the **privilege** command instead to return a command to its default privilege level. For example, to remove the **privilege exec level** *reload* command from the configuration and return the **reload** command to its default privilege of 15, use the **privilege exec reset** *reload* command.

SUMMARY STEPS

1. **enable** *password*
2. **configure terminal**
3. **enable secret level** *level password*
4. **privilege exec level** *level command-string*
5. **privilege exec all level** *level command-string*
6. **end**

DETAILED STEPS

Step 1 **enable** *password*

Enters privileged EXEC mode. Enter the password when prompted.

```
Router> enable
```

Step 2 **configure terminal**

Enters global configuration mode.

```
Router# configure terminal
```

Step 3 **enable secret level** *level password*

Configures a new enable secret password for privilege level 7.

```
Router(config)# enable secret level 7 Zy72sKj
```

Step 4 **privilege exec level** *level command-string*

Changes the privilege level of the **clear counters** command from privilege level 15 to privilege level 7.

```
Router(config)# privilege exec level 7 clear counters
```

Step 5 **privilege exec all level** *level command-string*

Changes the privilege level of the **reload** command from privilege level 15 to privilege level 7.

```
Router(config)# privilege exec all level 7 reload
```

Step 6 **end**

Exits global configuration mode.

```
Router(config)# end
```

Verifying the Configuration for the First-Line Technical Support Staff

This task describes how to verify that the network device is configured correctly for the first-line technical support staff.

Prerequisites

The following commands must have been modified to run at privilege level 7 for this task:

- **clear counters**
- **reload**

SUMMARY STEPS

1. **enable** *level password*
2. **show privilege**
3. **clear counters**
4. **clear** *ip route* *
5. **reload in time**
6. **reload cancel**
7. **disable**
8. **show privilege**

DETAILED STEPS

Step 1 **enable** *level password*

Logs the user into the networking device at the privilege level specified for the level argument.

```
Router> enable 7 Zy72sKj
```

Step 2 **show privilege**

Displays the privilege level of the current CLI session

```
Router# show privilege
Current privilege level is 7
```

Step 3 **clear counters**

The clear counters command clears the interface counters. This command has been changed from privilege level 15 to privilege level 7.

```
Router# clear counters
Clear "show interface" counters on all interfaces [confirm]
Router#
02:41:37: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console
```

Step 4 **clear ip route** *

The *ip route* argument string for the **clear** command should not be allowed because it was not changed from privilege level 15 to privilege level 7.

```
Router# clear ip route *
         ^
% Invalid input detected at '^' marker.

Router#
```

Step 5 **reload in time**

The reload command causes the networking device to reboot.

```
Router# reload in 10
Reload scheduled in 10 minutes by console
Proceed with reload? [confirm]
Router#


***
*** --- SHUTDOWN in 0:10:00 ---
***

02:59:50: %SYS-5-SCHEDULED_RELOAD: Reload requested for 23:08:30 PST Sun Mar 20
```

Step 6 **reload cancel**

The **reload cancel** terminates a reload that was previously setup with the the **reload in** *time* command.

```
Router# reload cancel

***
*** --- SHUTDOWN ABORTED ---
***

04:34:08: %SYS-5-SCHEDULED_RELOAD_CANCELLED:  Scheduled reload cancelled at 15:38:46 PST
Sun Mar 27 2005
```

Step 7 **disable**

Exits the current privilege level and returns to privilege level 1.

```
Router# disable
```

Step 8 **show privilege**

Displays the privilege level of the current CLI session

```
Router> show privilege
Current privilege level is 1
```

Troubleshooting Tips

If your configuration does not work the way that you want it to and you want to remove the privilege commands from the configuration, use the **reset** keyword for the **privilege** command to return the commands to their default privilege level. For example, to remove the command **privilege exec level** *reload* command from the configuration and return the **reload** command to its default privilege of 15 use the **privilege exec reset** *reload* command.

What to Do Next

If you want to add an additional level of security by requiring that the first level technical staff use a login name, proceed to the "Configuring and Verifying the Networking Device to Require a Username for the First-Line Technical Support Staff" section on page 30.

Configuring and Verifying the Networking Device to Require a Username for the First-Line Technical Support Staff

This task configures the networking device to require that the first-line technical support staff login to the networking device with a login name of admin. The admin username configured in this task is assigned the privilege level 0f 7 which will allow users who log in with this name to run the commands that were reassigned to privilege level 7 in the previous task. When a user successfully logs in with the admin username, the CLI session will automatically enter privilege level 7.

Enhanced Username Password Security

Before Cisco IOS Releases 12.0(18)S and 12.2(8)T, two types of passwords were associated with usernames: Type 0, which is a clear text password visible to any user who has access to privileged mode on the router, and type 7, which has a password encrypted by the **service password encryption** command.

In Cisco IOS Releases 12.0(18)S, 12.2(8)T, and later releases, the new **secret** keyword for the **username** command allows you to configure Message Digest 5 (MD5) encryption for username passwords.

Prerequisites

The following commands must have been modified to run at privilege level 7 for this task:

- **clear counters**
- **reload**

See the "Configuring the Networking Device for the First-Line Technical Support Staff" section on page 25 for instructions on how to change the privilege level for a command.

Restrictions

MD5 encryption for the **username** command is not supported in versions of Cisco IOS software prior to Cisco IOS Releases 12.0(18)S and 12.2(8)T.

You must not have the aaa-new model command enabled on the networking device. You must not have the **login local** command configured for the local CLI sessions over the console port or the remote CLI sessions.

Note For clarity, only the arguments and keywords that are relevant for each step are shown in the syntax for the steps in this task. Refer to the Cisco IOS command reference book for your Cisco IOS release for further information on the additional arguments and keywords that can be used with these commands.

SUMMARY STEPS

1. **enable** *password*
2. **configure terminal**

3. **username *username* privilege** *level* **secret** *password*
4. **end**
5. **disable**
6. **login username** *password*
7. **show privilege**
8. **clear counters**
9. **clear ip route** *
10. **reload in 10**
11. **reload cancel**
12. **disable**
13. **show privilege**

DETAILED STEPS

Step 1 **enable** *t6D77CdKq*

Enters privileged EXEC mode. Enter the password when prompted.

```
Router> enable
```

Step 2 **configure terminal**

Enters global configuration mode.

```
Router# configure terminal
```

Step 3 **username *username* privilege** *level* **secret** *password*

Creates a username and applies MD5 encryption to the *password* text string.

```
Router(config)# username admin privilege 7 secret Kd65xZa
```

Step 4 **end**

Exits global configuration mode.

```
Router(config)# end
```

Step 5 **disable**

Exits the current privilege level and returns to user EXEC mode.

```
Router# disable
```

Step 6 **login** *username*

Logs in the user. Enter the username and password you configured in step 3 when prompted.

```
Router> login admin
```

Step 7 **show privilege**

The **show privilege** command displays the privilege level of the CLI session.

```
Router# show privilege

Current privilege level is 7
```

Step 8 **clear counters**

The **clear counters** command clears the interface counters. This command has been changed from privilege level 15 to privilege level 7.

```
Router# clear counters
Clear "show interface" counters on all interfaces [confirm]
Router#
02:41:37: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console
```

Step 9 **clear** *ip route* *

The *ip route* argument string for the **clear** command is not allowed because it was not changed from privilege level 15 to privilege level 7.

```
Router# clear ip route *
         ^
% Invalid input detected at '^' marker.

Router#
```

Step 10 **reload in** ***time***

The reload command causes the networking device to reboot.

```
Router# reload in 10
Reload scheduled in 10 minutes by console
Proceed with reload? [confirm]
Router#


***
*** --- SHUTDOWN in 0:10:00 ---
***

02:59:50: %SYS-5-SCHEDULED_RELOAD: Reload requested for 23:08:30 PST Sun Mar 20
```

Step 11 **reload cancel**

The **reload cancel** command terminates a reload that was previously setup with the the **reload in** *time* command.

```
Router# reload cancel

***
*** --- SHUTDOWN ABORTED ---
***

04:34:08: %SYS-5-SCHEDULED_RELOAD_CANCELLED:  Scheduled reload cancelled at 15:38:46 PST
Sun Mar 27 2005
```

Step 12 **disable**

Exits the current privilege level and returns to user EXEC mode.

```
Router# disable
```

Step 13 **show privilege**

Displays the privilege level of the current CLI session

```
Router> show privilege
Current privilege level is 1
```

Recovering from a Lost or Misconfigured Password for Local CLI Sessions

There are three methods that can be used to recover from a lost or misconfigured password for local CLI sessions over console port. The method that you will use depends on the current configuration of your networking device.

- Networking Device Is Configured to Allow Remote CLI Sessions, page 33
- Networking Device Is Not Configured to Allow Remote CLI Sessions and the Local CLI Session Password Has Not Been Saved to the Startup Configuration File, page 33
- Networking Device is not Configured to Allow Remote CLI Sessions and the Local CLI Session Password Has Been Saved to the Startup Configuration File, page 33

Networking Device Is Configured to Allow Remote CLI Sessions

The fastest method to recover from a lost, or misconfigured password for local CLI sessions is to establish a remote CLI session with the networking device and repeat the "Configuring and Verifying a Password for Local CLI Sessions" section on page 18. Your networking device must be configured to allow remote CLI sessions and you must know the remote CLI session password to perform this procedure.

Networking Device Is Not Configured to Allow Remote CLI Sessions and the Local CLI Session Password Has Not Been Saved to the Startup Configuration File

If you cannot establish a remote session to your networking device, and you have not saved the misconfigured local CLI session password to the startup configuration, you can restart the networking device. When the networking device starts up again it will read the startup configuration file. The previous local CLI session password is restored.

Caution Restarting a networking device will cause it to stop forwarding traffic. This will also cause an interruption in any services that are running on the networking device, such as a DHCP server service, to stop. You should only restart a networking device during a period of time that has been allocated for network maintenance.

Networking Device is not Configured to Allow Remote CLI Sessions and the Local CLI Session Password Has Been Saved to the Startup Configuration File

If you can not establish a remote CLI session with the networking device, and you have saved the misconfigured local CLI session password to the startup configuration, or you have lost the local CLI session password, you must perform a password recovery procedure. Password recovery procedures are device specific. You must locate the document that describes the procedure for your type of networking device.

There are three methods for locating a password recovery procedure document for your networking device:

- Many networking devices have a Password Recovery subsection in the Troubleshoot and Alerts section of their product support page on Cisco's Technical Support website (http://www.cisco.com/tac). Navigate to the Troubleshoot and Alerts section of the product support page for your networking device and look for the Password Recovery subsection.

- If you do not find the password recovery document for your networking device on its product support page try searching for the text string **"password recovery"** on Cisco's Technical Support website (http://www.cisco.com/tac). Enclose the text string in double quotes. You can improve the search results by adding an additional text string that matches the model number, series number or platform name for your networking device, For example searching on the string **"password recovery" 12000** will provide search results that give documents with the words **password**, **recovery**" and **12000** in the title a higher ranking.
- If the product support page for your networking device does not have a password recovery document, and you can not find the correct document by searching for it, you can try the Cisco's Network Professionals Connection (http://www.cisco.com/go/netpro).

Recovering from a Lost or Misconfigured Password for Remote CLI Sessions

There are three methods that can be used to recover from a lost, or misconfigured remote CLI session password. The method that you will use depends on the current configuration of your networking device.

- Networking Device Is Configured to Allow Local CLI Sessions, page 34
- Networking Device Is Not Configured to Allow Local CLI Sessions and the Remote CLI Session Password Has Not Been Saved to the Startup Configuration File, page 34
- Networking Device Is Not Configured to Allow Local CLI Sessions and the Remote CLI Session Password Has Been Saved to the Startup Configuration File, page 35

Networking Device Is Configured to Allow Local CLI Sessions

The fastest method to recover from a lost, or misconfigured password for remote CLI sessions is to establish a local CLI session with the networking device and repeat the "Configuring and Verifying a Password for Remote CLI Sessions" section on page 15. Your networking device must be configured to allow local CLI sessions and you must know the local CLI session password to perform this procedure.

Networking Device Is Not Configured to Allow Local CLI Sessions and the Remote CLI Session Password Has Not Been Saved to the Startup Configuration File

If you cannot establish a local CLI session to your networking device, and you have not saved the misconfigured remote CLI session password to the startup configuration, you can restart the networking device. When the networking device starts up again it will read the startup configuration file. The previous remote CLI session password is restored.

Caution Restarting a networking device will cause it to stop forwarding traffic. This will also cause an interruption in any services that are running on the networking device, such as a DHCP server service, to stop. You should only restart a networking device during a period of time that has been allocated for network maintenance.

Networking Device Is Not Configured to Allow Local CLI Sessions and the Remote CLI Session Password Has Been Saved to the Startup Configuration File

If you can not establish a local CLI session with the networking device, and you have saved the misconfigured remote CLI session password to the startup configuration, or you have lost the remote CLI session password, you must perform a password recovery procedure. Password recovery procedures are device specific. You must locate the document that describes the procedure for your type of networking device.

There are three methods for locating a password recovery procedure document for your networking device:

- Many networking devices have a Password Recovery subsection in the Troubleshoot and Alerts section of their product support page on Cisco's Technical Support website (http://www.cisco.com/tac). Navigate to the Troubleshoot and Alerts section of the product support page for your networking device and look for the Password Recovery subsection.
- If you do not find the password recovery document for your networking device on its product support page try searching for the text string **"password recovery"** on Cisco's Technical Support website (http://www.cisco.com/tac). Enclose the text string in double quotes. You can improve the search results by adding an additional text string that matches the model number, series number or platform name for your networking device, For example searching on the string **"password recovery" 12000** will provide search results that give documents with the words **password**, **recovery**" and **12000** in the title a higher ranking.
- If the product support page for your networking device does not have a password recovery document, and you can not find the correct document by searching for it, you can try the Cisco's Network Professionals Connection (http://www.cisco.com/go/netpro).

Recovering from a Lost or Misconfigured Passwords for Privileged EXEC Mode

There are two methods that can be used to recover from a lost, or misconfigured Privileged EXEC Mode password. The method that you will use depends on the current configuration of your networking device.

- A Misconfigured Privileged EXEC Mode Password Has Not Been Saved to the Startup Configuration File, page 35
- A Misconfigured Privileged EXEC Mode Password Has Been Saved to the Startup Configuration File, or the Privileged EXEC Mode Password Has Been Lost, page 36

A Misconfigured Privileged EXEC Mode Password Has Not Been Saved to the Startup Configuration File

If you have not saved the misconfigured privileged EXEC mode password to the startup configuration, you can restart the networking device. When the networking device starts up again it will read the startup configuration file. The previous privileged EXEC mode password is restored.

Caution Restarting a networking device will cause it to stop forwarding traffic. This will also cause an interruption in any services that are running on the networking device, such as a DHCP server service, to stop. You should only restart a networking device during a period of time that has been allocated for network maintenance.

