Cisco IOS XE Security Configuration Guide: Securing the Data Plane

Release 2

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 XE Security Configuration Guide: Securing the Data Plane

About Cisco IOS XE Software Documentation

Last Updated: December 1, 2009

This document describes the objectives, audience, conventions, and organization used in Cisco IOS XE software 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 XE documentation describe the tasks and commands available to configure and maintain Cisco networking devices.

Audience

The Cisco IOS XE documentation set is intended 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 XE documentation set is also intended for those users experienced with Cisco IOS XE software who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS XE release.

Documentation Conventions

In Cisco IOS XE 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 XE 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 contains the following topics:

Typographic Conventions

Cisco IOS XE 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 XE 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.
...	An ellipsis (three consecutive nonbolded periods without spaces) after a syntax element indicates that the element can be repeated.
|	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 XE software uses the following 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 the Cisco IOS XE software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Reader Alert Conventions

Cisco IOS XE documentation 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 XE documentation set, how it is organized, and how to access it on Cisco.com. Listed are configuration guides, command references, and supplementary references and resources that comprise the documentation set.

Cisco IOS XE Documentation Set

The Cisco IOS XE documentation set 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 XE software. Review release notes before other documents to learn whether updates have been made to a feature.
- Sets of configuration guides and command references organized by technology and published for each standard Cisco IOS XE release.
  - Configuration guides—Compilations of documents that provide conceptual and task-oriented descriptions of Cisco IOS XE features.
  - Command references—Alphabetical compilations of command pages that provide detailed information about the commands used in the Cisco IOS XE features and the processes that comprise the related configuration guides. For each technology, there is a single command reference that covers all Cisco IOS XE releases and that is updated at each standard release.
- Command reference book for **debug** commands.
- Lists of all the commands in a specific release and all commands that are new, modified, removed, or replaced in the release.
- Reference book for system messages for all Cisco IOS XE releases.

Cisco IOS XE 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 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.

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.

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 XE commands that are supported in many different software releases and on many different platforms. The books are organized by technology. For information about all Cisco IOS XE 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.

Cisco IOS XE Supplementary Documents and Resources

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

Configuration Guides, Command References, and Supplementary Resources

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

Table 2 lists documents and resources that supplement the Cisco IOS XE 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.

For additional information about configuring and operating specific networking devices, and to access Cisco IOS documentation, go to the Product/Technologies Support area of Cisco.com at the following location:

http://www.cisco.com/go/techdocs

Table 1 Cisco IOS XE Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
• *Cisco ASR 1000 Series Aggregation Services Routers SIP and SPA Software Configuration Guide*	Configuration and troubleshooting of SPA interface processors (SIPs) and shared port adapters (SPAs) that are supported on the Cisco ASR 1000 Series Router.
• *Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide*	Overview of software functionality that is specific to the Cisco ASR 1000 Series Aggregation Services Routers.
• *Cisco IOS XE Access Node Control Protocol Configuration Guide* • *Cisco IOS Access Node Control Protocol Command Reference*	Communication protocol between digital subscriber line access multiplexers (DSLAMs) and a broadband remote access server (BRAS).
• *Cisco IOS XE Asynchronous Transfer Mode Configuration Guide* • *Cisco IOS Asynchronous Transfer Mode Command Reference*	LAN ATM, multiprotocol over ATM (MPoA), and WAN ATM.
• *Cisco IOS XE Broadband Access Aggregation and DSL Configuration Guide* • *Cisco IOS Broadband Access Aggregation and DSL Command Reference*	PPP over Ethernet (PPPoE).
• *Cisco IOS XE Carrier Ethernet Configuration Guide* • *Cisco IOS Carrier Ethernet Command Reference*	IEEE 802.3ad Link Bundling; Link Aggregation Control Protocol (LACP) support for Ethernet and Gigabit Ethernet links and EtherChannel bundles; LACP support for stateful switchover (SSO), in service software upgrade (ISSU), Cisco nonstop forwarding (NSF), and nonstop routing (NSR) on Gigabit EtherChannel bundles; and IEEE 802.3ad Link Aggregation MIB.
• *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.

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

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
• *Cisco IOS XE DECnet Configuration Guide* • *Cisco IOS DECnet Command Reference*	DECnet protocol.
• *Cisco IOS XE Dial Technologies Configuration Guide* • *Cisco IOS Dial Technologies Command Reference*	Asynchronous communications, dial backup, dialer technology, Multilink PPP (MLP), PPP, and virtual private dialup network (VPDN).
• *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 XE Intelligent Services Gateway Configuration Guide* • *Cisco IOS Intelligent Services Gateway Command Reference*	Subscriber identification, service and policy determination, session creation, session policy enforcement, session life-cycle management, accounting for access and service usage, and session state monitoring.
• *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 XE IP Addressing Services Configuration Guide* • *Cisco IOS IP Addressing Services Command Reference*	IP addressing, 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 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, TCP, Web Cache Communication Protocol (WCCP), User Datagram Protocol (UDP), and Virtual Router Redundancy Protocol (VRRP).
• *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).
• *Cisco IOS XE IP Routing: BFD Configuration Guide*	Bidirectional forwarding detection (BFD).
• *Cisco IOS XE IP Routing: BGP Configuration Guide* • *Cisco IOS IP Routing: BGP Command Reference*	Border Gateway Protocol (BGP), multiprotocol BGP, multiprotocol BGP extensions for IP multicast.
• *Cisco IOS XE IP Routing: EIGRP Configuration Guide* • *Cisco IOS IP Routing: EIGRP Command Reference*	Enhanced Interior Gateway Routing Protocol (EIGRP).
• *Cisco IOS XE IP Routing: ISIS Configuration Guide* • *Cisco IOS IP Routing: ISIS Command Reference*	Intermediate System-to-Intermediate System (IS-IS).

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

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
• *Cisco IOS XE IP Routing: ODR Configuration Guide* • *Cisco IOS IP Routing: ODR Command Reference*	On-Demand Routing (ODR).
• *Cisco IOS XE IP Routing: OSPF Configuration Guide* • *Cisco IOS IP Routing: OSPF Command Reference*	Open Shortest Path First (OSPF).
• *Cisco IOS XE IP Routing: Protocol-Independent Configuration Guide* • *Cisco IOS IP Routing: Protocol-Independent Command Reference*	IP routing protocol-independent features and commands. Generic policy-based routing (PBR) features and commands are included.
• *Cisco IOS XE IP Routing: RIP Configuration Guide* • *Cisco IOS IP Routing: RIP Command Reference*	Routing Information Protocol (RIP).
• *Cisco IOS XE IP SLAs Configuration Guide* • *Cisco IOS IP SLAs Command Reference*	Cisco IOS IP Service Level Agreements (IP SLAs).
• *Cisco IOS XE IP Switching Configuration Guide* • *Cisco IOS IP Switching Command Reference*	Cisco Express Forwarding.
• *Cisco IOS XE IPv6 Configuration Guide* • *Cisco IOS IPv6 Command Reference*	For a list of IPv6 features, protocols, and technologies, go to the IPv6 “Start Here” document at the following URL: http://www.cisco.com/en/US/docs/ios/ios_xe/ipv6/configuration/guide/ip6-roadmap_xe.html
• *Cisco IOS XE ISO CLNS Configuration Guide* • *Cisco IOS ISO CLNS Command Reference*	ISO Connectionless Network Service (CLNS).
• *Cisco IOS XE LAN Switching Configuration Guide* • *Cisco IOS LAN Switching Command Reference*	VLANs and multilayer switching (MLS).
• *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 XE NetFlow Configuration Guide* • *Cisco IOS NetFlow Command Reference*	Network traffic data analysis, aggregation caches, and export features.
• *Cisco IOS XE Network Management Configuration Guide* • *Cisco IOS Network Management Command Reference*	Basic system management, system monitoring and logging, Cisco IOS Scripting with Tool Control Language (Tcl), Cisco networking services (CNS), Embedded Event Manager (EEM), Embedded Syslog Manager (ESM), HTTP, Remote Monitoring (RMON), and SNMP.
• *Cisco IOS XE Novell IPX Configuration Guide* • *Cisco IOS Novell IPX Command Reference*	Novell Internetwork Packet Exchange (IPX) protocol.

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

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
• *Cisco IOS XE Quality of Service Solutions Configuration Guide* • *Cisco IOS Quality of Service Solutions Command Reference*	Class-based weighted fair queueing (CBWFQ), low latency queueing (LLQ), Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC), Network-Based Application Recognition (NBAR), priority queueing, Multilink PPP (MLP) for QoS, header compression, Resource Reservation Protocol (RSVP), weighted fair queueing (WFQ), and weighted random early detection (WRED).
• *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; public key infrastructure (PKI); RADIUS; and TACACS+.
• *Cisco IOS XE Security Configuration Guide: Secure Connectivity*	Internet Key Exchange (IKE) for IPsec VPNs; security for VPNs with IPsec; VPN availability features (reverse route injection, IPsec preferred peer, and real-time resolution for the IPsec tunnel peer); IPsec data plane features; IPsec management plane features; Public Key Infrastructure (PKI); Dynamic Multipoint VPN (DMVPN); Easy VPN; and Cisco Group Encrypted Transport VPN (GET VPN).
• *Cisco IOS XE Security Configuration Guide: Securing the Data Plane*	Access Control Lists (ACLs); Firewalls: Context-Based Access Control (CBAC) and Zone-Based Firewall; Cisco IOS Intrusion Prevention System (IPS); Flexible Packet Matching; Unicast Reverse Path Forwarding (uRPF); Threat Information Distribution Protocol (TIDP) and TMS.
• *Cisco IOS XE Security Configuration Guide: Securing User Services*	AAA (includes Network Admission Control [NAC]); Security Server Protocols (RADIUS and TACACS+); Secure Shell (SSH); Secure Access for Networking Devices (includes Autosecure and Role-Based CLI access); Lawful Intercept.
• *Cisco IOS XE Service Advertisement Framework Configuration Guide* • *Cisco IOS Service Advertisement Framework Command Reference*	Cisco Service Advertisement Framework.
• *Cisco IOS XE VPDN Configuration Guide* • *Cisco IOS VPDN Command Reference*	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, and tunnel authentication via RADIUS on tunnel terminator.
• *Cisco IOS XE Wide-Area Networking Configuration Guide* • *Cisco IOS Wide-Area Networking Command Reference*	Frame Relay; L2VPN Pseudowire Redundancy; and Media-Independent PPP and Multilink PPP.

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

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
• *Cisco Unified Border Element (Enterprise) Configuration Guide* • *Cisco IOS Voice Command Reference*	The Cisco Unified Border Element (Enterprise) on the Cisco ASR 1000 brings a scalable option for enterprise customers. Running as a process on the Cisco ASR 1000 and utilizing the high-speed RTP packet processing path, the Cisco Unified Border Element (Enterprise) is used as an IP-to-IP gateway by enterprises and commercial customers to interconnect SIP and H.323 voice and video networks. The Cisco UBE (Enterprise) provides a network-to-network demarcation interface for signaling interworking, media interworking, address and port translations, billing, security, quality of service (QoS), and bandwidth management.
• *Cisco Unified Border Element (SP Edition) Configuration Guide: Distributed Model* • *Cisco Unified Border Element (SP Edition) Command Reference: Distributed Model*	The Cisco Unified Border Element (SP Edition) is a session border controller (SBC) that is VoIP-enabled and deployed at the edge of networks. For Cisco IOS XE Release 2.3 and earlier releases, Cisco Unified Border Element (SP Edition) is supported only in the distributed mode. Operating in the distributed mode, the SBC is a toolkit of functions that can be used to deploy and manage VoIP services, such as signaling interworking, network hiding, security, and quality of service.
• *Cisco Unified Border Element (SP Edition) Configuration Guide: Unified Model* • *Cisco Unified Border Element (SP Edition) Command Reference: Unified Model*	The Cisco Unified Border Element (SP Edition) is a highly scalable, carrier-grade session border controller (SBC) that is designed for service providers and that is generally deployed at the border of the enterprise or SP networks to enable the easy deployment and management of VoIP services. Cisco Unified Border Element (SP Edition) is integrated into Cisco routing platforms and can use a large number of router functions to provide a very feature-rich and intelligent SBC application. Formerly known as Integrated Session Border Controller, Cisco Unified Border Element (SP Edition) provides a network-to-network demarcation interface for signaling interworking, media interworking, address and port translations, billing, security, quality of service, call admission control, and bandwidth management. For Cisco IOS XE Release 2.4 and later releases, Cisco Unified Border Element (SP Edition) can operate in two modes or deployment models: unified and distributed. The configuration guide documents the features in the unified mode.

Table 2 lists documents and resources that supplement the Cisco IOS XE software configuration guides and command references.

Table 2 Cisco IOS XE Software Supplementary Documents and Resources

Document Title or Resource	Description
Cisco IOS Master Command List, All Releases	Alphabetical list of all the commands documented in all Cisco IOS XE software releases.
Cisco IOS Debug Command Reference	Alphabetical list of **debug** commands including brief descriptions of use, command syntax, and usage guidelines.
Cisco IOS XE system messages	List of Cisco IOS XE system messages and descriptions. System messages may indicate problems with your system, may be informational only, or may help diagnose problems with communications lines, internal hardware, or the system software.
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 XE software releases.
MIBs	Files used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS XE software 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 XE 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 updated 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 XE software 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 XE Software

Last Updated: December 1, 2009

This document provides basic information about the command-line interface (CLI) in 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 "Part 1: Using the Cisco IOS Command-Line Interface (CLI)" of the *Cisco IOS XE Configuration Fundamentals Configuration Guide*.

For information about the software documentation set, see the "About 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/go/techdocs.

