Cisco 1000 Series Connected Grid Routers Security Software Configuration Guide

June 2014

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number: OL-25632-03

Security Overview

Cisco Connected Grid OS software (hereafter referred to as Cisco CG-OS software) on the Cisco 1000 Series Connected Grid Routers (hereafter referred to as Cisco CG-OS router) supports security features that can protect your network against degradation or failure. These features can also protect against data loss or compromise resulting from intentional attacks and from unintended but damaging mistakes by network users.

This chapter includes the following sections:

Authentication, Authorization, and Accounting

Authentication, authorization, and accounting (AAA) is an architectural framework for configuring a set of three independent security functions in a consistent, modular manner.

- Authentication—Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption. Authentication is the process of identifying a user before that user is allowed access to the network and network services. Configuring AAA authentication involves defining a named list of authentication methods and then applying that list to various interfaces.
- Authorization—Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared with the information contained in a database for a given user, and the result is returned to AAA to determine the user's actual capabilities and restrictions.
- Accounting—Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes. Accounting enables you to track the services that users are accessing, as well as the amount of network resources that they are consuming.

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

Related Topics

Chapter 4, "Configuring AAA"

RADIUS and TACACS+ Security Protocols

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

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

- RADIUS—A distributed client/server system that allows you to secure networks against unauthorized access. In the Cisco implementation, RADIUS clients run on the Cisco CG-OS router and send authentication and accounting requests to a central RADIUS server that contains all user-authentication and network-service access information.
- TACACS+—A security application implemented through AAA that provides a centralized validation of users who are attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. TACACS+ provides for separate and modular authentication, authorization, and accounting facilities. TACACS+ specifically requires Command Authorization and Configuration Authorization for Device Administration to the router.

Related Topics

Chapter 2, "Configuring RADIUS"

Chapter 3, "Configuring TACACS+"

SSHv2 and Telnet

You can use the Secure Shell version 2 (SSHv2) server to enable an SSHv2 client to make a secure, encrypted connection to the Cisco CG-OS router. SSHv2 uses strong encryption for authentication.

- The SSHv2 server in Cisco CG-OS software is interoperable with publicly and commercially available SSHv2 clients.
- The SSHv2 client in Cisco CG-OS software works with publicly and commercially available SSHv2 servers.

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site and then passes the keystrokes from one device to the other. Telnet can accept either an IP address or a domain name as the remote device address.

Related Topics

Chapter 5, "Configuring SSHv2 and Telnet"

PKI

The Public Key Infrastructure (PKI) allows the Cisco CG-OS router to obtain and use digital certificates for secure communication in the network and provides manageability and scalability for applications, such as SSHv2, that support digital certificates.

Related Topics

Chapter 6, “Configuring PKI”

User Accounts and Roles

You can create and manage user accounts and assign roles that limit access to operations on the Cisco CG-OS router. This definition and assignment process is knows as role-based access control (RBAC).

Related Topics

Chapter 7, “Configuring User Accounts and RBAC”

IKEv2 and IPSec

Internet Key Exchange version 2 (IKEv2) and Cisco IP Security (IPSec) allow configuration of secure communications between a source (Cisco CG-OS router) and destination router over a virtual tunnel.

Related Topics

Chapter 8, “Configuring IKEv2 and IPSec”

IP ACLs

IP ACLs are ordered sets of rules that you can use to filter traffic based on IPv4 information in the Layer 3 header of packets. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When Cisco CG-OS software determines that an IP ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether a packet is permitted or denied, or if there is no match, Cisco CG-OS software applies the applicable default rule. Cisco CG-OS software continues processing packets that are permitted and drops packets that are denied.

Related Topics

Chapter 9, “Configuring IP ACLs”

Control-Plane Policing

To prevent the Cisco CG-OS router from Denial of Service (DoS) attacks, the system employs control-plane policing (CoPP or CPP). CoPP increases security on the router by protecting the system from unnecessary or DoS traffic and gives priority to important control-plane and management traffic.

Related Topics

Chapter 10, “Configuring Control-Plane Policing”

Zero Touch Configuration

Zero Touch Deployment is an ease-of-use feature that automatically registers (enrolls) and distributes X.509 certificates and provisioning information over secure connections within a connected grid network.

Related Topics

Chapter 11, “Zero Touch Deployment”

Configuring RADIUS

This chapter describes how to configure the Remote Access Dial-In User Service (RADIUS) protocol on the Cisco 1000 Series Connected Grid Routers (hereafter referred to as Cisco CG-OS router).

This chapter includes the following sections:

Information About RADIUS

The RADIUS distributed client/server system allows you to secure networks against unauthorized access. In the Cisco implementation, RADIUS clients run on the Cisco CG-OS router and send authentication and accounting requests to a central RADIUS server that contains all user-authentication and network-service access information.

This section includes the following topics:

RADIUS Network Environments

RADIUS is implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

You can use RADIUS in the following network environments that require access security:

- Networks with multiple-vendor network devices, each supporting RADIUS. For example, network devices from several vendors can use a single RADIUS server-based security database.
- Networks that require resource accounting. You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet Service Provider (ISP) might use a freeware-based version of the RADIUS access control and accounting software to meet special security and billing needs.
- Networks that support authentication profiles. Using the RADIUS server in your network, you can configure authentication, authorization, and accounting (AAA) authentication and set up per-user profiles. Per-user profiles enable the Cisco CG-OS router to better manage ports by using their existing RADIUS solutions and to efficiently manage shared resources by offering different Service-Level Agreements (SLAs).

RADIUS Operation

When a user attempts to log in to a Cisco CG-OS router and authenticate by using a remote RADIUS server, the following process occurs:

1. Server prompts the user for username and password.
2. Cisco CG-OS router sends entered username and encrypted password over the network to the RADIUS server.
3. The user receives one of the following responses from the RADIUS server:
   - ACCEPT—Server authenticates the user.
   - REJECT—Server does not authenticate the user and the Cisco CG-OS router prompts the user to reenter the username and password, or access is denied.
   - CHALLENGE—Server requests and collects additional information from the user.
   - CHANGE PASSWORD—Server requests that the user select a new password.

The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

- Services that the user can access, including Telnet, rlogin, or Local-Area Transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 or IPv6 address, access list, and user timeouts.

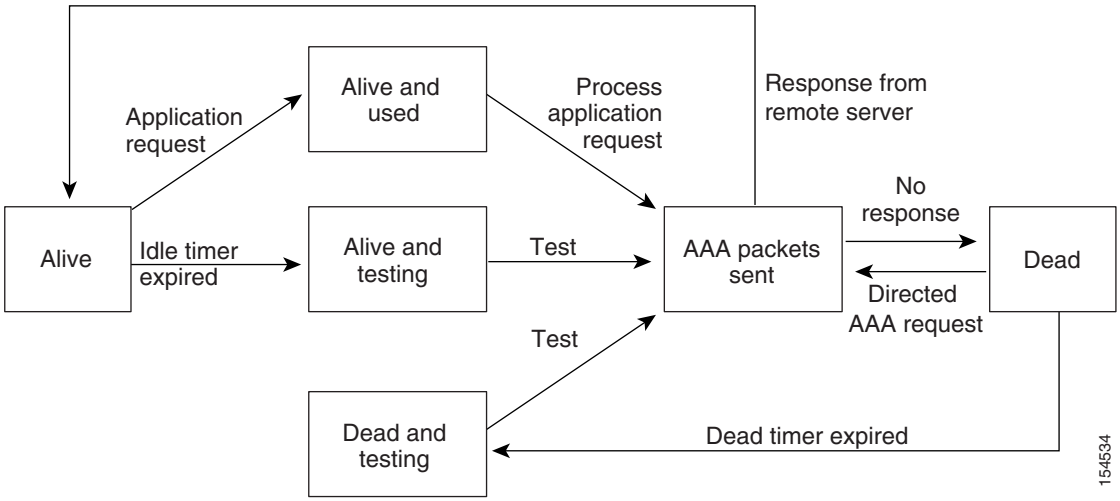
RADIUS Server Monitoring

An unresponsive RADIUS server can cause a delay in processing AAA requests. You can configure the Cisco CG-OS router to periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco CG-OS router marks unresponsive RADIUS servers as dead and does not send AAA requests to any dead RADIUS servers. The Cisco CG-OS router

periodically monitors the dead RADIUS servers and marks them as in the alive state when they respond. This monitoring process verifies that a RADIUS server is in a working state before real AAA requests are sent its way.

Whenever a RADIUS server changes to the dead state, the Cisco CG-OS router displays an error message that a failure is taking place. Figure 2-1 shows the RADIUS server states.

Figure 2-1 RADIUS Server States



Note The Cisco CG-OS router performs RADIUS server monitoring by sending a test authentication request to the RADIUS server. (See Configuring Periodic RADIUS Server Monitoring, page 2-14.)

Vendor-Specific Attributes

The Internet Engineering Task Force (IETF) draft standard specifies attribute 26 as the method for communicating Vendor-Specific Attributes (VSAs) between the network access server and the RADIUS server. VSAs allow vendors to support their own extended attributes that are not suitable for general use.

The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol: attribute separator value *
```

The protocol is a Cisco attribute for a particular type of authorization, the separator is an equal (=) sign for mandatory attributes, and an asterisk (`*`) indicates optional attributes.

When you use RADIUS servers for authentication on a Cisco CG-OS router, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, with authentication results. This authorization information is specified through VSAs.

Cisco CG-OS software supports the following VSA protocol options:

- Shell—Protocol used in access-accept packets to provide user profile information.
- Accounting—Protocol used in accounting-request packets. If a value contains any white spaces, enclose the value within double quotation marks.

Cisco CG-OS software supports the following attributes:

- Roles—Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space. For example, if the user belongs to roles network-operator and vdc-admin, the value field is “network-operator vdc-admin.” This subattribute, which the RADIUS server sends in the VSA portion of the Access-Accept frames, can only be used with the shell protocol value. The following examples show the roles attribute that the Cisco Access Control Server (ACS) supports:

  ```
  shell:roles="network-operator vdc-admin"

  shell:roles*"network-operator vdc-admin"
  ```

  The following examples show the roles attribute that is supported by FreeRADIUS:

  ```
  Cisco-AVPair = "shell:roles=\"network-operator vdc-admin\""

  Cisco-AVPair = "shell:roles*\"network-operator vdc-admin\""
  ```

Note When you specify a VSA as shell:roles*"network-operator vdc-admin" or "shell:roles*\"network-operator vdc-admin\"", this VSA is flagged as an optional attribute and other Cisco devices ignore this attribute.

- accountinginfo—Stores accounting information in addition to the attributes covered by the standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the Cisco CG-OS router. It can be used only with the accounting protocol data units (PDUs).

Prerequisites for RADIUS

Obtain IPv4 or IPv6 addresses or hostnames for the RADIUS servers.

Obtain keys from the RADIUS servers.

Ensure that the Cisco CG-OS router is is recognized as a RADIUS client on the AAA servers.

Guidelines and Limitations

You can configure a maximum of 64 RADIUS servers on the Cisco CG-OS router.

When you have a user account configured on the local Cisco CG-OS router that has the same name as a remote user account on an AAA server, Cisco CG-OS software applies the user roles for the local user account to the remote user, instead of the user roles configured on the AAA server.

Default Settings

Table 2-1 lists the default settings for RADIUS parameters.

Table 2-1 Default RADIUS Parameters

Parameters	Default
Server roles	Authentication and accounting
Dead timer interval	0 minutes
Retransmission count	1
Retransmission timer interval	5 seconds
Authentication UDP port	1812
Accounting UDP port	1813
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

Configuring RADIUS Servers

This section includes the following topics:

RADIUS Server Configuration Process

To configure RADIUS servers, follow these steps:

Step 1 Establish the RADIUS server connections to the Cisco CG-OS router. (See Configuring RADIUS Servers, page 2-6.)

Step 2 Configure the RADIUS secret keys for the RADIUS servers. (See Configuring Global RADIUS Keys, page 2-7.)

Step 3 If needed, configure RADIUS server groups with subsets of the RADIUS servers for AAA authentication methods. (See Configuring RADIUS Server Groups, page 2-9 and Configuring AAA, page 4-6.)

Step 4 When needed, configure any of the following optional parameters:

- Dead-time interval (See Configuring the Dead-Time Interval, page 2-15.)
- Allow specification of a RADIUS server at login (See Configuring the Global RADIUS Transmission Retry Count and Timeout Interval, page 2-11.)
- Transmission retry count and timeout interval (See Configuring the Global RADIUS Transmission Retry Count and Timeout Interval, page 2-11.)
- Accounting and authentication attributes (See Configuring Accounting and Authentication Attributes for RADIUS Servers, page 2-13.)

Step 5 (Optional) Configure periodic RADIUS server monitoring. (See Configuring Periodic RADIUS Server Monitoring, page 2-14.)

Configuring RADIUS Servers

To access a remote RADIUS server, you must define the IP address or hostname of the RADIUS server on the Cisco CG-OS router. You can configure up to 64 RADIUS servers.

Note

- By default, when you define a RADIUS server IP address or hostname on the Cisco CG-OS router, the RADIUS server becomes a member of the default RADIUS server group.
- You can also add the RADIUS server to another RADIUS server group. For information about creating RADIUS server groups, see Configuring RADIUS Server Groups, page 2-9.

BEFORE YOU BEGIN

Ensure that the server is a member of a server group. Refer to Configuring RADIUS Server Groups, page 2-9.

Ensure that the server is configured to authenticate RADIUS traffic.

Ensure that the Cisco CG-OS router is recognized as a RADIUS client on the AAA servers.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*}	Specifies the IPv4 or IPv6 address or hostname for a RADIUS server to use for authentication.
Step 3	**show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 4	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to define an IPv4 address or hostname for those RADIUS servers that the Cisco CG-OS router wants to access.

```
router# configure terminal
router(config)# radius-server host 10.10.1.1
router(config)# copy running-config startup-config
```

Configuring Global RADIUS Keys

You can configure RADIUS keys for all servers used by the Cisco CG-OS router. A RADIUS key is a shared secret text string between the Cisco CG-OS router and the RADIUS server hosts. To configure a RADIUS key specific to a RADIUS server, see Configuring a Key for a Specific RADIUS Server, page 2-8.

BEFORE YOU BEGIN

Obtain the RADIUS key values for the remote RADIUS servers.

Configure the RADIUS key on the remote RADIUS servers.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server key** [**0** | **7**] *key-value*	Specifies a RADIUS key for all RADIUS servers. You can specify that the *key-value* be in clear-text (**0**) format or be encrypted (**7**). Cisco CG-OS software encrypts a clear-text key before saving it to the running configuration. The maximum length is 63 characters. The default format is clear-text. By default, no RADIUS key is configured.

	Command	Purpose
Step 3	**show radius-server [groups | sorted | statistics]**	(Optional) Displays the RADIUS server configuration or the specified options. **groups**–Displays RADIUS server group configuration. **sorted**–Lists RADIUS servers sorted by name. **statistics**–Displays RADIUS statistics. **Note** The Cisco CG-OS router saves the RADIUS keys in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted RADIUS keys. Enter the **show** command in the EXEC mode to display all options above.
Step 4	**copy running-config startup-config**	(Optional) Saves the configuration change.

EXAMPLE

This example shows how to configure a global key for all RADIUS servers with which the Cisco CG-OS router communicates.