A Misconfigured Privileged EXEC Mode Password Has Been Saved to the Startup Configuration File, or the Privileged EXEC Mode Password Has Been Lost

If you have saved the misconfigured privileged EXEC mode password to the startup configuration, or you have lost the privileged EXEC mode password, you must perform a password recovery procedure. Password recovery procedures are device specific. You must locate the document that describes the procedure for your type of networking device.

There are three methods for locating a password recovery procedure document for your networking device:

- Many networking devices have a Password Recovery subsection in the Troubleshoot and Alerts section of their product support page on Cisco's Technical Support website (http://www.cisco.com/tac). Navigate to the Troubleshoot and Alerts section of the product support page for your networking device and look for the Password Recovery subsection.
- If you do not find the password recovery document for your networking device on its product support page try searching for the text string **"password recovery"** on Cisco's Technical Support website (http://www.cisco.com/tac). Enclose the text string in double quotes. You can improve the search results by adding an additional text string that matches the model number, series number or platform name for your networking device, For example searching on the string **"password recovery" 12000** will provide search results that give documents with the words **password**, **recovery**" and **12000** in the title a higher ranking.
- If the product support page for your networking device does not have a password recovery document, and you can not find the correct document by searching for it, you can try the Cisco's Network Professionals Connection (http://www.cisco.com/go/netpro).

Configuration Examples for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

This section contains the following configuration examples:

- Configuring and Verifying a Networking Device to Allow Non Administrative Users to Clear Remote CLI Sessions: Example, page 37
- Configuring and Verifying a Networking Device to Allow Non Administrative Users to View the Running Configuration Automatically: Example, page 38
- Configuring and Verifying a Networking Device to Allow Non Administrative Users to Shutdown and Enable Interfaces: Example, page 38

Configuring and Verifying a Networking Device to Allow Non Administrative Users to Clear Remote CLI Sessions: Example

The following example shows how to configure a networking device to allow a non administrative user to clear remote CLI session virtual terminal (VTY) lines.

The first section is an an excerpt of the running configuration for this example. The following sections show you how this example is used.

The following section is an excerpt of the running-configuration:

```
!
privilege exec level 7 clear line
!
no aaa new-model
!
!
username admin privilege 7 secret 5 $1$tmIw$1aM7sadKhWMpkVTzxNw1J.
!
privilege exec level 7 clear line
!
! the privilege exec level 7 clear command below is entered automatically
! when you enter the privilege exec level 7 clear line command above, do
! not enter it again
!
privilege exec level 7 clear
!
```

The following section using the **login** command shows the user logging in to the networking device with the username of admin:

```
R1> login
Username: admin
Password:
```

The following section using the **show privilege** command shows that the current privilege level is 7:

```
R1# show privilege
Current privilege level is 7
R1#
```

The following section using the **show user** command shows that two users (admin and root) are currently logged in to the networking device:

```
R1# show user
    Line       User       Host(s)              Idle       Location
*  0 con 0     admin      idle                 00:00:00
   2 vty 0     root       idle                 00:00:17 172.16.6.2

  Interface    User               Mode         Idle     Peer Address
```

The following section using the **clear line** *2* command terminates the remote CLI session in use by the username root:

```
R1# clear line  2
[confirm]
 [OK]
```

The following section using the **show user** command shows that admin is the only user currently logged in to the networking device:

```
R1# show user
    Line       User       Host(s)              Idle       Location
*  0 con 0     admin      idle                 00:00:00

  Interface    User               Mode         Idle     Peer Address
```

Configuring and Verifying a Networking Device to Allow Non Administrative Users to View the Running Configuration Automatically: Example

The following example shows how to configure the networking device to allow a non administrative users (no access to privileged EXEC mode) to view the running configuration automatically. This example requires that the username is configured for privilege level 15 because many of the commands in the configuration file can be viewed only by users who have access to privilege level 15.

The solution is to temporarily allow the user access to privilege level 15 while running the **show running-config** command and then terminating the CLI session when the end of the configuration file has been viewed. In this example the networking device will automatically terminate the CLI session when the end of the configuration file has been viewed. No further configuration steps are required.

Caution You must include the **noescape** keyword for the **username** command to prevent the user from entering an escape character that will terminate viewing the configuration file and leave the session running at privilege level 15.

```
!
!
username viewconf privilege 15 noescape secret 5 $1$zA9C$TDWD/Q0zwp/5xRwRqdgc/.
username viewconf autocommand show running-config
!
```

Configuring and Verifying a Networking Device to Allow Non Administrative Users to Shutdown and Enable Interfaces: Example

The following example shows how to configure a networking device to allow non administrative users to shutdown and enable interfaces.

The first section is an an excerpt of the running configuration for this example. The following sections show you how this example is used.

The following section is an excerpt of the running-configuration:

```
!
no aaa new-model
!
username admin privilege 7 secret 5 $1$tmIw$1aM7sadKhWMpkVTzxNw1J.
!
privilege interface all level 7 shutdown
privilege interface all level 7 no shutdown
privilege configure level 7 interface
privilege exec level 7 configure terminal
!
! the privilege exec level 7 configure command below is entered automatically
! when you enter the privilege exec level 7 configure terminal command above, do
! not enter it again
!
```

```
privilege exec level 7 configure
!
```

The following section using the **login** command shows the user logging in to the networking device with the username of admin:

```
R1> login
Username: admin
Password:
```

The following section using the **show privilege** command shows that the current privilege level is 7:

```
R1# show privilege
Current privilege level is 7
R1#
```

The following section using the **show user** command shows that admin is the only user currently logged in to the networking device:

```
R1# show user
    Line       User       Host(s)              Idle       Location
*  0 con 0     admin      idle                 00:00:00

  Interface    User               Mode         Idle     Peer Address
```

The following section shows that the admin user is permitted to shutdown and enable an interface:

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# interface ethernet 1/0
R1(config-if)# shutdown
R1(config-if)# no shutdown
R1(config-if)# exit
R1#
```

Where to Go Next

Once you have established a baseline of security for your networking devices you can consider more advanced options such as:

- Role-Based CLI Access—The role-based CLI access feature offers a more comprehensive set of options than the **privilege** command (described in this document) for network managers who want to allow different levels of technical support staff to have different levels of access to CLI commands.
- AAA Security—Many Cisco networking devices offer an advanced level of security using authentication, authorization and accounting (AAA) features. All of the tasks described in this document, and other - more advanced security features - can be implemented using AAA on the networking device in conjunction with a remote TACACS+ or RADIUS server. For information how to configure AAA security features that can be run locally on a networking device, or for information on how to configure remote AAA security using TACACS+ or RADIUS servers, see the *Cisco IOS Security Configuration Guide,* Release 12.4.

Additional References

The following sections provide references related to Configuring Security with Passwords and, Login Usernames for CLI Sessions on Networking Devices.

Related Documents

Related Topic	Document Title
Managing user access to CLI commands and configuration information	*Role-Based CLI Access*
AAA Security Features	*Cisco IOS Security Configuration Guide*, Release 12.4
Configuring MD5 secure neighbor authentication for protocols such as OSPF and BGP	*Neighbor Router Authentication: Overview and Guidelines*
Assigning privilege levels with TACACS+ and RADIUS	*How to Assign Privilege Levels with TACACS+ and RADIUS*

Standards

Standard	Title
No new or modified RFCs are supported by this functionality, and support for existing RFCs has not been modified.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this functionality, and support for existing RFCs has not been modified.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

Table 70 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or 12.0(3)S or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific command was introduced, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 70 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 70 Feature Information for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

Feature Name	Releases	Feature Configuration Information
Enhanced Password Security	12.0(18)S 12.2(8)T	Using the Enhanced Password Security feature, you can configure MD5 encryption for username passwords. MD5 encryption is a one-way hash function that makes reversal of an encrypted password impossible, providing strong encryption protection. Using MD5 encryption, you cannot retrieve clear text passwords. MD5 encrypted passwords cannot be used with protocols that require that the clear text password be retrievable, such as Challenge Handshake Authentication Protocol (CHAP). The following sections provide information about this feature: • Configuring and Verifying the Networking Device to Require a Username for the First-Line Technical Support Staff, page 30
Privilege Command Enhancement	12.0(22)S 12.2(13)T	The keyword **all** was added to the **privilege** command as a wild card to reduce the number of times you ned to enter the **privilege** command when you are changing the privilege level of several keywords for the same command. The following sections provide information about this feature: • Privilege Command Enhancement, page 26

Appendixes

RADIUS Attributes

RADIUS Attributes Overview and RADIUS IETF Attributes

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Remote Authentication Dial-In User Service (RADIUS) attributes are used to define specific authentication, authorization, and accounting (AAA) elements in a user profile, which is stored on the RADIUS daemon. This appendix lists the RADIUS attributes currently supported.

In This Appendix

This appendix contains the following sections:

RADIUS Attributes Overview

This section contains information important to understanding how RADIUS attributes exchange AAA information between a client and server and includes the following sections:

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- RADIUS Files
- Supporting Documentation

IETF Attributes Versus VSAs

RADIUS Internet Engineering Task Force (IETF) attributes are the original set of 255 standard attributes that are used to communicate AAA information between a client and a server. Because IETF attributes are standard, the attribute data is predefined and well known; thus all clients and servers who exchange AAA information via IETF attributes must agree on attribute data such as the exact meaning of the attributes and the general bounds of the values for each attribute.

RADIUS vendor-specific attributes (VSAs) derived from one IETF attribute—vendor-specific (attribute 26). Attribute 26 allows a vendor to create an additional 255 attributes however they wish. That is, a vendor can create an attribute that does not match the data of any IETF attribute and encapsulate it behind attribute 26; thus, the newly created attribute is accepted if the user accepts attribute 26.

For more information on VSAs, refer to the section "RADIUS Vendor-Specific Attributes (VSA) and RADIUS Disconnect-Cause Attribute Values" later in this appendix.

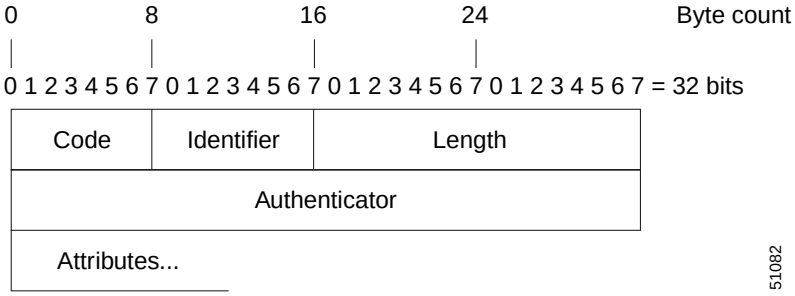
RADIUS Packet Format

The data between a RADIUS server and a RADIUS client is exchanged in RADIUS packets. The data fields are transmitted from left to right.

Figure 125 shows the fields within a RADIUS packet.

Note For a diagram of VSAs, which is an extension of Figure 125, refer to Figure 1.

Figure 125 RADIUS Packet Diagram



Each RADIUS packet contains the following information:

- Code—The code field is one octet; it identifies one of the following types of RADIUS packets:
  - Access-Request (1)
  - Access-Accept (2)
  - Access-Reject (3)
  - Accounting-Request (4)
  - Accounting-Response (5)

- Identifier—The identifier field is one octet; it helps the RADIUS server match requests and responses and detect duplicate requests.
- Length—The length field is two octets; it specifies the length of the entire packet.
- Authenticator—The authenticator field is 16 octets. The most significant octet is transmitted first; it is used to authenticate the reply from the RADIUS server. Two types of authenticators are as follows:
  - Request-Authentication: Available in Access-Request and Accounting-Request packets
  - Response-Authenticator: Available in Access-Accept, Access-Reject, Access-Challenge, and Accounting-Response packets

RADIUS Packet Types

The following list defines the various types of RADIUS packet types that can contain attribute information:

Access-Request—Sent from a client to a RADIUS server. The packet contains information that allows the RADIUS server to determine whether to allow access to a specific network access server (NAS), which will allow access to the user. Any user performing authentication *must* submit an Access-Request packet. Once an Access-Request packet is received, the RADIUS server *must* forward a reply.

Access-Accept—Once a RADIUS server receives an Access-Request packet, it must send an Access-Accept packet if all attribute values in the Access-Request packet are acceptable. Access-Accept packets provide the configuration information necessary for the client to provide service to the user.

Access-Reject—Once a RADIUS server receives an Access-Request packet, it must send an Access-Reject packet if any of the attribute values are not acceptable.

Access-Challenge—Once the RADIUS server receives an Access-Accept packet, it can send the client an Access-Challenge packet, which requires a response. If the client does not know how to respond or if the packets are invalid, the RADIUS server discards the packets. If the client responds to the packet, a new Access-Request packet should be sent with the original Access-Request packet.

Accounting-Request—Sent from a client to a RADIUS accounting server, which provides accounting information. If the RADIUS server successfully records the Accounting-Request packet, it must submit an Accounting Response packet.

Accounting-Response—Sent by the RADIUS accounting server to the client to acknowledge that the Accounting-Request has been received and recorded successfully.

RADIUS Files

Understanding the types of files used by RADIUS is important for communicating AAA information from a client to a server. Each file defines a level of authentication or authorization for the user: The dictionary file defines which attributes the user's NAS can implement; the clients file defines which users are allowed to make requests to the RADIUS server; the users files defines which user requests the RADIUS server will authenticate based on security and configuration data.

Dictionary File

A dictionary file provides a list of attributes that are dependent upon which attributes your NAS supports. However, you can add your own set of attributes to your dictionary for custom solutions. It defines attribute values, thereby allowing you to interpret attribute output such as parsing requests. A dictionary file contains the following information:

- Name—The ASCII string “name” of the attribute, such as User-Name.
- ID—The numerical “name” of the attribute; for example, User-Name attribute is attribute 1.
- Value type—Each attribute can be specified as one of the following five value types:
  - abinary—0 to 254 octets.
  - date—32-bit value in big endian order. For example, seconds since 00:00:00 GMT, JAN. 1, 1970.
  - ipaddr—4 octets in network byte order.
  - integer—32-bit value in big endian order (high byte first).
  - string—0 to 253 octets.

When the data type for a particular attribute is an integer, you can optionally expand the integer to equate to some string. The follow sample dictionary includes an integer-based attribute and its corresponding values:

```
# dictionary sample of integer entry
#
ATTRIBUTE      Service-Type         6                integer
VALUE          Service-Type         Login            1
VALUE          Service-Type         Framed           2
VALUE          Service-Type         Callback-Login   3
VALUE          Service-Type         Callback-Framed  4
VALUE          Service-Type         Outbound         5
VALUE          Service-Type         Administrative   6
VALUE          Service-Type         NAS-Prompt       7
VALUE          Service-Type         Authenticate-Only 8
VALUE          Service-Type         Callback-NAS-Prompt 9
VALUE          Service-Type         Call-Check       10
VALUE          Service-Type         Callback-Administrative 11
```

Clients File

A clients file is important because it contains a list of RADIUS clients that are allowed to send authentication and accounting requests to the RADIUS server. To receive authentication, the name and authentication key the client sends the server must be an exact match with the data contained in clients file.

The following is an example of a clients file. The key, as shown in this example, must be the same as the **radius-server key** *SomeSecret* command.