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 settings that you can change on a console port or 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 ASR 1000 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	The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS XE 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 XE 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 use a method that is configured to connect to the Cisco IOS XE 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 XE state. • Replace or roll back the configuration. • Provide methods of restarting the Cisco IOS XE software or other processes. • Reboot hardware, such as the entire router, an RP, an ESP, a SIP, a SPA, or 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
```

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 command history feature saves the commands that 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 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**.

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** form is documented in the command pages of command references. The **default** form is generally documented in the command pages only when the **default** form performs a different function than the plain and **no** forms of the command. To see what **default** commands are available on your system, enter **default ?** in the appropriate command mode.

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 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. You can use output modifiers to filter this output to show only the information that you want to see.

The following three output modifiers are available:

- **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 *System Messages for Cisco IOS XE* document.

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

- "Part 1: Using the Cisco IOS Command-Line Interface (CLI)" of the *Cisco IOS XE Configuration Fundamentals Configuration Guide*

  http://www.cisco.com/en/US/docs/ios/ios_xe/fundamentals/configuration/guide/2_xe/cf_xe_book.html

  or

  "Using Cisco IOS XE Software" chapter of the *Cisco ASR 1000 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/go/techdocs
- Support area on Cisco.com (also search for documentation by task or product)

  http://www.cisco.com/en/US/support/index.html
- 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 XE 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 XE 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\

Access Control Lists (ACLs)

IP Access List Overview

First Published: August 18, 2006
Last Updated: July 31, 2009

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.

Finding Feature Information

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 IP Access Lists" section on page 13.

Use Cisco Feature Navigator to find information about platform support 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

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

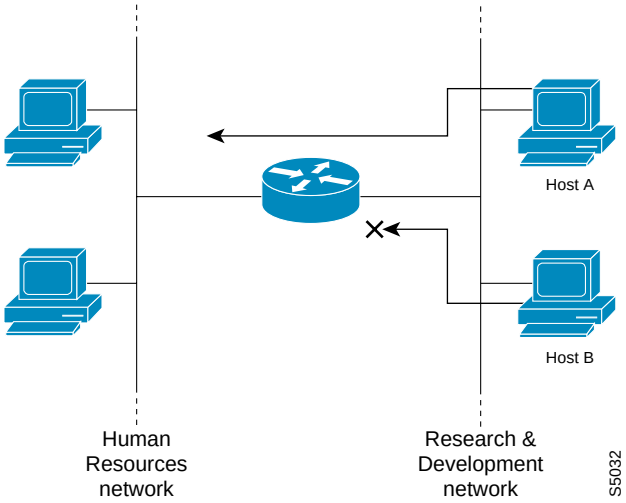
To enable the Cisco IOS XE software to receive incoming remote shell (rsh) protocol and remote copy (rcp) protocol requests, customers must configure an authentication database to control access to the router. Access lists can simplify the identification of local users, remote hosts, and remote users 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 XE 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 interface gigabitEthernet 0/1/0 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 gigabitEthernet 0/1/0
 ip access-group branchoffices in
```

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.
- 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 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
- 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 Access Lists” section on page 10 and the module “Refining an IP Access List”)

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 XE 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 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.

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.

- 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.

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*
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, or Noncontiguous Ports"

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE software 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 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 IP Access Lists

Table 2 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which 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 XE software release that introduced support for a given feature in a given Cisco IOS XE software release train. Unless noted otherwise, subsequent releases of that Cisco IOS XE software release train also support that feature.

Table 2 Feature Information for IP Access Lists

Feature Name	Releases	Feature Configuration Information
IP Named Access Control Lists	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

Creating an IP Access List and Applying It to an Interface

First Published: August 18, 2006
Last Updated: July 31, 2009

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

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 Creating IP Access Lists" section on page 20.

Use Cisco Feature Navigator to find information about platform support 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

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.
- The ASR 1000 is a hardware-based platform that uses TCAM (hardware) for ACL lookup. Therefore, where the ACE occurs in the access-list has no implications on performance. In other words, doing a lookup on the ACE is independent of where that ACE is present in the ACL.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- A packet will match the first ACE in the ACL. Thus, a **permit ip any any** will match all packets, ignoring all subsequent ACES.
- 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 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 13.

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.

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.

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, page 8
- Creating a Numbered Extended Access List, page 11

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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.

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 slot*/*subslot*/port[**.***subinterface-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.

	Command or Action	Purpose
Step 3	**interface** *type slot/subslot/port*[*.subinterface-number*] **Example:** `Router(config)# interface gigabitethernet 0/0/0`	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.

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 gigabitethernet 0 and the workstation belonging to Smith is not allowed access:

```
interface gigabitethernet0/0/0
 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 gigabitethernet interface 0, but the Main subnet is allowed access:

```
interface gigabitethernet0/0/0
 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 gigabitethernet0/0/0
 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 XE 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 gigabitethernet0/0/0
 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 gigabitethernet interface 0:

```
interface gigabitethernet0/0/0
 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 gigabitethernet0/0/0
 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 gigabitethernet 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 gigabitethernet 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 gigabitethernet0/0/0
 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 gigabitethernet0/0/0
 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 gigabitethernet0/0/0
 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
```

Additional References

The following sections provide references related to IP access lists.

Related Documents

Related Topic	Document Title
• Order of access list entries • Access list entries based on time of day or week • Packets with noninitial fragments	"Refining an IP Access List"
Filtering on IP Options, TCP flags, or noncontiguous	"Creating an IP Access List to Filter IP Options, TCP Flags, or Noncontiguous Ports"
Controlling logging-related parameters	http://www.cisco.com/web/about/security/intelligence/acl-logging.html
Command descriptions related to IP access lists	*Cisco IOS Security Command Reference*

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 XE software 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

Feature Information for Creating IP Access Lists

Table 1 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which 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 XE software release that introduced support for a given feature in a given Cisco IOS XE software release train. Unless noted otherwise, subsequent releases of that Cisco IOS XE software release train also support that feature.

Table 1 Feature Information for Creating IP Access Lists

Feature Name	Releases	Feature Configuration Information
Commented IP Access List Entries	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following sections provide information about this feature: • Helpful Hints for Creating IP Access Lists, page 2 • Access List Remarks, page 3 • Creating a Standard Access List to Filter on Source Address, page 4 • Creating an Extended Access List, page 8 • Applying the Access List to an Interface, page 13 No commands were introduced or modified for this feature.

Table 1 Feature Information for Creating IP Access Lists (continued)

Feature Name	Releases	Feature Configuration Information
Standard IP Access List Logging	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following sections provide information about this feature: • Helpful Hints for Creating IP Access Lists, page 2 • Creating a Standard Access List to Filter on Source Address, page 4 • Creating an Extended Access List, page 8 • Applying the Access List to an Interface, page 13 No commands were introduced or modified for this feature.
ACL Performance Enhancement	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following sections provide information about this feature: • Helpful Hints for Creating IP Access Lists, page 2 • Additional IP Access List Features, page 3 • Creating a Standard Access List to Filter on Source Address, page 4 • Creating an Extended Access List, page 8 No commands were introduced or modified for this feature.

Creating an IP Access List to Filter IP Options, TCP Flags, or Noncontiguous Ports

First Published: August 18, 2006
Last Updated: July 31, 2009

This module describes how to use an IP access list to filter IP packets that contain certain IP options, TCP flags, or noncontiguous ports.

Finding Feature Information

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 Using an IP Access List to Filter Packets" section on page 15.

Use Cisco Feature Navigator to find information about platform support 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

How to Create an IP Access List to Filter IP Options, TCP Flags, or Noncontiguous Ports

This section includes the following 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.

	Command or Action	Purpose
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. Without this feature, when multiple flags are specified on the access control entry (ACE), the packet will be allowed if one of the flags is a match . 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.
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 XE ACLs.
- Before this feature was supported, 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 now be used:

  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 version of Cisco IOS XE 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.

	Command or Action	Purpose
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.
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.
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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.

Configuration Examples for Filtering IP Options, TCP Flags, and Noncontiguous Ports

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
```

Additional References

The following sections provide references related to the IP access list filtering features 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.	*ACL IP Options Selective Drop*
QoS commands	*Cisco IOS Quality of Service Solutions Command Reference*

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE 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 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 Using an IP Access List to Filter Packets

Table 2 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which 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 XE software release that introduced support for a given feature in a given Cisco IOS XE software release train. Unless noted otherwise, subsequent releases of that Cisco IOS XE software release train also support that feature.

Table 2 Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, or Noncontiguous Ports

Feature Name	Releases	Feature Configuration Information
ACL Support for Filtering IP Options	Cisco IOS XE Release 2.1	This feature allows you to filter packets having IP options, in order to prevent routers from becoming saturated with spurious packets. The following sections provide information about this feature: • Filtering Packets That Contain IP Options, page 2 • Filtering Packets That Contain IP Options: Example, page 12 No commands were introduced or modified for this feature.

Table 2 Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, or Noncontiguous Ports

Feature Name	Releases	Feature Configuration Information
ACL TCP Flags Filtering	Cisco IOS XE Release 2.1	This feature provides a flexible mechanism for filtering on TCP flags. It 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. The following sections provide information about this feature: • Filtering Packets That Contain TCP Flags, page 5 • Filtering Packets That Contain TCP Flags: Example, page 13 No commands were introduced or modified for this feature.
ACL—Named ACL Support for Noncontiguous Ports on an Access Control Entry	Cisco IOS XE Release 2.1	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. The following sections provide information about this feature: • 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 13 • Consolidating Some Existing Access List Entries into One Access List Entry with Noncontiguous Ports: Example, page 13 No commands were introduced or modified for this feature.

Refining an IP Access List

First Published: August 18, 2006
Last Updated: July 31, 2009

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

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 Refining an IP Access List" section on page 15.

Use Cisco Feature Navigator to find information about platform support 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

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.
- 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.

	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)# 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 access lists
  - Data confidentiality with IP Security Protocol (IPsec)
- When provider access rates vary by time of day, it is possible to automatically reroute traffic cost effectively.
- 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.

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.

	Command or Action	Purpose
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.
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 Fast Ethernet interface 0/0/0.

```
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 fastethernet 0/0/0
 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	The chapter "Performing Basic System Management" in the *Cisco IOS XE Network Management Configuration Guide*, Release 2
Network management command descriptions	*Cisco IOS Network Management Command Reference*

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 XE software 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

Feature Information for Refining an IP Access List

Table 1 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which 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 XE software release that introduced support for a given feature in a given Cisco IOS XE software release train. Unless noted otherwise, subsequent releases of that Cisco IOS XE software release train also support that feature.

Table 1 Feature Information for Refining an IP Access List

Feature Name	Releases	Feature Configuration Information
Time-Based Access Lists	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following sections provide information about this feature: • Access List Sequence Numbers, page 2 • Revising an Access List Using Sequence Numbers, page 2 • Restricting an Access List Entry to a Time of Day or Week, page 6 • Filtering Noninitial Fragments of Packets, page 7 No commands were introduced or modified for this feature.

IP Access List Entry Sequence Numbering

First Published: January 30, 2003
Last Updated: July 31, 2009

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.

Finding Feature Information

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 IP Access List Entry Sequence Numbering" section on page 12.

Use Cisco Feature Navigator to find information about platform support 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

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 a permit statement and a deny statement 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 packet is dropped. 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.
- 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.
- 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.
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.

Examples

Review the output of the **show ip access-lists** command to see that the access list includes the new entries:

```
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
```

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 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 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 10.4.4.4 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 0.0.0.0, wildcard bits 0.0.0.255
```

Additional References

Related Documents

Related Topic	Document Title
Configuring IP access lists	*Creating an IP Access List and Applying It to an Interface*
IP access list commands	*Cisco IOS Security Command Reference*

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 XE software 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

Feature Information for IP Access List Entry Sequence Numbering

Table 1 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which 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 XE software release that introduced support for a given feature in a given Cisco IOS XE software release train. Unless noted otherwise, subsequent releases of that Cisco IOS XE software release train also support that feature.

Table 1 Feature Information for IP Access List Entry Sequence Numbering

Feature Name	Releases	Feature Information
IP Access List Entry Sequence Numbering	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Aggregation Services Series Routers. The following sections provide information about this feature: • Purpose of IP Access Lists, page 2 • How an IP Access List Works, page 2 • IP Access List Entry Sequence Numbering, page 4 • Sequencing Access-List Entries and Revising the Access List, page 5 The following commands were introduced or modified: **ip access-list resequence**, **deny (IP)**, **permit (IP)**.

ACL IP Options Selective Drop

First Published: July 22, 2002
Last Updated: July 31, 2009

The ACL IP Options Selective Drop feature allows Cisco routers to filter packets containing IP options or to mitigate the effects of IP options on a router or downstream routers by dropping these packets or ignoring the processing of the IP options.

Finding Feature Information

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 ACL IP Options Selective Drop" section on page 7.

Use Cisco Feature Navigator to find information about platform support 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

Restrictions for ACL IP Options Selective Drop

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 modes.

Information About ACL IP Options Selective Drop

Before you configure the ACL IP Options Selective Drop feature, you should understand the concepts in the following sections:

- Using ACL IP Options Selective Drop, page 2
- Benefits of Using ACL IP Options Selective Drop, page 2

Using ACL IP Options Selective Drop

The ACL IP Options Selective Drop feature allows a router to filter IP options packets, thereby mitigating the effects of these packets on a router and downstream routers, and perform the following actions:

- Drop all IP options packets that it receives and prevent options from going deeper into the network.
- Ignore IP options packets destined for the router and treat them as if they had no IP options.

For many users, dropping the packets is the best solution. However, in environments in which some IP options may be legitimate, reducing the load that the packets present on the routers is sufficient. Therefore, users may prefer to skip options processing on the router and forward the packet as though it were pure IP.

Benefits of Using ACL IP Options Selective Drop

- Drop mode filters packets from the network and relieves downstream routers and hosts of the load from options packets.
- Drop mode minimizes loads to the Route Processor (RP) for options that require RP processing on distributed systems. Previously, the packets were always routed to or processed by the RP CPU. Now, the ignore and drop forms prevent the packets from impacting the RP performance.

How to Configure ACL IP Options Selective Drop

This section contains the following configuration information:

- Configuring ACL IP Options Selective Drop, page 3

Configuring ACL IP Options Selective Drop

This section describes how to configure the ACL IP Options Selective Drop feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip options** {**drop** | **ignore**}
4. **exit**
5. **show ip traffic**

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 options** {**drop**	**ignore**} **Example:** `Router(config)# ip options drop`	Drops or ignores IP options packets that are sent to the router.
Step 4	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show ip traffic** **Example:** `Router# show ip traffic`	(Optional) Displays statistics about IP traffic.

Configuration Examples for ACL IP Options Selective Drop

This section provides the following configuration examples:

Configuring ACL IP Options Selective Drop: Example

The following example shows how to configure the router (and downstream routers) to drop all options packets that enter the network:

```
Router(config)# ip options drop

% Warning:RSVP and other protocols that use IP Options packets may not function in drop or
ignore modes.
end
```

Verifying ACL IP Options Selective Drop: Example

The following sample output is displayed after using the **ip options drop** command:

```
Router# show ip traffic

IP statistics:
  Rcvd:  428 total, 323 local destination
         0 format errors, 0 checksum errors, 0 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 cipso, 0 ump
         0 other, 30 ignored
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 fragments, 0 couldn't fragment
  Bcast: 0 received, 0 sent
  Mcast: 323 received, 809 sent
  Sent:  809 generated, 591 forwarded
  Drop:  0 encapsulation failed, 0 unresolved, 0 no adjacency
         0 no route, 0 unicast RPF, 0 forced drop, 0 unsupported-addr
         0 options denied, 0 source IP address zero
```

Additional References

The following sections provide references related to ACL IP Options Selective Drop.

Related Documents

Related Topic	Document Title
IP access list commands	*Cisco IOS Security Command Reference*
Using access lists for filtering IP options	*Creating an IP Access List to Filter IP Options, TCP Flags, or Noncontiguous Ports*

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 XE software 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

Feature Information for ACL IP Options Selective Drop

Table 1 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which 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 XE software release that introduced support for a given feature in a given Cisco IOS XE software release train. Unless noted otherwise, subsequent releases of that Cisco IOS XE software release train also support that feature.

Table 1 Feature Information for ACL IP Options Selective Drop

Feature Name	Releases	Feature Information
ACL IP Options Selective Drop	Cisco IOS XE Release 2.1	The ACL IP Options Selective Drop feature allows Cisco routers to filter packets containing IP options or to mitigate the effects of IP options on a router or downstream routers by dropping these packets or ignoring the processing of the IP options. This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following sections provide information about this feature: • Using ACL IP Options Selective Drop, page 2 • Benefits of Using ACL IP Options Selective Drop, page 2 • Configuring ACL IP Options Selective Drop, page 3 The following command was introduced: **ip options**

Configuring Unicast Reverse Path Forwarding

Last Updated: July 31, 2009

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.

Finding Feature Information

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 Unicast Reverse Path Forwarding" section on page 18.

Use Cisco Feature Navigator to find information about platform support 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 Unicast Reverse Path Forwarding

Unicast RPF requires Cisco Express Forwarding to function properly on the router.

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.

Restrictions for Unicast Reverse Path Forwarding

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 Cisco Express Forwarding.

Information About Unicast Reverse Path Forwarding

Before you configure Unicast RPF, you should understand the following concepts:

- About Unicast Reverse Path Forwarding, page 3
- How Unicast RPF Works, page 3
- Unicast RPF Implementing Principles, page 6

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.

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 is enabled on the router, because the lookup relies on the presence of the Forwarding Information Base (FIB). Cisco Express Forwarding 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 Cisco Express Forwarding 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 Cisco Express Forwarding 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, page 4
- Per-Interface Statistics, page 4

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 1 illustrates how Unicast RPF and Cisco Express Forwarding 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 1 Unicast RPF Validating IP Source Addresses

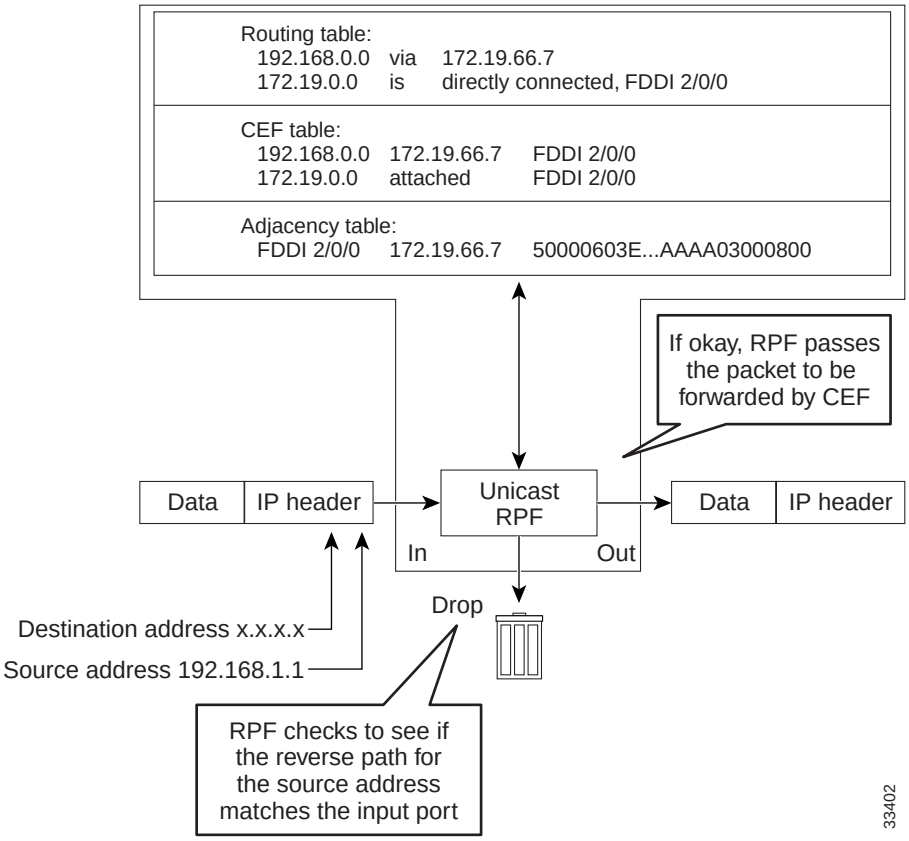
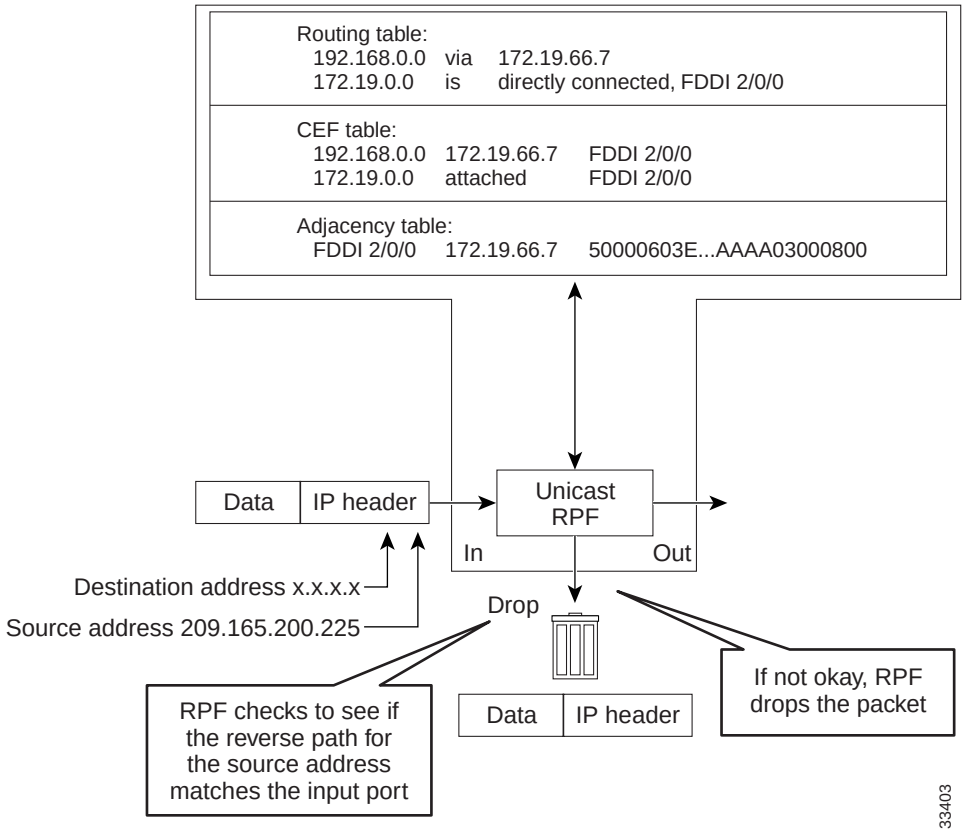


Figure 2 illustrates how Unicast RPF drops packets that fail validation. In this example, a customer has sent a packet having a source address of 209.165.200.225, which is received at interface FDDI 2/0/0. Unicast RPF checks the FIB to see if 209.165.200.225 has a return path to FDDI 2/0/0. If there 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 2 Unicast RPF Dropping Packets That Fail Verification



Unicast RPF Implementing Principles

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.

Ingress and Egress Filtering Policy for Unicast RPF

Unicast RPF can be more effective at mitigating spoofing attacks when combined with a policy of ingress and egress filtering using Cisco IOS XE 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, *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*.

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, page 8
- Network Access Server Application (Applying Unicast RPF in PSTN/ISDN PoP Aggregation Routers), page 9

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.

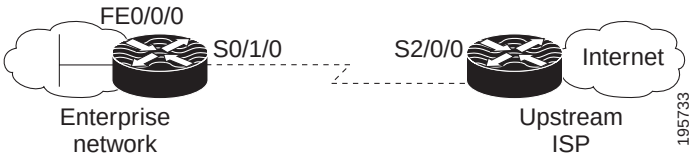
ACLs work well for many single-homed customers; however, there are trade-offs when ACLs are used as ingress filters, including two commonly referenced limitations:

- 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 Cisco Express Forwarding 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 3 illustrates an enterprise network that has a single link to an upstream ISP. In this example, Unicast RPF is applied at interface S0/1/0 on the enterprise router for protection from malformed packets arriving from the Internet. Unicast RPF is also applied at interface S2/0/0 on the ISP router for protection from malformed packets arriving from the enterprise network.

Figure 3 Enterprise Network Using Unicast RPF for Ingress Filtering



Using the topography in Figure 3, a typical configuration (assuming that Cisco Express Forwarding 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 2/0/0
  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 2/0/0
```

The gateway router configuration of the enterprise network (assuming that Cisco Express Forwarding is turned on) would look similar to the following:

```
ip cef
interface FastEthernet 0/0/0
 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 0/1/0
 description 128K HDLC link to ExampleCorp Internet Inc WT50314E  C0
 bandwidth 128
 ip unnumbered FastEthernet 0/0/0
 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 0/1/0
```

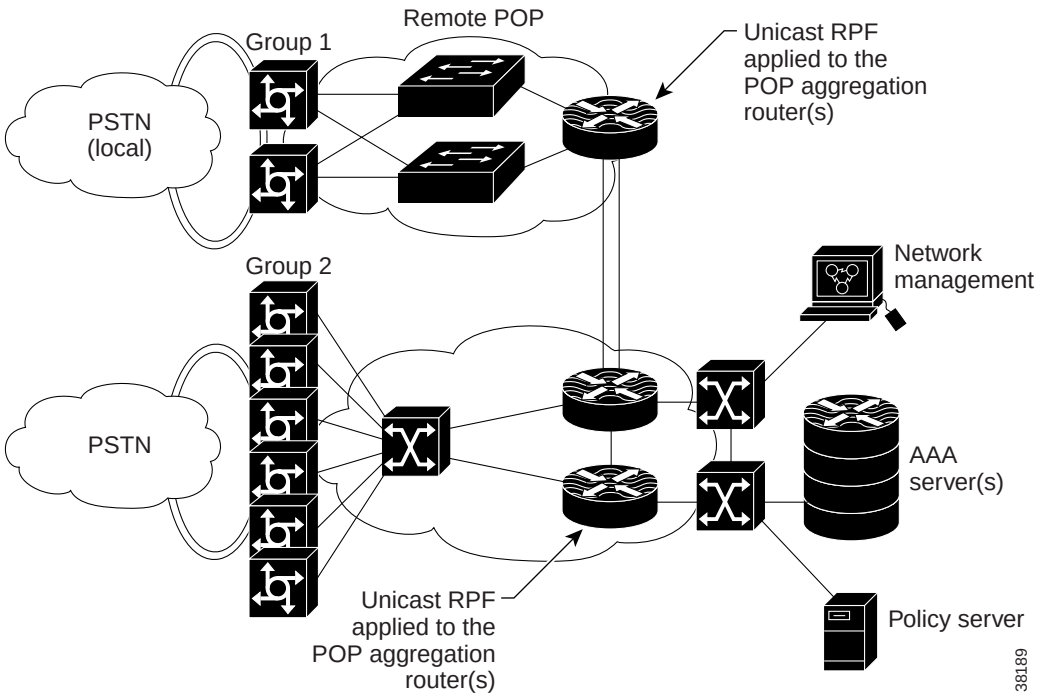
Notice that Unicast RPF works with a single default route. There are no additional routes or routing protocols. Network 192.168.10.0/22 is a connected network. Hence, packets coming from the Internet with a source address in the range 192.168.10.0/22 will be dropped by Unicast RPF.