```
router# configure terminal
router(config)# radius-server key 0 PlIjUhYg
router(config)# copy running-config startup-config
```

Configuring a Key for a Specific RADIUS Server

You can configure a key on the Cisco CG-OS router for a specific RADIUS server. A RADIUS key is a secret text string shared between the Cisco CG-OS router and a specific RADIUS server.

BEFORE YOU BEGIN

Configure one or more RADIUS server hosts. (See Configuring RADIUS Servers, page 2-6.)

Obtain the key value for the remote RADIUS server.

Configure the key on the RADIUS server.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** [**0** | **7**] *key-value*	Specifies a RADIUS key for a specific RADIUS server. You can specify that the *key-value* is in clear-text (**0**) format or is encrypted (**7**). The Cisco CG-OS software encrypts a clear-text key before saving it to the running configuration. The maximum length is 63 characters. The default format is clear-text. The specified RADIUS server uses this RADIUS key rather than the global RADIUS key.
Step 3	**show radius-server**	(Optional) Displays the RADIUS server configuration. **Note** Cisco CG-OS software saves the RADIUS keys in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted RADIUS keys.
Step 4	**copy running-config startup-config**	(Optional) Saves this configuration change.

EXAMPLE

This example shows how to configure a shared key on a RADIUS server:

```
router# configure terminal
router(config)# radius-server host 10.10.1.1 key 0 PlIjUhYg
router(config)# copy running-config startup-config
```

Configuring RADIUS Server Groups

You can specify one or more remote AAA servers for authentication using server groups. All members of a group must use the RADIUS protocol. The Cisco CG-OS router contacts the servers in the order in which they are configured. You can configure up to 100 server groups on the Cisco CG-OS router.

You can configure these server groups at any time but they only take effect when you apply them to an AAA service. For information on AAA services, see Remote AAA Services, page 4-2.

BEFORE YOU BEGIN

Ensure that all servers that you want to add to the group are RADIUS servers.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa group server radius** *group-name*	Creates a RADIUS server group and enters the RADIUS server group configuration submode for that group. The *group-name* argument is a case-sensitive alphanumeric string with a maximum length of 127 characters.
Step 3	**server** {*ipv4-address* | *ipv6-address* | *host-name*}	Configures the RADIUS server as a member of the RADIUS server group. **Tip** When the specified RADIUS server is not found, enter the **radius-server host** command to identify the server and then retry this command.
Step 4	**deadtime** *minutes*	(Optional) Configures the monitoring dead time. The range is from 1 through 1440. The default is 0 minutes. **Note** When the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value. (See Configuring the Dead-Time Interval, page 2-15.)
Step 5	**source-interface** *interface*	(Optional) Configures a source interface to access the RADIUS servers in the server group. You can use Ethernet, cellular, and WiMax interfaces, and loopback interfaces. The default is the global source interface. (See Configuring the Global Source Interface for RADIUS Server Groups, page 2-10.)
Step 6	**show radius-server groups** [*group-name*]	(Optional) Displays the RADIUS server group configuration.
Step 7	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to create a RADIUS server group.

```
router# configure terminal
router(config)# aaa group server radius RadServer
router(config-radius)# server 10.10.1.1
router(config-radius)# copy running-config startup-config
```

Configuring the Global Source Interface for RADIUS Server Groups

You can configure a global source interface for RADIUS server groups to use when accessing RADIUS servers. To configure a different source interface for a specific RADIUS server group, see Configuring RADIUS Server Groups, page 2-9. By default, Cisco CG-OS software uses any available interface.

BEFORE YOU BEGIN

Configure at least one server group.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip radius source-interface** *interface*	Configures the global source interface for all RADIUS server groups configured on the device.
Step 3	**show radius-server [groups | sorted | statistics]**	(Optional) Displays the RADIUS server configuration information.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

```
router# configure terminal
router(config)# ip radius source-interface mgmt 0
router(config)# copy running-config startup-config
```

Configuring the Global RADIUS Transmission Retry Count and Timeout Interval

You can configure a global retransmission retry count and timeout interval for all RADIUS servers. By default, a Cisco CG-OS router retries transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server.

The timeout interval determines how long the Cisco CG-OS router waits for responses from RADIUS servers before declaring a timeout failure.

BEFORE YOU BEGIN

Configure at least one server group.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server retransmit** *count*	Specifies the number of times that a router retransmits data to a RADIUS server before it reverts to local authentication. Sets the retransmission count for all RADIUS servers. The count range is from 1 to 5. The default retransmission count is 1.
Step 3	**radius-server timeout** *seconds*	Specifies the transmission timeout interval for RADIUS servers. The range is from 1 to 60 seconds. The default timeout interval is 5 seconds.

	Command	Purpose
Step 4	**show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure the global RADIUS parameters, transmission retry count and timeout interval.

```
router# configure terminal
router(config)# radius-server retransmit 3
router(config)# radius-server timeout 10
router(config)# copy running-config startup-config
```

Configuring the RADIUS Transmission Retry Count and Timeout Interval for a Specific Server

By default, the Cisco CG-OS router retries a transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. You can also set a timeout interval that the Cisco CG-OS router waits for responses from RADIUS servers before declaring a timeout failure and reporting it to the system log.

BEFORE YOU BEGIN

Configure at least one RADIUS server. (See Configuring RADIUS Servers, page 2-6.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **retransmit** *count*	Specifies the retransmission count for a specific server. The default is the global value. **Note** The retransmission count value specified for a specific RADIUS server overrides the global count value specified for all RADIUS servers.
Step 3	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **timeout** *seconds*	Specifies the transmission timeout interval for a specific server. The default is the global value. **Note** The timeout interval value specified for a specific RADIUS server overrides the global interval value specified for all RADIUS servers.

	Command	Purpose
Step 4	**show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure the RADIUS parameters, transmission retry count and timeout interval, for a specific server.

```
router# configure terminal
router(config)# radius-server host server1 retransmit 3
router(config)# radius-server host server1 timeout 10
router(config)# copy running-config startup-config
```

Configuring Accounting and Authentication Attributes for RADIUS Servers

You can specify that a RADIUS server only be used for accounting purposes or only be used for authentication purposes. By default, RADIUS servers perform both accounting and authentication.

You can also specify the destination UDP port numbers for RADIUS accounting and authentication messages when there is a conflict with the default port.

BEFORE YOU BEGIN

Configure at least one RADIUS server. (See Configuring RADIUS Servers, page 2-6.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **acct-port** *udp-port*	(Optional) Specifies a UDP port to use for RADIUS accounting messages. The range is from 0 to 65535. The default UDP port is 1813.
Step 3	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **accounting**	(Optional) Specifies the RADIUS server for accounting purposes only. The default is both accounting and authentication.
Step 4	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **auth-port** *udp-port*	(Optional) Specifies a UDP port to use for RADIUS authentication messages. The range is from 0 to 65535. The default UDP port is 1812.
Step 5	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **authentication**	(Optional) Specifies the RADIUS server for authentication purposes only. The default is both accounting and authentication.

	Command	Purpose
Step 6	**show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 7	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure one RADIUS server to perform only accounting and another to perform only authentication.

```
router# configure terminal
router(config)# radius-server host 10.10.1.1 accounting
router(config)# radius-server host 10.10.2.2 authentication
router(config)# copy running-config startup-config
```

Configuring Periodic RADIUS Server Monitoring

You can monitor the availability of RADIUS servers. These parameters include the username and password to use for the server and an idle timer.

The idle timer specifies the interval during which a RADIUS server receives no requests before the Cisco CG-OS router sends out a test packet. You can configure this option to test servers periodically.

Note For security reasons, Cisco recommends that you do not configure a test username that is the same as an existing user name in the RADIUS database.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, the Cisco CG-OS router does not perform periodic RADIUS server monitoring.

BEFORE YOU BEGIN

Configure at least one RADIUS server. (See Configuring RADIUS Servers, page 2-6.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}	Specifies parameters for server monitoring. The default username password is **test**. The valid range for the idle timer is from 0 to 1440 minutes. The default value for the idle timer is 0 minutes. **Note** For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.

	Command	Purpose
Step 3	**radius-server dead-time** *minutes*	Specifies the number of minutes before the Cisco CG-OS router checks a RADIUS server that was previously unresponsive. The valid range is from 1 to 1440 minutes. The default value is 0 minutes.
Step 4	**show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure RADIUS monitoring parameters.

```
router# configure terminal
router(config)# radius-server host 10.10.1.1 test username user1 password Ur2Gd2BH
idle-time 3
router(config)# radius-server dead-time 5
router(config)# copy running-config startup-config
```

Configuring the Dead-Time Interval

You can configure the dead-time interval for all RADIUS servers. The dead-time interval specifies the time that the Cisco CG-OS router waits after declaring a RADIUS server as dead, before sending out a test packet to determine if the server is now alive. The default value is 0 minutes.

Note When the dead-time interval is 0 minutes, RADIUS servers are not marked as dead even if they are not responding. You can configure the dead-time interval for a RADIUS server group. (See Configuring RADIUS Server Groups, page 2-9.)

BEFORE YOU BEGIN

Configure at least one RADIUS server. (See Configuring RADIUS Servers, page 2-6.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server deadtime** *minutes*	Configures the dead-time interval. The range is from 1 to 1440 minutes. The default value is 0 minutes.
Step 3	**show radius-server**	(Optional) Displays the RADIUS server configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure the dead interval for all RADIUS servers.

```
router# configure terminal
router(config)# radius-server deadtime 5
router(config)# copy running-config startup-config
```

Manually Monitoring RADIUS Server or Groups

You can manually issue a test message to a RADIUS server or to a server group.

BEFORE YOU BEGIN

Configure at least one RADIUS server and server group. (See Configuring RADIUS Servers, page 2-6 and Configuring RADIUS Server Groups, page 2-9.)

DETAILED STEPS

	Command	Purpose
Step 1	**test aaa server radius** {*ipv4-address* | *ipv6-address* | *host-name*} *username password*	Sends a test message to a RADIUS server to confirm availability.
Step 1	**test aaa group** *group-name username password*	Sends a test message to a RADIUS server group to confirm availability.

EXAMPLE

This example shows how to configure a test message to be sent to a RADIUS server.

```
router# test aaa server radius 10.10.1.1 user1 Ur2Gd2BH

This example shows how to configure a test message to be sent to a RADIUS server group.

router# test aaa group RadGroup user2 As3He3CI
```

Verifying Configuration

To display RADIUS configuration information, enter any or all of the following commands.

Command	Purpose
show running-config radius [**all**]	Displays the RADIUS configuration in the running configuration.
show startup-config radius	Displays the RADIUS configuration in the startup configuration.
show radius-server [*host-name* | *ipv4-address* | *ipv6-address*] [**groups** | **sorted** | **statistics**]	Displays all configured RADIUS server parameters or a subset using the optional parameters.

For detailed information about the fields in the output from these commands, see the Command Lookup Tool on Cisco.com.

Monitoring Statistics

BEFORE YOU BEGIN

Configure at least one RADIUS server. (See Configuring RADIUS Servers, page 2-6.)

DETAILED STEPS

To display the statistics that the Cisco CG-OS router maintains for RADIUS server activity, enter the command below.

Command	Purpose
show radius-server statistics {*hostname* | *ipv4-address* | *ipv6-address*}	Displays statistics for RADIUS servers.

Configuration Example

The following example shows how to configure a RADIUS server:

```
radius-server key 7 "ToIkLhPpG"
radius-server host 10.10.1.1 key 7 "ShMoMhTl" authentication accounting
aaa group server radius RadServer
    server 10.10.1.1
```

Where to Go Next

You can now configure AAA authentication methods to include the RADIUS server groups. (See Chapter 4, “Configuring AAA”.)

Configuring TACACS+

This chapter describes how to configure the Terminal Access Controller Access Control System Plus (TACACS+) protocol on the Cisco 1000 Series Connected Grid Routers (hereafter referred to as Cisco CG-OS router).

This chapter includes the following sections:

Information About TACACS+

The TACACS+ security protocol provides centralized validation of users attempting to gain access to the Cisco CG-OS router. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before you can configure and employ the TACACS+ features on your Cisco CG-OS router.

TACACS+ provides for separate authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service—authentication, authorization, and accounting—independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The TACACS+ client/server protocol uses TCP (TCP port 49) for transport requirements. The Cisco CG-OS router provides centralized authentication using the TACACS+ protocol.

This section includes the following topics:

TACACS+ Advantages

TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the Cisco CG-OS router can authorize access without authenticating.
- Uses the TCP transport protocol to send data between the AAA client and server, making reliable transfers with a connection-oriented protocol.
- Encrypts the entire protocol payload between the Cisco CG-OS router and the AAA server to ensure higher data confidentiality. The RADIUS protocol only encrypts passwords.

TACACS+ Operation for User Login

When a user attempts a Password Authentication Protocol (PAP) login to a Cisco CG-OS router using TACACS+, the following actions occur:

1. When the Cisco CG-OS router establishes a connection, it contacts the TACACS+ daemon to obtain the username and password.

Note TACACS+ allows an arbitrary conversation between the daemon and the user until the daemon receives enough information to authenticate the user. This action is usually done by prompting for a username and password combination, but may include prompts for other items, such as the maiden name of your mother.

2. The Cisco CG-OS router will eventually receive one of the following responses from the TACACS+ daemon:
   a. ACCEPT—User authentication succeeds and service begins. When the Cisco CG-OS router requires user authorization, authorization begins.
   b. REJECT—User authentication fails. The TACACS+ daemon either denies further access to the user or prompts the user to retry the login sequence.
   c. ERROR—An error occurs at some time during authentication either at the daemon or in the network connection between the daemon and the Cisco CG-OS router. When the Cisco CG-OS router receives an ERROR response, the Cisco CG-OS router tries to use an alternative method for authenticating the user.

   After authentication, the user also undergoes an additional authorization phase when authorization is enabled on the Cisco CG-OS router. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.
3. If TACACS+ authorization is required, the Cisco CG-OS router again contacts the TACACS+ daemon and it returns an ACCEPT or REJECT authorization response. An ACCEPT response contains attributes that are used to direct the EXEC or NETWORK session for that user and determines the services that the user can access.

   Services include the following:

- Telnet, rlogin, Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services
- Connection parameters, including the host or client IP address (IPv4 or IPv6), access list, and user timeouts

Default TACACS+ Server Encryption Type and Secret Key

You must configure the TACACS+ secret key to authenticate the Cisco CG-OS router to the TACACS+ server. A secret key is a secret text string shared between the Cisco CG-OS router and the TACACS+ server host. The length of the key is restricted to 63 characters and can include any printable ASCII characters (white spaces are not allowed). You can configure a global secret key for all TACACS+ server configurations on the Cisco CG-OS router to use.

You can override the global secret key assignment by explicitly using the **key** option when configuring an individual TACACS+ server.

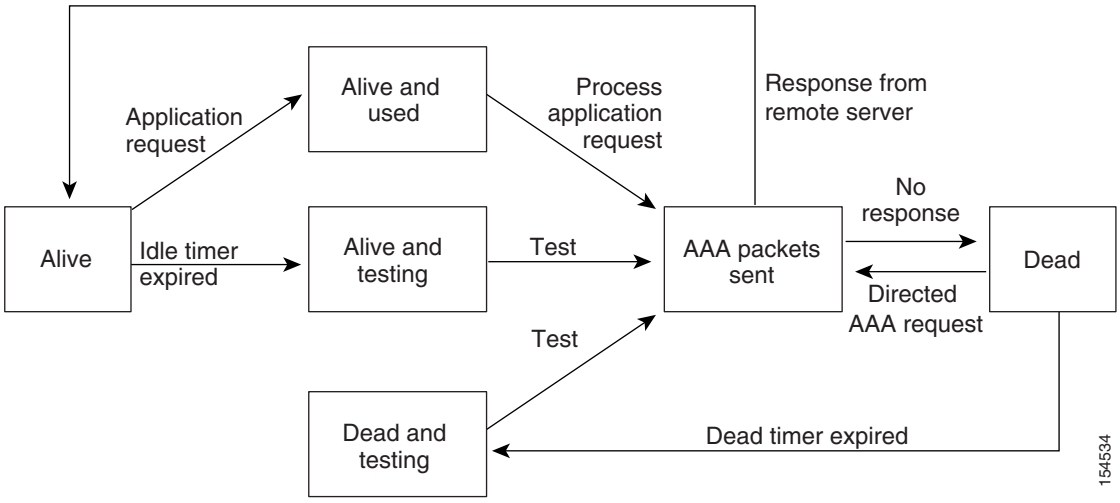
TACACS+ Server Monitoring

An unresponsive TACACS+ server can delay the processing of AAA requests.

A Cisco CG-OS router periodically monitors a TACACS+ server to check whether it is responding (or alive) to save time in processing AAA requests. The router marks an unresponsive TACACS+ server as dead and does not continue to send AAA requests to that dead TACACS+ server.

A Cisco CG-OS router periodically monitors dead TACACS+ servers and brings them to the alive state once they are responding. Continual monitoring ensures that a TACACS+ server is in a working state before it receives real AAA requests. Whenever an TACACS+ server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the Cisco CG-OS router displays an error message that a failure is taking place before it can impact performance. (See Figure 3-1.)

Figure 3-1 TACACS+ Server States



Note The Cisco CG-OS router initiates TACACS+ server monitoring by sending a test authentication request to the TACACS+ server.

Vendor-Specific Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific attributes (VSAs) between the network access server and the TACACS+ server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use.

This section includes the following topics:

Cisco VSA Format

The Cisco TACACS+ implementation supports one vendor-specific option using the format recommended in the IETF specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Cisco attribute for a particular type of authorization, separator is an equal sign (=) for mandatory attributes, and an asterisk (*) indicates optional attributes.

When you use TACACS+ servers for authentication on a Cisco CG-OS router, the TACACS+ protocol directs the TACACS+ server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

The Cisco CG-OS software supports the following VSA protocol options:

- Shell—Protocol used in access-accept packets to provide user profile information.
- Accounting—Protocol used in accounting-request packets. If a value contains any white spaces, enclose the value within double quotation marks.

The Cisco CG-OS software supports the following attributes:

- roles—Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space. For example, if the user belongs to roles network-operator and vdc-admin, the value field would be “network-operator vdc-admin.” This subattribute, which the TACACS+ server sends in the VSA portion of the Access-Accept frames, can only be used with the shell protocol value. The following examples show the roles attribute as supported by Cisco ACS:

  ```
  shell:roles="network-operator vdc-admin"

  shell:roles*"network-operator vdc-admin"
  ```

Note When you specify a VSA as shell:roles*“network-operator vdc-admin”, this VSA is flagged as an optional attribute and other Cisco devices ignore this attribute.

- accountinginfo—Stores accounting information in addition to the attributes covered by the standard TACACS+ accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the TACACS+ client on the Cisco CG-OS router. It can be used only with the accounting protocol data units (PDUs).

Cisco TACACS+ Privilege Levels

TACACS+ servers support privilege levels for specifying the permissions that users have when logging onto a Cisco CG-OS router. For the maximum privilege level 15, the Cisco CG-OS software applies the network-admin role in the default VDC or the vdc-admin role for non-default VDCs. All other privilege levels are translated to the vdc-operator role. For more information on user roles, see Chapter 7, "Configuring User Accounts and RBAC."

Note When you specify a user role in the cisco-av-pair, that takes precedence over the privilege level.

Note Although references to a default VDC might be seen in CLI displays, the Cisco CG-OS router does not support the configuration of more than one VDC. The Cisco CG-OS router only supports a default VDC.

Prerequisites for TACACS+

Obtain the IPv4 or IPv6 addresses or hostnames for the TACACS+ servers.

Obtain the secret keys from the TACACS+ servers, if any.

Ensure that the Cisco CG-OS router is recognized as a TACACS+ client on the AAA servers.

Guidelines and Limitations

Configure a maximum of 64 TACACS+ servers on the Cisco CG-OS router.

When you have a user account configured on the local Cisco CG-OS router that has the same name as a remote user account on an AAA server, the Cisco CG-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.

Default Settings

Table 3-1 lists the default settings for TACACS+ parameters.

Table 3-1 Default TACACS+ Parameters

Parameters	Default
TACACS+	Disabled
Dead timer interval	0 minutes
Timeout interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

Configuring TACACS+

This section includes the following topics:

TACACS+ Server Configuration Process

To configure TACACS+ servers, follow these steps:

Step 1 Enable TACACS+. (See Enabling TACACS+, page 3-7.)

Step 2 Establish the TACACS+ server connections to the Cisco CG-OS router. (See Configuring TACACS+ Server Hosts, page 3-7.)

Step 3 Configure the secret keys for the TACACS+ servers. (See Configuring Global TACACS+ Keys, page 3-8 and Configuring a Key for a Specific TACACS+ Server, page 3-9.)

Step 4 When needed, configure TACACS+ server groups with subsets of the TACACS+ servers for AAA authentication methods. (See Configuring TACACS+ Server Groups, page 3-10 and Configuring AAA, page 4-6.)

Step 5 When needed, configure any of the following optional parameters:

- Dead-time interval (See Configuring the Dead-Time Interval, page 3-15.)
- TACACS+ server specification allowed at user login (See Configuring the Global TACACS+ Timeout Interval, page 3-12.)
- Timeout interval (See Configuring the Global TACACS+ Timeout Interval, page 3-12.)
- TCP port (See Configuring TCP Ports, page 3-13.)

Step 6 When needed, configure periodic TACACS+ server monitoring. (See Configuring Periodic TACACS+ Server Monitoring, page 3-14.)

Enabling TACACS+

By default, the TACACS+ feature is disabled on the Cisco CG-OS router. You must enable the TACACS+ feature to access the configuration and verification commands for authentication.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**feature tacacs+**	Enables TACACS+.
Step 3	**show feature**	(Optional) Displays the enabled status of the feature.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to enable the TACACS+ feature on the Cisco CG-OS router before configuring commands that support authentication.

```
router# configure terminal
router(config)# feature tacacs+
router(config)# copy running-config startup-config
```

Configuring TACACS+ Server Hosts

To access a remote TACACS+ server, you must configure the IP address or the hostname for the TACACS+ server on the Cisco CG-OS router. You can configure up to 64 TACACS+ servers.

Note By default, when you configure a TACACS+ server IP address or hostname on the Cisco CG-OS router, the TACACS+ server is added to the default TACACS+ server group. You can also add the TACACS+ server to another TACACS+ server group. For information about creating TACACS+ server groups, see Configuring TACACS+ Server Groups, page 3-10).

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

Obtain the IPv4 or IPv6 addresses or the hostnames for the remote TACACS+ servers.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*}	Specifies the IPv4 or IPv6 address or hostname for a TACACS+ server.
Step 3	**show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure an IP address for the TACACS+ server on the Cisco CG-OS router.

```
router# configure terminal
router(config)# tacacs-server host 10.10.2.2
router(config)# copy running-config startup-config
```

Configuring Global TACACS+ Keys

You can configure secret TACACS+ keys at the global level for all servers used by the Cisco CG-OS router. A secret key is a secret text string shared between the Cisco CG-OS router and the TACACS+ server hosts.

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

Obtain the secret key values for the remote TACACS+ servers.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**tacacs-server key** [**0** | **7**] *key-value*	Specifies a TACACS+ key for all TACACS+ server. You can specify that the *key-value* is in clear text (**0**) format or is encrypted (**7**). The Cisco CG-OS software encrypts a clear-text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters. By default, no secret key is configured.

	Command	Purpose
Step 3	**show tacacs-server**	(Optional) Displays the TACACS+ server configuration. **Note** The Cisco CG-OS router saves the secret keys in an encrypted form in the running configuration. Use the **show running-config** command to display the encrypted secret keys.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure a global TACACS+ key for the Cisco CG-OS router.

```
router# configure terminal
router(config)# tacacs-server key 0 QsEfThUkO
router(config)# copy running-config startup-config
```

Configuring a Key for a Specific TACACS+ Server

You can configure secret keys for a TACACS+ server. A secret key is a shared secret text string between the Cisco CG-OS router and the TACACS+ server host to allow secure communication.

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

Obtain the secret key values for the remote TACACS+ servers.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** [**0** | **7**] *key-value*	Specifies a secret key for a specific TACACS+ server. You can specify that the *key-value* is in clear text (**0**) format or is encrypted (**7**). Cisco CG-OS software encrypts a clear text key before saving it to the running configuration. The maximum length is 63 characters. The default format is clear text. When a secret key is configured, it supersedes the global secret key.

	Command	Purpose
Step 3	**show tacacs-server**	(Optional) Displays the TACACS+ server configuration. **Note** The secret keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted secret keys.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure a key for a specific TACACS+ server.

```
router# configure terminal
router(config)# tacacs-server host 10.10.1.1 key 0 PlIjUhYg
router(config)# copy running-config startup-config
```

Configuring TACACS+ Server Groups

You can specify one or more remote AAA servers to authenticate users using server groups. All members of a group must use the TACACS+ protocol. The Cisco CG-OS router attempts access to the servers in the same order in which you configure them.

You can configure these server groups at any time but they only take effect when you apply them to an AAA service. For information on AAA services, see Remote AAA Services, page 4-2.

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa group server tacacs+** *group-name*	Creates a TACACS+ server group and enters the TACACS+ server group configuration mode for that group.
Step 3	**server** {*ipv4-address* | *ipv6-address* | *host-name*}	Configures the TACACS+ server as a member of the TACACS+ server group. **Tip** If the specified TACACS+ server is not found, configure it using the **tacacs-server host** command and retry this command.

	Command	Purpose
Step 4	**deadtime** *minutes*	Configures the monitoring dead time. The range is from 1 through 1440. The default is 0 minutes. **Note** The recommended value is one (1) minute. A value greater than one incurs greater delay in authentication with the external AAA server. **Note** When the dead-time interval for a TACACS+ server group is greater than zero (0), that value takes precedence over the global dead-time value. (See Configuring the Dead-Time Interval, page 3-15.)
Step 5	**source-interface** *interface*	(Optional) Configures a source interface to access the TACACS+ servers in the server group. You can use Ethernet interfaces, loopback interfaces, or the management interface (mgmt 0). The default is the global source interface.
Step 6	**show tacacs-server groups**	(Optional) Displays the TACACS+ server group configuration.
Step 7	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure a TACACS+ server group for authentication.

```
router# configure terminal
router(config)# aaa group server tacacs+ TacServer
router(config-tacacs+)# server 10.10.2.2
router(config-tacacs+)# deadtime 1
router(config-tacacs+)# copy running-config startup-config
```

Configuring the Global Source Interface for TACACS+ Server Groups

You can configure a global source interface for TACACS+ server groups to use when accessing TACACS+ servers. To configure a different source interface for a specific TACACS+ server group, see Configuring TACACS+ Server Groups, page 3-10. By default, the Cisco CG-OS software uses any available interface.

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip tacacs source-interface** *interface*	Configures the global source interface for all TACACS+ server groups configured on the device.

	Command	Purpose
Step 3	**show tacacs-server**	(Optional) Displays the TACACS+ server configuration information.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure a global source interface for TACACS+ server groups.

```
router# configure terminal
router(config)# ip tacacs source-interface mgmt 0
router(config)# copy running-config startup-config
```

Configuring the Global TACACS+ Timeout Interval

You can set a global timeout interval for all TACACS+ servers. The timeout interval determines how long the Cisco CG-OS router waits for responses from TACACS+ servers before declaring a timeout failure.

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**tacacs-server timeout** *seconds*	Specifies the timeout interval for TACACS+ servers. The range is from 1 to 60 seconds. The default timeout interval is 5 seconds.
Step 3	**show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure a global TACACS+ timeout interval for the Cisco CG-OS router.

```
router# configure terminal
router(config)# tacacs-server timeout 10
router(config)# copy running-config startup-config
```

Configuring the Timeout Interval for a Server

The timeout interval determines how long the Cisco CG-OS router waits for responses from a TACACS+ server before declaring a timeout failure.

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **timeout** *seconds*	Specifies the timeout interval for a specific server. The default is the global value. **Note** When you configure a timeout interval value for a TACACS+ server, that value overrides any global timeout interval value configured for all TACACS+ servers.
Step 3	**show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure a timeout for a TACACS+ server.

```
router# configure terminal
router(config)# tacacs-server host server1 timeout 10
router(config)# copy running-config startup-config
```

Configuring TCP Ports

You can configure another TCP port for the TACACS+ servers if there are conflicts with another application. By default, the Cisco CG-OS router uses port 49 for all TACACS+ requests.

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **port** *tcp-port*	Specifies the TCP port to use for TACACS+ messages to the server. The range is from 1 to 65535. The default TCP port is 49.
Step 3	**show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

```
router# configure terminal
router(config)# tacacs-server host 10.10.1.1 port 2
router(config)# copy running-config startup-config
```

Configuring Periodic TACACS+ Server Monitoring

You can monitor the availability of TACACS+ servers. The monitoring parameters include the username and password to use for the server and an idle timer. The idle timer specifies the interval in which a TACACS+ server receives no requests before the Cisco CG-OS router sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Note To protect network security, Cisco recommends that you use a username that is not the same as an existing username in the TACACS+ database.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, the Cisco CG-OS router does not perform periodic TACACS+ server monitoring.

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}	Specifies parameters for server monitoring. The default username is test and the default password is test. The valid range is from 0 to 1440 minutes. The default value for the idle timer is 0 minutes. **Note** For periodic TACACS+ server monitoring, the idle timer value must be greater than 0.
Step 3	**tacacs-server deadtime** *minutes*	Specifies the number of minutes before the Cisco CG-OS router checks a TACACS+ server that was previously unresponsive. The valid range is from 0 to 1440 minutes. The default value is 0 minutes.
Step 4	**show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure a dead time interval to monitor the availability of a TACACS+ server and how to configure a unique monitoring username and password.

```
router(config)# configure terminal
router(config)# tacacs-server host 10.10.1.1 test username user1 password Ur2Gd2BH
idle-time 3
router(config)# tacacs-server deadtime 5
router(config)# copy running-config startup-config
```

Configuring the Dead-Time Interval

You can configure a global dead-time interval for all TACACS+ servers. The dead-time interval specifies the time that the Cisco CG-OS router waits, after declaring a TACACS+ server is dead, before sending out a test packet to determine if the server is now alive.

Note When the dead timer interval is 0 minutes, TACACS+ servers are not marked as dead even if they are not responding. You can configure the dead-timer per server group. (See Configuring TACACS+ Server Groups, page 3-10.)

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**tacacs-server deadtime** *minutes*	Configures the global dead time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes
Step 3	**show tacacs-server**	(Optional) Displays the TACACS+ server configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to deliver a global deadtime interval that applies to all TACACS+ servers.

```
router(config)# configure terminal
router(config)# tacacs-server deadtime 5
router(config)# copy running-config startup-config
```

Enabling ASCII Authentication

You can enable ASCII authentication on the TACACS+ server.

Note Only TACACS+ servers support ASCII authentication.

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa authentication login ascii-authentication**	Enables ASCII authentication. The default is disabled.
Step 3	**show aaa authentication login ascii-authentication**	(Optional) Displays the TACACS+ ASCII authentication configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to enable ASCII authentication on the TACACS+ server.

```
router# configure terminal
router(config)# aaa authentication login ascii-authentication
router(config)# copy running-config startup-config
```

Manually Monitoring TACACS+ Servers or Groups

You can manually issue a test message to a TACACS+ server or to a server group.

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

DETAILED STEPS

Enter one or both of the commands below as applicable.

Command	Purpose
test aaa server tacacs+ {*ipv4-address* | *ipv6-address* | *host-name*} *username password*	Sends a test message to a TACACS+ server to confirm availability.
test aaa group *group-name username password*	Sends a test message to a TACACS+ server group to confirm availability.

EXAMPLE

This example shows how to configure a command to send a manual test of a TACACS+ server.

```
router# test aaa server tacacs+ 10.10.1.1 user1 Ur2Gd2BH
```

This example shows how to configure a command to send a manual test of a TACACS+ server group.

```
router# test aaa group TacGroup user2 As3He3CI
```

Disabling TACACS+

You can disable TACACS+.

Caution When you disable TACACS+, the Cisco CG-OS router automatically discards all related running configurations.

BEFORE YOU BEGIN

Ensure that an alternate TACACS+ server resource is available for those devices that are authenticated by the server before it is taken out of service.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**no feature tacacs+**	Disables TACACS+.
Step 3	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to disable a TACACS+ server.