```
#Client Name        Key
#----------------  ---------------
10.1.2.3:256        test
nas01               bananas
nas02               MoNkEys
nas07.foo.com       SomeSecret
```

Users File

A RADIUS users file contains an entry for each user that the RADIUS server will authenticate; each entry, which is also referred to as a user profile, establishes an attribute the user can access.

The first line in any user profile is always a "user access" line; that is, the server must check the attributes on the first line before it can grant access to the user. The first line contains the name of the user, which can be up to 252 characters, followed by authentication information such as the password of the user.

Additional lines, which are associated with the user access line, indicate the attribute reply that is sent to the requesting client or server. The attributes sent in the reply must be defined in the dictionary file.

When looking at a user file, please note the the data to the left of the equal (=) character is an attribute defined in the dictionary file, and the data to the right of the equal character is the configuration data.

Note A blank line cannot appear anywhere within a user profile.

The following is an example of a RADIUS user profile (Merit Daemon format). In this example, the user name is cisco.com, the password is cisco, and the user can access five tunnel attributes.

```
# This user profile includes RADIUS tunneling attributes
cisco.com  Password="cisco" Service-Type=Outbound
     Tunnel-Type = :1:L2TP
     Tunnel-Medium-Type = :1:IP
     Tunnel-Server-Endpoint = :1:10.0.0.1
     Tunnel-Password = :1:"welcome"
     Tunnel-Assignment-ID = :1:"nas"
```

Supporting Documentation

For more information on RADIUS IETF and Vendor-Proprietary Attributes, refer to the following documents:

- Cisco AAA Implementation Case Study
- "Configuring RADIUS" "Configuring Authentication," "Configuring Authorization" and "Configuring Accounting" chapters in this book.

  Refer to these chapters for information on how RADIUS is used with AAA.
- IETF RADIUS RFCs
  - RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*
  - RFC 2866, *RADIUS Accounting*
  - RFC 2867, *RADIUS Accounting Modifications for Tunnel Protocol Support*
  - RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*
  - RFC 2869, *RADIUS Extensions*
- RADIUS Vendor-Specific Attributes Voice Implementation Guide

RADIUS IETF Attributes

Note In the Cisco IOS Release 12.2 for RADIUS tunnel attributes, 32 tagged tunnel sets are supported for L2TP.

This section contains the following sections:

- Supported RADIUS IETF Attributes
- Comprehensive List of RADIUS Attribute Descriptions

Supported RADIUS IETF Attributes

Table 71 lists Cisco-supported IETF RADIUS attributes and the Cisco IOS release in which they are implemented. In cases where the attribute has a security server-specific format, the format is specified.

Refer to Table 72 for a description of each listed attribute.

Note Attributes implemented in special (AA) or early development (T) releases will be added to the next mainline image.

Table 71 Supported RADIUS IETF Attributes

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1	12.2
1	User-Name	yes	yes	yes	yes	yes	yes	yes	yes
2	User-Password	yes	yes	yes	yes	yes	yes	yes	yes
3	CHAP-Password	yes	yes	yes	yes	yes	yes	yes	yes
4	NAS-IP Address	yes	yes	yes	yes	yes	yes	yes	yes
5	NAS-Port	yes	yes	yes	yes	yes	yes	yes	yes
6	Service-Type	yes	yes	yes	yes	yes	yes	yes	yes
7	Framed-Protocol	yes	yes	yes	yes	yes	yes	yes	yes
8	Framed-IP-Address	yes	yes	yes	yes	yes	yes	yes	yes
9	Framed-IP-Netmask	yes	yes	yes	yes	yes	yes	yes	yes
10	Framed-Routing	yes	yes	yes	yes	yes	yes	yes	yes
11	Filter-Id	yes	yes	yes	yes	yes	yes	yes	yes
12	Framed-MTU	yes	yes	yes	yes	yes	yes	yes	yes
13	Framed-Compression	yes	yes	yes	yes	yes	yes	yes	yes
14	Login-IP-Host	yes	yes	yes	yes	yes	yes	yes	yes
15	Login-Service	yes	yes	yes	yes	yes	yes	yes	yes
16	Login-TCP-Port	yes	yes	yes	yes	yes	yes	yes	yes
18	Reply-Message	yes	yes	yes	yes	yes	yes	yes	yes
19	Callback-Number	no	no	no	no	no	no	yes	yes

Table 71 Supported RADIUS IETF Attributes (continued)

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1	12.2
20	Callback-ID	no	no	no	no	no	no	no	no
22	Framed-Route	yes	yes	yes	yes	yes	yes	yes	yes
23	Framed-IPX-Network	no	no	no	no	no	no	no	no
24	State	yes	yes	yes	yes	yes	yes	yes	yes
25	Class	yes	yes	yes	yes	yes	yes	yes	yes
26	Vendor-Specific	yes	yes	yes	yes	yes	yes	yes	yes
27	Session-Timeout	yes	yes	yes	yes	yes	yes	yes	yes
28	Idle-Timeout	yes	yes	yes	yes	yes	yes	yes	yes
29	Termination-Action	no	no	no	no	no	no	no	no
30	Called-Station-Id	yes	yes	yes	yes	yes	yes	yes	yes
31	Calling-Station-Id	yes	yes	yes	yes	yes	yes	yes	yes
32	NAS-Identifier	no	no	no	no	no	no	no	yes
33	Proxy-State	no	no	no	no	no	no	no	no
34	Login-LAT-Service	yes	yes	yes	yes	yes	yes	yes	yes
35	Login-LAT-Node	no	no	no	no	no	no	no	yes
36	Login-LAT-Group	no	no	no	no	no	no	no	no
37	Framed-AppleTalk-Link	no	no	no	no	no	no	no	no
38	Framed-AppleTalk- Network	no	no	no	no	no	no	no	no
39	Framed-AppleTalk-Zone	no	no	no	no	no	no	no	no
40	Acct-Status-Type	yes	yes	yes	yes	yes	yes	yes	yes
41	Acct-Delay-Time	yes	yes	yes	yes	yes	yes	yes	yes
42	Acct-Input-Octets	yes	yes	yes	yes	yes	yes	yes	yes
43	Acct-Output-Octets	yes	yes	yes	yes	yes	yes	yes	yes
44	Acct-Session-Id	yes	yes	yes	yes	yes	yes	yes	yes
45	Acct-Authentic	yes	yes	yes	yes	yes	yes	yes	yes
46	Acct-Session-Time	yes	yes	yes	yes	yes	yes	yes	yes
47	Acct-Input-Packets	yes	yes	yes	yes	yes	yes	yes	yes
48	Acct-Output-Packets	yes	yes	yes	yes	yes	yes	yes	yes
49	Acct-Terminate-Cause	no	no	no	yes	yes	yes	yes	yes
50	Acct-Multi-Session-Id	no	yes	yes	yes	yes	yes	yes	yes
51	Acct-Link-Count	no	yes	yes	yes	yes	yes	yes	yes
52	Acct-Input-Gigawords	no	no	no	no	no	no	no	no
53	Acct-Output-Gigawords	no	no	no	no	no	no	no	no
55	Event-Timestamp	no	no	no	no	no	no	no	yes
60	CHAP-Challenge	yes	yes	yes	yes	yes	yes	yes	yes
61	NAS-Port-Type	yes	yes	yes	yes	yes	yes	yes	yes

Table 71 Supported RADIUS IETF Attributes (continued)

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1	12.2
62	Port-Limit	yes	yes	yes	yes	yes	yes	yes	yes
63	Login-LAT-Port	no	no	no	no	no	no	no	no
64	Tunnel-Type[1]	no	no	no	no	no	no	yes	yes
65	Tunnel-Medium-Type[1]	no	no	no	no	no	no	yes	yes
66	Tunnel-Client-Endpoint	no	no	no	no	no	no	yes	yes
67	Tunnel-Server-Endpoint[1]	no	no	no	no	no	no	yes	yes
68	Acct-Tunnel-Connection-ID	no	no	no	no	no	no	yes	yes
69	Tunnel-Password[1]	no	no	no	no	no	no	yes	yes
70	ARAP-Password	no	no	no	no	no	no	no	no
71	ARAP-Features	no	no	no	no	no	no	no	no
72	ARAP-Zone-Access	no	no	no	no	no	no	no	no
73	ARAP-Security	no	no	no	no	no	no	no	no
74	ARAP-Security-Data	no	no	no	no	no	no	no	no
75	Password-Retry	no	no	no	no	no	no	no	no
76	Prompt	no	no	no	no	no	no	yes	yes
77	Connect-Info	no	no	no	no	no	no	no	yes
78	Configuration-Token	no	no	no	no	no	no	no	no
79	EAP-Message	no	no	no	no	no	no	no	no
80	Message-Authenticator	no	no	no	no	no	no	no	no
81	Tunnel-Private-Group-ID	no	no	no	no	no	no	no	no
82	Tunnel-Assignment-ID[1]	no	no	no	no	no	no	yes	yes
83	Tunnel-Preference	no	no	no	no	no	no	no	yes
84	ARAP-Challenge-Response	no	no	no	no	no	no	no	no
85	Acct-Interim-Interval	no	no	no	no	no	no	yes	yes
86	Acct-Tunnel-Packets-Lost	no	no	no	no	no	no	no	no
87	NAS-Port-ID	no	no	no	no	no	no	no	no
88	Framed-Pool	no	no	no	no	no	no	no	no
90	Tunnel-Client-Auth-ID[2]	no	no	no	no	no	no	no	yes
91	Tunnel-Server-Auth-ID	no	no	no	no	no	no	no	yes
200	IETF-Token-Immediate	no	no	no	no	no	no	no	no

1. This RADIUS attribute complies with the following two draft IETF documents: RFC 2868 *RADIUS Attributes for Tunnel Protocol Support* and RFC 2867 *RADIUS Accounting Modifications for Tunnel Protocol Support.*
2. This RADIUS attribute complies withRFC 2865 and RFC 2868.

Comprehensive List of RADIUS Attribute Descriptions

Table 72 lists and describes IETF RADIUS attributes. In cases where the attribute has a security server-specific format, the format is specified.

Table 72 RADIUS IETF Attributes

Number	IETF Attribute	Description
1	User-Name	Indicates the name of the user being authenticated by the RADIUS server.
2	User-Password	Indicates the user's password or the user's input following an Access-Challenge. Passwords longer than 16 characters are encrypted using RFC 2865 specifications.
3	CHAP-Password	Indicates the response value provided by a PPP Challenge-Handshake Authentication Protocol (CHAP) user in response to an Access-Challenge.
4	NAS-IP Address	Specifies the IP address of the network access server that is requesting authentication. The default value is 0.0.0.0/0.
5	NAS-Port	Indicates the physical port number of the network access server that is authenticating the user. The NAS-Port value (32 bits) consists of one or two 16-bit values (depending on the setting of the **radius-server extended-portnames** command). Each 16-bit number should be viewed as a 5-digit decimal integer for interpretation as follows: For asynchronous terminal lines, async network interfaces, and virtual async interfaces, the value is **00ttt**, where **ttt** is the line number or async interface unit number. For ordinary synchronous network interface, the value is **10xxx**. For channels on a primary rate ISDN interface, the value is **2ppcc**. For channels on a basic rate ISDN interface, the value is **3bb0c**. For other types of interfaces, the value is **6nnss**.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
6	Service-Type	Indicates the type of service requested or the type of service to be provided. • In a request: Framed for known PPP or SLIP connection. Administrative-user for **enable** command. • In response: Login—Make a connection. Framed—Start SLIP or PPP. Administrative User—Start an EXEC or **enable ok**. Exec User—Start an EXEC session. Service type is indicated by a particular numeric value as follows: • 1: Login • 2: Framed • 3: Callback-Login • 4: Callback-Framed • 5: Outbound • 6: Administrative • 7: NAS-Prompt • 8: Authenticate Only • 9: Callback-NAS-Prompt
7	Framed-Protocol	Indicates the framing to be used for framed access. No other framing is allowed. Framing is indicated by a numeric value as follows: • 1: PPP • 2: SLIP • 3: ARA • 4: Gandalf-proprietary single-link/multilink protocol • 5: Xylogics-proprietary IPX/SLIP
8	Framed-IP-Address	Indicates the IP address to be configured for the user, by sending the IP address of a user to the RADIUS server in the access-request. To enable this command, use the **radius-server attribute 8 include-in-access-req** command in global configuration mode.
9	Framed-IP-Netmask	Indicates the IP netmask to be configured for the user when the user is a router to a network. This attribute value results in a static route being added for Framed-IP-Address with the mask specified.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
10	Framed-Routing	Indicates the routing method for the user when the user is a router to a network. Only "None" and "Send and Listen" values are supported for this attribute. Routing method is indicated by a numeric value as follows: • 0: None • 1: Send routing packets • 2: Listen for routing packets • 3: Send routing packets and listen for routing packets
11	Filter-Id	Indicates the name of the filter list for the user and is formatted as follows: %d, %d.in, or %d.out. This attribute is associated with the most recent service-type command. For login and EXEC, use %d or %d.out as the line access list value from 0 to 199. For Framed service, use %d or %d.out as interface output access list, and %d.in for input access list. The numbers are self-encoding to the protocol to which they refer.
12	Framed-MTU	Indicates the maximum transmission unit (MTU) that can be configured for the user when the MTU is not negotiated by PPP or some other means.
13	Framed-Compression	Indicates a compression protocol used for the link. This attribute results in a "/compress" being added to the PPP or SLIP autocommand generated during EXEC authorization. Not currently implemented for non-EXEC authorization. Compression protocol is indicated by a numeric value as follows: • 0: None • 1: VJ-TCP/IP header compression • 2: IPX header compression
14	Login-IP-Host	Indicates the host to which the user will connect when the Login-Service attribute is included. (This begins immediately after login.)
15	Login-Service	Indicates the service that should be used to connect the user to the login host. Service is indicated by a numeric value as follows: • 0: Telnet • 1: Rlogin • 2: TCP-Clear • 3: PortMaster • 4: LAT
16	Login-TCP-Port	Defines the TCP port with which the user is to be connected when the Login-Service attribute is also present.
18	Reply-Message	Indicates text that might be displayed to the user via the RADIUS server. You can include this attribute in user files; however, you cannot exceed a maximum of 16 Replyp-Message entries per profile.
19	Callback-Number	Defines a dialing string to be used for callback.
20	Callback-ID	Defines the name (consisting of one or more octets) of a place to be called, to be interpreted by the network access server.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
22	Framed-Route	Provides routing information to be configured for the user on this network access server. The RADIUS RFC format (net/bits [router [metric]]) and the old style dotted mask (net mask [router [metric]]) are supported. If the router field is omitted or 0, the peer IP address is used. Metrics are currently ignored. This attribute is access-request packets.
23	Framed-IPX-Network	Defines the IPX network number configured for the user.
24	State	Allows state information to be maintained between the network access server and the RADIUS server. This attribute is applicable only to CHAP challenges.
25	Class	(Accounting) Arbitrary value that the network access server includes in all accounting packets for this user if supplied by the RADIUS server.
26	Vendor-Specific	Allows vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named “cisco-avpair.” The value is a string of the format: `protocol : attribute sep value` “Protocol” is a value of the Cisco “protocol” attribute for a particular type of authorization. “Attribute” and “value” are an appropriate AV pair defined in the Cisco TACACS+ specification, and “sep” is “=” for mandatory attributes and “*” for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS. For example: `cisco-avpair= ”ip:addr-pool=first“` `cisco-avpair= ”shell:priv-lvl=15“` The first example causes Cisco’s “multiple named ip address pools” feature to be activated during IP authorization (during PPP’s IPCP address assignment). The second example causes a user logging in from a network access server to have immediate access to EXEC commands. Table 71 lists supported vendor-specific RADIUS attributes (IETF attribute 26). The “TACACS+ Attribute-Value Pairs” appendix provides a complete list of supported TACACS+ attribute-value (AV) pairs that can be used with IETF attribute 26. (RFC 2865)
27	Session-Timeout	Sets the maximum number of seconds of service to be provided to the user before the session terminates. This attribute value becomes the per-user “absolute timeout.”