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 Cisco Express Forwarding, 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 4 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 4 Unicast RPF Applied to PSTN/ISDN Customer Connections



Routing Table Requirements

To work correctly, Unicast RPF needs proper information in the Cisco Express Forwarding tables. This requirement does not mean that the router must have the entire Internet routing table. The amount of routing information needed in the Cisco Express Forwarding 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

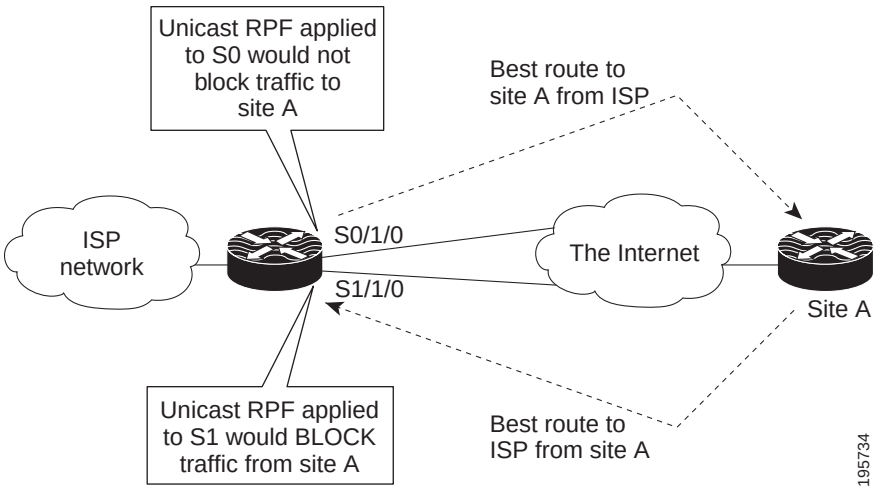
Unicast RPF should not be used on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry (see Figure 5), meaning multiple routes to the source of a packet. Unicast RPF should be applied only where there is natural or configured symmetry. As long as administrators 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 5 illustrates how Unicast RPF can block legitimate traffic in an asymmetrical routing environment.

Figure 5 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.

How to Configure Unicast Unicast Reverse Path Forwarding

The following sections describe how to configure and verify Unicast RPF.

- Configuring Unicast RPF, page 11
- Verifying Unicast RPF, page 13
- Monitoring and Maintaining Unicast RPF, page 14

Configuring Unicast RPF

To configure Unicast RPF, perform the following task.

Prerequisites for Configuring RPF

To use Unicast RPF, you must configure the router for Cisco Express Forwarding switching or Cisco Express Forwarding distributed switching. There is no need to configure the input interface for Cisco Express Forwarding switching because Unicast RPF has been implemented as a search through the FIB using the source IP address. As long as Cisco Express Forwarding 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 Cisco Express Forwarding be turned on globally in the router—Unicast RPF will not work without Cisco Express Forwarding.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]
4. **interface** *slot*/*subslot*/*port*[**.***subinterface-number*]
5. **ip verify unicast reverse-path** *list*
6. **exit**
7. Repeat Steps 4 and 5 for each interface on which you want to apply Unicast RPF.
8. **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	**ip cef** [**distributed**] **Example:** `Router(config)# ip cef distributed`	Enables Cisco Express Forwarding or distributed Cisco Express Forwarding on the router.
Step 4	`Router(config-if)#` **interface** *slot/subslot/port*[*.subinterface-number*] **Example:** `Router(config-if)# interface FastEthernet 0/0/0`	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.

	Command or Action	Purpose
Step 5	**ip verify unicast reverse-path *list*** **Example:** `Router(config-if)# ip verify unicast reverse-path 197`	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 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode. Repeat Steps 2 and 3 for each interface on which you want to apply Unicast RPF.
Step 7	Repeat Steps 4 and 5 for each interface on which you want to apply Unicast RPF.	–
Step 8	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

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# 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

HSRP Failure

Failure to disable Unicast RPF before disabling Cisco Express Forwarding can cause Hot Standby Router Protocol (HSRP) failure. If you want to disable Cisco Express Forwarding on the router, you must first disable Unicast RPF. To disable Unicast RPF, see the section "Monitoring and Maintaining Unicast RPF."

Dropped Boot Requests

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 Cisco Express Forwarding, you must first disable Unicast RPF. Failure to disable Unicast RPF before disabling Cisco Express Forwarding can cause HSRP failure. If you want to disable Cisco Express Forwarding 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 fastethernet0/1/1

   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 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 FastEthernet0/1/1 to check packets arriving at that interface.

For example, packets with a source address of 192.168.201.10 arriving at interface FastEthernet0/1/1 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 FastEthernet0 /1/2 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 fasteth0/1/1
 ip address 192.168.200.1 255.255.255.0
 ip verify unicast reverse-path 197
!
int fasteth0/1/2
 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
```

Additional References

The following sections provide references related to the Unicast Reverse Path Forwardingfeature.

Related Documents

Related Topic	Document Title
Cisco Express Forwarding concepts and tasks	*Cisco IOS XE IP Switching Configuration Guide*, Release 2
Unicast RPF command descriptions	*Cisco IOS Security Command Reference*

RFCs

RFC	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
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 Unicast Reverse Path Forwarding

Table 1 lists the release history for this feature.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which 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 XE software release that introduced support for a given feature in a given Cisco IOS XE software release train. Unless noted otherwise, subsequent releases of that Cisco IOS XE software release train also support that feature.

Table 1 Feature Information for Unicast Reverse Path Forwarding

Feature Name	Releases	Feature Information
Unicast Reverse Path Forwarding	Cisco IOS XE Release 2.1	Unicast RPF helps to mitigate problems that are caused by malformed or forged IP source addresses that are passing through a router. The following sections provide information about this feature: • About Unicast Reverse Path Forwarding, page 3 • How Unicast RPF Works, page 3 • Unicast RPF Implementing Principles, page 6 • Configuring Unicast RPF, page 11 No commands were introduced or modified for this feature.

Flexible Packet Matching

First Published: October 31, 2006
Last Updated: July 31, 2009

Flexible Packet Matching (FPM) is the next generation access control list (ACL) pattern matching tool, providing more thorough and customized packet filters. FPM enables users to match on arbitrary bits of a packet at an arbitrary depth in the packet header and payload. FPM removes constraints to specific fields that had limited packet inspection.

FPM is useful because it enables users to create their own stateless packet classification criteria and to define policies with multiple actions (such as drop, log, or send Internet Control Message Protocol [ICMP] unreachable[1]) to immediately block new viruses, worms, and attacks.

Finding Feature Information

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 Flexible Packet Matching" section on page 11.

Use Cisco Feature Navigator to find information about platform support 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

1. Send ICMP unreachable is currently not supported on the Supervisor Engine 32 PISA.

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

Prerequisites for Flexible Packet Matching

Although access to an XML editor is not required, XML will ease the creation of protocol header description files (PHDFs).

Restrictions for Flexible Packet Matching

- FPM can search for patterns up to 32 bytes in length within the first 256 bytes of the packet.
- A maximum of 32 classes are supported in a policy-map.
- For IP option packets, FPM inspects only the fields in the Layer 2 header and the first 20 bytes of the IP header.
- For noninitial IP fragments, FPM inspects only the fields in the Layer 2 header and the first 20 bytes of the IP header.
- FPM cannot be used to mitigate an attack that requires stateful classification.
- Because FPM is stateless, it cannot keep track of port numbers being used by protocols that dynamically negotiate ports. Thus, when using FPM, port numbers must be explicitly specified.
- FPM cannot perform IP fragmentation or TCP flow reassembly.
- FPM inspects only IPv4 unicast packets.
- FPM cannot classify packets with IP options.
- FPM does not support multicast packet inspection.
- FPM is not supported on tunnel and MPLS interfaces.
- Noninitial fragments will not be matched by the FPM engine.
- Offset can be only a constant in a match start construct.
- FPM cannot match across packets.
- Mapping of FPM policies to control-plane is not supported.

Information About Flexible Packet Matching

Flexible Packet Matching Functional Overview

FPM allows customers to create their own filtering policies that can immediately detect and block new viruses and attacks.

A filtering policy is defined via the following tasks:

- Load a PHDF (for protocol header field matching)
- Define a class map and define the protocol stack chain (traffic class)
- Define a service policy (traffic policy)
- Apply the service policy to an interface

Protocol Header Description File

Protocol headers are defined in separate files called PHDFs; the field names that are defined within the PHDFs are used for defining the packet filters. A PHDF is a file that allows the user to leverage the flexibility of XML to describe almost any protocol header. The important components of the PHDF are the version, the XML file schema location, and the protocol field definitions. The protocol field definitions name the appropriate field in the protocol header, allow for a comment describing the field, provide the location of the protocol header field in the header (the offset is relative to the start of the protocol header), and provide the length of the field. Users can choose to specify the measurement in bytes or in bits.

Note The total length of the header must be specified at the end of each PHDF.

Note When redundant sup PHDF files are used by FPM policy, the files should also be on standby sup's corresponding disk. If the files are not available FPM policy will not work after the switch over.

Users can write their own custom PHDFs via XML for existing or proprietary protocols. However, the following standard PHDFs can also be loaded onto the router via the **load protocol** command: ip.phdf, ether.phdf, tcp.phdf, and udp.phdf.

Note Because PHDFs are defined via XML, they are not shown in a running configuration. However, you can use the **show protocol phdf** command to verify the loaded PHDF.

Standard PHDFs are available on Cisco.com at the following URL:
http://www.cisco.com/cgi-bin/tablebuild.pl/fpm

Filter Description

A filter description is a definition of a traffic class that can contain the header fields defined in a PHDF (using the **match field** command). If a PHDF is not loaded, the traffic class can be defined via the datagram header start (Layer 2) or the network header start (Layer 3) (using the **match start** command). If a PHDF has been loaded onto the router, the class specification begins with a list of the protocol headers in the packet.

A filter definition also includes the policy map; that is, after a class map has been defined, a policy map is needed to bind the match to an action. A policy map is an ordered set of classes and associated actions, such as drop, log, or send ICMP unreachable.

For information on how to configure a class map and a policy map for FPM, see the following section "How to Configure a Flexible Packet Matching Traffic Class and Traffic Policy."

How to Configure a Flexible Packet Matching Traffic Class and Traffic Policy

This section contains the following procedures that should be followed when configuring a FPM traffic class and traffic policy within your network:

- Creating a Traffic Class for Flexible Packet Matching, page 4
- Creating a Traffic Policy for Flexible Packet Matching, page 6

Creating a Traffic Class for Flexible Packet Matching

Perform this task to create an FPM traffic class; that is, create a stateless packet classification criteria that, when used in conjunction with an appropriately defined policy, can mitigate network attacks.

Note If the PHDF protocol fields are referenced in the access-control classmap, the stack classmap is required in order to make FPM work properly

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **load protocol** *location*:*filename*
4. **class-map** [**type** {**stack** | **access-control**}] *class-map-name* [**match-all** | **match-any**]
5. **description** *character-string*
6. **match field** *protocol protocol-field* {**eq** [*mask*] | **neq** [*mask*] | **gt** | **lt** | **range** *range* | **regex** *string*} *value* [**next** *next-protocol*]
7. **match start** {**l2-start** | **l3-start**} **offset** *number* **size** *number* {**eq** | **neq** | **gt** | **lt** | **range** *range* | **regex** *string*} *value* [*value2*]
8. **exit**
9. **show class-map** [**type** {**stack** | **access-control**}] [*class-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	**load protocol** *location*:*filename* **Example:** `Router(config)# load protocol disk2:udp.phdf`	(Optional) Loads a PHDF onto a router. • The specified location must be local to the router. **Note** If a PHDF is not loaded, only the **match start** command can be used; that is, you cannot issue the **match field** command. **Note** PHDF files should be manually copied (via the **load protocol** command) to the active and standby route processor (RP) file systems.
Step 4	**class-map** [**type** {**stack** | **access-control**}] *class-map-name* [**match-all** | **match-any**] **Example:** `Router(config)# class-map type access-control slammer match-all`	Creates a class map to be used for matching packets to a specified class and enters class-map configuration mode. • **type stack** —Enables FPM to determine the correct protocol stack in which to examine. • **type access-control**—Determines the exact pattern to look for in the protocol stack of interest. • *class-map-name*—Can be a maximum of 40 alphanumeric characters. • If **match-all** or **match-any** are not specified, traffic must match all the match criterion to be classified as part of the traffic class.
Step 5	**description** *character-string* **Example:** `Router(config-cmap)# description “match on slammer packets”`	(Optional) Adds a description to the class map.
Step 6	**match field** *protocol protocol-field* {**eq** [*mask*] | **neq** [*mask*] | **gt** | **lt** | **range** *range* | **regex** *string*} *value* [**next** *next-protocol*] **Example:** `Router(config-cmap)# match field udp dest-port eq 0x59A`	(Optional) Configures the match criteria for a class map on the basis of the fields defined in the PHDFs.

	Command or Action	Purpose
Step 7	**match start** {**l2-start** | **l3-start**} **offset** *number* **size** *number* {**eq** | **neq** | **gt** | **lt** | **range** *range* | **regex** *string*} *value* [*value2*] **Example:** `Router(config-cmap)# match start l3-start offset 224 size 4 eq 0x4011010`	(Optional) Configures the match criteria for a class map on the basis of the datagram header (Layer 2) or the network header (Layer 3).

Step 8	**exit**
	
	Example:
	`Router(config-cmap)# exit`
	
	Example:
	`Router(config)# exit`
		Exits class-map configuration mode and global configuration mode.

Step 9	**show class-map** [**type** {**stack** | **access-control**}] [*class-map-name*] **Example:** `Router# show class-map type access-control slammer`	(Optional) Displays all configured FPM class maps.

Troubleshooting Tips

To track all FPM events, issue the **debug fpm event** command.

The following sample output is from the **debug fpm event** command:

```
*Jun 21 09:22:21.607: policy-classification-inline(): matches class: class-default *Jun 21
09:22:21.607: packet-access-control(): policy-map: fpm-policy, dir: input, match. retval:
0x0, ip-flags: 0x80000000
```

What to Do Next

After you have defined at least one class map for your network, you must create a traffic policy and apply that policy to an interface as shown in the following task "Creating a Traffic Policy for Flexible Packet Matching."

Creating a Traffic Policy for Flexible Packet Matching

Perform this task to create an FPM traffic policy and apply the policy to a given interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** [**type access-control**] *policy-map-name*
4. **description** *character-string*
5. **class** *class-name*

6. **drop**
7. **service-policy** *policy-map-name*
8. **exit**
9. **interface** *type name*
10. **service-policy** [**type access-control**] {**input** | **output**} *policy-map-name*
11. **exit**
12. **show policy-map interface** [**type access-control**] *interface-name* [**input** | **output**]

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	**policy-map** [**type access-control**] *policy-map-name* **Example:** `Router(config)# policy-map type access-control fpm-udp-policy`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy and enters policy-map configuration mode.
Step 4	**description** *character-string* **Example:** `Router(config-pmap)# description “policy for UDP based attacks”`	(Optional) Adds a description to the policy map.
Step 5	**class** *class-name* **Example:** `Router(config-pmap)# class slammer`	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy.
Step 6	**drop** **Example:** `Router(config-pmap)# drop`	(Optional) Configures a traffic class to discard packets belonging to a specific class. If this command is issued, note the following restrictions: • Discarding packets is the only action that can be configured in a traffic class. • When a traffic class is configured with the **drop** command, a “child” (nested) policy cannot be configured for this specific traffic class through the **service policy** command. • Discarding packets cannot be configured for the default class specified via the c**lass class-default** command.

	Command or Action	Purpose
Step 7	**service-policy** *policy-map-name* **Example:** `Router(config-pmap-c)# service policy fpm-udp-policy`	Creates hierarchical service policies.

Step 8	**exit**
	
	Example:
	`Router(config-pmap-c)# exit`
	
	Example:
	`Router(config-pmap)# exit`
		Exits policy-map class configuration mode and policy-map configuration mode.

Step 9	**interface** *type number* **Example:** `Router(config)# interface gigabitEthernet 0/1/0`	Configures an interface type and enters interface configuration mode.
Step 10	**service-policy** [**type access-control**] {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy type access-control input fpm-policy`	Specifies the type and the name of the traffic policy to be attached to the input or output direction of an interface.

Step 11	**exit**
	
	Example:
	`Router(config-if)# exit`
	
	Example:
	`Router(config)# exit`
		Exits interface configuration and global configuration modes.

Step 12	**show policy-map interface** [**type access-control**] *interface-name* [**input** | **output**] **Example:** `Router# show policy-map interface type access-control interface gigabit 0/1`	(Optional) Verifies the FPM configuration.

Configuration Examples for an FPM Configuration

This section contains the following configuration example:

- Configuring and Verifying FPM on ASR Platform: Example, page 9

Configuring and Verifying FPM on ASR Platform: Example

The following example shows how to configure FPM on the ASR platform.

```
load protocol bootflash:ip.phdf
load protocol bootflash:tcp.phdf
class-map type stack match-all ip_tcp
 match field IP protocol eq 6 next TCP

class-map type access-control match-all test_class
 match field TCP dest-port gt 10
 match start l3-start offset 40 size 32 regex "ABCD"

policy-map type access-control child
 class test_class
  drop

policy-map type access-control parent
 class ip_tcp
  service-policy child

interface GigabitEthernet0/3/0
 ip address 10.1.1.1 255.0.0.0
 service-policy type access-control input parent
```

In the following sample output, all TCP packets are seen under the class-map “ip_tcp” and all packets matching the specific pattern are seen under the class-map “test_class.” TCP packets without the specific pattern are seen under the child policy “class-default,” while all non-TCP packets are seen under the parent policy “class-default.” (The counter is 0 in this example.)

```
Router# show policy-map type access-control interface GigabitEthernet0/3/0

GigabitEthernet0/3/0
 Service-policy access-control input: parent

  Class-map: ip_tcp (match-all)
  2024995578 packets, 170099628552 bytes
  5 minute offered rate 775915000 bps
  Match: field IP version eq 4
  Match: field IP ihl eq 5
  Match: field IP protocol eq 6 next TCP

 Service-policy access-control : child

 Class-map: test_class (match-all)
  1598134279 packets, 134243279436 bytes
  5 minute offered rate 771012000 bps, drop rate 771012000 bps
  Match: field TCP dest-port gt 10
  Match: start l3-start offset 40 size 32 regex "ABCD"
 drop

 Class-map: class-default (match-any)
  426861294 packets, 35856348696 bytes
  5 minute offered rate 4846000 bps, drop rate 0 bps
  Match: any

 Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
Router#
```

Additional References

The following sections provide references related to Flexible Packet Matching.

Related Documents

Related Topic	Document Title
Complete suite of QoS commands	*Cisco IOS Quality of Service Solutions Command Reference*
Information about commands listed in this document	Cisco IOS Master Command List, All Releases

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE software 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

Feature Information for Flexible Packet Matching

Table 1 lists the release history for this feature.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which 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 XE software release that introduced support for a given feature in a given Cisco IOS XE software release train. Unless noted otherwise, subsequent releases of that Cisco IOS XE software release train also support that feature.

Table 1 Feature Information for Flexible Packet Matching

Feature Name	Releases	Feature Information
Flexible Packet Matching	Cisco IOS XE Release 2.2	FPM is a packet classification feature that allows users to define one or more classes of network traffic by pairing a rich set of standard matching operators with user-defined protocol header fields. The following commands were introduced or modified: **class (policy-map)**, **class-map**, **debug fpm event,** **description (class-map)**, **load protocol**, **match field**, **match start**, **policy-map**, **service-policy**, **show class-map**, **show policy-map interface**, **show protocol phdf**.

Zone-Based Policy Firewall

Zone-Based Policy Firewall

First Published: February 22, 2006
Last Updated: November 25, 2009

This module describes the Cisco IOS XE unidirectional firewall policy between groups of interfaces known as zones.

Finding Feature Information

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 Zone-Based Policy Firewall" section on page 28.

Use Cisco Feature Navigator to find information about platform support 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 Zone-Based Policy Firewall

Before you create zones, think about what should constitute the zones. The general guideline is that you should group together interfaces that are similar when they are viewed from a security perspective.

Restrictions for Zone-Based Policy Firewall

- Application-level maps (also referred to as Layer 7 class maps) are currently not supported in Cisco IOS XE software.
- In a WAAS and Cisco IOS XE firewall configuration, all packets processed by a Wide Area Application Engine (WAE) device must go over the Cisco IOS XE firewall in both directions to support the Web Cache Coordination Protocol (WCCP) generic routing encapsulation (GRE) redirect. This situation occurs when the Layer 2 redirect is not available. If Layer 2 redirect is configured on the WAE, the system defaults to the GRE redirect to continue to function.
- In a WAAS and Cisco IOS XE firewall configuration, WCCP does not support traffic redirection using policy-based routing (PBR).

Information About Zone-Based Policy Firewall

To configure a zone-based policy firewall, you should understand the following concepts:

Top-level Class Maps and Policy Maps

Top-level class maps allow you to identify the traffic stream at a high level. This is accomplished by using **match access-group** and **match protocol** commands. Top-level class maps are also referred to as Layer 3 and Layer 4 class-maps.

Top-level policy maps allow you to define high-level actions such as **inspect**, **drop**, and **pass**. You can attach the policy maps to a target (zone-pair).

Note Only inspect type policies can be configured on a zone-pair.

Zones

A zone is a group of interfaces that have similar functions or features. They provide a way for you to specify *where* a Cisco IOS XE software firewall is applied.

For example, on a router, interfaces Ethernet 0/0 and Ethernet 0/1 may be connected to the local LAN. These two interfaces are similar because they represent the internal network, so they can be grouped into a zone for firewall configurations.

By default, the traffic between interfaces in the same zone is not be subjected to any policy. The traffic passes freely.

Firewall zones are used for security features.

Note Zones may not span interfaces in different virtual routing and forwarding (VRF) instances.

Security Zones

A security zone is a group of interfaces to which a policy can be applied.

Grouping interfaces into zones involves two procedures:

- Creating a zone so that interfaces can be attached to it
- Configuring an interface to be a member of a given zone

By default, traffic flows among interfaces that are members of the same zone.

When an interface is a member of a security zone, all traffic to and from that interface (except traffic going to the router or initiated by the router) is dropped. To permit traffic to and from a zone-member interface, you must make that zone part of a zone-pair and then apply a policy to that zone-pair. If the policy permits traffic (via **inspect** or **pass** actions), traffic can flow through the interface.

Basic rules to consider when setting up your zones are as follows:

- Traffic from a zone interface to a non-zone interface or from a non-zone interface to a zone interface is always dropped.
- Traffic between two zone interfaces is inspected if there is a zone-pair relationship for each zone and if there is a configured policy for that zone-pair.
- By default, all traffic between two interfaces in the same zone is always allowed as if the “pass” action is configured.
- A zone-pair can be configured with a zone as both the source and the destination zones. An inspect policy can be configured on this zone-pair to inspect or drop the traffic between two interfaces in the same zone.

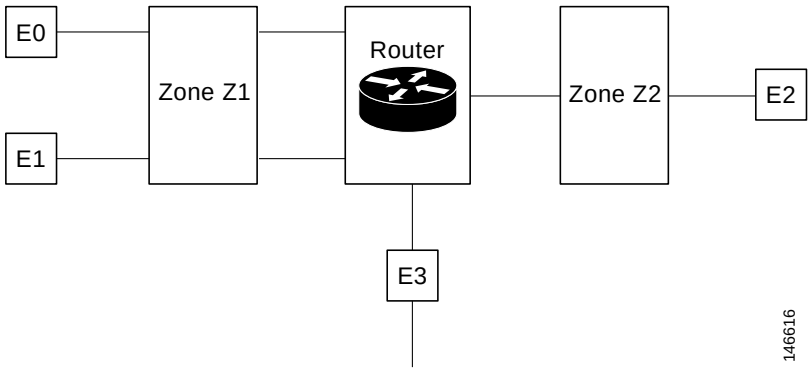
For traffic to flow among all the interfaces in a router, all the interfaces must be a member of one security zone or another.

It is not necessary for all router interfaces to be members of security zones.

Figure 1 illustrates the following:

- Interfaces E0 and E1 are members of security zone Z1.
- Interface E2 is a member of security zone Z2.
- Interface E3 is not a member of any security zone.

Figure 1 Security Zone Restrictions



The following situations exist:

- In no same-zone zone-pair and policy are configured, traffic flows freely between interfaces E0 and E1 because they are members of the same security zone (Z1).
- If no policies are configured, traffic will not flow between any other interfaces (for example, E0 and E2, E1 and E2, E3 and E1, and E3 and E2).
- Traffic can flow between E0 or E1 and E2 only when an explicit policy permitting traffic is configured between zone Z1 and zone Z2.
- Traffic can never flow between E3 and E0/E1/E2 because E3 is not part of any security zone.

Virtual Interfaces As Members of Security Zones

A virtual template interface is a logical interface configured with generic configuration information for a specific purpose or for configuration common to specific users, plus router-dependent information. The template contains Cisco IOS XE software interface commands that are applied to virtual access interfaces, as needed. To configure a virtual template interface, use the **interface virtual-template** command.

Virtual interfaces can be members of a security zone. The virtual template interface is a member of a zone and all virtual access interfaces created from the template are members of that zone.

Zone member information is acquired from a RADIUS server and then the dynamically created interface is made a member of that zone.

The **zone-member security** command puts the dynamic interface into the corresponding zone.

Zone-Pairs

A zone-pair allows you to specify a unidirectional firewall policy between two security zones.

To define a zone-pair, use the **zone-pair security** command. The direction of the traffic is specified by specifying a source and destination zone. The source and destination zones of a zone-pair must be security zones.

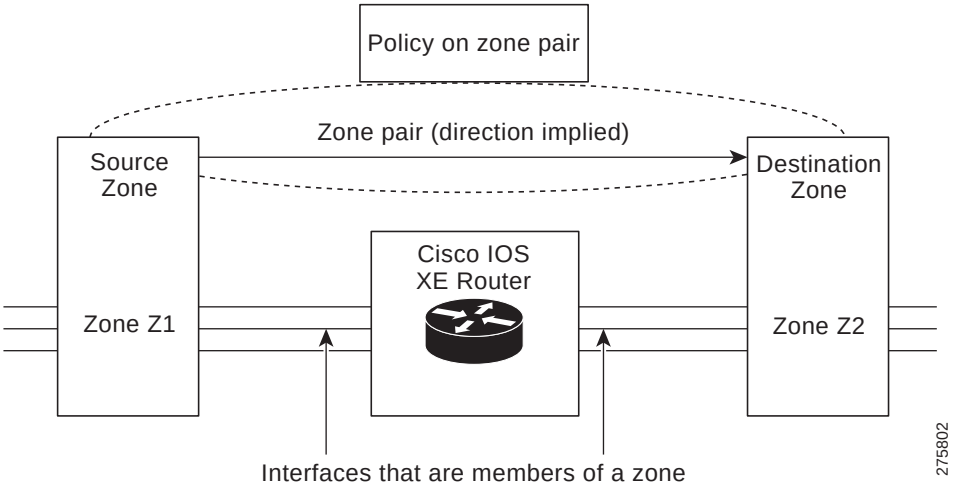
If desired, you can select the default self zone as either the source or the destination zone. The self zone is a system-defined zone. It does not have any interfaces as members. A zone-pair that includes the self zone, along with the associated policy, applies to traffic directed to the router or traffic generated by the router. It does not apply to traffic through the router.

The most common usage of firewalls is to apply them to traffic through a router, so you usually need at least two zones (that is, you cannot use the self zone).

To permit traffic between zone-member interfaces, you must configure a policy permitting (or inspecting) traffic between that zone and another zone. To attach a firewall policy map to the target zone-pair, use the **service-policy type inspect** command.

Figure 2 shows the application of a firewall policy to traffic flowing from zone z1 to zone z2, which means that the ingress interface for the traffic is a member of zone z1 and the egress interface is a member of zone z2.

Figure 2 Zone Pairs



If there are two zones and you require policies for traffic going in both directions (from z1 to z2 and z2 to z1), you must configure two zone-pairs (one for each direction).

If a policy is not configured between a pair of zones, traffic is dropped. However, it is not necessary to configure a zone-pair and a service policy solely for return traffic. Return traffic is allowed, by default, if a service policy permits the traffic in the forward direction. In the above example, it is not mandatory that you configure a zone-pair source Z2 destination Z1 solely for allowing return traffic from Z2 to Z1. The service policy on the Z1-Z2 zone-pair takes care of it.

Zones and Inspection

Zone-based policy firewalls examine the source and destination zones from the ingress and egress interfaces for a firewall policy. It is not necessary that all traffic flowing to or from an interface be inspected; you can designate that individual flows in a zone-pair be inspected through your policy map that you apply across the zone-pair. The policy map will contain class-maps that specify the individual flows.

You can also configure **inspect** parameters like TCP thresholds and timeouts on a per-flow basis.

Zones and ACLs

- Pinholes are not punched for return traffic in interface ACLs.
- ACLs applied to interfaces that are members of zones are processed before the policy is applied on the zone-pair. So, you must relax interface ACLs when there are policies between zones so that they cannot interfere with the policy firewall traffic.

Overview of Security Zone Firewall Policies

A class is a way of identifying a set of packets based on its contents. Normally you define a class so that you can apply an action on the identified traffic that reflects a policy. A class is designated via class maps.

An action is a specific functionality. It typically is associated with a traffic class. For example, **inspect**, **drop**, and **pass** are actions.

To create firewall policies, you should complete the following tasks:

- Define a match criteria (**class map**)
- Associate actions to the match criteria (**policy map**)
- Attach the policy map to a zone pair (**service policy**)

The **class-map** command creates a class map to be used for matching packets to a specified class. Packets arriving at the targets (such as the input interface, output interface, or zone-pair), determined by how the **service-policy** command is configured, are checked against the match criteria configured for a class map to determine if the packet belongs to that class.

The **policy-map** command creates or modifies a policy map that can be attached to one or more targets to specify a service policy. Use the **policy-map** command to specify the name of the policy map to be created, added to, or modified before you can configure policies for classes whose match criteria are defined in a class map.

Class Maps and Policy Maps for Zone-Based Policy Firewalls

Quality of Service (QoS) class maps have numerous match criteria; firewalls have fewer match criteria. Firewall class maps have type **inspect**; this information controls what shows up under firewall class maps.

A policy is an association of traffic classes and actions. It specifies what actions should be performed on the defined traffic classes. An action is a specific function, and it is typically associated with a traffic class. For example, **inspect** and **drop** are actions.

Layer 3 and Layer 4 Class Maps and Policy Maps

Layer 3 and Layer 4 class maps are used to identify traffic streams on which different actions should be performed.

A Layer 3 or Layer 4 policy map is sufficient for the basic inspection of traffic.

The following example shows how to configure class map c1 with match criteras of ACL 101 and the FTP protocol, and create an inspect policy map named p1 to specify that packets will be dropped on the traffic at c1:

```
Router(config)# class-map type inspect match-all c1
Router(config-cmap)# match access-group 101
Router(config-cmap)# match protocol ftp

Router(config)# policy-map type inspect p1
Router(config-pmap)# class type inspect c1
Router(config-pmap-c)# drop
```

Supported Protocols

The following protocols are currently supported:

- Domain Name Server (DNS)
- File Transfer Protocol (FTP)
- H.323
- ICMP
- Lightweight Directory Access Protocol (LDAP)
- LDAP over TLS/SSL (LDAPS)
- Real-time Streaming Protocol (RTSP)
- Session Initiation Protocol (SIP)
- Skinny Client Control Protocol (SCCP)
- TCP
- Trivial File Transfer Protocol (TFTP)
- UDP

Class-Map Configuration Restriction

If traffic meets multiple match criteria, the match criteria must be applied in the order of specific to less specific. For example, consider the following class map example:

```
class-map type inspect match-any my-test-cmap
 match protocol ftp
 match protocol tcp
```

In this example, FTP traffic must first encounter the **match protocol ftp** command to ensure that the traffic will be handled by the service-specific capabilities of FTP inspection. If the “match” lines were reversed so traffic encountered the **match protocol tcp** command before it was compared to the **match protocol ftp** command, the traffic would simply be classified as TCP traffic and inspected according to the capabilities of the Firewall’s TCP Inspection component.

Class-Default Class Map

In addition to user-defined classes, there is a system-defined class map named class-default that represents all packets that do not match any of the user-defined classes in a policy. It always is the last class in a policy map.

You can define explicit actions for this group of packets. If you do not configure any actions for class-default in an inspect policy, the default action is **drop**.

WAAS Support for the Cisco IOS XE Firewall

The WAAS firewall software provides an integrated firewall that optimizes security-compliant WANs and application acceleration solutions with the following benefits:

- Optimizes a WAN through full stateful inspection capabilities
- Simplifies Payment Card Industry (PCI) compliance
- Protects transparent WAN accelerated traffic

- Integrates WAAS networks transparently
- Supports the Network Management Equipment (NME) WAE modules or standalone WAAS device deployment

WAAS has an automatic discovery mechanism that uses TCP options during the initial three-way handshake used to identify WAE devices transparently. After automatic discovery, optimized traffic flows (paths) experience a change in the TCP sequence number to allow endpoints to distinguish between optimized and nonoptimized traffic flows.

Note Paths are synonymous with connections.

WAAS allows the Cisco IOS XE firewall to automatically discover optimized traffic by enabling the sequence number to change without compromising the stateful Layer 4 inspection of TCP traffic flows that contain internal firewall TCP state variables. These variables are adjusted for the presence of WAE devices.

If the Cisco IOS XE firewall notices that a traffic flow has successfully completed WAAS automatic discovery, it permits the initial sequence number shift for the traffic flow and maintains the Layer 4 state on the optimized traffic flow.

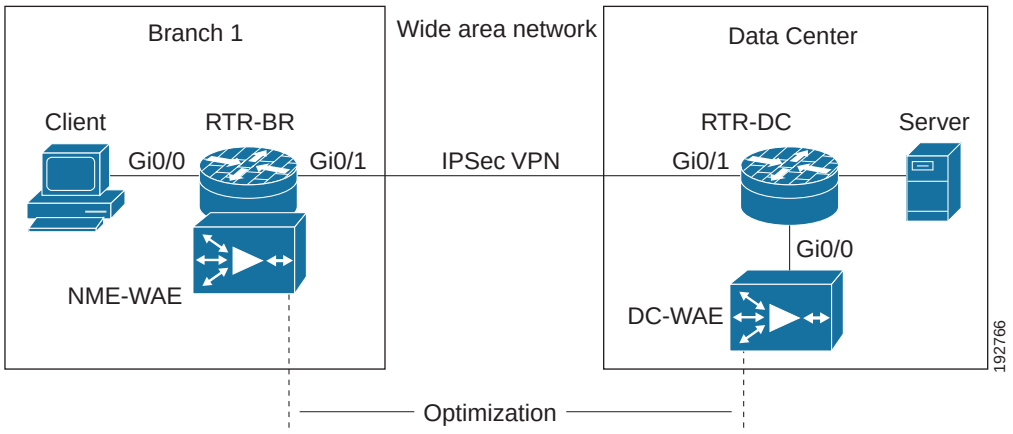
WAAS Traffic Flow Optimization Deployment Scenarios

The following sections describe three different WAAS traffic flow optimization scenarios for branch office deployments. WAAS traffic flow optimization works with the Cisco IOS XE firewall feature on a Cisco Integrated Services Router (ISR).

- WAAS Branch Deployment with an Off-Path device, page 9
- WAAS Branch Deployment with an Inline Device, page 10

Figure 3 shows an example of an end-to-end WAAS traffic flow optimization with the Cisco IOS XE firewall. In this particular deployment, a NME-WAE device is on the same router as the Cisco IOS XE firewall. WCCP is used to redirect traffic for interception.

Figure 3 End-to-End WAAS Optimization Path

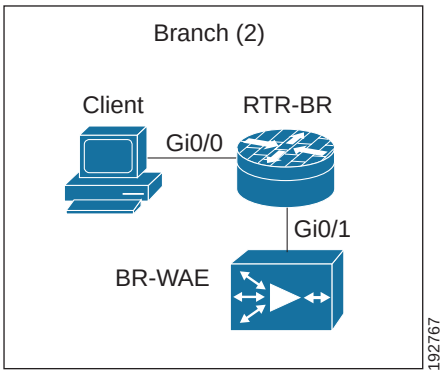


WAAS Branch Deployment with an Off-Path device

A WAE device can be either an NME-WAE that is installed on an ISR as an integrated service engine (as shown in Figure 3) or a standalone WAE device.

Figure 4 shows a WAAS branch deployment that uses WCCP to redirect traffic to an off-path, standalone WAE device for traffic interception. The configuration for this option is the same as the WAAS branch deployment with an NME-WAE.

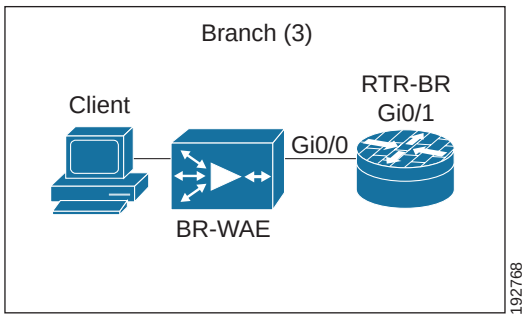
Figure 4 WAAS Off-Path Branch Deployment



WAAS Branch Deployment with an Inline Device

Figure 5 shows a WAAS branch deployment that has an inline WAE device that is physically in front of the ISR router. Since the WAE device is in front of the router, Layer 7 inspection on the client side is not supported because the Cisco IOS XE firewall receives WAAS optimized packets.

Figure 5 WAAS Inline Path Branch Deployment



An edge WAAS device with the Cisco IOS XE firewall is applied at branch office sites that must inspect traffic moving to and from a WAN connection. The Cisco IOS XE firewall monitors traffic for optimization indicators (TCP options and subsequent TCP sequence number changes) and allows optimized traffic to pass, while still applying Layer 4 stateful inspection and deep packet inspection to all traffic, maintaining security while accommodating WAAS optimization advantages.

Note If the WAE device is in the inline location, the device enters its bypass mode after the automatic discovery process. Although the router is not directly involved in WAAS optimization, the router must be aware that WAAS optimization is applied to the traffic in order to apply the Cisco IOS XE firewall inspection to network traffic and make allowances for optimization activity if optimization indicators are present.

How to Configure Zone-Based Policy Firewall

This section contains the following configuration tasks:

- Configuring Layer 3 and Layer 4 Firewall Policies, page 10
- Configuring an Inspect Parameter Map, page 13
- Configuring NetFlow Event Logging, page 16
- Creating Security Zones, Zone-Pairs, and Attaching a Policy Map to a Zone-Pair, page 17
- Configuring the Firewall with WAAS and WCCP, page 20

Configuring Layer 3 and Layer 4 Firewall Policies

Layer 3 and Layer 4 policies are “top level” policies that are attached to the target (zone-pair). Use the following tasks to configure Layer 3 and Layer 4 firewall policies:

- Configuring a Class Map for a Layer 3 and Layer 4 Firewall Policy, page 11
- Creating a Policy Map for a Layer 3 and Layer 4 Firewall Policy, page 12

Configuring a Class Map for a Layer 3 and Layer 4 Firewall Policy

Use this task to configure a class map for classifying network traffic.

Note You must perform at least one step from Step 4, 5, or 6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** [**match any** | **match all**] *class-map-name*
4. **match access-group** {*access-group* | **name** *access-group-name*}
5. **match protocol** *protocol_name*
6. **match class-map** *class-map-name*
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	**class-map type inspect** [**match-any** | **match-all**] *class-map-name* **Example:** `Router(config)# class-map type inspect match-all c1`	Creates a Layer 3 or Layer 4 inspect type class map. Enters class-map configuration mode.
Step 4	**match access-group** {*access-group* | **name** *access-group-name*} **Example:** `Router(config-cmap)# match access-group 101`	Configures the match criteria for a class map based on the ACL name or number.
Step 5	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol ftp`	Configures the match criteria for a class map on the basis of a specified protocol.

	Command or Action	Purpose
Step 6	**match class-map** *class-map-name* **Example:** `Router(config-cmap)# match class-map c1`	Specifies a previously defined class as the match criteria for a class map.
Step 7	**exit** **Example:** `Router(config-cmap)# exit`	Returns to global configuration mode.

Creating a Policy Map for a Layer 3 and Layer 4 Firewall Policy

Use this task to create a policy map for a Layer 3 and Layer 4 firewall policy that will be attached to zone-pairs.

If you are creating an inspect type policy map, note that only the following actions are allowed: drop, inspect, and pass.

Note You must perform at least one step from Step 5, 6, or 7.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type inspect** *policy-map-name*
4. **class type inspect** *class-name*
5. **inspect** [*parameter-map-name*]
6. **drop** [**log**]
7. **pass**
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.

	Command or Action	Purpose
Step 3	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect p1`	Creates a Layer 3 and Layer 4 inspect type policy map. Enters policy-map configuration mode.
Step 4	**class type inspect** *class-name* **Example:** `Router(config-pmap)# class type inspect c1`	Specifies the traffic (class) on which an action is to be performed.
Step 5	**inspect** [*parameter-map-name*] **Example:** `Router(config-pmap-c)# inspect inspect-params`	Enables Cisco IOS XE stateful packet inspection.
Step 6	**drop** [**log**] **Example:** `Router(config-pmap-c)# drop`	(Optional) Drops packets that are matched with the defined class. **Note** The actions **drop** and **pass** are exclusive, and the actions **inspect** and **drop** are exclusive; that is, you cannot specify both of them.
Step 7	**pass** **Example:** `Router(config-pmap-c)# pass`	(Optional) Allows packets that are matched with the defined class.
Step 8	**exit** **Example:** `Router(config-pmap-c)# exit`	Returns to policy-map configuration mode.

Configuring an Inspect Parameter Map

An inspect parameter map is optional if you are using an inspect type policy. If you do not configure a parameter map, the software uses default parameters. Parameters associated with the inspect action apply to all nested actions (if any). If parameters are specified in both the top and lower levels, those in the lower levels override those in the top levels

Note Changes to the parameter map are not reflected on connections already established through the firewall. Changes are applicable only to new connections permitted to the firewall. To ensure that your firewall enforces policies strictly, clear all the connections allowed in the firewall after you change the parameter map. To clear existing connections, use the **clear zone-pair inspect sessions** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect** *parameter-map-name*
4. **alert** {**on** | **off**}

5. **audit-trail** {**on** | **off**}
6. **dns-timeout** *seconds*
7. **icmp idle-timeout** *seconds*
8. **max-incomplete** {**low** *number-of-connections* | **high** *number-of-connections*}
9. **one-minute** {**low** *number-of-connections* | **high** *number-of-connections*}
10. **sessions maximum** *sessions*
11. **tcp finwait-time** *seconds*
12. **tcp idle-time** *seconds*
13. **tcp max-incomplete host** *threshold* [**block-time** *minutes*]
14. **tcp synwait-time** *seconds*
15. **tcp window-scale-enforcement loose**
16. **udp idle-time** *seconds*
17. **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	**parameter-map type inspect** *parameter-map-name* **Example:** `Router(config)# parameter-map type inspect eng-network-profile`	Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action. Enters parameter-map type inspect configuration mode.
Step 4	**alert** {**on** | **off**} **Example:** `Router(config-profile)# alert on`	(Optional) Turns on and off stateful packet inspection alert messages that are displayed on the console.
Step 5	**audit-trail** {**on** | **off**} **Example:** `Router(config-profile)# audit-trail on`	(Optional) Turns audit trail messages on or off.
Step 6	**dns-timeout** *seconds* **Example:** `Router(config-profile)# dns-timeout 60`	(Optional) Specifies the domain name system (DNS) idle timeout (the length of time for which a DNS lookup session will continue to be managed while there is no activity).

	Command or Action	Purpose
Step 7	**icmp idle-timeout** *seconds* **Example:** `Router(config-profile)# icmp idle-timeout 90`	(Optional) Configures the timeout for Internet Control Message Protocol (ICMP) sessions.

Step 8	**max-incomplete** {**low** *number-of-connections* | **high** *number-of-connections*}
	
	Example:
	`Router(config-profile)# max-incomplete low 800`
	`Router(config-profile)# max-incomplete high 10000`
		(Optional) Defines the number of existing half-open sessions that will cause the Cisco IOS firewall to start and stop deleting half-open sessions.

Step 9	**one-minute** {**low** *number-of-connections* | **high** *number-of-connections*}
	
	Example:
	`Router(config-profile)# one-minute low 300`
	`Router(config-profile)# one-minute high 400`
		(Optional) Defines the number of new unestablished sessions that will cause the system to start deleting half-open sessions and stop deleting half-open sessions.

Step 10	**sessions maximum** *sessions* **Example:** `Router(config-profile)# sessions maximum 200`	(Optional) Sets the maximum number of allowed sessions that can exist on a zone-pair. You may want to use this command to limit the bandwidth used by the sessions. • *sessions*—Maximum number of allowed sessions. Range: 1 to 2147483647.
Step 11	**tcp finwait-time** *seconds* **Example:** `Router(config-profile)# tcp finwait-time 5`	(Optional) Specifies how long a TCP session will be managed after the firewall detects a FIN-exchange.
Step 12	**tcp idle-time** *seconds* **Example:** `Router(config-profile)# tcp idle-time 90`	(Optional) Configures the timeout for TCP sessions.
Step 13	**tcp max-incomplete host** *threshold* [**block-time** *minutes*] **Example:** `Router(config-profile)# tcp max-incomplete host 500 block-time 10`	(Optional) Specifies threshold and blocking time values for TCP host-specific DoS detection and prevention.
Step 14	**tcp synwait-time** *seconds* **Example:** `Router(config-profile)# tcp synwait-time 3`	(Optional) Specifies how long the software will wait for a TCP session to reach the established state before dropping the session.
Step 15	**tcp window-scale-enforcement loose** **Example:** `Router(config-profile)# tcp window-scale-enforcement loose`	(Optional) Disables the window scale option check in the parameter map for a TCP packet that has an invalid window scale option under the Zone Based Firewall (ZBF)