```
router# configure terminal
router(config)# no feature tacacs+
router(config)# copy running-config startup-config
```

Monitoring TACACS+ Statistics

You can display the statistics that the Cisco CG-OS router maintains for TACACS+ activity by using the command in the table below.

BEFORE YOU BEGIN

Enable TACACS+. (See Enabling TACACS+, page 3-7.)

Command	Purpose
show tacacs-server statistics {*hostname* | *ipv4-address* | *ipv6-address*}	Displays the TACACS+ statistics.

Verifying TACACS+ Configuration

To display TACACS+ configuration information, enter any or all of the following commands:

Command	Purpose
show feature	Displays the enabled status of the feature.
show running-config tacacs [**all**]	Displays the TACACS+ configuration in the running configuration.
show startup-config tacacs	Displays the TACACS+ configuration in the startup configuration.
show tacacs-server [*host-name* | *ipv4-address* | *ipv6-address*] [**groups** | **sorted** | **statistics**]	Displays all configured TACACS+ server parameters. **Note** The Cisco CG-OS router does not support the directed-response option of this command and it is not shown in the command.

Configuration Example

The following example shows how to configure TACACS+:

```
feature tacacs+
tacacs-server key 7 "ToIkLhPpG"
tacacs-server host 10.10.2.2 key 7 "ShMoMhTl"
aaa group server tacacs+ TacServer
    server 10.10.2.2
```

Where to Go Next

You can now configure AAA authentication methods to include the TACACS+ server groups. (See Chapter 4, “Configuring AAA”.)

Configuring AAA

This chapter describes how to configure Authentication, Authorization, and Accounting (AAA) on Cisco 1000 Series Connected Grid Routers (hereafter referred to as the Cisco CG-OS router).

This chapter includes the following sections:

Information About AAA

This section includes the following topics:

AAA Security Services

The AAA feature allows you to verify the identity of, grant access to, and track the actions of users managing the Cisco CG-OS router. The Cisco CG-OS router supports Remote Access Dial-In User Service (RADIUS) or Terminal Access Controller Access Control device Plus (TACACS+) protocols.

Based on the user ID and password combination that you provide, the Cisco CG-OS router performs local authentication or authorization using the local database or remote authentication or authorization using one or more AAA servers. A pre-shared secret key provides security for communication between the Cisco CG-OS router and AAA servers. You can configure a common secret key for all AAA servers or for only a specific AAA server.

AAA security provides the following services:

- Authentication—Identifies users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption.

  Authentication is the process of verifying the identity of the person or device accessing the Cisco CG-OS router, which is based on the user ID and password combination provided by the entity trying to access the Cisco CG-OS router. The Cisco CG-OS routers allow you to perform local authentication (using the local lookup database) or remote authentication (using one or more RADIUS or TACACS+ servers).

- Authorization—Provides access control.

  AAA authorization is the process of assembling a set of attributes that describe what the user is authorized to perform. Authorization in Cisco CG-OS is provided by attributes that are downloaded from AAA servers. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights with the appropriate user.

- Accounting—Provides the method for collecting information, logging the information locally on the Cisco CG-OS router, and sending the information to the AAA server for billing, auditing, and reporting.

  The accounting feature tracks and maintains a log of every management session used to access the Cisco CG-OS router. You can use this information to generate reports for troubleshooting and auditing purposes. You can store accounting logs locally on the Cisco CG-OS router or send them to remote AAA servers.

Note Cisco CG-OS software supports authentication, authorization, and accounting independently. For example, you can configure authentication and authorization without configuring accounting.

Benefits of Using AAA

AAA provides the following benefits:

- Increased flexibility and control of access configuration
- Scalability
- Standardized authentication methods, such as RADIUS and TACACS+ security
- Multiple backup devices

Remote AAA Services

Remote AAA services provided through RADIUS and TACACS+ protocols have the following advantages over local AAA services on the Cisco CG-OS router:

- AAA servers are already deployed widely across enterprises and can be easily used for AAA services.

- It is more efficient to define and manage user attributes for Cisco CG-OS routers within centralized AAA servers, which can be a shared resource for multiple routers rather than configuring local AAA services on each Cisco CG-OS router independently. Additionally, AAA Server Groups can provide additional redundancy.

AAA Server Groups

You can specify remote AAA servers for authentication, authorization, and accounting by using server groups. A server group is a set of remote AAA servers that implement the same AAA protocol. The purpose of a server group is to provide for failover servers in case a remote AAA server fails to respond. If the first remote server in the group fails to respond, then the next remote server in the group is queried until one of the servers sends a response. If all the AAA servers in the server group fail to respond, then that server group option is considered a failure. When required, you can specify multiple server groups. If the Cisco CG-OS router encounters errors from the servers in the first group, it tries the servers in the next server group.

AAA Service Configuration Options

AAA configuration in the Cisco CG-OS router is service-based, which means that you can have separate AAA configurations for the following services:

- User Telnet or Secure Shell version 2 (SSHv2) login authentication
- Console login authentication
- User management session accounting

Table 4-1 provides the relevant CLI command for each AAA service configuration option.

Table 4-1 AAA Service Configuration Commands

AAA Service Configuration Option	Related Command
Telnet or SSH login	**aaa authentication login default**
Console login	**aaa authentication login console**
User session accounting	**aaa accounting default**

You can specify the following authentication methods for the AAA services:

- RADIUS server groups—Uses the global pool of RADIUS servers for authentication
- Specified server groups—Uses specified RADIUS or TACACS+ server groups for authentication
- Local—Uses the local username or password database for authentication
- None—Uses only the username

Note If the chosen authentication method employs all RADIUS servers, rather than a specific server group, the Cisco CG-OS router chooses the RADIUS server from the global pool of configured RADIUS servers, in the order of configuration. Servers from this global pool can also be configured within a RADIUS server group on the Cisco CG-OS router.

Table 4-2 shows the AAA authentication methods that you can configure for the AAA services.

Table 4-2 AAA Authentication Methods for AAA Services

AAA Service	AAA Methods
Console login authentication	Server groups, local, and none
User login authentication	Server groups, local, and none
User management session accounting	Server groups and local

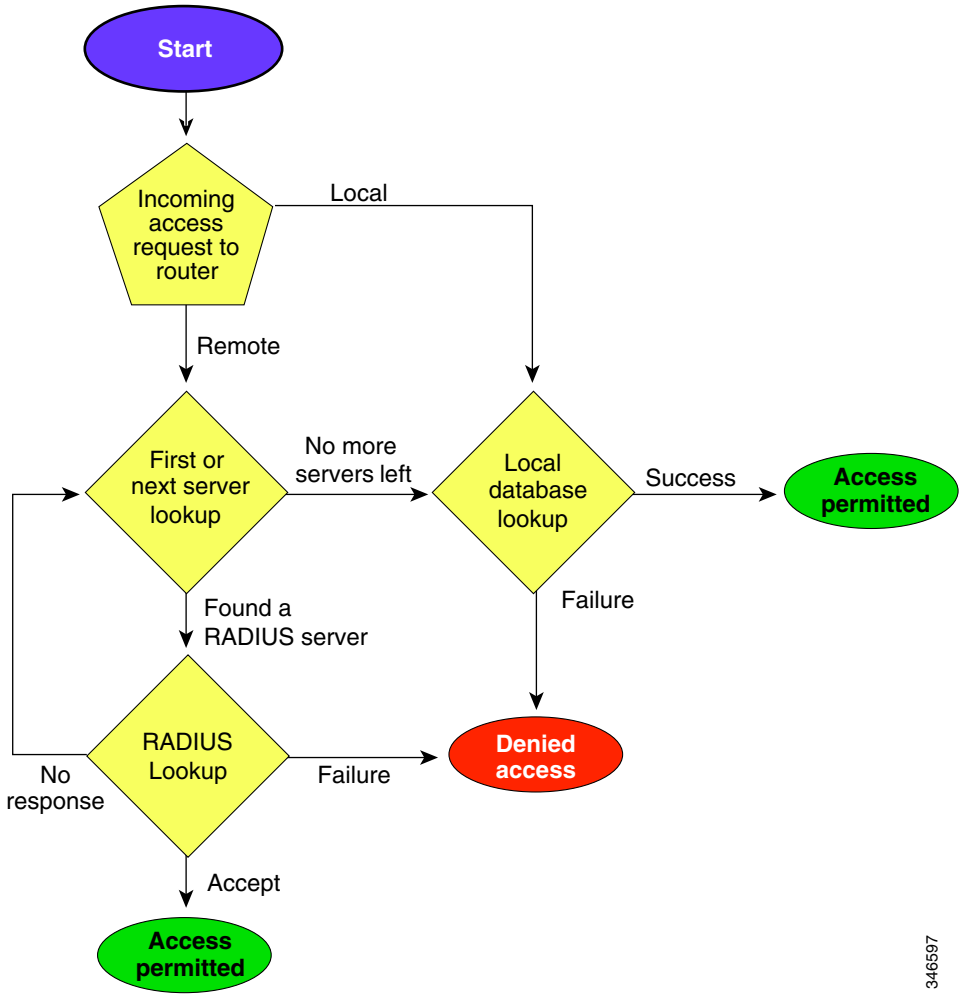
Note For console login authentication and user login authentication, and user management session accounting, the Cisco CG-OS router queries each option in the order specified. The local option is the default method when other configured options fail.

Authentication and Authorization Process for User Login

Figure 4-1 shows a flow chart of the authentication and authorization process for user login. The following list explains the process:

1. When you log in to one of the required Cisco CG-OS routers, you can use the Telnet, SSHv2, or console login options. Cisco recommends employing SSHv2 for increased security.
2. When you configure the AAA server groups using the server group authentication method, the Cisco CG-OS router sends an authentication request to the first AAA server in the group as follows:
   - If the AAA server fails to respond, then the Cisco CG-OS router queries the next AAA server and so on until a remote AAA server responds to the authentication request.
   - If all AAA servers in the server group fail to respond, then the Cisco CG-OS router contacts servers in the next server group.
   - If all configured methods fail, then the local database on the Cisco CG-OS router is used for authentication.
3. When the Cisco CG-OS router successfully authenticates through a remote AAA server, the following possibilities apply:
   - If the AAA server protocol is RADIUS, then the server downloads an authentication response to the Cisco CG-OS router that includes user roles, which are part of the cisco-av-pair attribute.
   - If the AAA server protocol is TACACS+, then the Cisco CG-OS router sends another request to the same server to get the user roles specified as custom attributes for the shell.
   - If the user roles are not successfully retrieved from the remote AAA server by the Cisco CG-OS router, then the Cisco CG-OS router assigns the user the *vdc-operator* role. For more information on user roles, refer to Chapter 7, "Configuring User Accounts and RBAC."
4. When the Cisco CG-OS router successfully authenticates your username and password, the Cisco CG-OS router logs you in and assigns you the roles configured in the local database.

Figure 4-1 Authorization and Authentication Flow for User Login



Note "No more servers left" means that there is no response from any server within available server groups.

Prerequisites for AAA

Ensure that at least one RADIUS or TACACS+ server is IP reachable. (See the Configuring RADIUS Servers, page 2-6 and Configuring TACACS+ Server Hosts, page 3-7.)

Ensure that the Cisco CG-OS router is recognized as a client of the AAA servers.

Ensure that you configure the pre-share secret key on the Cisco CG-OS router and the remote AAA servers.

Ensure that the remote server responds to AAA requests from the Cisco CG-OS router. (See Manually Monitoring RADIUS Server or Groups, page 2-16 and the Manually Monitoring TACACS+ Servers or Groups, page 3-16.)

Guidelines and Limitations for AAA

The Cisco CG-OS software does not support all-numeric usernames, whether created with TACACS+ or RADIUS, or created locally, and does not create local users with all-numeric names. When an all-numeric username exists on an AAA server and it is entered during login, the Cisco CG-OS router does not log in the user.

When you have a user account configured on a local Cisco CG-OS router that has the same name as a remote user account on an AAA server, Cisco CG-OS applies the user roles for the local user account to the remote user, instead of the user roles configured on the AAA server.

Default Settings

Table 4-3 lists the default settings for AAA parameters.

Table 4-3 Default AAA Parameters

Parameters	Default
Console authentication method	Local
Default authentication method	Local
Login authentication failure messages	Disabled
Default accounting method	Local
Accounting log display length	250 KB

Configuring AAA

This section includes the following topics:

Process for Configuring AAA

To configure AAA authentication and accounting, follow these steps:

Step 1 When you want to use remote RADIUS or TACACS+ servers for authentication, and to configure the hosts on your Cisco CG-OS router, refer to Chapter 2, “Configuring RADIUS” and Chapter 3, “Configuring TACACS+”).

Step 2 Enable the Default User Role for Authentication. (See Enabling the Default User Role for Authentication, page 4-8.)

Step 3 Enable the Login Authentication Failure Messages. (See Enabling Login Authentication Failure Messages, page 4-8.)

Step 4 Configure default login authentication methods for user logins. (See Configuring Default Login Authentication Methods, page 4-7.)

Step 5 Configure default AAA accounting default methods. (See Configuring AAA Accounting Default Methods, page 4-9.)

Configuring Default Login Authentication Methods

The authentication methods include the following:

- Global pool of RADIUS servers
- Named subset of RADIUS or TACACS+ servers
- Local database on the Cisco CG-OS router (default)

BEFORE YOU BEGIN

Configure RADIUS or TACACS+ server groups.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa authentication login default** {**group** *group-list* [**none**]| **local**	Configures the default authentication methods. *group-list*—Space-separated list of server groups that can include any configured RADIUS or TACACS+ server group name. **local—**Specifies the local database of the Cisco CG-OS router for authentication. **none**—Uses no authentication. The default login method is **local**, which the Cisco CG-OS router uses when no methods are configured or when all the configured methods fail to respond.
Step 3	**show aaa authentication**	(Optional) Displays the configuration of the default login authentication methods.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure default login authentication methods for the Cisco CG-OS router.

```
router# configure terminal
router(config)# aaa authentication login default group va_reston2
```

```
router(config)# copy running-config startup-config
```

Enabling the Default User Role for Authentication

You can enable the default user role that allows remote users who do not have a user role to log in to the Cisco CG-OS router through a RADIUS or TACACS+ server. The default user role on the Cisco CG-OS router is *network-operator*. For more information on user roles, see Chapter 7, "Configuring User Accounts and RBAC."

Note Although references to a default VDC might be seen in CLI displays, the Cisco CG-OS router does not support the configuration of more than one VDC. The Cisco CG-OS router only supports a default VDC.

BEFORE YOU BEGIN

Configure RADIUS or TACACS+ servers or server groups.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa user default-role**	Enables the default user role for AAA authentication. The default is enabled. You can disable the default user role feature by using the **no** form of this command.
Step 3	**show aaa user default-role**	(Optional) Displays the AAA default user role configuration as either enabled or disabled on the Cisco CG-OS router.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to enable the default user role of *network-operator* for remote authentication to the Cisco CG-OS router through a AAA (RADIUS or TACACS+) server.

```
router# configure terminal
router(config)# aaa user default-role
router(config)# copy running-config startup-config
```

Enabling Login Authentication Failure Messages

When you enable login failure messages on the Cisco CG-OS router, the following messages display when access to remote AAA servers fails and the local user database takes precedence:

```
Remote AAA servers unreachable; local authentication done
Remote AAA servers unreachable; local authentication failed
```

BEFORE YOU BEGIN

Configure RADIUS or TACACS+ servers or server groups.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa authentication login error-enable**	Enables login authentication failure messages. The default is disabled.
Step 3	**show aaa authentication login error-enable**	(Optional) Displays whether the login failure message configuration is enabled or disabled.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to enable authentication failure messages on the Cisco CG-OS router that will appear on a user (client) terminal when authentication with a RADIUS or TACACS+ server fails.

```
router# configure terminal
router(config)# aaa authentication login error-enable
router(config)# copy running-config startup-config
```

Configuring AAA Accounting Default Methods

The Cisco CG-OS router supports TACACS+ and RADIUS methods for accounting and reports user activity to TACACS+ or RADIUS security servers in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs, which are stored on the designated AAA server.

When you activate AAA accounting, the Cisco CG-OS router reports these attributes as accounting records, which are then stored in an accounting log on the defined AAA security server.

You can create default method lists defining specific accounting methods, which include the following:

- RADIUS server group—Specifies a global pool of RADIUS servers for accounting.
- Specified server group—Uses a specified RADIUS or TACACS+ server group for accounting.
- Local—Uses the local username or password database on the Cisco CG-OS router for accounting.

Note When you configure server groups and the server groups do not respond, by default, the local database on the Cisco CG-OS router is used for authentication.

BEFORE YOU BEGIN

Configure RADIUS or TACACS+ server groups.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa accounting default** {**group** *server-group-name* | **local**}	Configures the default accounting method. *server-group-name*– List the server groups on which you want to store accounting logs. **radius**–Uses the global pool of RADIUS servers for accounting. **local**– Uses the local database of the Cisco CG-OS router for accounting. The default method is **local**, which is used when you do not configure any options or when all the configured server groups fail to respond.
Step 3	**show aaa accounting**	(Optional) Displays the configured default AAA accounting method.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure the Cisco CG-OS router to use default accounting methods employed by RADIUS servers.