28	Idle-Timeout	Sets the maximum number of consecutive seconds of idle connection allowed to the user before the session terminates. This attribute value becomes the per-user “session-timeout.”
29	Termination-Action	Termination is indicated by a numeric value as follows: • 0: Default • 1: RADIUS request
30	Called-Station-Id	(Accounting) Allows the network access server to send the telephone number the user called as part of the Access-Request packet (using Dialed Number Identification Service [DNIS] or similar technology). This attribute is only supported on ISDN, and modem calls on the Cisco AS5200 if used with PRI.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
31	Calling-Station-Id	(Accounting) Allows the network access server to send the telephone number the call came from as part of the Access-Request packet (using Automatic Number Identification or similar technology). This attribute has the same value as "remote-addr" from TACACS+. This attribute is only supported on ISDN, and modem calls on the Cisco AS5200 if used with PRI.
32	NAS-Identifier	String identifying the network access server originating the Access-Request. Use the **radius-server attribute 32 include-in-access-req** global configuration command to send RADIUS attribute 32 in an Access-Request or Accounting-Request. By default, the FQDN is sent in the attribute when the format is not specified.
33	Proxy-State	Attribute that can be sent by a proxy server to another server when forwarding Access-Requests; this must be returned unmodified in the Access-Accept, Access-Reject or Access-Challenge and removed by the proxy server before sending the response to the network access server.
34	Login-LAT-Service	Indicates the system with which the user is to be connected by LAT. This attribute is only available in the EXEC mode.
35	Login-LAT-Node	Indicates the node with which the user is to be automatically connected by LAT.
36	Login-LAT-Group	Identifies the LAT group codes that this user is authorized to use.
37	Framed-AppleTalk-Link	Indicates the AppleTalk network number that should be used for serial links to the user, which is another AppleTalk router.
38	Framed-AppleTalk-Network	Indicates the AppleTalk network number that the network access server uses to allocate an AppleTalk node for the user.
39	Framed-AppleTalk-Zone	Indicates the AppleTalk Default Zone to be used for this user.
40	Acct-Status-Type	(Accounting) Indicates whether this Accounting-Request marks the beginning of the user service (start) or the end (stop).
41	Acct-Delay-Time	(Accounting) Indicates how many seconds the client has been trying to send a particular record.
42	Acct-Input-Octets	(Accounting) Indicates how many octets have been received from the port over the course of this service being provided.
43	Acct-Output-Octets	(Accounting) Indicates how many octets have been sent to the port in the course of delivering this service.
44	Acct-Session-Id	(Accounting) A unique accounting identifier that makes it easy to match start and stop records in a log file. Acct-Session ID numbers restart at 1 each time the router is power cycled or the software is reloaded. To send this attribute in access-request packets, use the **radius-server attribute 44 include-in-access-req** command in global configuration mode.
45	Acct-Authentic	(Accounting) Indicates how the user was authenticated, whether by RADIUS, the network access server itself, or another remote authentication protocol. This attribute is set to "radius" for users authenticated by RADIUS; "remote" for TACACS+ and Kerberos; or "local" for local, enable, line, and if-needed methods. For all other methods, the attribute is omitted.
46	Acct-Session-Time	(Accounting) Indicates how long (in seconds) the user has received service.
47	Acct-Input-Packets	(Accounting) Indicates how many packets have been received from the port over the course of this service being provided to a framed user.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
48	Acct-Output-Packets	(Accounting) Indicates how many packets have been sent to the port in the course of delivering this service to a framed user.
49	Acct-Terminate-Cause	(Accounting) Reports details on why the connection was terminated. Termination causes are indicated by a numeric value as follows: 1. User request 2. Lost carrier 3. Lost service 4. Idle timeout 5. Session timeout 6. Admin reset 7. Admin reboot 8. Port error 9. NAS error 10. NAS request 11. NAS reboot 12. Port unneeded 13. Port pre-empted 14. Port suspended 15. Service unavailable 16. Callback 17. User error 18. Host request **Note** For attribute 49, Cisco IOS supports values 1 to 6, 9, 12, and 15 to 18.
50	Acct-Multi-Session-Id	(Accounting) A unique accounting identifier used to link multiple related sessions in a log file. Each linked session in a multilink session has a unique Acct-Session-Id value, but shares the same Acct-Multi-Session-Id.
51	Acct-Link-Count	(Accounting) Indicates the number of links known in a given multilink session at the time an accounting record is generated. The network access server can include this attribute in any accounting request that might have multiple links.
52	Acct-Input-Gigawords	Indicates how many times the Acct-Input-Octets counter has wrapped around 2^32 over the course of the provided service.
53	Acct-Output-Gigawords	Indicates how many times the Acct-Output-Octets counter has wrapped around 2^32 while delivering service.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
55	Event-Timestamp	Records the time that the event occurred on the NAS; the timestamp sent in attribute 55 is in seconds since January 1, 1970 00:00 UTC. To send RADIUS attribute 55 in accounting packets, use the **radius-server attribute 55 include-in-acct-req** command. **Note** Before the Event-Timestamp attribute can be sent in accounting packets, you *must* configure the clock on the router. (For information on setting the clock on your router, refer to section "Performing Basic System Management" in the chapter "System Management" of the *Cisco IOS Configuration Fundamentals Configuration Guide*.) To avoid configuring the clock on the router every time the router is reloaded, you can enable the **clock calendar-valid** command. (For information on this command, refer to the chapter "Basic System Management Commands" in the *Cisco IOS Configuration Fundamentals Command Reference*.
60	CHAP-Challenge	Contains the Challenge Handshake Authentication Protocol challenge sent by the network access server to a PPP CHAP user.
61	NAS-Port-Type	Indicates the type of physical port the network access server is using to authenticate the user. Physical ports are indicated by a numeric value as follows: • 0: Asynchronous • 1: Synchronous • 2: ISDN-Synchronous • 3: ISDN-Asynchronous (V.120) • 4: ISDN-Asynchronous (V.110) • 5: Virtual
62	Port-Limit	Sets the maximum number of ports provided to the user by the NAS.
63	Login-LAT-Port	Defines the port with which the user is to be connected by LAT.
64	Tunnel-Type[1]	Indicates the tunneling protocol(s) used. Cisco IOS software supports two possible values for this attribute: L2TP and L2F. If this attribute is not set, L2F is used as a default.
65	Tunnel-Medium-Type[1]	Indicates the transport medium type to use to create a tunnel. This attribute has only one available value for this release: IP. If no value is set for this attribute, IP is used as the default.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
66	Tunnel-Client-Endpoint	Contains the address of the initiator end of the tunnel. It *may* be included in both Access-Request and Access-Accept packets to indicate the address from which a new tunnel is to be initiated. If the Tunnel-Client-Endpoint attribute is included in an Access-Request packet, the RADIUS server should take the value as a hint; the server is not obligated to honor the hint, however. This attribute *should* be included in Accounting-Request packets that contain Acct-Status-Type attributes with values of either Start or Stop, in which case it indicates the address from which the tunnel was initiated. This attribute, along with the Tunnel-Server-Endpoint and Acct-Tunnel-Connection-ID attributes, may be used to provide a globally unique means to identify a tunnel for accounting and auditing purposes. An enhancement has been added for the network access server to accept a value of 127.0.0.X for this attribute such that: 127.0.0.0 would indicate that loopback0 IP address is to be used 127.0.0.1 would indicate that loopback1 IP address is to be used ... 127.0.0.X would indicate that loopbackX IP address is to be used for the actual tunnel client endpoint IP address. This enhancement adds scalability across multiple network access servers.
67	Tunnel-Server-Endpoint[1]	Indicates the address of the server end of the tunnel. The format of this attribute varies depending on the value of Tunnel-Medium-Type. Because this release only supports IP as a tunnel medium type, the IP address or the host name of LNS is valid for this attribute.
68	Acct-Tunnel-Connection-ID	Indicates the identifier assigned to the tunnel session. This attribute *should* be included in Accounting-Request packets that contain an Acct-Status-Type attribute having the value Start, Stop, or any of the values described above. This attribute, along with the Tunnel-Client-Endpoint and Tunnel-Server-Endpoint attributes, may be used to provide a means to uniquely identify a tunnel session for auditing purposes.
69	Tunnel-Password[1]	Defines the password to be used to authenticate to a remote server. This attribute is converted into different AAA attributes based on the value of Tunnel-Type: AAA_ATTR_l2tp_tunnel_pw (L2TP), AAA_ATTR_nas_password (L2F), and AAA_ATTR_gw_password (L2F). By default, all passwords received are encrypted, which can cause authorization failures when a NAS attempts to decrypt a non-encrypted password. To enable attribute 69 to receive non-encrypted passwords, use the **radius-server attribute 69 clear** global configuration command.
70	ARAP-Password	Identifies an Access-Request packet containing a Framed-Protocol of ARAP.
71	ARAP-Features	Includes password information that the NAS should send to the user in an ARAP "feature flags" packet.
72	ARAP-Zone-Access	Indicates how the ARAP zone list for the user should be used.
73	ARAP-Security	Identifies the ARAP Security Module to be used in an Access-Challenge packet.
74	ARAP-Security-Data	Contains the actual security module challenge or response. It can be found in Access-Challenge and Access-Request packets.
75	Password-Retry	Indicates how many times a user may attempt authentication before being disconnected.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
76	Prompt	Indicates to the NAS whether it should echo the user's response as it is entered or not echo it. (0=no echo, 1=echo)
77	Connect-Info	Provides additional call information for modem calls. This attribute is generated in start and stop accounting records.
78	Configuration-Token	Indicates a type of user profile to be used. This attribute should be used in large distributed authentication networks based on proxy. It is sent from a RADIUS Proxy Server to a RADIUS Proxy Client in an Access-Accept; it should not be sent to a NAS.
79	EAP-Message	Encapsulates Extended Access Protocol (EAP) packets that allow the NAS to authenticate dial-in users via EAP without having to understand the EAP protocol.
80	Message-Authenticator	Prevents spoofing Access-Requests using CHAP, ARAP, or EAP authentication methods.
81	Tunnel-Private-Group-ID	Indicates the group ID for a particular tunneled session.
82	Tunnel-Assignment-ID[1]	Indicates to the tunnel initiator the particular tunnel to which a session is assigned.
83	Tunnel-Preference	Indicates the relative preference assigned to each tunnel. This attribute should be included if more than one set of tunneling attributes is returned by the RADIUS server to the tunnel initiator.
84	ARAP-Challenge-Response	Contains the response to the challenge of the dial-in client.
85	Acct-Interim-Interval	Indicates the number of seconds between each interim update in seconds for this specific session. This value can only appear in the Access-Accept message.
86	Acct-Tunnel-Packets-Lost	Indicates the number of packets lost on a given link. This attribute should be included in Accounting-Request packets that contain an Acct-Status-Type attribute having the value Tunnel-Link-Stop.
87	NAS-Port-ID	Contains a text string which identifies the port of the NAS that is authenticating the user.
88	Framed-Pool	Contains the name of an assigned address pool that should be used to assign an address for the user. If a NAS does not support multiple address pools, the NAS should ignore this attribute.
90	Tunnel-Client-Auth-ID	Specifies the name used by the tunnel initiator (also known as the NAS) when authenticating tunnel setup with the tunnel terminator. Supports L2F and L2TP protocols.
91	Tunnel-Server-Auth-ID	Specifies the name used by the tunnel terminator (also known as the Home Gateway) when authenticating tunnel setup with the tunnel initiator. Supports L2F and L2TP protocols.
200	IETF-Token-Immediate	Determines how RADIUS treats passwords received from login-users when their file entry specifies a hand-held security card server. The value for this attribute is indicated by a numeric value as follows: • 0: No, meaning that the password is ignored. • 1: Yes, meaning that the password is used for authentication.

1. This RADIUS attribute complies with the following two IETF documents: RFC 2868, *RADIUS Attributes for Tunnel Protocol Support* and RFC 2867, *RADIUS Accounting Modifications for Tunnel Protocol Support*.

RADIUS Vendor-Specific Attributes (VSA) and RADIUS Disconnect-Cause Attribute Values

First Published: September 23, 2005
Last Updated: September 26, 2008

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (attribute 26). Attribute 26 encapsulates vendor specific attributes, thereby, allowing vendors to support their own extended attributes otherwise not suitable for general use.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your Cisco IOS software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for RADIUS Vendor-Specific Attributes (VSA) and RADIUS Disconnect-Cause Attribute Values" section on page 14.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About RADIUS Vendor-Specific Attributes (VSA) and RADIUS Disconnect-Cause Attribute Values

The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of the following format:

```
protocol : attribute sep value *
```

"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization; protocols that can be used include IP, IPX, VPDN, VOIP, SHELL, RSVP, SIP, AIRNET, OUTBOUND. "Attribute" and "value" are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS.

For example, the following AV pair causes Cisco's "multiple named ip address pools" feature to be activated during IP authorization (during PPP's IPCP address assignment):

```
cisco-avpair= ″ip:addr-pool=first“
```

If you insert an "*", the AV pair "ip:addr-pool=first" becomes optional. Note that any AV pair can be made optional.