	Command or Action	Purpose
Step 16	**udp idle-time** *seconds* **Example:** `Router(config-profile)# udp idle-time 75`	(Optional) Configures the idle timeout of User Datagram Protocol (UDP) sessions going through the firewall.
Step 17	**exit** **Example:** `Router(config-profile)# exit`	Returns to global configuration mode.

Configuring NetFlow Event Logging

Global parameter maps are used for Netflow event logging. With Netflow event logging enabled, logs are sent to an off-box high-speed log collector. By default, this functionality is not enabled. (If this functionality is not enabled, firewall logs are sent to a logger buffer located in the route processor or console.)

For more information on syslog logging, see the chapter "Troubleshooting and Fault Management."

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type inspect global**
4. **log dropped-packets**
5. **log flow-export v9 udp destination** *ipv4-address port*
6. **log flow-export template timeout-rate** *seconds*
7. **exit**
8. **exit**
9. **show parameter-map type inspect global**

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	**parameter-map type inspect global** **Example:** `Router(config)# parameter-map type inspect global`	Configures a global parameter map. You must enter this command to access the CLI that enables packet logging.
Step 4	**log dropped-packets** **Example:** `Router(config-profile)#`	Enables dropped packet logging.
Step 5	**log flow-export v9 udp destination** *ipv4-address port* **Example:** `Router(config-profile)# log flow-export v9 udp destination 1.1.1.1 5000`	Enables Netflow event logging and provides the collector's IP address and port.
Step 6	**log flow-export template timeout-rate** *seconds* **Example:** `Router(config-profile)# log flow-export template timeout-rate 5000`	Specifies the template timeout value.
Step 7	**exit**	Exits parameter-map configuration mode.
Step 8	**exit**	Exits global configuration mode.
Step 9	**show parameter-map type inspect global** **Example:** `Router# show parameter-map type inspect global`	Displays logging configurations.

Creating Security Zones, Zone-Pairs, and Attaching a Policy Map to a Zone-Pair

You need two security zones to create a zone-pair. However, you can create only one security zone and use a system-defined security zone called "self." Note that if you select a self zone, you cannot configure inspect policing.

Use this process to complete the following tasks:

- Create at least one security zone
- Define zone-pairs
- Assign interfaces to security zones
- Attach a policy map to a zone-pair.

Tip Before you create zones, think about what should constitute the zones. The general guideline is that you should group together interfaces that are similar when they are viewed from a security perspective.

Security Zone Restrictions

- An interface can be a member of only one security zone.
- When an interface is a member of a security zone, all traffic to and from that interface is blocked unless you configure an explicit interzone policy on a zone-pair involving that zone.
- Traffic cannot flow between an interface that is a member of a security zone and an interface that is not a member of a security zone because a policy can be applied only between two zones.
- For traffic to flow among all the interfaces in a router, all the interfaces must be members of one security zone or another. This is particularly important because after you make an interface a member of a security zone, a policy action (such as **inspect** or **pass**) must explicitly allow packets. Otherwise, packets are dropped.
- If an interface on a router cannot be part of a security zone or firewall policy, you may have to put that interface in a security zone and configure a “pass all” policy (that is, a “dummy” policy) between that zone and other zones to which a traffic flow is desired.
- You cannot apply an access control list (ACL) between security zones or on a zone-pair.
- An ACL cannot be applied between security zones and zone-pairs. Include the ACL configuration in a class map, and use policy maps to drop traffic.
- An ACL on an interface that is a zone member should not be restrictive (strict).
- Traffic between interfaces in the same security zone is not subjected to any policy; the traffic passes freely.
- The source and the destination zones in a zone pair must be the type security.
- The same zone cannot be defined as both the source and the destination.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *zone-name*
4. **description** *line-of-description*
5. **exit**
6. **zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*]
7. **description** *line-of-description*
8. **exit**
9. **interface** *type number*
10. **zone-member security** *zone-name*
11. **exit**
12. **zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**]} **destination** [**self** | *destination-zone-name*]
13. **service-policy type inspect** *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	**zone security** *zone-name* **Example:** `Router(config)# zone security zone1`	Creates a security zone to which interfaces can be assigned. Enters security zone configuration mode.
Step 4	**description** *line-of-description* **Example:** `Router(config-sec-zone)# description Internet Traffic`	(Optional) Describes the zone.
Step 5	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 6	**zone-pair security** *zone-pair name* {**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Router(config)# zone-pair security zp source z1 destination z2`	Creates a zone-pair. **Note** To apply a policy, you must configure a zone-pair Enters security zone configuration mode.
Step 7	**description** *line-of-description* **Example:** `Router(config-sec-zone)# description accounting network`	(Optional) Describes the zone-pair.
Step 8	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface for configuration. Enters interface configuration mode.

	Command or Action	Purpose
Step 10	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security zone1`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone-pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 11	**exit** **Example:** `Router(config-if)# exit`	Returns to interface configuration mode.
Step 12	**zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**]} **destination** [**self** | *destination-zone-name*] **Example:** `Router(config)# zone-pair security zp source z1 destination z2`	Creates a zone-pair. Enters security zone-pair configuration mode.
Step 13	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect p2`	Attaches a firewall policy map to the destination zone-pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.

Configuring the Firewall with WAAS and WCCP

Use the task in this section to configure firewall inspection parameters with WAAS.

Note In Cisco IOS XE software, WAAS support is always enabled and WAAS processing is always discovered. Thus, the **ip inspect waas enable** is not necessary and therefore not supported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip wccp** *service-id*
4. **class-map type inspect** *class-name*
5. **match protocol** *protocol-name* [**signature**]
6. **exit**
7. **policy-map type inspect** *policy-map-name*
8. **class class-default**
9. **class-map type inspect** *class-name*

10. **inspect**
11. **exit**
12. **exit**
13. **zone security** *zone-name*
14. **description** *line-of-description*
15. **exit**
16. **zone-pair security** *zone-pair name* {**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*]
17. **description** *line-of-description*
18. **exit**
19. **interface** *type number*
20. **description** *line-of-description*
21. **zone-member security** *zone-name*
22. **ip address** *ip-address*
23. **ip wccp** {*service-id* {**group-listen** | **redirect** {**in** | **out**}} | **redirect exclude in** | **web-cache** {**group-listen** | **redirect** {**in** | **out**}}
24. **exit**
25. **zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**]} **destination** [**self** | *destination-zone-name*]
26. **service-policy type inspect** *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 wccp** *service-id* **Example:** `Router(config)# ip wccp 61`	Enters the WCCP dynamically defined service identifier number.
Step 4	**class-map type inspect** *class-name* **Example:** `Router(config)# class-map type inspect most-traffic`	Creates an inspect type class map for the traffic class and enters class-map configuration mode. **Note** The **class-map type inspect most-traffic** command is hidden.

	Command or Action	Purpose
Step 5	**match protocol** *protocol-name* [**signature**] **Example:** `Router(config-cmap)# match protocol http`	Configures the match criteria for a class map on the basis of a specified protocol and enters security zone configuration mode. Only Cisco IOS stateful packet inspection supported protocols can be used as match criteria in inspect type class maps. • **signature**—Signature-based classification for peer-to-peer (P2P) packets is enabled.
Step 6	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 7	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect p1`	Creates a Layer 3 and Layer 4 inspect type policy map and enters policy-map configuration mode.
Step 8	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Specifies the matching of the system default class. If the system default class is not to be specified, then unclassified packets are matched.
Step 9	**class-map type inspect** *class-name* **Example:** `Router(config-pmap)# class type inspect most-traffic`	Specifies the firewall traffic (class) map on which an action is to be performed.
Step 10	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables Cisco IOS stateful packet inspection.
Step 11	**exit** **Example:** `Router(config-pmap-c)# exit`	Returns to policy map configuration mode.
Step 12	**exit** **Example:** `Router(config-pmap)# exit`	Returns to global configuration mode.
Step 13	**zone security** *zone-name* **Example:** `Router(config)# zone security zone1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.

	Command or Action	Purpose
Step 14	**description** *line-of-description* **Example:** `Router(config-sec-zone)# description Internet Traffic`	(Optional) Describes the zone.
Step 15	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 16	**zone-pair security** *zone-pair name* {**source** *source-zone-name* | **self**] **destination** [**self** | *destination-zone-name*] **Example:** `Router(config)# zone-pair security zp source z1 destination z2`	Creates a zone-pair and enters security zone configuration mode. **Note** To apply a policy, you must configure a zone-pair
Step 17	**description** *line-of-description* **Example:** `Router(config-sec-zone)# description accounting network`	(Optional) Describes the zone-pair.
Step 18	**exit** **Example:** `Router(config-sec-zone)# exit`	Returns to global configuration mode.
Step 19	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface and enters interface configuration mode.
Step 20	**description** *line-of-description* **Example:** `Router(config-if)# description`	(Optional) Describes the interface.
Step 21	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security zone1`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic into and out of that interface (except traffic bound for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone-pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.

	Command or Action	Purpose
Step 22	**ip address** *ip-address* **Example:** `Router(config-if)# ip address 10.70.0.1 255.255.255.0`	Assigns the interface IP address for the security zone and enters IP configuration mode.
Step 23	**ip wccp** {*service-id* {**group-listen** | **redirect** {**in** | **out**}} | **redirect exclude in** | **web-cache** {**group-listen** | **redirect** {**in** | **out**}} **Example:** `Router(config-if)# ip wccp 61 redirect in`	Specifies the following WCCP parameters on the interface: • The *service-id* argument defines a service identifier number from 1 to 254. • The **redirect exclude in** keywords are used to exclude inbound packets from outbound redirection. • The **web-cache** keyword is used to define the standard web caching service. • The **group-listen** keyword is used for discovering multicast WCCP protocol packets. • The **in** keyword is used to redirect to a cache engine the appropriate inbound packets. • The **out** keyword is used to redirect to a cache engine the appropriate outbound packets.
Step 24	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 25	**zone-pair security** *zone-pair-name* {**source** *source-zone-name* | **self**]} **destination** [**self** | *destination-zone-name*] **Example:** `Router(config)# zone-pair security zp source z1 destination z2`	Creates a zone-pair and enters security zone-pair configuration mode.
Step 26	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect p2`	Attaches a firewall policy map to the destination zone-pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.

Configuration Examples for Zone-Based Policy Firewall

This section contains the following configuration examples:

Zone-Based Policy Configuration on an ASR 1000 Series Router: Example

The following example shows how to configure a basic zone-based policy on a Cisco IOS XE ASR 1000 series router:

```
class-map type inspect match-all tcp-traffic
match protocol tcp
match access-group 199
policy-map type inspect mypolicy
class type inspect tcp-traffic
inspect
zone security z1
description finance department networks
zone security z2
description engineering services network
interface ethernet0
zone-member security z1
interface ethernet1
zone-member security z2
zone-pair security zp source z1 destination z2
service-policy type inspect mypolicy
```

Firewall Configuration with WAAS and WCCP: Example

The following example provides an end-to-end WAAS traffic flow optimization configuration for the firewall that uses WCCP to redirect traffic to a WAE device for traffic interception.

The following configuration example prevents traffic from being dropped between security zone members because the integrated-service-engine interface is configured on a different zone and each security zone member is assigned an interface.