```
router# configure terminal
router(config)# aaa accounting default group va_reston3
router(config)# copy running-config startup-config
```

Using AAA Server VSAs

You can use Vendor-Specific Attributes (VSAs) to specify user roles on AAA servers.

This section includes the following topics:

About VSAs

The Internet Engineering Task Force (IETF) draft standard specifies attribute 26 as the method for communicating VSAs between the network access server and the RADIUS server. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Cisco attribute for a particular type of authorization, the separator is = (equal sign) for mandatory attributes, and * (asterisk) indicates optional attributes.

When you use RADIUS servers for authentication on the Cisco CG-OS router, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

VSA Format

Cisco CG-OS supports the following VSA protocol options:

- Shell—Protocol used in access-accept packets to provide user-profile information.
- Accounting—Protocol used in accounting-request packets. When a value contains any white spaces, put it within double quotation marks.

Cisco CG-OS supports the following attributes:

- roles—Lists all the roles assigned to the user. The value field is a string that stores the list of group names delimited by white space. For example, if you belong to roles network-operator, the value field would be “network-operator.” This subattribute is sent in the VSA portion of the Access-Accept frames from the RADIUS server, and it can only be used with the shell protocol value. These examples use the roles attribute:

  ```
  shell:roles=“network-operator vdc-admin”
  shell:roles*“network-operator” vdc-admin
  ```

  The following examples show the roles attribute as supported by FreeRADIUS:

  ```
  Cisco-AVPair = “shell:roles=\“network-operator vdc-admin\””
  Cisco-AVPair = “shell:roles*\“network-operator vdc-admin\””
  ```

Note When you specify a VSA as shell:roles*"network-operator" vdc-admin or "shell:roles*\"network-operator vdc-admin\"", this VSA is flagged as an optional attribute and other Cisco devices ignore this attribute.

- accountinginfo—Stores accounting information in addition to the attributes covered by the standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the Cisco CG-OS router, and it can only be used with the accounting protocol-related PDUs.

Specifying User Roles on AAA Servers

You can use the VSA cisco-av-pair on AAA servers to specify user role mapping for the Cisco CG-OS router using this format:

```
shell:roles="roleA roleB …"
```

If you do not specify the role option in the cisco-av-pair attribute, the default user role is network-operator.

For more information on user roles, see Chapter 7, “Configuring User Accounts and RBAC.”

Displaying and Clearing the Local AAA Accounting Log

The Cisco CG-OS router maintains a local log for the AAA accounting activity.

You can display the contents of the log or clear the contents of the log by entering one of the commands below:

Command	Purpose
show accounting log [*size* | **start-time** *year month day hh:mm:ss*]	Displays the contents of the AAA accounting log on the Cisco CG-OS router. *size*–Use to limit command output from the accounting log. The range is from 0 to 250000 bytes. By default, the command output contains up to 250000 bytes of the accounting log. **start-seqnum**–Specifies for the log output.**start-time**–Specifies
clear accounting log	Clears the contents of the AAA accounting log on the Cisco CG-OS router. Enter command at the EXEC level.

Note The AAA accounting log is local to the Cisco CG-OS router.

Verifying Configuration

To display AAA configuration information, enter any or all of the following commands:

Command	Purpose
show aaa accounting	Displays AAA accounting configuration.
show aaa authentication [**login error-enable**]	Indicates if the AAA authentication login error-enable option is enabled or disabled on the Cisco CG-OS router.
show aaa groups	Displays the AAA server group names configured on the Cisco CG-OS router.
show running-config aaa [all]	Displays the AAA configuration in the running configuration.
show startup-config aaa	Displays the AAA configuration in the startup configuration.

For detailed information about the fields in the output from these commands, see the Command Lookup Tool on Cisco.com.

Configuration Example

The following example shows how to configure AAA:

```
aaa authentication login default group va_reston2
aaa accounting default group va_reston3
```

Configuring SSHv2 and Telnet

This chapter describes how to configure Secure Shell Protocol version 2 (SSHv2) and Telnet on the Cisco1000 Series Connected Grid Routers (hereafter referred to as Cisco CG-OS router).

This chapter includes the following sections:

Information About SSHv2 and Telnet

This section includes the following topics:

SSHv2 Server

You can use the SSHv2 server to enable an SSH client to make a secure, encrypted connection to the Cisco CG-OS router. SSHv2 uses strong encryption for authentication. The SSHv2 server in the Cisco CG-OS software can interoperate with publicly and commercially available SSHv2 clients.

The user authentication mechanisms supported for SSHv2 are RADIUS, TACACS+, and the use of locally stored usernames and passwords on the Cisco CG-OS router.

SSHv2 Client

The SSHv2 client feature is an application that runs over the SSHv2 protocol to provide device authentication and encryption. The SSHv2 client enables the Cisco CG-OS router to make a secure, encrypted connection to any other device that runs the SSHv2 server. This connection provides an encrypted outbound connection. With authentication and encryption, the SSHv2 client allows for a secure communication over an insecure network.

The SSHv2 client in Cisco CG-OS works with publicly and commercially available SSHv2 servers.

SSHv2 Server Keys

SSHv2 requires server keys for secure communications to the Cisco CG-OS router. You can use SSHv2 server keys for the following SSHv2 options:

- SSHv2 version 2 using Rivest, Shamir, and Adelman (RSA) public-key cryptography
- SSHv2 version 2 using the Digital System Algorithm (DSA)

Be sure to have an SSHv2 server key-pair with the appropriate version before enabling the SSHv2 service. You can generate the SSHv2 server key-pair according to the SSHv2 client version used. The SSHv2 service accepts two types of key-pairs for use by SSHv2:

- The **dsa** option generates the DSA key-pair for the SSHv2 protocol.
- The **rsa** option generates the RSA key-pair for the SSHv2 protocol.

By default, Cisco CG-OS generates an RSA key using 1024 bits.

SSHv2 supports the following public key formats:

- OpenSSH
- IETF Secure Shell (SECSH)

Caution If you delete all of the SSHv2 keys, you cannot start the SSHv2 services.

SSHv2 Authentication Using Digital Certificates

SSHv2 authentication provides X.509 digital certificate support for host authentication. An X.509 digital certificate is a data item that ensures the origin and integrity of a message. It contains encryption keys for secured communications and is signed by a trusted certification authority (CA) to verify the identity of the presenter. The X.509 digital certificate support provides either DSA or RSA algorithms for authentication.

The certificate infrastructure uses the first certificate that supports the Secure Socket Layer (SSL) and is returned by the security infrastructure, either through query or notification. Verification of certificates is successful if the certificates are from any of the trusted CAs.

You can configure your device for either SSHv2 authentication using an X.509 certificate or SSHv2 authentication using a Public Key Certificate, but not both. If either of them is configured and the authentication fails, you are prompted for a password.

For more information on CAs and digital certificates, see Chapter 6, “Configuring PKI.”

Telnet Server

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site and then passes the keystrokes from one device to the other. Telnet can accept either an IP address or a domain name as the remote device address.

By default, the Telnet server is disabled on the Cisco CG-OS router.

Prerequisites

Configure IP on a Layer 3 interface, out-of-band on the mgmt 0 interface, or inband on an Ethernet interface.

Guidelines and Limitations

Cisco CG-OS supports only SSH version 2 (SSHv2).

You can configure the Cisco CG-OS router with either SSHv2 authentication using an X.509 certificate or SSHv2 authentication using a Public Key Certificate, but not both. Regardless of which authentication method is in use, the Cisco CG-OS router prompts the user for a password when authentication fails.

Cisco CG-OS supports a maximum of 60 concurrent SSHv2 and Telnet sessions.

Default Settings

Table 5-1 lists the default settings for SSHv2 and Telnet parameters.

Table 5-1 Default SSHv2 and Telnet Parameters

Parameters	Default
SSHv2 server	Enabled
SSHv2 server key	RSA key generated with 1024 bits
RSA key bits for generation	1024
Telnet server	Disabled
Telnet port number	23

Configuring SSHv2

This section includes the following sections:

- Clearing SSHv2 Sessions, page 5-8

Generating SSHv2 Server Keys

You can generate an SSHv2 server key based on your security requirements. The default SSHv2 server key is an RSA key that the Cisco CG-OS router generates using 1024 bits.

BEFORE YOU BEGIN

Ensure that you have met the prerequisites for SSHv2 summarized under Prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**no feature ssh**	Disables SSHv2. By default, SSHv2 is enabled on the Cisco CG-OS router.
Step 3	**ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}	Generates the SSHv2 server key. **dsa–**generates the DSA key-pair for the SSHv2 protocol. **rsa–**generates the RSA key-pair for the SSHv2 protocol. (Optional) The *bits* argument is the number of bits used to generate the key. The range is from 768 to 2048 and the default value is 1024. (Optional) Use the **force** keyword to replace an existing key.
Step 4	**feature ssh**	Enables SSHv2.
Step 5	**show ssh key**	(Optional) Displays the SSHv2 server keys.
Step 6	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to generate a SSHv2 server key on the Cisco CG-OS router.

```
router# configure terminal
router(config)# no feature ssh
router(config)# ssh key rsa 2048
router(config)# feature ssh
router(config)# copy running-config startup-config
```

Specifying the SSHv2 Public Keys for User Accounts

You can configure an SSHv2 public key to log in using an SSHv2 client without being prompted for a password. You can specify the SSHv2 public key in one of these formats:

- OpenSSH format
- IETF SECSH format

Specifying the SSHv2 Public Keys in OpenSSH Format

You can specify the SSHv2 public keys in OpenSSH format for user accounts.

BEFORE YOU BEGIN

Generate an SSHv2 public key in OpenSSH format.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**username** *username* **sshkey** *ssh-key*	Configures the SSHv2 public key in OpenSSH format.
Step 3	**show user-account**	(Optional) Displays the user account configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to specify SSHv2 public keys for user accounts.

```
router# configure terminal
router(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50rv7gsEPjhOBYmsi6PAVKui1nIf/
DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQW3g9igG30c6k6+XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH
3UD/vKyziEh5S4Tplx8=
router(config)# copy running-config startup-config
```

Specifying the SSHv2 Public Keys in IETF SECSH Format

You can specify the SSHv2 public keys in IETF SECSH format for user accounts.

BEFORE YOU BEGIN

Generate an SSHv2 public key in IETF SCHSH format.

DETAILED STEPS

	Command	Purpose
Step 1	**copy** *server-file* **bootflash:***filename*	Downloads the file containing the SSHv2 key in IETF SECSH format from a server. The server can be FTP, secure copy (SCP), secure FTP (SFTP), or TFTP. **Note** Only a network-admin or vdc-admin can perform this task.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**username** *username* **sshkey file bootflash:***filename*	Configures the SSHv2 public key in IETF SECSH format.
Step 4	**show user-account**	(Optional) Displays the user account configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to specify the SSHv2 public keys in IETF SECSH format.

```
router# copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub
router(config)# configure terminal
router(config)# username User1 sshkey file bootflash:secsh_file.pub
router(config)# copy running-config startup-config
```

Starting SSHv2 Sessions

You can start SSHv2 sessions using IPv4 or IPv6 to connect to remote devices from the Cisco CG-OS router.

Note Cisco CG-OS supports a maximum of 60 concurrent SSHv2 and Telnet sessions.

BEFORE YOU BEGIN

Obtain the hostname for the remote device and, if needed, the username on the remote device.

Enable the SSHv2 server on the remote device.

DETAILED STEPS

	Command	Purpose
Step 1	**ssh** [*username*@]{*ipv4-address* | *hostname*}	Creates an SSHv2 IPv4 session to a remote device using IPv4.
	ssh6 [*username*@]{*ipv6-address* | *hostname*}	Creates an SSHv2 IPv6 session to a remote device using IPv6.

EXAMPLE

This example shows how to create an SSHv2 IPv4 session to a remote device.

```
router# ssh 10.10.1.1
```

This example shows how to how to create an SSHv2 IPv6 session to a remote device.

```
router# ssh6 HostA
```

Clearing SSHv2 Hosts

When you download a file from a server using SCP or SFTP, or when you start an SSHv2 session from the Cisco CG-OS router to a remote host, you establish a trusted SSHv2 relationship with that server. You can clear the list of trusted SSHv2 servers for your user account.

Command	Purpose
clear ssh hosts	Clears the SSHv2 host sessions.

Disabling the SSHv2 Server

By default, the SSHv2 server is enabled on the Cisco CG-OS router. You can disable the SSHv2 server to prevent SSHv2 access to the Cisco CG-OS router.

Command	Purpose
no feature ssh	Disables SSH.

Note To reenable SSHv2, you must first generate an SSHv2 server key. (See Generating SSHv2 Server Keys, page 5-4.)

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**no feature ssh**	Disables the SSHv2 server. Feature is enabled by default.
Step 3	**show ssh server**	(Optional) Displays the SSHv2 server configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to disable SSHv2 on the Cisco CG-OS router.

```
router# configure terminal
router(config)# no feature ssh
router(config)# copy running-config startup-config
```

Deleting SSHv2 Server Keys

BEFORE YOU BEGIN

Disable the SSHv2 server. (See Disabling the SSHv2 Server, page 5-7.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**no feature ssh**	Disables the SSHv2 server.
Step 3	**no ssh key** [**dsa** | **rsa**]	Deletes the SSHv2 server key. The default is to delete all the SSHv2 keys.
Step 4	**show ssh key**	(Optional) Displays the SSHv2 server key configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to delete SSHv2 server keys.

```
router# configure terminal
router(config)# no feature ssh
router(config)# no ssh key rsa
router(config)# copy running-config startup-config
```

Clearing SSHv2 Sessions

You can clear SSHv2 sessions from the Cisco CG-OS router.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**show users**	Displays user session information.
Step 2	**clear line** *vty-line*	Clears a user SSHv2 session. *vty-line*–virtual terminal line.

EXAMPLE

```
router# configure terminal
router(config)# show users
router(config)# clear line pts/12
```

Configuring Telnet

This section includes the following topics:

Enabling the Telnet Server

You can enable the Telnet server on the Cisco CG-OS router. By default, the Telnet server is disabled.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**feature telnet**	Enables the Telnet server. The default is disabled.
Step 3	**show telnet server**	(Optional) Displays the Telnet server configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to enable Telnet on the Cisco CG-OS router.

```
router# configure terminal
router(config)# feature telnet
```

```
router(config)# copy running-config startup-config
```

Starting Telnet Sessions to Remote Devices

You can start Telnet sessions to connect to remote devices from the Cisco CG-OS router. You can start Telnet sessions by using either IPv4 or IPv6.

Note Cisco CG-OS supports a maximum of 60 concurrent SSHv2 and Telnet sessions.

BEFORE YOU BEGIN

Obtain the hostname or IP address for the remote device and, if needed, the username on the remote device.

Enable the Telnet server on the Cisco CG-OS router. (See Enabling the Telnet Server, page 5-9.)

Enable the Telnet server on the remote device.

DETAILED STEPS

	Command	Purpose
Step 1	**telnet** {*ipv4-address* | *host-name*} [*port-number*]	Starts a Telnet session to a remote device using IPv4. The default port number is 23. The range is from 1 to 65535.
	telnet6 {*ipv6-address* | *host-name*} [*port-number*]	Starts a Telnet session to a remote device using IPv6. The default port number is 23. The range is from 1 to 65535.

EXAMPLE

This example shows how to configure a Telnet session to a remote device that is using IPv4.

```
router# telnet 10.10.1.1
```

This example shows how to configure a Telnet session to a remote device that is using IPv6.

```
router# telnet 2001:0DB8::ABCD:1 management
```

Clearing Telnet Sessions

You can clear Telnet sessions from the Cisco CG-OS router.

BEFORE YOU BEGIN

Telnet server must be enabled on the Cisco CG-OS router.

DETAILED STEPS

	Command	Purpose
Step 1	**show users**	Displays user session information.
Step 2	**clear line** *vty-line*	Clears a user Telnet session.

EXAMPLE

This example shows how to clear a Telnet session.