```
cisco-avpair= ″ip:addr-pool*first“
```

The following example shows how to cause a user logging in from a network access server to have immediate access to EXEC commands:

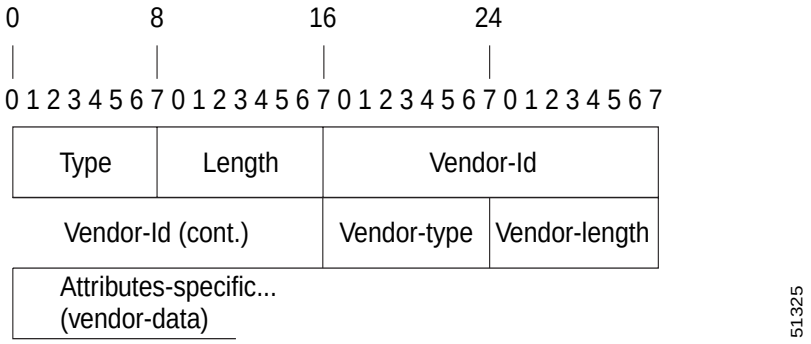
```
cisco-avpair= ″shell:priv-lvl=15“
```

Attribute 26 contains the following three elements:

- Type
- Length
- String (also known as data)
  - Vendor-Id
  - Vendor-Type
  - Vendor-Length
  - Vendor-Data

Figure 1 shows the packet format for a VSA encapsulated "behind" attribute 26.

Figure 1 VSA Encapsulated Behind Attribute 26



Note It is up to the vendor to specify the format of their VSA. The Attribute-Specific field (also known as Vendor-Data) is dependent on the vendor's definition of that attribute.

Table 2 lists supported vendor-specific RADIUS attributes (IETF attribute 26). Table 1 describes significant fields listed in the Table 2.

Table 1 Vendor-Specific Attributes Table Field Descriptions

Field	Description
Number	All attributes listed in the following table are extensions of IETF attribute 26.
Vendor-Specific Command Codes	A defined code used to identify a particular vendor. Code 9 defines Cisco VSAs, 311 defines Microsoft VSAs, and 529 defines Ascend VSAs.
Sub-Type Number	The attribute ID number. This number is much like the ID numbers of IETF attributes, except it is a “second layer” ID number encapsulated behind attribute 26.
Attribute	The ASCII string name of the attribute.
Description	Description of the attribute.

Table 2 Vendor-Specific RADIUS IETF Attributes

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
MS-CHAP Attributes				
26	311	1	MSCHAP-Response	Contains the response value provided by a PPP MS-CHAP user in response to the challenge. It is only used in Access-Request packets. This attribute is identical to the PPP CHAP Identifier. (RFC 2548)
26	311	11	MSCHAP-Challenge	Contains the challenge sent by a network access server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets. (RFC 2548)
VPDN Attributes				
26	9	1	l2tp-cm-local-window-size	Specifies the maximum receive window size for L2TP control messages. This value is advertised to the peer during tunnel establishment.

Table 2 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	l2tp-drop-out-of-order	Respects sequence numbers on data packets by dropping those that are received out of order. This does not ensure that sequence numbers will be sent on data packets, just how to handle them if they are received.
26	9	1	l2tp-hello-interval	Specifies the number of seconds for the hello keepalive interval. Hello packets are sent when no data has been sent on a tunnel for the number of seconds configured here.
26	9	1	l2tp-hidden-avp	When enabled, sensitive AVPs in L2TP control messages are scrambled or hidden.
26	9	1	l2tp-nosession-timeout	Specifies the number of seconds that a tunnel will stay active with no sessions before timing out and shutting down.
26	9	1	tunnel-tos-reflect	Copies the IP ToS field from the IP header of each payload packet to the IP header of the tunnel packet for packets entering the tunnel at the LNS.
26	9	1	l2tp-tunnel-authen	If this attribute is set, it performs L2TP tunnel authentication.
26	9	1	l2tp-tunnel-password	Shared secret used for L2TP tunnel authentication and AVP hiding.
26	9	1	l2tp-udp-checksum	This is an authorization attribute and defines whether L2TP should perform UDP checksums for data packets. Valid values are “yes” and “no.” The default is no.
Store and Forward Fax Attributes				
26	9	3	Fax-Account-Id-Origin	Indicates the account ID origin as defined by system administrator for the **mmoip aaa receive-id** or the **mmoip aaa send-id** commands.
26	9	4	Fax-Msg-Id=	Indicates a unique fax message identification number assigned by Store and Forward Fax.
26	9	5	Fax-Pages	Indicates the number of pages transmitted or received during this fax session. This page count includes cover pages.
26	9	6	Fax-Coverpage-Flag	Indicates whether or not a cover page was generated by the off-ramp gateway for this fax session. True indicates that a cover page was generated; false means that a cover page was not generated.
26	9	7	Fax-Modem-Time	Indicates the amount of time in seconds the modem sent fax data (x) and the amount of time in seconds of the total fax session (y), which includes both fax-mail and PSTN time, in the form x/y. For example, 10/15 means that the transfer time took 10 seconds, and the total fax session took 15 seconds.

Table 2 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	8	Fax-Connect-Speed	Indicates the modem speed at which this fax-mail was initially transmitted or received. Possible values are 1200, 4800, 9600, and 14400.
26	9	9	Fax-Recipient-Count	Indicates the number of recipients for this fax transmission. Until e-mail servers support Session mode, the number should be 1.
26	9	10	Fax-Process-Abort-Flag	Indicates that the fax session was aborted or successful. True means that the session was aborted; false means that the session was successful.
26	9	11	Fax-Dsn-Address	Indicates the address to which DSNs will be sent.
26	9	12	Fax-Dsn-Flag	Indicates whether or not DSN has been enabled. True indicates that DSN has been enabled; false means that DSN has not been enabled.
26	9	13	Fax-Mdn-Address	Indicates the address to which MDNs will be sent.
26	9	14	Fax-Mdn-Flag	Indicates whether or not message delivery notification (MDN) has been enabled. True indicates that MDN had been enabled; false means that MDN had not been enabled.
26	9	15	Fax-Auth-Status	Indicates whether or not authentication for this fax session was successful. Possible values for this field are success, failed, bypassed, or unknown.
26	9	16	Email-Server-Address	Indicates the IP address of the e-mail server handling the on-ramp fax-mail message.
26	9	17	Email-Server-Ack-Flag	Indicates that the on-ramp gateway has received a positive acknowledgment from the e-mail server accepting the fax-mail message.
26	9	18	Gateway-Id	Indicates the name of the gateway that processed the fax session. The name appears in the following format: hostname.domain-name.
26	9	19	Call-Type	Describes the type of fax activity: fax receive or fax send.
26	9	20	Port-Used	Indicates the slot/port number of the Cisco AS5300 used to either transmit or receive this fax-mail.
26	9	21	Abort-Cause	If the fax session aborts, indicates the system component that signaled the abort. Examples of system components that could trigger an abort are FAP (Fax Application Process), TIFF (the TIFF reader or the TIFF writer), fax-mail client, fax-mail server, ESMTP client, or ESMTP server.
H323 Attributes				
26	9	23	Remote-Gateway-ID (h323-remote-address)	Indicates the IP address of the remote gateway.

Table 2 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	24	Connection-ID (h323-conf-id)	Identifies the conference ID.
26	9	25	Setup-Time (h323-setup-time)	Indicates the setup time for this connection in Coordinated Universal Time (UTC) formerly known as Greenwich Mean Time (GMT) and Zulu time.
26	9	26	Call-Origin (h323-call-origin)	Indicates the origin of the call relative to the gateway. Possible values are originating and terminating (answer).
26	9	27	Call-Type (h323-call-type)	Indicates call leg type. Possible values are **telephony** and **VoIP**.
26	9	28	Connect-Time (h323-connect-time)	Indicates the connection time for this call leg in UTC.
26	9	29	Disconnect-Time (h323-disconnect-time)	Indicates the time this call leg was disconnected in UTC.
26	9	30	Disconnect-Cause (h323-disconnect-cause)	Specifies the reason a connection was taken offline per Q.931 specification.
26	9	31	Voice-Quality (h323-voice-quality)	Specifies the impairment factor (ICPIF) affecting voice quality for a call.
26	9	33	Gateway-ID (h323-gw-id)	Indicates the name of the underlying gateway.
Large Scale Dialout Attributes				
26	9	1	callback-dialstring	Defines a dialing string to be used for callback.
26	9	1	data-service	No description available.
26	9	1	dial-number	Defines the number to dial.
26	9	1	force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available.
26	9	1	map-class	Allows the user profile to reference information configured in a map class of the same name on the network access server that dials out.
26	9	1	send-auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication.

Table 2 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	send-name	PPP name authentication. To apply for PAP, do not configure the **ppp pap sent-name password** command on the interface. For PAP, "preauth:send-name" and "preauth:send-secret" will be used as the PAP username and PAP password for outbound authentication. For CHAP, "preauth:send-name" will be used not only for outbound authentication, but also for inbound authentication. For a CHAP inbound case, the NAS will use the name defined in "preauth:send-name" in the challenge packet to the caller box. **Note** The send-name attribute has changed over time: Initially, it performed the functions now provided by both the send-name and remote-name attributes. Because the remote-name attribute has been added, the send-name attribute is restricted to its current behavior.
26	9	1	send-secret	PPP password authentication. The vendor-specific attributes (VSAs) "preauth:send-name" and "preauth:send-secret" will be used as the PAP username and PAP password for outbound authentication. For a CHAP outbound case, both "preauth:send-name" and "preauth:send-secret" will be used in the response packet.
26	9	1	remote-name	Provides the name of the remote host for use in large-scale dial-out. Dialer checks that the large-scale dial-out remote name matches the authenticated name, to protect against accidental user RADIUS misconfiguration. (For example, dialing a valid phone number but connecting to the wrong router.)
Miscellaneous Attributes				
26	9	2	Cisco-NAS-Port	Specifies additional vendor specific attribute (VSA) information for NAS-Port accounting. To specify additional NAS-Port information in the form an Attribute-Value Pair (AVPair) string, use the **radius-server vsa send** global configuration command. **Note** This VSA is typically used in Accounting, but may also be used in Authentication (Access-Request) packets.
26	9	1	min-links	Sets the minimum number of links for MLP.

Table 2 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	proxyacl#<n>	Allows users to configure the downloadable user profiles (dynamic ACLs) by using the authentication proxy feature so that users can have the configured authorization to permit traffic going through the configured interfaces.
26	9	1	spi	Carries the authentication information needed by the home agent to authenticate a mobile node during registration. The information is in the same syntax as the **ip mobile secure host <addr>** configuration command. Basically it contains the rest of the configuration command that follows that string, verbatim. It provides the Security Parameter Index (SPI), key, authentication algorithm, authentication mode, and replay protection timestamp range.

For more information on configuring your NAS to recognize and use VSAs, refer to the section "Configuring Router to Use Vendor-Specific RADIUS Attributes" of the chapter *"Configuring RADIUS."*

RADIUS Disconnect-Cause Attribute Values

Disconnect-cause attribute values specify the reason a connection was taken offline. The attribute values are sent in Accounting request packets. These values are sent at the end of a session, even if the session fails to be authenticated. If the session is not authenticated, the attribute can cause stop records to be generated without first generating start records.

Table 3 lists the cause codes, values, and descriptions for the Disconnect-Cause (195) attribute.

Note The Disconnect-Cause is incremented by 1000 when it is used in RADIUS AVPairs; for example, disc-cause 4 becomes 1004.

Table 3 Disconnect-Cause Attribute Values

Cause Code	Value	Description
0	No-Reason	No reason is given for the disconnect.
1	No-Disconnect	The event was not disconnected.
2	Unknown	Reason unknown.
3	Call-Disconnect	The call has been disconnected.
4	CLID-Authentication-Failure	Failure to authenticate number of the calling-party.
9	No-Modem-Available	A modem in not available to connect the call.

Table 3 Disconnect-Cause Attribute Values (continued)

Cause Code	Value	Description
10	No-Carrier	No carrier detected. **Note** Codes 10, 11, and 12 can be sent if there is a disconnection during initial modem connection.
11	Lost-Carrier	Loss of carrier.
12	No-Detected-Result-Codes	Failure to detect modem result codes.
20	User-Ends-Session	User terminates a session. **Note** Codes 20, 22, 23, 24, 25, 26, 27, and 28 apply to EXEC sessions.
21	Idle-Timeout	Timeout waiting for user input. Codes 21, 100, 101, 102, and 120 apply to all session types.
22	Exit-Telnet-Session	Disconnect due to exiting Telnet session.
23	No-Remote-IP-Addr	Could not switch to SLIP/PPP; the remote end has no IP address.
24	Exit-Raw-TCP	Disconnect due to exiting raw TCP.
25	Password-Fail	Bad passwords.
26	Raw-TCP-Disabled	Raw TCP disabled.
27	Control-C-Detected	Control-C detected.
28	EXEC-Process-Destroyed	EXEC process destroyed.
29	Close-Virtual-Connection	User closes a virtual connection.
30	End-Virtual-Connection	Virual connected has ended.
31	Exit-Rlogin	User exists Rlogin.
32	Invalid-Rlogin-Option	Invalid Rlogin option selected.
33	Insufficient-Resources	Insufficient resources.
40	Timeout-PPP-LCP	PPP LCP negotiation timed out. **Note** Codes 40 through 49 apply to PPP sessions.
41	Failed-PPP-LCP-Negotiation	PPP LCP negotiation failed.
42	Failed-PPP-PAP-Auth-Fail	PPP PAP authentication failed.
43	Failed-PPP-CHAP-Auth	PPP CHAP authentication failed.
44	Failed-PPP-Remote-Auth	PPP remote authentication failed.
45	PPP-Remote-Terminate	PPP received a Terminate Request from remote end.
46	PPP-Closed-Event	Upper layer requested that the session be closed.
47	NCP-Closed-PPP	PPP session closed because there were no NCPs open.
48	MP-Error-PPP	PPP session closed because of an MP error.
49	PPP-Maximum-Channels	PPP session closed because maximum channels were reached.
50	Tables-Full	Disconnect due to full terminal server tables.
51	Resources-Full	Disconnect due to full internal resources.
52	Invalid-IP-Address	IP address is not valid for Telnet host.
53	Bad-Hostname	Hostname cannot be validated.

Table 3 Disconnect-Cause Attribute Values (continued)

Cause Code	Value	Description
54	Bad-Port	Port number is invalid or missing.
60	Reset-TCP	TCP connection has been reset. **Note** Codes 60 through 67 apply to Telnet or raw TCP sessions.
61	TCP-Connection-Refused	TCP connection has been refused by the host.
62	Timeout-TCP	TCP connection has timed out.
63	Foreign-Host-Close-TCP	TCP connection has been closed.
64	TCP-Network-Unreachable	TCP network is unreachable.
65	TCP-Host-Unreachable	TCP host is unreachable.
66	TCP-Network-Admin Unreachable	TCP network is unreachable for administrative reasons.
67	TCP-Port-Unreachable	TCP port in unreachable.
100	Session-Timeout	Session timed out.
101	Session-Failed-Security	Session failed for security reasons.
102	Session-End-Callback	Session terminated due to callback.
120	Invalid-Protocol	Call refused because the detected protocol is disabled.
150	RADIUS-Disconnect	Disconnected by RADIUS request.
151	Local-Admin-Disconnect	Administrative disconnect.
152	SNMP-Disconnect	Disconnected by SNMP request.
160	V110-Retries	Allowed V.110 retries have been exceeded.
170	PPP-Authentication-Timeout	PPP authentication timed out.
180	Local-Hangup	Disconnected by local hangup.
185	Remote-Hangup	Disconnected by remote end hangup.
190	T1-Quiesced	Disconnected because T1 line was quiesced.
195	Call-Duration	Disconnected because the maximum duration of the call was exceeded.
600	VPN-User-Disconnect	Call disconnected by client (through PPP). Code is sent if the LNS receives a PPP terminate request from the client.
601	VPN-Carrier-Loss	Loss of carrier. This can be the result of a physical line going dead. Code is sent when a client is unable to dial out using a dialer.
602	VPN-No-Resources	No resources available to handle the call. Code is sent when the client is unable to allocate memory (running low on memory).
603	VPN-Bad-Control-Packet	Bad L2TP or L2F control packets. This code is sent when an invalid control packet, such as missing mandatory Attribute-Value pairs (AVP), from the peer is received. When using L2TP, the code will be sent after six retransmits; when using L2F, the number of retransmits is user configurable. **Note** VPN-Tunnel-Shut will be sent if there are active sessions in the tunnel.

Table 3 Disconnect-Cause Attribute Values (continued)

Cause Code	Value	Description
604	VPN-Admin-Disconnect	Administrative disconnect. This can be the result of a VPN soft shutdown, which is when a client reaches maximum session limit or exceeds maximum hopcount. Code is sent when a tunnel is brought down by issuing the **clear vpdn tunnel** command.
605	VPN-Tunnel-Shut	Tunnel teardown or tunnel setup has failed. Code is sent when there are active sessions in a tunnel and the tunnel goes down. **Note** This code is *not* sent when tunnel authentication fails.
606	VPN-Local-Disconnect	Call is disconnected by LNS PPP module. Code is sent when the LNS sends a PPP terminate request to the client. It indicates a normal PPP disconnection initiated by the LNS.
607	VPN-Session-Limit	VPN soft shutdown is enabled. Code is sent when a call has been refused due to any of the soft shutdown restrictions previously mentioned.
608	VPN-Call-Redirect	VPN call redirect is enabled.

For Q.850 cause codes and descriptions, see the section “Internal Cause Codes for SIP and H.323” in the chapter “Cause Codes and Debug Values” of the *Cisco IOS Voice Troubleshooting and Monitoring*.

Additional References

The following sections provide references related to RADIUS Vendor-Specific Attributes (VSA) and RADIUS Disconnect-Cause Attribute Values.

Related Documents

Related Topic	Document Title
Security Features	*Cisco IOS Security Configuration Guide, Release 12.4*
Security Server Protocols	Part 2: Security Server Protocols in the *Cisco IOS Security Configuration Guide, Release 12.4*
RADIUS Configuration	Configuring RADIUS

Standards

Standard	Title
Internet Engineering Task Force (IETF) Internet Draft: Network Access Servers Requirements	*Network Access Servers Requirements: Extended RADIUS Practices*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2865	*Remote Authentication Dial In User Service (RADIUS)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for RADIUS Vendor-Specific Attributes (VSA) and RADIUS Disconnect-Cause Attribute Values

Table 4 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 4 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 4 Feature Information for RADIUS Vendor-Specific Attributes (VSA) and RADIUS Disconnect-Cause Attribute Values

Feature Name	Releases	Feature Information
RADIUS Vendor-Specific Attributes (VSA) and RADIUS Disconnect-Cause Attribute Values	12.0(30)S3s 12.3(11)YS1 12.2(33)SRC	This document discusses the Internet Engineering Task Force (IETF) draft standard, which specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (attribute 26). Attribute 26 encapsulates vendor specific attributes, thereby, allowing vendors to support their own extended attributes otherwise not suitable for general use. This feature was introduced into Cisco IOS Release 12.0(30)S3s. This feature was integrated into Cisco IOS Release 12.3(11)YS1 This feature was integrated into Cisco IOS Release 12.2(33)SRC.
Accounting of VPDN Disconnect Cause	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.
Vendor-Specific RADIUS Attributes	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

RADIUS Attribute 8 (Framed-IP-Address) in Access Requests

First Published: August 12, 2002
Last Updated: January 10, 2008

The RADIUS Attribute 8 (Framed-IP-Address) in Access Requests feature makes it possible for a network access server (NAS) to provide the RADIUS server with a hint of the user IP address in advance of user authentication. An application can be run on the RADIUS server to use this hint and build a table (map) of user names and addresses. Using the mapping information, service applications can begin preparing user login information to have available upon successful user authentication.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your Cisco IOS software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for RADIUS Attribute 8 (Framed-IP-Address) in Access Requests" section on page 7.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for RADIUS Attribute 8 (Framed-IP-Address) in Access Requests

Sending RADIUS attribute 8 in the RADIUS access requests assumes that the login host has been configured to request its IP address from the NAS server. It also assumes that the login host has been configured to accept an IP address from the NAS.

The NAS must be configured with a pool of network addresses on the interface supporting the login hosts.

Information About RADIUS Attribute 8 (Framed-IP-Address) in Access Requests

How This Feature Works

When a network device dials in to a NAS that is configured for RADIUS authentication, the NAS begins the process of contacting the RADIUS server in preparation for user authentication. Typically, the IP address of the dial-in host is not communicated to the RADIUS server until after successful user authentication. Communicating the device IP address to the server in the RADIUS access request allows other applications to begin to take advantage of that information.

As the NAS is setting up communication with the RADIUS server, the NAS assigns an IP address to the dial-in host from a pool of IP addresses configured at the specific interface. The NAS sends the IP address of the dial-in host to the RADIUS server as attribute 8. At that time, the NAS sends other user information, such as the user name, to the RADIUS server.

After the RADIUS server receives the user information from the NAS, it has two options:

- If the user profile on the RADIUS server already includes attribute 8, the RADIUS server can override the IP address sent by the NAS with the IP address defined as attribute 8 in the user profile. The address defined in the user profile is returned to the NAS.
- If the user profile does not include attribute 8, the RADIUS server can accept attribute 8 from the NAS, and the same address is returned to the NAS.

The address returned by the RADIUS server is saved in memory on the NAS for the life of the session. If the NAS is configured for RADIUS accounting, the accounting start packet sent to the RADIUS server includes the same IP address as in attribute 8. All subsequent accounting packets, updates (if configured), and stop packets will also include the same IP address provided in attribute 8.

Benefits

The RADIUS Attribute 8 (Framed-IP-Address) in Access Requests feature makes it possible to run applications on the RADIUS server that builds mapping tables of users and IP addresses. The server can then use the mapping table information in other applications, such as preparing customized user login pages in advance of a successful user authentication with the RADIUS server.

How to Configure RADIUS Attribute 8 (Framed-IP-Address) in Access Requests

This section contains the following procedures:

- Configuring RADIUS Attribute 8 in Access Requests, page 3 (required)
- Verifying RADIUS Attribute 8 in Access Requests, page 4

Configuring RADIUS Attribute 8 in Access Requests

To send RADIUS attribute 8 in the access request, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server attribute 8 include-in-access-req**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**radius-server attribute 8 include-in-access-req** **Example:** `Router(config)# radius-server attribute 8 include-in-access-req`	Sends RADIUS attribute 8 in access-request packets.

Verifying RADIUS Attribute 8 in Access Requests

To verify that RADIUS attribute 8 is being sent in access requests, perform the following steps. Attribute 8 should be present in all PPP access requests.

SUMMARY STEPS

1. **enable**
2. **more system:running-config**
3. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`more system:running-config` **Example:** `Router# more system:running-config`	Displays the contents of the current running configuration file. (Note that the **more system:running-config** command has replaced the **show running-config** command.)
Step 3	`debug radius` **Example:** `Router# debug radius`	Displays information associated with RADIUS. The output of this command shows whether attribute 8 is being sent in access requests.

Configuration Examples for RADIUS Attribute 8 (Framed-IP-Address) in Access Requests

This section provides the following configuration example:

- NAS Configuration That Sends the IP Address of the Dial-in Host to the RADIUS Server in the RADIUS Access Request

NAS Configuration That Sends the IP Address of the Dial-in Host to the RADIUS Server in the RADIUS Access Request

The following example shows a NAS configuration that sends the IP address of the dial-in host to the RADIUS server in the RADIUS access request. The NAS is configured for RADIUS authentication, authorization, and accounting (AAA). A pool of IP addresses (async1-pool) has been configured and applied at interface Async1.