```
ip wccp 61
ip wccp 62
class-map type inspect most-traffic
match protocol icmp
match protocol ftp
match protocol tcp
match protocol udp
policy—map type inspect p1
class type inspect most—traffic
inspect
class class—default
zone security zone-hr
zone security zone-outside
zone security z-waas
zone—pair security hr—out source zone-hr destination zone-outside
service—policy type inspect p1
zone—pair security out—hr source zone-outside destination zone-hr
service—policy type inspect p1
zone—pair security eng—out source zone-eng destination zone-outside
service—policy type inspect p1

interface GigabitEthernet0/0
description Trusted interface
ipaddress 10.70.0.1 255.255.255.0
ip wccp 61 redirect in
zone—member security zone-hr

interface GigabitEthernet0/0
description Trusted interface
ipaddress 10.71.0.2 255.255.255.0
```

```
ip wccp 61 redirect in
zone—member security zone-eng

interface GigabitEthernet0/1
description Untrusted interface
ipaddress 10.72.2.3 255.255.255.0
ip wccp 62 redirect in
zone—member security zone-outside

interface Integrated—Service—Enginel/0
ipaddress 10.70.100.1 255.255.255.252
ip wccp redirect exclude in
zone—member security z-waas
```

SCCP-Enabled Firewall Configuration: Example

The following configuration example shows how to enable SCCP in a firewall configuration.

Note When you enable SCCP (via the **match protocol** command) in a firewall configuration, you must also enable TFTP (via the **match protocol** command); otherwise, the IP phones that use SCCP cannot communicate with the Call Manager.

```
class-map type inspect match-any c_appl
  match protocol sccp
  match protocol tftp
policy-map type inspect p1
 class type inspect c_appl
  inspect
 class class-default
zone security z_in
zone security z_out
zone-pair security in2out source z_in destination z_out
 service-policy type inspect p1
interface gigabitethernet0/1/5
zone-member security z_in
interface gigabitethernet0/1/6
zone-member security z_out
```

LDAP-Enabled Firewall Configuration: Example

The following example shows how to configure a firewall policy to inspect Layer 4 LDAP messages:

Interface Configuration

```
interface GigabitEthernet 0/1/5
 ip address 132.1.1.1 255.255.255.0
 zone-member security private
 no shut
interface GigabitEthernet 0/1/6
 ip address 212.1.1.1 255.255.255.0
 zone-member security internet
 no shut
```

Global Firewall Configuration

```
zone security private
zone security internet
class-map type inspect match-any internet-traffic-class
 match protocol ldap
 match protocol ldaps
 match protocol ldap-admin
policy-map type inspect private-internet-policy
 class type inspect internet-traffic-class
 inspect
 class class-default
zone-pair security private-internet source private destination internet
 service-policy type inspect private-internet-policy
```

Additional References

The following sections provide references related to Zone-Based Policy Firewall.

Related Documents

Related Topic	Document Title
Security commands	*Cisco IOS Security Command Reference*
Quality of Service commands	*Cisco IOS Quality of Service Solutions Command Reference*,

Standards

Standard	Title
No new or modified standards are supported by this release.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC4511	*Lightweight Directory Access Protocol (LDAP): The Protocol*

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 Zone-Based Policy Firewall

Table 1 lists the release history for this feature.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which 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 XE software release that introduced support for a given feature in a given Cisco IOS XE software release train. Unless noted otherwise, subsequent releases of that Cisco IOS XE software release train also support that feature.

Table 1 Feature Information for Zone-Based Policy Firewall

Feature Name	Releases	Feature Configuration Information
Zone-Based Policy Firewall	Cisco IOS XE Release 2.1	This feature provides a Cisco IOS XE software unidirectional firewall policy between groups of interfaces known as zones.
NAT High Speed Logging (HSL) Support	Cisco IOS XE Release 2.1	This feature introduces support for firewall High Speed Logging (HSL) using NetFlow v9 as export format. The following section provides information about this feature: Configuring NetFlow Event Logging. The following commands were introduced to support this feature: **parameter-map type inspect global**, **log dropped-packet**, **log flow-export v9 udp destination**, **log flow-export template timeout-rate**

Table 1 Feature Information for Zone-Based Policy Firewall (continued)

Feature Name	Releases	Feature Configuration Information
Firewall—SCCP Video ALG Support	Cisco IOS XE Release 2.4	SSCCP enables voice communication between two Skinny clients through the use of a Cisco CM. This feature enables Cisco Firewalls to inspect Skinny control packets that are exchanged between a Skinny client and the CM. The following section provides information about this feature: SCCP-Enabled Firewall Configuration: Example The **match protocol** command was enhanced to support the **sccp** keyword.
Firewall—NetMeeting Directory (LDAP) ALG Support	Cisco IOS XE Release 2.4	LDAP is an application protocol that is used for querying and updating information stored on directory servers. This feature enables Cisco Firewalls to support Layer 4 LDAP inspection by default. The following section provides information about this feature: • LDAP-Enabled Firewall Configuration: Example The **match protocol** command was enhanced to support the **ldap, ldaps, and ldap-admin** keywords.

Cisco Firewall—SIP Enhancements: ALG

First Published: April 14, 2008
Last Updated: August 27, 2009

Enhanced Session Initiation Protocol (SIP) inspection in the Cisco firewall provides basic SIP inspect functionality (SIP packet inspection and pinholes opening) as well as protocol conformance and application security. These enhancements give the user a more control than in previous releases on what policies and security checks to apply to SIP traffic and the capability to filter out unwanted messages or users.

The development of additional SIP functionality in Cisco IOS XE software provides increased support for Cisco Call Manager (CCM), Cisco Call Manager Express (CCME), and Cisco IP-IP Gateway based voice/video systems. The Application Layer Gateway (ALG) SIP enhancement also supports RFC 3261 and its extensions.

Finding Feature Information

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 Cisco Firewall—SIP Enhancements: ALG" section on page 8.

Use Cisco Feature Navigator to find information about platform support 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 Cisco Firewall—SIP Enhancements: ALG

Your system must be running Cisco IOS XE Release 2.4 or a later Cisco IOS XE software release

Restrictions for Cisco Firewall—SIP Enhancements: ALG

DNS Name Resolution

Although SIP methods can have Domain Name System (DNS) names instead of raw IP addresses, this feature currently does not support DNS names.

Cisco ASR 1000 Series Routers

This feature was implemented without support for AIC on the Cisco ASR 1000 series routers. This release includes support for the following commands only: **class-map type inspect**, **class type inspect**, **match protocol**, and **policy-map type inspect**.

Information About Cisco Firewall—SIP Enhancements: ALG

Firewall and SIP Overviews

Cisco IOS XE Firewall

The Cisco IOS XE firewall extends the concept of static access control lists (ACLs) by introducing dynamic ACL entries that open on the basis of the necessary application ports on a specific application and close these ports at the end of the application session. The Cisco IOS XE firewall achieves this functionality by inspecting the application data, checking for conformance of the application protocol, extracting the relevant port information to create the dynamic ACL entries, and closing these ports at the end of the session. The Cisco IOS XE firewall is designed to easily allow a new application inspection whenever support is needed.

Session Initiation Protocol

SIP is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions could include Internet telephone calls, multimedia distribution, and multimedia conferences. SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a request that invokes a particular method or function on the server and at least one response.

SIP invitations used to create sessions carry session descriptions that allow participants to agree on a set of compatible media types. SIP makes use of elements called proxy servers to help route requests to the user's current location, authenticate and authorize users for services, implement provider call-routing policies, and provide features to users. SIP also provides a registration function that allows users to upload their current locations for use by proxy servers. SIP runs on top of several different transport protocols.

Firewall for SIP Functionality Description

The firewall for SIP support feature allows SIP signaling requests to traverse directly between gateways or through a series of proxies to the destination gateway or phone. After the initial request, if the Record-Route header field is not used, subsequent requests can traverse directly to the destination gateway address as specified in the Contact header field. Thus, the firewall is aware of all surrounding proxies and gateways and allows the following functionality:

- SIP signaling responses can travel the same path as SIP signaling requests.
- Subsequent signaling requests can travel directly to the endpoint (destination gateway).
- Media endpoints can exchange data between each other.

SIP UDP and TCP Support

RFC 3261 is the current RFC for SIP, which replaces RFC 2543. This feature supports the SIP User Datagram Protocol (UDP) and the TCP format for signaling.

SIP Inspection

This section describes the deployment scenarios supported by the Cisco Firewall—SIP ALG Enhancements feature.

Cisco IOS XE Firewall Between SIP Phones and CCM

The Cisco IOS XE firewall is located between CCM or CCME and SIP phones. SIP phones are registered to CCM or CCME through the firewall, and any SIP calls from or to the SIP phones pass through the firewall.

Cisco IOS XE Firewall Between SIP Gateways

The Cisco IOS XE firewall is located between two SIP gateways, which can be CCM, CCME, or a SIP proxy. Phones are registered with SIP gateways directly. The firewall sees the SIP session or traffic only when there is a SIP call between phones registered to different SIP gateways. In some scenarios an IP-IP gateway can also be configured on the same device as the firewall. With this scenario all the calls between the SIP gateways are terminated in the IP-IP gateway.

Cisco IOS XE Firewall with Local CCME and Remote CCME/CCCM

The Cisco IOS XE firewall is located between two SIP gateways, which can be CCM, CCME, or a SIP proxy. One of the gateways is configured on the same device as the firewall. All the phones registered to this gateway are locally inspected by the firewall. The firewall also inspects SIP sessions between the two gateways when there is a SIP call between them. With this scenario the firewall locally inspects SIP phones on one side and SIP gateways on the other side.

Cisco IOS XE Firewall with Local CCME

The Cisco IOS XE firewall and CCME is configured on the same device. All the phones registered to the CCME are locally inspected by the firewall. Any SIP call between any of the phones registered will also be inspected by the Cisco IOS XE firewall.

How to Configure Cisco Firewall—SIP Enhancements: ALG

Perform the following task to configure Cisco IOS XE Firewall—SIP Enhancements: ALG

- Enabling SIP Inspection on Cisco ASR Series Routers, page 4

Enabling SIP Inspection on Cisco ASR Series Routers

To enable SIP packet inspection, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **match protocol** *protocol-name*
6. **exit**
7. **policy-map type inspect** *policy-map-name*
8. **class type inspect** *class-map-name*
9. **inspect**
10. **service-policy** *policy-map-name*
11. **class class-default**
12. **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.

class-map type inspect match-any *class-map-name* Router(config)# class-map type inspect match-any sip_class1	Creates an inspect type class map and enters class-map configuration mode.
protocol-name Router(config-cmap)# match protocol sip	Configures the match criterion for a class map on the basis of the named protocol.
protocol-name Router(config-cmap)# match protocol tftp	Configures the match criterion for a class map on the basis of the named protocol.
exit Router(config-cmap)# exit	Exits class-map configuration mode.
policy-map type inspect *policy-map-name* Router(config)# policy-map type inspect sip_policy	Creates an inspect type policy map and enters policy-map configuration mode.
class type inspect *class-map-name* Router(config-pmap)# class type inspect sip_class1	Specifies the class on which the action is performed and enters policy-map-class configuration mode.
inspect Router(config-pmap-c)# inspect	Enables stateful packet inspection.
service-policy *policy-map-name* Router(config-pmap-c)# service-policy policy_2	Attaches the policy map to the service policy for the interface or virtual circuit.
class class-default Router(config-pmap-c)# class class-default	Specifies that these policy map settings apply to the predefined default class. If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
exit Router(config-pmap-c)# exit	Exits policy-map-class configuration mode.

Troubleshooting Tips

- **clear zone-pair**
- **debug cce**
- **debug ip inspect**
- **debug policy-map type inspect**
- **show policy-map type inspect zone-pair**
- **show zone-pair security**

Configuration Examples for Cisco Firewall—SIP Enhancements: ALG

-

Firewall and SIP Configuration on Cisco ASR 1000 Series Routers: Example

```
zone security z_in
zone security z_out

interface fastethernet0/3/6
zone-member security z_in
interface fastethernet0/3/7
zone-member security z_out

zone-pair security in2out source z_in destination z_out
service-policy type inspect p1
```

Additional References

Related Documents

Related Topic	Document Title
	Cisco IOS Security Command Reference
	Guide to Cisco Systems VoIP Infrastructure Solution for SIP

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3261	SIP: Session Initiation Protocol

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

Table 1 lists the release history for this feature.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which 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 XE software release that introduced support for a given feature in a given Cisco IOS XE software release train. Unless noted otherwise, subsequent releases of that Cisco IOS XE software release train also support that feature.

Table 1 Feature Information for Cisco Firewall—SIP Enhancements: ALG

Feature Name	Releases	Feature Information
		class type inspect, class-map type inspect match protocol, policy-map type inspect