```
router# show users
router(config)# clear line pts/12
```

Verifying the SSHv2 and Telnet Configuration

To display the SSHv2 and Telnet configuration information, enter any or all of the following commands:

Command	Purpose
show ssh key [dsa | rsa]	Displays SSHv2 server key-pair information.
show running-config security [**all**]	Displays the SSHv2 and user account configuration in the running configuration. The **all** keyword displays the default values for the SSHv2 and user accounts.
show ssh server	Displays the SSHv2 server configuration.
show telnet server	Displays the Telnet server configuration.

Configuration Example

```
configure terminal
  no feature ssh
  ssh key rsa
    generating rsa key(1024 bits).....
    generated rsa key
  feature ssh
  show ssh key
    rsa Keys generated: Tues Jan 29 00:10:39 2013

    ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAvWhEBsF55oaPHNDBnpXOTw6+/OdHoLJZKr+MZm99n2U0
    ChzZG4svRWmHuJY4PeDWl0e5yE3g3EO3pjDDmt923siNiv5aSga60K36lr39HmXL6VgpRVn1XQFiBwn4
    na+H1d3Q0hDt+uWEA0tka2uOtXlDhliEmn4HVXOjGhFhoNE=

    bitcount:1024
    fingerprint:
    51:6d:de:1c:c3:29:50:88:df:cc:95:f0:15:5d:9a:df
    ************************************
    could not retrieve dsa key information
    ************************************
```

```
  username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50rv7gsEPjhOBYmsi6PAVKui1nIf/
DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQW3g9igG30c6k6+XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH
3UD/vKyziEh5S4Tplx8=

  copy running-config startup-config
```

Configuring PKI

This chapter describes the Public Key Infrastructure (PKI) support on the Cisco 1000 Series Connected Grid Routers (hereafter referred to as the Cisco CG-OS router). PKI allows the Cisco CG-OS router to obtain and use digital certificates for secure communication in the network.

This chapter includes the following sections:

Information About PKI

This section provides information about PKI and includes the following topics:

CAs and Digital Certificates

Certificate authorities (CAs) manage certificate requests and issue certificates to participating entities such as hosts, network devices, or users. The CAs provide centralized key management for the participating entities.

Digital signatures, based on public key cryptography, digitally authenticate devices and individual users. In public key cryptography, such as the RSA encryption system, each device or user has a key-pair that contains both a private key and a public key. The private key is kept secret and is known only to the owning device or user. However, the public key is known to everybody. Anything encrypted with one of the keys can be de-encrypted with the other. A signature is formed when data is encrypted with a sender's private key. The receiver verifies the signature by de-encrypting the message with the sender's public key. This process relies on the receiver having a copy of the sender's public key and knowing with a high degree of certainty that it really does belong to the sender and not to someone pretending to be the sender.

Digital certificates link the digital signature to the sender. A digital certificate contains information to identify a user or device, such as the name, serial number, company, department, or IP address. It also contains a copy of the entity's public key. The CA that signs the certificate is a third party that the receiver explicitly trusts to validate identities and to create digital certificates.

To validate the signature of the CA, the receiver must first know the CA's public key. Typically this process is handled out of band or through an operation done at installation. For instance, most web browsers are configured with the public keys of several CAs by default.

Trust Model, Trustpoints, and Identity CAs

The PKI trust model is hierarchical with multiple configurable trusted CAs. You can configure each participating device with a list of trusted CAs so that a peer certificate obtained during the security protocol exchanges can be authenticated if it was issued by one of the locally trusted CAs. The Cisco CG-OS software locally stores the self-signed root certificate of the trusted CA (or certificate chain for a subordinate CA). The process of securely obtaining a trusted CA root certificate (or the entire chain in the case of a subordinate CA) and storing it locally is called *CA authentication*.

The information about a trusted CA that you have configured is called the *trustpoint* and the CA itself is called a *trustpoint CA*. This information consists of a CA certificate (or certificate chain in case of a subordinate CA) and certificate revocation checking information.

The Cisco CG-OS router can also enroll with a trustpoint to obtain an identity certificate to associate with a key-pair. This trustpoint is called an *identity CA*.

RSA Key-Pairs and Identity Certificates

You can obtain an identity certificate by generating one or more RSA key-pairs and associating each RSA key-pair with a trustpoint CA where the Cisco CG-OS router intends to enroll. The Cisco CG-OS router needs only one identity per CA, which consists of one key-pair and one identity certificate per CA.

The Cisco CG-OS software allows you to generate RSA key-pairs with a configurable key size (or modulus). The default key size is 2048 bits. You can also configure an RSA key-pair label. The default key label is the device fully qualified domain name (FQDN).

The following list summarizes the relationship between trustpoints, RSA key-pairs, and identity certificates:

- A trustpoint corresponds to a specific CA that the Cisco CG-OS router trusts for peer certificate verification for any application.

- A Cisco CG-OS router can have many trustpoints and all applications on the device can trust a peer certificate issued by any of the trustpoint CAs.
- A trustpoint is not restricted to a specific application.
- A Cisco CG-OS router enrolls with the CA that corresponds to the trustpoint to obtain an identity certificate. You can enroll your Cisco CG-OS router with multiple trustpoints, which means that you can obtain a separate identity certificate from each trustpoint. Applications employ the identity certificates as determined by those purposes specified in the certificate by the issuing CA. The purpose of a certificate is stored in the certificate as a certificate extension.
- When enrolling with a trustpoint, you must specify a RSA key-pair to be certified. This key-pair must be generated and associated to the trustpoint before generating the enrollment request. The association between the trustpoint, key-pair, and identity certificate is valid until it is explicitly removed by deleting the certificate, key-pair, or trustpoint.
- The subject name in the identity certificate is the fully qualified domain name (FQDN) for the Cisco CG-OS router.
- You can generate one or more RSA key-pairs on a device and each can be associated to one or more trustpoints. But no more than one key-pair can be associated to a trustpoint, which means only one identity certificate is allowed from a CA.
- You do not need more than one identity certificate from a trustpoint or more than one key-pair to be associated to a trustpoint. A CA certifies a given identity (or name) only once and does not issue multiple certificates with the same name. If you need more than one identity certificate for a CA and if the CA allows multiple certificates with the same names, you must define another trustpoint for the same CA, associate another key-pair to it, and have it certified.

Multiple Trusted CA Support

The Cisco CG-OS router can trust multiple CAs by configuring multiple trustpoints and associating each with a distinct CA. With multiple trusted CAs, you do not have to enroll a device with the specific CA that issued the certificate to a peer. Instead, you can configure the device with multiple trusted CAs that the peer trusts. The Cisco CG-OS router can then use a configured trusted CA to verify certificates received from a peer that were not issued by the same CA defined in the identity of the peer device.

PKI Enrollment Support

Enrollment is the process of obtaining an identity certificate for the device that is used for applications, in this case the Cisco CG-OS router, and the certificate authority (CA).

Cisco recommends that you employ an intermediate router such as the Cisco 3945 Integrated Services Router (Cisco ISR) as the Registration Authority (functioning as a CA proxy) for obtaining certificates for the Cisco CG-OS router from the CA.

The Cisco CG-OS router performs the following steps when performing the PKI enrollment process:

1. Generates an RSA private and public key-pair.
2. Generates a certificate request in standard format and forwards it to the CA.

Note The CA administrator might be required to manually approve the enrollment request at the CA server when the request is received by the CA.

3. Receives the issued certificate back from the CA, signed with the private key of the CA.
4. Writes the certificate into a nonvolatile storage area on the Cisco CG-OS router.

Multiple RSA Key-Pair and Identity CA Support

Multiple identity CAs enable the Cisco CG-OS router to enroll with more than one trustpoint, which results in multiple identity certificates, each from a distinct CA. With this feature, the Cisco CG-OS router can participate in applications with many peers using certificates issued by CAs that are acceptable to those peers.

The multiple RSA key-pair feature allows the Cisco CG-OS router to maintain a distinct key-pair for each CA with which it is enrolled. It can match policy requirements for each CA without conflicting with the requirements specified by the other CAs, such as the key length. The Cisco CG-OS router can generate multiple RSA key-pairs and associate each key-pair with a distinct trustpoint. Thereafter, when enrolling with a trustpoint, the associated key-pair is used to construct the certificate request.

Peer Certificate Verification

PKI support on a Cisco CG-OS router can verify peer certificates. The Cisco CG-OS software verifies certificates received from peers during security exchanges for applications. The applications verify the validity of the peer certificates. The Cisco CG-OS software performs the following steps when verifying peer certificates:

1. Verifies that the peer certificate is issued by one of the locally-trusted CAs.
2. Verifies that the peer certificate is valid (not expired) with respect to current time.

Import and Export Support for Certificates and Associated Key-Pairs

As part of the CA authentication and enrollment process, the subordinate CA certificate (or certificate chain) and identity certificates can be imported in standard PEM (base64) format.

The complete identity information in a trustpoint can be exported to a file in the password-protected PKCS#12 standard format. It can be later imported to the same device (for example, after a system crash) or to a replacement device. The information in a PKCS#12 file consists of the RSA key-pair, the identity certificate, and the CA certificate (or chain).

Prerequisites

You must configure the Registration Authority (RA) to proxy for the CA server before you configure the Cisco CG-OS router. (See Configuring the Registration Authority, page 6-11.)

Guidelines and Limitations

The maximum number of key-pairs you can configure on the Cisco CG-OS router is 16.

The maximum number of trustpoints you can declare on the Cisco CG-OS router is 16.

The maximum number of identify certificates you can configure on the Cisco CG-OS router is 16.

The maximum number of certificates in a CA certificate chain is 10.

The maximum number of trustpoints you can authenticate to a specific CA is 10.

When generating certificates for the Cisco CG-OS router, a different RSA key-pair must be defined for the registration authority (RA) and the certification authority (CA).

Default Settings

Table 6-1 lists the default settings for PKI parameters.

Table 6-1 Default PKI Parameters

Parameters	Default
trustpoint	None
RSA key-pair	None
RSA key-pair label	Device FQDN
RSA key-pair modulus	2048
RSA key-pair exportable	Disabled

Configuring Certificate Enrollment

The Cisco CG-OS router supports the following types of certificate enrollment:

- Simple Certificate Enrollment Protocol (SCEP)–Allows automatic enrollment of the certificates on the Cisco CG-OS router without user intervention. (See Auto Enrollment Using SCEP, page 6-5.)
- Cut-and-paste enrollment–Supports manual enrollment between the Cisco CG-OS router and CA and requires that a user manually cut-and-paste the certificate requests and resulting certificates to manage the enrollment steps between the Cisco CG-OS router and the CA. (See Manual Enrollment, page 6-17.)
- Self-signed certificate–Allows the Cisco CG-OS router to create its own self-signed certificate. (See Configuring Self-Signed Certificates on the Cisco CG-OS Router, page 6-23.)
- Importing identity information in PKCS#12 format–Imports the certificate and RSA key-pair into the Cisco CG-OS router. (See Importing Identity Information in PKCS#12 Format, page 6-25.)

Auto Enrollment Using SCEP

This section describes the process of configuring the Cisco CG-OS router to communicate and exchange certificates with a Windows CA server to allow automatic enrollment of certificates.

Additionally, this section provides details on how to configure a Cisco ISR to serve as Registration Authority (RA) and proxy for the Windows CA server (which is the Cisco recommended configuration). This section does not provide details on configuring the Windows CA server.

This section includes the following topics:

- Configuring the Cisco CG-OS Router, page 6-6
- Configuring the Registration Authority, page 6-11

Configuring the Cisco CG-OS Router

This section includes the following topics:

Configuring the Cisco CG-OS Router Hostname and IP Domain Name

You must configure the hostname and IP domain name of the Cisco CG-OS router if you have not yet configured it because the Cisco CG-OS software uses the fully qualified domain name (FQDN) of the Cisco CG-OS router as the subject in the identity certificate. Additionally, the Cisco CG-OS software uses the device FQDN as a default key label when you do not specify a label during key-pair generation. For example, a certificate named DeviceA.example.com is based on a device hostname of DeviceA and a device IP domain name of example.com.

You must configure the hostname and IP domain name for both the Cisco CG-OS router and the Registration Authority.

Caution Changing the hostname or IP domain name after generating the certificate can invalidate the certificate.

BEFORE YOU BEGIN

Confirm the IP domain name and hostname with the system administrator.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**hostname** *hostname*	Configures the hostname of the Cisco CG-OS router.
Step 3	**ip domain-name** *name*	Configures the IP domain name of the Cisco CG-OS router.
Step 4	**show hosts**	(Optional) Displays the IP domain name.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure the hostname and IP domain name for the Cisco CG-OS router.

```
router# configure terminal
router(config)# hostname router_cgr01
router_cgr01(config)# ip domain-name yourcompany.com
router_cgr01(config)# copy running-config startup-config
```

Caution You must configure the Registration Authority **before** configuring auto-enrollment on the Cisco CG-OS router. (See Configuring the Registration Authority, page 6-11.) After configuring the Registration Authority (RA), you can then continue configuring the Cisco CG-OS router.

Creating an Enrollment Profile on the Cisco CG-OS Router

Specifies the use of a RA as the trustpoint source for the Cisco CG-OS router and the system that authenticates the certificate for the Cisco CG-OS router.

BEFORE YOU BEGIN

Configure the router acting as the RA. (See Configuring the Registration Authority, page 6-11.)

Configure the server acting as the CA. (See your Windows server manual.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 1	**crypto ca profile enrollment** *profile_name*	Creates or updates an existing enrollment profile.
Step 2	**enrollment url** *url*	Defines the URL of the RA that serves as CA proxy and enters the enrollment profile configuration submode. The RA must be directly connected to the Cisco CG-OS router.
Step 3	**enrollment credential {IDevID | trustpoint** *trustpoint*}	(Optional) Specifies the method of authentication identity from the existing trustpoint. When employing this command, you must use an existing trustpoint. *trustpoint*–Identifies the name of the trustpoint. IDev ID–IEEE 802.11AR security device identity assigned by Cisco. Cisco CG-OS router locates the value when IDevID is entered for this command.
Step 4	**exit**	Exits the enrollment profile configuration mode and returns the Cisco CG-OS router to the global configuration mode.

EXAMPLE

This example shows how to create an enrollment profile for the RA on the Cisco CG-OS router.

```
router_cgr01# configure terminal
router_cgr01(config)# crypto ca profile enrollment IOS_CA_RA_Profile
router_cgr01(config-enroll-profile)# enrollment url http://192.168.20.16
router_cgr01(config-enroll-profile)# enrollment credential trustpoint blueCA
router_cgr01(config-enroll-profile)# exit
router_cgr01(config)#
```

Generating an RSA Public and Private Key-Pair on the Cisco CG-OS Router

Tip When you do not configure the RSA public and private key-pair, the Cisco CG-OS router automatically generates the key-pair with a default length of 2048 bits. In this case, the key-pair is non-exportable and the PKS#12 format cannot be used for backup and restore. If you want to set a default length other than 2048 bits and want to have an exportable key-pair, follow the steps in this section.

The Cisco CG-OS router can generate RSA key-pairs to sign and/or encrypt and de-encrypt the security payload during security protocol exchanges for applications. The RSA key-pair must be generated for the Cisco CG-OS router before obtaining a certificate for the Cisco CG-OS router.

BEFORE YOU BEGIN

Create the enrollment profile. (See Creating an Enrollment Profile on the Cisco CG-OS Router, page 6-7.)

DETAILED STEPS

	Command	Purpose
Step 1	**crypto key generate rsa** [**label** *label-string*] [**exportable**] [**modulus** *size*]	Generates a RSA key-pair. The maximum number of key-pairs on a device is 16. *label-string*–An alphanumeric, case sensitive string with a maximum length of 64 characters. The default label string is the hostname and the FQDN separated by a period character (.). *size*–Values are 1024, 1536, and 2048. The default modulus size is 2048 bits. **Note** Cisco recommends that you use the 2048 setting. **Note** The security policy on the Cisco CG-OS router and on the CA (where enrollment is planned) must be considered when deciding the appropriate key modulus. By default, the key-pair is not exportable when the default value of 2048 is in use. Only exportable key-pairs can be exported in the PKCS#12 format. **Caution** You cannot change the exportability of a key-pair.
Step 2	**show crypto key mypubkey rsa**	(Optional) Displays the generated key.
Step 3	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to generate a RSA key-pair on the Cisco CG-OS router.