```
aaa new-model
aaa authentication login default group radius
aaa authentication ppp default group radius
aaa authorization network default group radius
```

```
aaa accounting network default start-stop group radius
!
ip address-pool local
!
interface Async1
 peer default ip address pool async1-pool
!
ip local pool async1-pool 209.165.200.225 209.165.200.229
!
radius-server host 172.31.71.146 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server attribute 8 include-in-access-req
radius-server key radhost<xxx>: Example
```

Additional References

The following sections provide references related to the RADIUS Attribute 8 (Framed-IP-Address) in Access Requests feature.

Related Documents

Related Topic	Document Title
Configuring authentication and configuring RADIUS	"Configuring Authentication" and "Configuring RADIUS" chapters, *Cisco Security Configuration Guide*
RFC 2138 (RADIUS)	RFC 2138, *Remote Authentication Dial In User Service (RADIUS)*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Security Command Reference* at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Cisco IOS Master Commands list.

- **radius-server attribute 8 include-in-access-req**

Feature Information for RADIUS Attribute 8 (Framed-IP-Address) in Access Requests

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for RADIUS Attribute 8 (Framed-IP-Address) in Access Requests

Feature Name	Releases	Feature Information
RADIUS Attribute 8 (Framed-IP-Address) in Access Requests	12.2(11)T 12.2(28)SB 12.2(33)SRC	The following sections provide information about this feature: • Information About RADIUS Attribute 8 (Framed-IP-Address) in Access Requests, page 2 • How to Configure RADIUS Attribute 8 (Framed-IP-Address) in Access Requests, page 3 • Configuration Examples for RADIUS Attribute 8 (Framed-IP-Address) in Access Requests, page 4 The following commands were introduced or modified: **radius-server attribute 8 include-in-access-req**.
Sticky IP	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

RADIUS Tunnel Attribute Extensions

Feature History

Release	Modification
12.1(5)T	This feature was introduced.
12.2(4)B3	This feature was integrated into Cisco IOS Release 12.2(4)B3.
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2(13)T.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This feature module describes the RADIUS Tunnel Attribute Extensions feature. It includes the following sections:

Feature Overview

The RADIUS Tunnel Attribute Extensions feature introduces RADIUS attribute 90 (Tunnel-Client-Auth-ID) and RADIUS attribute 91 (Tunnel-Server-Auth-ID). Both attributes help support the provision of compulsory tunneling in virtual private networks (VPNs) by allowing the user to specify authentication names for the network access server (NAS) and the RADIUS server.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

How It Works

Once a NAS has set up communication with a RADIUS server, you can enable a tunneling protocol. Some applications of tunneling protocols are voluntary, but others involve compulsory tunneling; that is, a tunnel is created without any action from the user and without allowing the user any choice in the matter. In those cases, new RADIUS attributes are needed to carry the tunneling information from the NAS to the RADIUS server to establish authentication. These new RADIUS attributes are listed in Table 79.

Note In compulsory tunneling, any security measures in place apply only to traffic between the tunnel endpoints. Encryption or integrity protection of tunneled traffic must not be considered as a replacement for end-to-end security.

Table 79 RADIUS Tunnel Attributes

Number	IETF RADIUS Tunnel Attribute	Equivalent TACACS+ Attribute	Supported Protocols	Description
90	Tunnel-Client-Auth-ID	tunnel-id	• Layer 2 Forwarding (L2F) • Layer 2 Tunneling Protocol (L2TP)	Specifies the name used by the tunnel initiator (also known as the NAS[1]) when authenticating tunnel setup with the tunnel terminator.
91	Tunnel-Server-Auth-ID	gw-name	• Layer 2 Forwarding (L2F) • Layer 2 Tunneling Protocol (L2TP)	Specifies the name used by the tunnel terminator (also known as the Home Gateway[2]) when authenticating tunnel setup with the tunnel initiator.

1. When L2TP is used, the NAS is referred to as an L2TP access concentrator (LAC).
2. When L2TP is used, the Home Gateway is referred to as an L2TP network server (LNS).

RADIUS attribute 90 and RADIUS attribute 91 are included in the following situations:

- If the RADIUS server accepts the request and the desired authentication name is different from the default, they must be included it.
- If an accounting request contains Acct-Status-Type attributes with values of either start or stop and pertains to a tunneled session, they should be included in.

Benefits

The RADIUS Tunnel Attribute Extensions feature allows you to specify a name (other than the default) of the tunnel initiator and the tunnel terminator. Thus, you can establish a higher level of security when setting up VPN tunneling.

Restrictions

Your RADIUS server must support tagged attributes to use RADIUS tunnel attributes 90 and 91.

Related Documents

The following documents provide information related to the RADIUS Tunnel Attribute Extensions feature:

- The chapters “Configuring Authentication” and “Configuring RADIUS” in the *Cisco IOS Security Configuration Guide*, Release 12.2
- The appendix “RADIUS Attributes” in the *Cisco IOS Security Configuration Guide*, Release 12.2
- The chapter “Configuring Virtual Private Networks” in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
- RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

Supported Platforms

Cisco IOS Release 12.1(5)T Only

- AS5300
- AS5800

Cisco IOS Releases 12.2(4)B3 and 12.2(13)T Only

Cisco 6400-NRP-1

Cisco 6400-NRP-2

Cisco 6400-NRP-2SV

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

- RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

Prerequisites

To use RADIUS attributes 90 and 91, you must complete the following tasks:

- Configure your NAS to support AAA.
- Configure your NAS to support RADIUS.
- Configure your NAS to support VPN.

Configuration Tasks

None

Verifying RADIUS Attribute 90 and RADIUS Attribute 91

To verify that RADIUS attribute 90 and RADIUS attribute 91 are being sent in access accepts and accounting requests, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug radius`	Displays information associated with RADIUS. The output of this command shows whether attribute 90 and attribute 91 are being sent in access accepts and accounting requests.

Configuration Examples

This section provides the following configuration examples:

- L2TP Network Server (LNS) Configuration Example
- RADIUS User Profile with RADIUS Tunneling Attributes 90 and 91 Example

L2TP Network Server (LNS) Configuration Example

The following example shows how to configure the LNS with a basic L2F and L2TP configuration using RADIUS tunneling attributes 90 and 91:

```
aaa new-model
aaa authentication login default none
aaa authentication login console none
aaa authentication ppp default local group radius
aaa authorization network default group radius if-authenticated
!
username l2f-cli-auth-id password 0 l2f-cli-pass
username l2f-svr-auth-id password 0 l2f-svr-pass
username l2tp-svr-auth-id password 0 l2tp-tnl-pass
!
vpdn enable
vpdn search-order domain
!
vpdn-group 1
accept-dialin
protocol l2f
virtual-template 1
terminate-from hostname l2f-cli-auth-id
local name l2f-svr-auth-id
!
vpdn-group 2
accept-dialin
protocol l2tp
virtual-template 2
terminate-from hostname l2tp-cli-auth-id
local name l2tp-svr-auth-id
!
interface Ethernet1/0
ip address 10.0.0.3 255.255.255.0
no ip route-cache
no ip mroute-cache
```

```
!
interface Virtual-Template1
ip unnumbered Ethernet1/0
ppp authentication pap
!
interface Virtual-Template2
ip unnumbered Ethernet1/0
ppp authentication pap
!
radius-server host 1.1.1.1 auth-port 1645 acct-port 1646
radius-server key <deleted>
!
```

RADIUS User Profile with RADIUS Tunneling Attributes 90 and 91 Example

The following is an example of a RADIUS user profile that includes RADIUS tunneling attributes 90 and 91. This entry supports two tunnels, one for L2F and the other for L2TP. The tag entries with :1 support L2F tunnels, and the tag entries with :2 support L2TP tunnels.