```
router_cgr01(config)# crypto key generate rsa label IOS_CA_RA_Key modulus 2048
router_cgr01(config)# copy running-config startup-config
```

Creating a Trustpoint on the Cisco CG-OS Router

Defines the trustpoint for all services requiring secure communications. This trustpoint will be used by the Cisco CG-OS router to obtain its certificates from the RA.

BEFORE YOU BEGIN

Configure the RA. (See Configuring the Registration Authority, page 6-11.)

Generate the key-pair for the Cisco CG-OS router. (See Generating an RSA Public and Private Key-Pair on the Cisco CG-OS Router, page 6-8.)

DETAILED STEPS

	Command	Purpose
Step 1	**crypto ca trustpoint** *name*	Declares a trustpoint that the Cisco CG-OS router can trust and enters trustpoint configuration mode. *name*–Alphanumeric, case sensitive, string with a maximum length of 64 characters. **Note** The maximum number of trustpoints that you can configure on the Cisco CG-OS router is 16.
Step 2	**enrollment profile** *name*	Ensures that the RA requests the RA mode Certificate Service (CS) certificate from the CA server.
Step 3	**rsakeypair** *rsa-keypair-label*	Enter the key-pair name generated for the Cisco CG-OS router and RA. (See Generating a RSA Public and Private Key-Pair on the RA, page 6-14.)
Step 4	**revocation-check none**	Invalidates revocation of compromised certificates.
Step 5	**serial-number**	Includes the serial number of the Cisco CG-OS router in the certificate. **Note** This command is only applicable to SCEP auto-enrollment.
Step 6	**ip-address** *ip_address*	Configures the IP address of the Cisco CG-OS router that is included in the certificate request.
Step 7	**subject-alt-name** *name*	Configures an additional user to be defined in the certificate request during enrollment. *name*–Limit of 512 characters.
Step 8	**enrollment retry count** *retry-count*	Defines the number of times that the Cisco CG-OS router attempts to contact the RA for CA authentication and enrollment before reporting a failed enrollment. *retry-count*–Range of values is 1 to 10. Default value is 3.
Step 9	**enrollment retry period** *retry-period*	Defines the period of time (in seconds) between the retry attempts of the Cisco CG-OS router to contact the RA for CA authentication. *retry-period*–Range of values is 1 to 10 seconds. Default value is 5 seconds.

	Command	Purpose
Step 10	**fingerprint** *hex-data*	Configures the expected thumbprint of the CA server certificate. **Note** Thumbprint information is found in the **Certificate > Details** window of the Windows CA Server. Matching is performed during CA authentication and enrollment without the need for user intervention. **Note** The Cisco CG-OS router only supports SHA1 fingerprints.
Step 11	**exit**	Exits the trustpoint configuration mode and returns the Cisco CG-OS router to the global configuration mode.

EXAMPLE

This example shows how to create a trustpoint for the Cisco CG-OS router.

```
router_cgr01(config)# crypto ca trustpoint IOS_CA_RA
router_cgr01(config-trustpoint)# enrollment profile IOS-CA_Profile
router_cgr01(config-trustpoint)# rsakeypair IOS_CA_RAKey serial-number
router_cgr01(config-trustpoint)# ip address 192.168.200.40
router_cgr01(config-trustpoint)# subject-alt name jsmith@anycompany.com
router_cgr01(config-trustpoint)# enrollment retry count 10
router_cgr01(config-trustpoint)# enrollment retry period 5
router_cgr01(config-trustpoint)# fingerprint
23:65:E8:DA:D9::06:FD:32:4C:18:78:2B:8D:06:6F:B9:67:C1:09:91
router_cgr01(config-trustpoint)# exit
router_cgr01(config)#
```

Authenticating the CA on the Cisco CG-OS Router

BEFORE YOU BEGIN

Configure the RA. (See Configuring the Registration Authority, page 6-11.)

Generate the key-pair for the Cisco CG-OS router. (See Generating an RSA Public and Private Key-Pair on the Cisco CG-OS Router, page 6-8.)

Create a trustpoint for the Cisco CG-OS router. (See Creating a Trustpoint on the Cisco CG-OS Router, page 6-9.)

Enroll the Cisco CG-OS router with the RA Serving as CA Proxy. (See Configuring the Registration Authority, page 6-11.)

DETAILED STEPS

	Command	Purpose
Step 1	**crypto ca authenticate** *name*	Allows the Cisco CG-OS router to authenticate with RA as the proxy for the CA server and receive its certificates.
Step 2	**show crypto ca certificates** *name*	Displays the name of the trustpoint, certificate start and expiration dates and fingerprint.

EXAMPLE

This example shows how to configure the Cisco CG-OS router to authenticate with the CA server and its certificates.

```
router_cgr01(config)# crypto ca authenticate IOS_CA_RA
Trustpoint CA authentication in progress. Please wait for a response...
router_cgr01(config)# 2013 Jan 29 11:27:57 router_cgr01 %$ VDC-1 %$
%CERT_ENROLL-2-CERT_EN_SCEP_CA_AUTHENTICATE_OK: Trustpoint IOS_CA_RA: CA certificates(s)
authenticated.
```

Enrolling the Cisco CG-OS Router to the CA

To enroll the Cisco CG-OS Router to the CA, enter the following command.

Command	Purpose
crypto ca enroll *trustpoint-label*	Identifies the name of the trustpoint.

Configuring the Registration Authority

The RA proxies as a CA server on behalf of the Cisco CG-OS router to obtain its certificates from the CA server.

Tip This section provides the tasks necessary to configure a Cisco ISR to serve as the Registration Authority (RA). If you already have a RA configured or are going to use a different system for the RA, then you do not need to complete the tasks in this section.

Caution You must configure a RA **before** configuring the Cisco CG-OS router.

Note For more information on the Cisco ISR, refer to the following URL: http://www.cisco.com/en/US/products/ps10536/index.html

Tip The Cisco ISR (recommended system for RA) operates with Cisco IOS rather than the Cisco CG-OS software so the command syntax differs for some configurations.

This section includes the following topics:

Configuring the RA Hostname and IP Domain Name

You must configure the hostname and IP domain name of the RA router if it is not yet configured.

Caution Changing the hostname or IP domain name after generating the certificate can invalidate the certificate.

BEFORE YOU BEGIN

Confirm the IP domain name and hostname of the RA with the system administrator.

DETAILED STEPS

To configure the hostname and IP domain name for the Cisco ISR using Cisco IOS, follow these steps:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**hostname** *hostname*	Configures the hostname of the RA.
Step 3	**enable secret** *password*	Specifies an encrypted password to prevent unauthorized access to the RA.
Step 4	**ip domain-name** *name*	Defines a default domain name that the RA uses to complete unqualified hostnames (such as those without a dotted-decimal domain name).
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

```
router# configure terminal
router(config)# hostname IOS_CA_RA
IOS_CA_RA(config)# enable secret No$AcceSs
IOS_CA_RA(config)# ip domain-name yourcompany.com
IOS_CA_RA(config)# copy running-config startup-config
```

Configuring the RA as Proxy for the CA Server

Configures the RA to acts as a proxy for the CA server on behalf of the Cisco CG-OS router.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**crypto pki server** *name*	Configures the RA as a CA server and RA for the third-party CA server. The name assigned to the RA must match the trustpoint name and the RSA key-pair assigned to the Cisco CG-OS router.

	Command	Purpose
Step 3	**database level {minimal | names | complete}**	Controls what type of data is stored in the certificate enrollment database. **minimal**–Stores enough information in the database to continue issuing new certificates without conflict. Default setting. **names**–Stores the serial number and subject name of each certificate in the database, which provides enough information for the administrator to find and revoke a particular certificate, if necessary. **complete**–Stores each issued certificate in the database.
Step 4	**grant auto**	Specifies that all enrollment requests from the Cisco CG-OS router to the RA be granted automatically.
Step 5	**hash {md5 | sha1}**	Specifies the cryptographic hash function that Cisco CG-OS uses for self-signed certificates. By default, Cisco CG-OS software uses md5.
Step 6	**mode ra [transparent]**	Enters the RA certificate server mode. The transparent keyword allows the proxy CA server in RA mode to interoperate with more than one type of CA server. Specifically it allows a transparent path from the RA (CA proxy) to the actual CA server that stores the certificates.
Step 7	**ip http server**	Enables the RA to start listening on port 80 (HTTP).
Step 8	exit	Exits to the global configuration mode.

EXAMPLE

This example shows how to configure the RA to serve as proxy for the CA server.

```
IOS_CA_RA# configure terminal
IOS_CA_RA(config)# crypto pki server IOS_CA_RA
IOS_CA_RA(cs-server)# grant auto trustpoint BlueCA
IOS_CA_RA(cs-server)# hash sha1
IOS_CA_RA(cs-server)# mode ra transparent
IOS_CA_RA(cs-server)# ip http server
IOS_CA_RA(cs-server)# exit
IOS_CA_RA(config)#
```

Creating an Enrollment Profile on the RA

Enrolls with the CA server on behalf of the Cisco CG-OS router to obtain the certificates from the CA server.

DETAILED STEPS

	Command	Purpose
Step 1	**crypto pki trustpoint** *name*	Declares a trustpoint that the RA should trust and enters trustpoint configuration mode. *name*–Alphanumeric, case sensitive, string with a maximum length of 64 characters. **Note** The maximum number of trustpoints that you can configure in Cisco IOS is 16.
Step 2	**enrollment url** *url*	Defines the address of the CA server. *url*–Specifies the address of the CA server. **Note** You can specify only one RSA key-pair per CA.
Step 3	**serial-number**	Includes the serial number of the RA in the certificate. **Note** This command is only applicable to SCEP auto-enrollment.
Step 4	**subject-name** *x-500-name*	Specifies the subject name in the certificate request. *x-500-name*–Limit of 512 characters.
Step 5	**exit**	Exits to global configuration mode.

EXAMPLE

This example shows how to enroll the RA with the CA server to obtain certificates for the Cisco CG-OS router.

```
IOS_CA_RA (config)# crypto pki trustpoint IOS_CA_RA
IOS_CA_RA (ca-trustpoint)# enrollment url http://172/27/165.157:80
IOS_CA_RA (ca-trustpoint)# serial-number
IOS_CA_RA (ca-trustpoint)# subject-name ou=ioscs RA
IOS_CA_RA (ca-trustpoint)# exit
IOS_CA_RA(config)#
```

Generating a RSA Public and Private Key-Pair on the RA

The RSA key-pair provides secure communication between the RA and the CA server.

The RA can generate RSA key-pairs to sign and/or encrypt and de-encrypt the security payload during security protocol exchanges for applications.

Note The RSA key-pair must be generated for the RA before obtaining a certificate for the RA.

Note When configuring the RSA key-pair and CA trustpoint name, you must use the same name within the RA to ensure that the certificate is generated and associated correctly.

BEFORE YOU BEGIN

Define the trustpoint (secure credentials) for all services requiring secure communications. (See Creating an Enrollment Profile on the RA, page 6-13.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**crypto key generate rsa** [**label** *label-string*] [**exportable**] [**modulus** *size*]	Generates a RSA key-pair. The maximum number of key-pairs on a device is 16. *label-string*–An alphanumeric, case sensitive string with a maximum length of 64 characters. The default label string is the hostname and the FQDN separated by a period character (.). *size*–Values are 512, 768, 1024, 1536, and 2048. The default modulus size is 512. **Note** Cisco recommends that you use the 2048 setting. **Note** The security policy on the Cisco CG-OS router and on the CA (where enrollment is planned) must be considered when deciding the appropriate key modulus. By default, the key-pair is not exportable. Only exportable key-pairs can be exported in the PKCS#12 format. **Caution** You cannot change the exportability of a key-pair.
Step 3	**show crypto key mypubkey rsa**	(Optional) Displays the generated key.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to generate an RSA key-pair for the RA.

```
IOS_CA_RA# configure terminal
IOS_CA_RA(config)# crypto key generate rsa label IOS_CA_RA_Key modulus 2048
IOS_CA_RA(config)# copy running-config startup-config
```

Creating a Trustpoint for the RA

Defines the trustpoint (secure credentials) for all services requiring secure communications. This trustpoint will be used by the RA to obtain certificates for the Cisco CG-OS router from the CA server.

You must associate the RA with a trustpoint.

BEFORE YOU BEGIN

Generate the RSA key-pair for the RA router. (See Generating a RSA Public and Private Key-Pair on the RA, page 6-14.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**crypto pki trustpoint** *name*	Declares a trustpoint that the device should trust and enters trustpoint configuration mode. *name*–Alphanumeric, case sensitive, string with a maximum length of 64 characters. **Note** The maximum number of trustpoints that you can configure on a device is 16.
Step 3	**enrollment mode ra**	Ensures that the RA router requests the RA mode Certificate Service (CS) certificate from the CA server.
Step 4	**enrollment url** *url*	Defines the address of the CA server. *url*–Specifies the address of the CA server. **Note** You can specify only one RSA key-pair per CA.
Step 5	**serial-number**	Includes the serial number of the RA in the certificate. **Note** This command is only applicable to SCEP auto-enrollment.
Step 6	**fingerprint** *hex-data*	Defines the thumbprint of the CA server. Information is found in the **Certificate > Details** window of the CA Server.
Step 7	**revocation-check none**	(Optional) Invalidates revocation of compromised certificates.
Step 8	**rsakeypair** *rsa-keypair-label*	Enter the RSA key-pair name generated for the RA and CA. (See Generating a RSA Public and Private Key-Pair on the RA, page 6-14.)
Step 9	**show crypto pki trustpoint** *name*	(Optional) Displays information on any configured trustpoints.
Step 10	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to associate the RA with a trustpoint.

```
IOS_CA_RA# configure terminal
IOS_CA_RA(config)# crypto pki trustpoint IOS_CA_RA
IOS_CA_RA(ca-trustpoint)# enrollment mode ra
IOS_CA_RA(ca-trustpoint)# enrollment url http://<CA_Server_IP
address>:80/certserv/mscep/mscep.dll
IOS_CA_RA(ca-trustpoint)# serial-number
IOS_CA_RA(ca-trustpoint)# fingerprint 2D830B4783130C2B7B64B338835D7516
IOS_CA_RA(ca-trustpoint)# revocation-check none
IOS_CA_RA(ca-trustpoint)# rsakeypair IOS_CA_RA_Key 2048
IOS_CA_RA(ca-trustpoint)# ip http server
IOS_CA_RA(ca-trustpoint)# copy running-config startup-config
```

Authenticating the RA

Begins authentication between the Cisco CG-OS router and the RA.

BEFORE YOU BEGIN

Generate the RSA key-pair for the RA. (See Generating a RSA Public and Private Key-Pair on the RA, page 6-14.)

Create a trustpoint. (See Creating a Trustpoint for the RA, page 6-15.)

DETAILED STEPS

	Command	Purpose
Step 1	**crypto pki server** *name*	Identifies the PKI server that was configured previously. (See Configuring the RA as Proxy for the CA Server, page 6-12.)
Step 2	**no shutdown**	Enables the PKI server.

EXAMPLE

```
IOS_CA_RA# configure terminal
IOS_CA_RA(config)# crypto pki server IOS_CA_RA
IOS_CA_RA(config)# no shutdown
```

Manual Enrollment

The Cisco CG-OS software supports certificate retrieval and enrollment using manual cut-and-paste. Cut-and-paste enrollment means that you must use a terminal to cut-and-paste the certificate requests and resulting certificates sent between the Cisco CG-OS router and the CA.

You must perform the following steps when using cut-and-paste in the manual enrollment process:

1. Create an enrollment certificate request, which the Cisco CG-OS router displays in base64-encoded text form.
2. Cut-and-paste the encoded certificate request text in an e-mail or in a web form and send it to the CA.
3. Receive the issued certificate (in base64-encoded text form) from the CA in an e-mail or in a web browser download.

4. Cut-and-paste the issued certificate into the Cisco CG-OS router using the certificate import facility.

This section describes the tasks that you must perform to allow the Cisco CG-OS router to assign digital certificates to itself by using manual cut-and-paste, and includes the following topics:

Creating a Trustpoint

Defines the trustpoint (secure credentials) for all services requiring secure communications.

BEFORE YOU BEGIN

Ensure that you have access to a terminal.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**crypto ca trustpoint** *name*	Declares a trustpoint that the Cisco CG-OS router can trust and enters trustpoint configuration mode. *name*–Alphanumeric, case sensitive, string with a maximum length of 64 characters. **Note** The maximum number of trustpoints that you can configure on the Cisco CG-OS router is 16.
Step 3	**enrollment terminal**	Enables cut-and-paste certificate enrollment on the Cisco CG-OS router.

EXAMPLE

This example shows how to create a trustpoint for the Cisco CG-OS router using manual cut-and-paste enrollment.