```
cisco.com Password = "cisco", Service-Type = Outbound
    Service-Type = Outbound,
    Tunnel-Type = :1:L2F,
    Tunnel-Medium-Type = :1:IP,
    Tunnel-Client-Endpoint = :1:"10.0.0.2",
    Tunnel-Server-Endpoint = :1:"10.0.0.3",
    Tunnel-Client-Auth-Id = :1:"l2f-cli-auth-id",
    Tunnel-Server-Auth-Id = :1:"l2f-svr-auth-id",
    Tunnel-Assignment-Id = :1:"l2f-assignment-id",
    Cisco-Avpair = "vpdn:nas-password=l2f-cli-pass",
    Cisco-Avpair = "vpdn:gw-password=l2f-svr-pass",
    Tunnel-Preference = :1:1,
    Tunnel-Type = :2:L2TP,
    Tunnel-Medium-Type = :2:IP,
    Tunnel-Client-Endpoint = :2:"10.0.0.2",
    Tunnel-Server-Endpoint = :2:"10.0.0.3",
    Tunnel-Client-Auth-Id = :2:"l2tp-cli-auth-id",
    Tunnel-Server-Auth-Id = :2:"l2tp-svr-auth-id",
    Tunnel-Assignment-Id = :2:"l2tp-assignment-id",
    Cisco-Avpair = "vpdn:l2tp-tunnel-password=l2tp-tnl-pass",
    Tunnel-Preference = :2:2
```

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, see the Cisco IOS Security Command Reference at http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html. For information about all Cisco IOS commands, see the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Master Command List.

Glossary

Layer 2 Forwarding (L2F)—A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

Layer 2 Tunnel Protocol (L2TP)—A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

L2TP access concentrator (LAC)—A network access server (NAS) to which the client directly connects and through which PPP frames are tunneled to the L2TP network server (LNS). The LAC need only implement the media over which L2TP is to operate to pass traffic to one or more LNSs. The LAC may tunnel any protocol carried within PPP. The LAC initiates incoming calls and receives outgoing calls. A LAC is analogous to an L2F network access server.

L2TP network server (LNS)—A termination point for L2TP tunnels, and an access point where PPP frames are processed and passed to higher-layer protocols. An LNS can operate on any platform that terminates PPP. The LNS handles the server side of the L2TP protocol. L2TP relies only on the single medium over which L2TP tunnels arrive. The LNS initiates outgoing calls and receives incoming calls. An LNS is analogous to a home gateway in L2F technology.

network access server (NAS)—A Cisco platform, or collection of platforms, such as an AccessPath system, that interfaces between the packet world (such as the Internet) and the circuit-switched world (such as the PSTN).

tunnel—A virtual pipe between the L2TP access concentrator (LAC) and L2TP network server (LNS) that can carry multiple PPP sessions.

virtual private network (VPN)—A system that permits dial-in networks to exist remotely to home networks, while giving the appearance of being directly connected. VPNs use L2TP and L2F to terminate the Layer 2 and higher parts of the network connection at the L2TP network server (LNS) instead of the L2TP access concentrator (LAC).

RADIUS Vendor-Proprietary Attributes

First Published: May 15, 2001
Last Updated: September 25, 2008

The IETF draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server. However, some vendors have extended the RADIUS attribute set for specific applications. This document provides Cisco IOS support information for these vendor-proprietary RADIUS attrubutes.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for RADIUS Vendor-Proprietary Attributes" section on page 13.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Supported Vendor-Proprietary RADIUS Attributes

Table 73 lists Cisco-supported vendor-proprietary RADIUS attributes and the Cisco IOS release in which they are implemented. In cases where the attribute has a security server-specific format, the format is specified. Refer to Table 74 for a list of descriptions.

Note Attributes implemented in special (AA) or early development (T) releases will be added to the next mainline image.

Table 73 Supported Vendor-Proprietary RADIUS Attributes

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1	12.2	12.3	12.4
17	Change-Password	no	no	yes	yes	yes	yes	yes	yes	no	no
21	Password-Expiration	no	no	yes	yes	yes	yes	yes	yes	no	no
68	Tunnel-ID	no	no	no	no	no	no	no	yes	yes	yes
108	My-Endpoint-Disc-Alias	no	no	no	no	no	no	no	no	no	no
109	My-Name-Alias	no	no	no	no	no	no	no	no	no	no
110	Remote-FW	no	no	no	no	no	no	no	no	no	no
111	Multicast-GLeave-Delay	no	no	no	no	no	no	no	no	no	no
112	CBCP-Enable	no	no	no	no	no	no	no	no	no	no
113	CBCP-Mode	no	no	no	no	no	no	no	no	no	no
114	CBCP-Delay	no	no	no	no	no	no	no	no	no	no
115	CBCP-Trunk-Group	no	no	no	no	no	no	no	no	no	no
116	Appletalk-Route	no	no	no	no	no	no	no	no	no	no
117	Appletalk-Peer-Mode	no	no	no	no	no	no	no	no	no	no
118	Route-Appletalk	no	no	no	no	no	no	no	no	no	no
119	FCP-Parameter	no	no	no	no	no	no	no	no	no	no
120	Modem-PortNo	no	no	no	no	no	no	no	no	no	no
121	Modem-SlotNo	no	no	no	no	no	no	no	no	no	no
122	Modem-ShelfNo	no	no	no	no	no	no	no	no	no	no
123	Call-Attempt-Limit	no	no	no	no	no	no	no	no	no	no
124	Call-Block-Duration	no	no	no	no	no	no	no	no	no	no
125	Maximum-Call-Duration	no	no	no	no	no	no	no	no	no	no
126	Router-Preference	no	no	no	no	no	no	no	no	no	no
127	Tunneling-Protocol	no	no	no	no	no	no	no	no	no	no
128	Shared-Profile-Enable	no	no	no	no	no	no	no	no	yes	yes
129	Primary-Home-Agent	no	no	no	no	no	no	no	no	no	no
130	Secondary-Home-Agent	no	no	no	no	no	no	no	no	no	no
131	Dialout-Allowed	no	no	no	no	no	no	no	no	no	no
133	BACP-Enable	no	no	no	no	no	no	no	no	no	no
134	DHCP-Maximum-Leases	no	no	no	no	no	no	no	no	no	no
135	Primary-DNS-Server	no	no	no	no	yes	yes	yes	yes	yes	yes
136	Secondary-DNS-Server	no	no	no	no	yes	yes	yes	yes	yes	yes
137	Ascend-Client-Assign-DNS	no	no	no	no	no	no	no	no	yes	yes

Table 73 Supported Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1	12.2	12.3	12.4
138	User-Acct-Type	no	no	no	no	no	no	no	no	no	no
139	User-Acct-Host	no	no	no	no	no	no	no	no	no	no
140	User-Acct-Port	no	no	no	no	no	no	no	no	no	no
141	User-Acct-Key	no	no	no	no	no	no	no	no	no	no
142	User-Acct-Base	no	no	no	no	no	no	no	no	no	no
143	User-Acct-Time	no	no	no	no	no	no	no	no	no	no
144	Assign-IP-Client	no	no	no	no	no	no	no	no	no	no
145	Assign-IP-Server	no	no	no	no	no	no	no	no	no	no
146	Assign-IP-Global-Pool	no	no	no	no	no	no	no	no	no	no
147	DHCP-Reply	no	no	no	no	no	no	no	no	no	no
148	DHCP-Pool-Number	no	no	no	no	no	no	no	no	no	no
149	Expect-Callback	no	no	no	no	no	no	no	no	no	no
150	Event-Type	no	no	no	no	no	no	no	no	no	no
151	Ascend-Session-Svr-Key	no	no	no	yes	no	no	yes	yes	yes	yes
152	Ascend-Multicast-Rate-Limit	no	no	no	yes	no	no	yes	yes	yes	yes
153	IF-Netmask	no	no	no	no	no	no	no	no	no	no
154	h323-Remote-Address	no	no	no	no	no	no	no	no	yes	yes
155	Ascend-Multicast-Client	no	no	no	yes	no	no	yes	yes	yes	yes
156	FR-Circuit-Name	no	no	no	no	no	no	no	no	no	no
157	FR-LinkUp	no	no	no	no	no	no	no	no	no	no
158	FR-Nailed-Grp	no	no	no	no	no	no	no	no	no	no
159	FR-Type	no	no	no	no	no	no	no	no	no	no
160	FR-Link-Mgt	no	no	no	no	no	no	no	no	no	no
161	FR-N391	no	no	no	no	no	no	no	no	no	no
162	FR-DCE-N392	no	no	no	no	no	no	no	no	no	no
163	FR-DTE-N392	no	no	no	no	no	no	no	no	no	no
164	FR-DCE-N393	no	no	no	no	no	no	no	no	no	no
165	FR-DTE-N393	no	no	no	no	no	no	no	no	no	no
166	FR-T391	no	no	no	no	no	no	no	no	no	no
167	FR-T392	no	no	no	no	no	no	no	no	no	no
168	Bridge-Address	no	no	no	no	no	no	no	no	no	no
169	TS-Idle-Limit	no	no	no	no	no	no	no	no	no	no
170	TS-Idle-Mode	no	no	no	no	no	no	no	no	no	no
171	DBA-Monitor	no	no	no	no	no	no	no	no	no	no
172	Base-Channel-Count	no	no	no	no	no	no	no	no	no	no
173	Minimum-Channels	no	no	no	no	no	no	no	no	no	no

Table 73 Supported Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1	12.2	12.3	12.4
174	IPX-Route	no	no	no	no	no	no	no	no	no	no
175	FT1-Caller	no	no	no	no	no	no	no	no	no	no
176	Ipsec-Backup-Gateway	no	no	no	no	no	no	no	no	yes	yes
177	rm-Call-Type	no	no	no	no	no	no	no	no	yes	yes
178	Group	no	no	no	no	no	no	no	no	no	no
179	FR-DLCI	no	no	no	no	no	no	no	no	no	no
180	FR-Profile-Name	no	no	no	no	no	no	no	no	no	no
181	Ara-PW	no	no	no	no	no	no	no	no	no	no
182	IPX-Node-Addr	no	no	no	no	no	no	no	no	no	no
183	Home-Agent-IP-Addr	no	no	no	no	no	no	no	no	no	no
184	Home-Agent-Password	no	no	no	no	no	no	no	no	no	no
185	Home-Network-Name	no	no	no	no	no	no	no	no	no	no
186	Home-Agent-UDP-Port	no	no	no	no	no	no	no	no	no	no
187	Multilink-ID	no	no	no	yes	yes	yes	yes	yes	yes	yes
188	Ascend-Num-In-Multilink	no	no	no	yes	yes	yes	yes	yes	yes	yes
189	First-Dest	no	no	no	no	no	no	no	no	no	no
190	Pre-Input-Octets	no	no	no	yes	yes	yes	yes	yes	no	no
191	Pre-Output-Octets	no	no	no	yes	yes	yes	yes	yes	no	no
192	Pre-Input-Packets	no	no	no	yes	yes	yes	yes	yes	no	no
193	Pre-Output-Packets	no	no	no	yes	yes	yes	yes	yes	no	no
194	Maximum-Time	no	no	yes	yes	yes	yes	yes	yes	no	no
195	Disconnect-Cause	no	no	yes	yes	yes	yes	yes	yes	yes	yes
196	Connect-Progress	no	no	no	no	no	no	yes	yes	yes	yes
197	Data-Rate	no	no	no	no	yes	yes	yes	yes	yes	yes
198	PreSession-Time	no	no	no	yes	yes	yes	yes	yes	yes	yes
199	Token-Idle	no	no	no	no	no	no	no	no	yes	yes
201	Require-Auth	no	no	no	no	no	no	no	no	yes	yes
202	Number-Sessions	no	no	no	no	no	no	no	no	no	no
203	Authen-Alias	no	no	no	no	no	no	no	no	no	no
204	Token-Expiry	no	no	no	no	no	no	no	no	no	no
205	Menu-Selector	no	no	no	no	no	no	no	no	no	no
206	Menu-Item	no	no	no	no	no	no	no	no	no	no
207	PW-Warntime	no	no	no	no	no	no	no	no	no	no
208	PW-Lifetime	no	no	yes	yes	yes	yes	yes	yes	yes	yes
209	IP-Direct	no	no	no	no	yes	yes	yes	yes	yes	yes
210	PPP-VJ-Slot-Compression	no	no	yes	yes	yes	yes	yes	yes	yes	yes

Table 73 Supported Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1	12.2	12.3	12.4
211	PPP-VJ-1172	no	no	no	no	no	no	no	no	no	no
212	PPP-Async-Map	no	no	no	no	no	no	no	no	no	no
213	Third-Prompt	no	no	no	no	no	no	no	no	no	no
214	Send-Secret	no	no	no	no	no	no	yes	yes	yes	yes
215	Receive-Secret	no	no	no	no	no	no	no	no	no	no
216	IPX-Peer-Mode	no	no	no	no	no	no	no	no	no	no
217	IP-Pool	no	no	yes	yes	yes	yes	yes	yes	yes	yes
218	Static-Addr-Pool	no	no	yes	yes	yes	yes	yes	yes	yes	yes
219	FR-Direct	no	no	no	no	no	no	no	no	no	no
220	FR-Direct-Profile	no	no	no	no	no	no	no	no	no	no
221	FR-Direct-DLCI	no	no	no	no	no	no	no	no	no	no
222	Handle-IPX	no	no	no	no	no	no	no	no	no	no
223	Netware-Timeout	no	no	no	no	no	no	no	no	no	no
224	IPX-Alias	no	no	no	no	no	no	no	no	no	no
225	Metric	no	no	no	no	no	no	no	no	no	no
226	PRI-Number-Type	no	no	no	no	no	no	no	no	no	no
227	Dial-Number	no	no	no	no	no	no	yes	yes	yes	yes
228	Route-IP	no	no	yes	yes	yes	yes	yes	yes	yes	yes
229	Route-IPX	no	no	no	no	no	no	no	no	no	no
230	Bridge	no	no	no	no	no	no	no	no	no	no
231	Send-Auth	no	no	no	no	no	no	yes	yes	yes	yes
232	Send-Passwd	no	no	no	no	no	no	no	no	no	no
233	Link-Compression	no	no	yes	yes	yes	yes	yes	yes	yes	yes
234	Target-Util	no	no	no	yes	no	yes	yes	yes	yes	yes
235	Maximum-Channels	no	no	yes	yes	yes	yes	yes	yes	yes	yes
236	Inc-Channel-Count	no	no	no	no	no	no	no	no	no	no
237	Dec-Channel-Count	no	no	no	no	no	no	no	no	no	no
238	Seconds-of-History	no	no	no	no	no	no	no	no	no	no
239	History-Weigh-Type	no	no	no	no	no	no	no	no	no	no
240	Add-Seconds	no	no	no	no	no	no	no	no	no	no
241	Remove-Seconds	no	no	no	no	no	no	no	no	no	no
242	Data-Filter	no	no	yes	yes	yes	yes	yes	yes	yes	yes
243	Call-Filter	no	no	no	no	no	no	no	no	yes	yes
244	Idle-Limit	no	no	yes	yes	yes	yes	yes	yes	yes	yes
245	Preempt-Limit	no	no	no	no	no	no	no	no	no	no
246	Callback	no	no	no	no	no	no	no	no	yes	yes

Table 73 Supported Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1	12.2	12.3	12.4
247	Data-Service	no	no	no	no	no	no	yes	yes	yes	yes
248	Force-56	no	no	no	no	no	no	yes	yes	yes	yes
249	Billing Number	no	no	no	no	no	no	no	no	no	no
250	Call-By-Call	no	no	no	no	no	no	no	no	no	no
251	Transit-Number	no	no	no	no	no	no	no	no	no	no
252	Host-Info	no	no	no	no	no	no	no	no	no	no
253	PPP-Address	no	no	no	no	no	no	no	no	no	no
254	MPP-Idle-Percent	no	no	no	no	no	no	no	no	no	no
255	Xmit-Rate	no	no	no	yes	yes	yes	yes	yes	yes	yes

Comprehensive List of Vendor-Proprietary RADIUS Attribute Descriptions

Table 74 lists and describes the known vendor-proprietary RADIUS attributes:

Table 74 Vendor-Proprietary RADIUS Attributes

Number	Vendor-Proprietary Attribute	Description
17	Change-Password	Specifies a request to change the password of a user.
21	Password-Expiration	Specifies an expiration date for a user's password in the user's file entry.
68	Tunnel-ID	(Ascend 5) Specifies the string assigned by RADIUS for each session using CLID or DNIS tunneling. When accounting is implemented, this value is used for accoutning.
108	My-Endpoint-Disc-Alias	(Ascend 5) No description available.
109	My-Name-Alias	(Ascend 5) No description available.
110	Remote-FW	(Ascend 5) No description available.
111	Multicast-GLeave-Delay	(Ascend 5) No description available.
112	CBCP-Enable	(Ascend 5) No description available.
113	CBCP-Mode	(Ascend 5) No description available.
114	CBCP-Delay	(Ascend 5) No description available.
115	CBCP-Trunk-Group	(Ascend 5) No description available.
116	Appletalk-Route	(Ascend 5) No description available.
117	Appletalk-Peer-Mode	(Ascend 5) No description available.
118	Route-Appletalk	(Ascend 5) No description available.
119	FCP-Parameter	(Ascend 5) No description available.
120	Modem-PortNo	(Ascend 5) No description available.
121	Modem-SlotNo	(Ascend 5) No description available.

Table 74 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
122	Modem-ShelfNo	(Ascend 5) No description available.
123	Call-Attempt-Limit	(Ascend 5) No description available.
124	Call-Block-Duration	(Ascend 5) No description available.
125	Maximum-Call-Duration	(Ascend 5) No description available.
126	Router-Preference	(Ascend 5) No description available.
127	Tunneling-Protocol	(Ascend 5) No description available.
128	Shared-Profile-Enable	(Ascend 5) No description available.
129	Primary-Home-Agent	(Ascend 5) No description available.
130	Secondary-Home-Agent	(Ascend 5) No description available.
131	Dialout-Allowed	(Ascend 5) No description available.
133	BACP-Enable	(Ascend 5) No description available.
134	DHCP-Maximum-Leases	(Ascend 5) No description available.
135	Primary-DNS-Server	Identifies a primary DNS server that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation.
136	Secondary-DNS-Server	Identifies a secondary DNS server that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation.
137	Client-Assign-DNS	No description available.
138	User-Acct-Type	No description available.
139	User-Acct-Host	No description available.
140	User-Acct-Port	No description available.
141	User-Acct-Key	No description available.
142	User-Acct-Base	No description available.
143	User-Acct-Time	No description available.
144	Assign-IP-Client	No description available.
145	Assign-IP-Server	No description available.
146	Assign-IP-Global-Pool	No description available.
147	DHCP-Reply	No description available.
148	DHCP-Pool-Number	No description available.
149	Expect-Callback	No description available.
150	Event-Type	No description available.
151	Session-Svr-Key	No description available.
152	Multicast-Rate-Limit	No description available.
153	IF-Netmask	No description available.
154	Remote-Addr	No description available.
155	Multicast-Client	No description available.

Table 74 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
156	FR-Circuit-Name	No description available.
157	FR-LinkUp	No description available.
158	FR-Nailed-Grp	No description available.
159	FR-Type	No description available.
160	FR-Link-Mgt	No description available.
161	FR-N391	No description available.
162	FR-DCE-N392	No description available.
163	FR-DTE-N392	No description available.
164	FR-DCE-N393	No description available.
165	FR-DTE-N393	No description available.
166	FR-T391	No description available.
167	FR-T392	No description available.
168	Bridge-Address	No description available.
169	TS-Idle-Limit	No description available.
170	TS-Idle-Mode	No description available.
171	DBA-Monitor	No description available.
172	Base-Channel-Count	No description available.
173	Minimum-Channels	No description available.
174	IPX-Route	No description available.
175	FT1-Caller	No description available.
176	Backup	No description available.
177	Call-Type	No description available.
178	Group	No description available.
179	FR-DLCI	No description available.
180	FR-Profile-Name	No description available.
181	Ara-PW	No description available.
182	IPX-Node-Addr	No description available.
183	Home-Agent-IP-Addr	Indicates the home agent's IP address (in dotted decimal format) when using Ascend Tunnel Management Protocol (ATMP).
184	Home-Agent-Password	With ATMP, specifies the password that the foreign agent uses to authenticate itself.
185	Home-Network-Name	With ATMP, indicates the name of the connection profile to which the home agent sends all packets.
186	Home-Agent-UDP-Port	Indicates the UDP port number the foreign agent uses to send ATMP messages to the home agent.

Table 74 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
187	Multilink-ID	Reports the identification number of the multilink bundle when the session closes. This attribute applies to sessions that are part of a multilink bundle. The Multilink-ID attribute is sent in authentication-response packets.
188	Num-In-Multilink	Reports the number of sessions remaining in a multilink bundle when the session reported in an accounting-stop packet closes. This attribute applies to sessions that are part of a multilink bundle. The Num-In-Multilink attribute is sent in authentication-response packets and in some accounting-request packets.
189	First-Dest	Records the destination IP address of the first packet received after authentication.
190	Pre-Input-Octets	Records the number of input octets before authentication. The Pre-Input-Octets attribute is sent in accounting-stop records.
191	Pre-Output-Octets	Records the number of output octets before authentication. The Pre-Output-Octets attribute is sent in accounting-stop records.
192	Pre-Input-Packets	Records the number of input packets before authentication. The Pre-Input-Packets attribute is sent in accounting-stop records.
193	Pre-Output-Packets	Records the number of output packets before authentication. The Pre-Output-Packets attribute is sent in accounting-stop records.
194	Maximum-Time	Specifies the maximum length of time (in seconds) allowed for any session. After the session reaches the time limit, its connection is dropped.
195	Disconnect-Cause	Specifies the reason a connection was taken offline. The Disconnect-Cause attribute is sent in accounting-stop records. This attribute also causes stop records to be generated without first generating start records if disconnection occurs before authentication is performed. For more information, refer to the table of Disconnect-Cause Attribute Values and their meanings.
196	Connect-Progress	Indicates the connection state before the connection is disconnected.
197	Data-Rate	Specifies the average number of bits per second over the course of the connection's lifetime. The Data-Rate attribute is sent in accounting-stop records.
198	PreSession-Time	Specifies the length of time, in seconds, from when a call first connects to when it completes authentication. The PreSession-Time attribute is sent in accounting-stop records.
199	Token-Idle	Indicates the maximum amount of time (in minutes) a cached token can remain alive between authentications.
201	Require-Auth	Defines whether additional authentication is required for class that has been CLID authenticated.

Table 74 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
202	Number-Sessions	Specifies the number of active sessions (per class) reported to the RADIUS accounting server.
203	Authen-Alias	Defines the RADIUS server's login name during PPP authentication.
204	Token-Expiry	Defines the lifetime of a cached token.
205	Menu-Selector	Defines a string to be used to cue a user to input data.
206	Menu-Item	Specifies a single menu-item for a user-profile. Up to 20 menu items can be assigned per profile.
207	PW-Warntime	(Ascend 5) No description available.
208	PW-Lifetime	Enables you to specify on a per-user basis the number of days that a password is valid.
209	IP-Direct	When you include this attribute in a user's file entry, a framed route is installed to the routing and bridging tables. **Note** Packet routing is dependent upon the entire table, not just this newly installed entry. The inclusion of this attribute does not guarantee that all packets should be sent to the specified IP address; thus, this attribute is not fully supported. These attribute limitations occur because the Cisco router cannot bypass all internal routing and bridging tables and send packets to a specified IP address.
210	PPP-VJ-Slot-Comp	Instructs the Cisco router not to use slot compression when sending VJ-compressed packets over a PPP link.
211	PPP-VJ-1172	Instructs PPP to use the 0x0037 value for VJ compression.
212	PPP-Async-Map	Gives the Cisco router the asynchronous control character map for the PPP session. The specified control characters are passed through the PPP link as data and used by applications running over the link.
213	Third-Prompt	Defines a third prompt (after username and password) for additional user input.
214	Send-Secret	Enables an encrypted password to be used in place of a regular password in outdial profiles.
215	Receive-Secret	Enables an encrypted password to be verified by the RADIUS server.
216	IPX-Peer-Mode	(Ascend 5) No description available.
217	IP-Pool-Definition	Defines a pool of addresses using the following format: X a.b.c Z; where X is the pool index number, a.b.c is the pool's starting IP address, and Z is the number of IP addresses in the pool. For example, 3 10.0.0.1 5 allocates 10.0.0.1 through 10.0.0.5 for dynamic assignment.
218	Assign-IP-Pool	Tells the router to assign the user and IP address from the IP pool.

Table 74 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
219	FR-Direct	Defines whether the connection profile operates in Frame Relay redirect mode.
220	FR-Direct-Profile	Defines the name of the Frame Relay profile carrying this connection to the Frame Relay switch.
221	FR-Direct-DLCI	Indicates the DLCI carrying this connection to the Frame Relay switch.
222	Handle-IPX	Indicates how NCP watchdog requests will be handled.
223	Netware-Timeout	Defines, in minutes, how long the RADIUS server responds to NCP watchdog packets.
224	IPX-Alias	Allows you to define an alias for IPX routers requiring numbered interfaces.
225	Metric	No description available.
226	PRI-Number-Type	No description available.
227	Dial-Number	Defines the number to dial.
228	Route-IP	Indicates whether IP routing is allowed for the user's file entry.
229	Route-IPX	Allows you to enable IPX routing.
230	Bridge	No description available.
231	Send-Auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication.
232	Send-Passwd	Enables the RADIUS server to specify the password that is sent to the remote end of a connection on outgoing calls.
233	Link-Compression	Defines whether to turn on or turn off "stac" compression over a PPP link. Link compression is defined as a numeric value as follows: • 0: None • 1: Stac • 2: Stac-Draft-9 • 3: MS-Stac
234	Target-Util	Specifies the load-threshold percentage value for bringing up an additional channel when PPP multilink is defined.
235	Maximum-Channels	Specifies allowed/allocatable maximum number of channels.
236	Inc-Channel-Count	No description available.
237	Dec-Channel-Count	No description available.
238	Seconds-of-History	No description available.
239	History-Weigh-Type	No description available.
240	Add-Seconds	No description available.
241	Remove-Seconds	No description available.

Table 74 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
242	Data-Filter	Defines per-user IP data filters. These filters are retrieved only when a call is placed using a RADIUS outgoing profile or answered using a RADIUS incoming profile. Filter entries are applied on a first-match basis; therefore, the order in which filter entries are entered is important.
243	Call-Filter	Defines per-user IP data filters. On a Cisco router, this attribute is identical to the Data-Filter attribute.
244	Idle-Limit	Specifies the maximum time (in seconds) that any session can be idle. When the session reaches the idle time limit, its connection is dropped.
245	Preempt-Limit	No description available.
246	Callback	Allows you to enable or disable callback.
247	Data-Svc	No description available.
248	Force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available.
249	Billing Number	No description available.
250	Call-By-Call	No description available.
251	Transit-Number	No description available.
252	Host-Info	No description available.
253	PPP-Address	Indicates the IP address reported to the calling unit during PPP IPCP negotiations.
254	MPP-Idle-Percent	No description available.
255	Xmit-Rate	(Ascend 5) No description available.

For more information on vendor-propritary RADIUS attributes, refer to the section "Configuring Router for Vendor-Proprietary RADIUS Server Communication" in the chapter "Configuring RADIUS."

Feature Information for RADIUS Vendor-Proprietary Attributes

Table 75 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 75 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 75 Feature Information for RADIUS Vendor-Proprietary Attributes

Feature Name	Releases	Feature Information
RADIUS Vendor-Proprietary Attributes	12.2(1)XE	The IETF draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server. However, some vendors have extended the RADIUS attribute set for specific applications. This document provides Cisco IOS support information for these vendor-proprietary RADIUS attrubutes. In 12.2(1) XE, this feature was introduced.