```
router_cgr01# configure terminal
router_cgr01(config)# crypto ca trustpoint CGRca
router_cgr01(ca-trustpoint))# enrollment terminal
router_cgr01(ca-trustpoint))# exit
router_cgr01(config)#
```

Authenticating the CA

The configuration process of trusting a CA is complete only when the CA is authenticated to the Cisco CG-OS router. You must authenticate your Cisco CG-OS router to the CA by obtaining the self-signed certificate of the CA in PEM format, which contains the public key of the CA. Because the certificate of the CA is self-signed (the CA signs its own certificate) the public key of the CA should be authenticated by contacting the CA administrator to compare the fingerprint of the CA certificate.

Note In order to have a valid certificate, you must know the identity of the root CA even if there are intermediate servers in the path. The full path is identified as the *certificate chain*. The maximum number of certificates in a CA certificate chain is 10. Be sure that you cut-and-past the full certificate chain.

BEFORE YOU BEGIN

Create a trustpoint. (See Creating a Trustpoint, page 6-18.)

Obtain the CA certificate or CA certificate chain.

DETAILED STEPS

	Command	Purpose
Step 1	**crypto ca authenticate** *trustpoint-name*	Cisco CG-OS router prompts you to cut-and-paste the certificate of the CA. Use the same name that you used when defining the trustpoint. The maximum number of trustpoints that you can authenticate to a specific CA is 10. **Note** For subordinate CA authentication, Cisco CG-OS requires the full chain of CA certificates ending in a self-signed CA because the CA chain is needed for certificate verification as well as for PKCS#12 format export.
Step 2	**exit**	Exits the configuration mode.
Step 3	**show crypto ca trustpoints**	(Optional) Displays the CA trustpoint CA information.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to authenticate a CA.

```
router_cgr01# configure terminal
router_cgr01(config)# crypto ca authenticate BlueCA
input (cut & paste) CA certificate (chain) in PEM format;
end the input with a line containing only END OF INPUT :
-----BEGIN CERTIFICATE-----
MIIC4jCCAoygAwIBAgIQBWDSiay0GZRPSRIljK0ZejANBgkqhkiG9w0BAQUFADCB
kDEgMB4GCSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklO
MRIwEAYDVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UE
```

```
ChMFQ2lzY28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBD
QTAeFw0wNTA1MDMyMjQ2MzdaFw0wNzA1MDMyMjU1MTdaMIGQMSAwHgYJKoZIhvcN
AQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UEBhMCSU4xEjAQBgNVBAgTCUth
cm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3J1MQ4wDAYDVQQKEwVDaXNjbzETMBEG
A1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBhcm5hIENBMFwwDQYJKoZIhvcN
AQEBBQADSwAwSAJBAMW/7b3+DXJPANBsIHHzluNccNM87ypyzwuoSNZXOMpeRXXI
OzyBAgiXT2ASFuUOwQ1iDM8rO/41jf8RxvYKvysCAwEAAaOBvzCBvDALBgNVHQ8E
BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJyjyRoMbrCNMRU2OyRhQ
GgsWbHEwawYDVR0fBGQwYjAuoCygKoYoaHR0cDovL3NzZS0wOC9DZXJ0RW5yb2xs
L0FwYXJuYSUyMENBLmNybDAwoC6gLIYqZmlsZTovL1xcc3NlLTA4XENlcnRFbnJv
bGxcQXBhcm5hJTIwQ0EuY3JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEB
BQUAA0EAHv6UQ+8nE399Tww+KaGr0g0NIJaqNgLh0AFcT0rEyuyt/WYGPzksF9Ea
NBG7E0oN66zex0EOEfG1Vs6mXp1//w==
-----END CERTIFICATE-----
END OF INPUT
Fingerprint(s): MD5 Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12


Do you accept this certificate? [yes/no]: yes
```

router_cgr01(config)# **exit**

router_cgr01# **copy running-config startup-config**

Generating an RSA Public and Private Key-Pair

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**crypto key generate rsa** [**label** *label-string*] [**exportable**] [**modulus** *size*]	Generates an RSA key-pair. The maximum number of key-pairs on a device is 16. *label-string*–An alphanumeric, case sensitive string with a maximum length of 64 characters. The default label string is the hostname and the FQDN separated by a period character (.). *size*–Values are 512, 768, 1024, 1536, and 2048. The default modulus size is 512. **Note** Cisco recommends that you use the 2048 setting. **Note** The security policy on the Cisco CG-OS router and on the CA (where enrollment is planned) must be considered when deciding the appropriate key modulus. By default, the key-pair is not exportable. Only the PKCS#12 format allows exportable key-pairs. **Caution** You cannot change the exportability of a key-pair.
Step 2	**show crypto key mypubkey rsa**	(Optional) Displays the generated key.

EXAMPLE

This example shows how to generate a RSA key-pair.

```
router_cgr01# configure terminal
router_cgr01(config)# crypto key generate rsa label BlueCA_Identity modulus 2048
router_cgr01(config-trustpoint)#
```

Associating the RSA Key-Pair to the Trustpoint

BEFORE YOU BEGIN

Generate an RSA key-pair. (See Generating an RSA Public and Private Key-Pair, page 6-20.)

DETAILED STEPS

	Command	Purpose
Step 1	**crypto ca trustpoint** *trustpoint-label*	Specifies a trustpoint CA and enters trustpoint configuration mode.
Step 2	**rsakeypair** *rsa-keypair-label*	Enter the keypair name generated for the Cisco CG-OS router and RA.

EXAMPLE

This example shows how to associate an RSA key-pair to a trustpoint.

```
router_cgr01# configure terminal
router_cgr01(config)# crypto ca trustpoint BlueCA
router_cgr01(config-trustpoint)# rsakeypair BlueCA_Identify
```

Generating Certificate Requests

You must generate a request to obtain identity certificates from the associated trustpoint CA for each of the RSA key-pairs of the Cisco CG-OS router. You must then cut-and-paste the displayed request into an e-mail or in a website form for the CA.

BEFORE YOU BEGIN

Create an association with the CA. (See Associating the RSA Key-Pair to the Trustpoint, page 6-21.)

Obtain the CA certificate or CA certificate chain.

DETAILED STEPS

	Command	Purpose
Step 1	**crypto ca enroll** *trustpoint-label*	Generates a certificate request for an authenticated CA. *trustpoint label*–Name of the trustpoint. The maximum size is 64 characters. **Note** You must remember the challenge password. It is not saved with the configuration. You must enter this password if your certificate needs to be revoked.

EXAMPLE

This example shows how to generate a certificate request.

```
router_cgr01# configure terminal
router_cgr01(config)# crypto ca enroll CGRca
Create the certificate request ..
 Create a challenge password. You will need to verbally provide this
  password to the CA Administrator in order to revoke your certificate.
  For security reasons your password will not be saved in the configuration.
  Please make a note of it.
  Password:nbv123
 The subject name in the certificate will be: CGRca.cisco.com
 Include the router serial number in the subject name? [yes/no]: no
 Include an IP address in the subject name [yes/no]: yes
ip address:10.22.31.162
 The certificate request will be displayed...
-----BEGIN CERTIFICATE REQUEST-----
MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUJ1DVaSMqNIgJ2kt8rl4lKY
0JC6ManNy4qxk8VeMXZSiLJ4JgTzKWdxbLDkTTysnjuCXGvjb+wj0hEhv/y51T9y
P2NJJ8ornqShrvFZgC7ysN/PyMwKcgzhbVpj+rargZvHtGJ91XTq4WoVkSCzXv8S
VqyH0vEvAgMBAAGgTzAVBgkqhkiG9w0BCQcxCBMGbmJ2MTIzMDYGCSqGSIb3DQEJ
DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ
KoZIhvcNAQEEBQADgYEAkT60KER6Qo8nj0sDXZVHSfJZh6K6JtDz3Gkd99GlFWgt
PftrNcWUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktExiI6Ul88nTOjglXMjja8
8a23bNDpNsM8rklwA6hWkrVL8NUZEFJxqbjfngPNTZacJCUS6ZqKCMetbKytUx0=
-----END CERTIFICATE REQUEST-----
```

Installing Identity Certificates

You can receive the identity certificate from the CA by email or through a web browser in base64 encoded text form. You must install the identity certificate from the CA by cutting and pasting the encoded text.

BEFORE YOU BEGIN

Generate a certificate request and verify receipt of the signed certificate from the CA.

DETAILED STEPS

	Command	Purpose
Step 1	**crypto ca import** *trustpoint-label* **certificate**	Cisco CG-OS router prompts you to cut-and-paste the identity certificate for the CA. The maximum number of identify certificates that you can configure on the Cisco CG-OS router is 16.

EXAMPLE

This example shows how to install the identify certificate named CGRca into the Cisco CG-OS router.

```
router_cgr01# configure terminal
router_cgr01#(config)# crypto ca import CGRca certificate
input (cut & paste) CA certificate (chain) in PEM format;
end the input with a line containing only END OF INPUT:
-----BEGIN CERTIFICATE-----
MIIDITCCAgmgAwIBAgIBCDANBgkqhkiG9w0BAQUFADBuMQswCQYDVQQGEwJVUzET
MBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxFjAUBgNVBAoT
DUNpc2NvIFN5c3RlbXMxDTALBgNVBAsTBFNHQlUxEDAOBgNVBAMTB0JsdWUgQ0Ew
HhcNMTEwNDI2MTkyNTI4WhcNMjEwNDIzMTkyNTI4WjAXMRUwEwYDVQQDEwxhbGV4
bC1NaWxhbjEwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCtBHACAgCO
7+79Lm7JJ/G+PhSfga3sFp2wRMQKmoCvmus9d7vkzrHQIfzs6uaiqPT8F3SQTbWU
btr01JDfJvvuVOd/VNrwZoP9kFJHU3uOEDY767SmG2lv9XO96GcKsa/Nj8nXM6FZ
7PhuZl9NQ6QYYjEkvBWsUsQjMfL8oFckTZvPKoCdjl5WuQ14Umo6N2ST7K+UsqsL
wTkP9fyGgKsQuYJg6PBSaEu+OqfWkkvy65QWVtGGRdOmOD8EKsKpVQecVQuZ86it
o2CPcYET83LOeu35H2TVhzyuv5lUaoor02hP6swyUXsQLVg6XeyVbwzf7R8t4M2W
VPs42bHwqRHZAgMBAAGjITAfMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcD
AjANBgkqhkiG9w0BAQUFAAOCAQEAZxfmKoJb9aT/eZT5v1cCwdsM0Wh9fjON3h3L
no2r1y83jKvYhpjNhLcoF5Q0sFCPRBpq8f9vjKGHHT9j4fkQZGmsJl+l5YztuF6J
rkFmf3tzqJU9Hu8d6zh4YtdqMz57DxhMC6PbvAg1bw6KSujzxT7J0HnpFog+QkaX
MTC9JCvRWna1VOo38VsA0XYUtMrXRN7XntKzb6D33dW7VYQCSmz/McNWbOL0zlTl
B7ZKzkOH/uqG3A0fj6+LDqFyprQ62RhLTSuNKViCliR91QOE0weXm8txwLBlM8Tg
q/a6HGFzsLhOa3X1qhv2oqA5QgPyS0ixMEdzh1cxkTzTgmDjow==
-----END OF CERTIFICATE-----
router_cgr01#(config)#
```

Configuring Self-Signed Certificates on the Cisco CG-OS Router

Creates self-signed certificate for the Cisco CG-OS router.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**crypto ca trustpoint** *name*	Declares a trustpoint that the Cisco CG-OS router can trust and enters trustpoint configuration mode. *name*–Alphanumeric, case sensitive, string with a maximum length of 64 characters. **Note** The maximum number of trustpoints that you can configure on the Cisco CG-OS router is 16.
Step 3	**revocation-check none**	(Optional) Invalidates revocation of compromised certificates.
Step 4	**enrollment selfsigned**	Generates a self-signed certificate on the Cisco CG-OS router.
Step 5	**rsakeypair** *rsa-keypair-label*	Enters the RSA key-pair name generated by the Cisco CG-OS router.
Step 6	**exit**	Exits to the global configuration mode.
Step 7	**crypto ca enroll** *trustpoint-label*	Generates a certificate request for the self-signed Cisco CG-OS router. *trustpoint label*–Name of the trustpoint. The maximum size is 64 characters.

EXAMPLE

This example shows how to create a self-signed certificate on the Cisco CG-OS router.

```
router_cgr01# configure terminal
router_cgr01#(config)# crypto ca trustpoint SelfSignedCA
router_cgr01(config-trustpoint)# revocation-check none
router_cgr01(config-trustpoint)# enrollment selfsigned
router_cgr01(config-trustpoint)# rsakeypair PrivateKeyForSelfSignedCA 2048
router_cgr01(config-trustpoint)# exit
router_cgr01(config)# crypto ca enroll SelfSignedCA

Create the certificate request ..
The subject name in the certificate will be the name of the router.
Include the router serial number in the subject name? [yes/no]:yes
The serial number in the certificate will be: CGR1240/K9+JSJ15380008
Include an IP address in the subject name [yes/no]:no
Include the Alternate Subject Name ? [yes/no]:no
```

Importing Identity Information in PKCS#12 Format

You can import the certificate and RSA key-pair to recover from a system crash on your Cisco CG-OS router or when you replace equipment on your Cisco CG-OS router.

Note You can use only the bootflash:*filename* format when specifying the import URL.

BEFORE YOU BEGIN

Ensure that the trustpoint is empty by checking that no RSA key-pair is associated with it and no CA is associated with the trustpoint using CA authentication.

DETAILED STEPS

	Command	Purpose
Step 1	**copy** *scheme*:**//***server*/[*url*/]*filename* **bootflash:***filename*	Copies the PKCS#12 format file from the remote server. For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *server*, *url*, and *filename* arguments are case sensitive.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**crypto ca import** *trustpoint-label* **pksc12 bootflash:***filename password*	Imports the identity certificate and associated key-pair and CA certificates for trustpoint CA. *password*–Identifies the export password of PKCS#12.
Step 4	**show crypto ca certificates**	(Optional) Displays the CA certificates.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to copy the PKCS#12 format file from the remote server and then import that file into the Cisco CG-OS router.

```
copy tftp:adminid.p12 bootflash:adminid.p12
router_cgr01# configure terminal
router_cgr01(config)# crypto ca import ConnectedGrid pkcs12 bootflash:adminid.p12 nbv123
router_cgr01(config)# copy running-config startup-config
```

Ensuring Trustpoint Configurations Persist Across Reboots

You can ensure that the trustpoint configuration persists across reboots of the Cisco CG-OS router.

The trustpoint configuration is a normal system configuration that persists across system reboots only if you copy it explicitly to the startup configuration. The certificates and RSA key-pairs associated with a trustpoint are automatically persistent if you have already copied the trustpoint configuration in the startup configuration. Conversely, if the trustpoint configuration is not copied to the startup configuration, the certificates and RSA key-pairs associated with it are not persistent since they require the corresponding trustpoint configuration after a reboot.

Always copy the running configuration to the startup configuration to ensure that the configured certificates and RSA key-pairs are persistent. Also, save the running configuration after deleting a certificate or key-pair to ensure that the deletions permanent.

The certificates associated with a trustpoint automatically become persistent when imported (that is, without explicitly copying to the startup configuration) if the specific trustpoint is already saved in startup configuration.

Cisco recommends that you create a password protected backup of the identity certificates and save it to an external server. (See Exporting Identity Information in PKCS#12 Format, page 6-26.)

Note Copying the configuration to an external server includes the certificates and RSA key-pairs.

Exporting Identity Information in PKCS#12 Format

You can export the identity certificate along with the RSA key-pair and CA certificate (or the entire chain in the case of a subordinate CA) of a trustpoint to a PKCS#12 file for backup purposes. You can import the certificate and RSA key-pair to recover from a system crash on your device.

Note You can use only the bootflash:*filename* format when specifying the export URL.

BEFORE YOU BEGIN

Generate an exportable RSA key-pair. (See Generating an RSA Public and Private Key-Pair on the Cisco CG-OS Router, page 6-8.)

Authenticate the CA. (See Authenticating the CA, page 6-19.)

Install an identity certificate. (See Installing Identity Certificates, page 6-23.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command	Purpose
Step 2	**crypto ca export** *trustpoint-label* **pkcs12 bootflash:***filename password*	Exports the identity certificate and associated key-pair and CA certificates for a trustpoint CA. The password is alphanumeric, case sensitive, and has a maximum length of 128 characters.
Step 3	**copy bootflash:***filename scheme***://**server**/**[*url*/]*filename*	Copies the PKCS#12 format file to a remote server. For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *server*, *url*, and *filename* arguments are case sensitive.

EXAMPLE

This example shows how to export an identity certificate to a PKCS#12 file for backup purposes.

```
router_cgr01# configure terminal
router_cgr01(config)# crypto ca export ConnectedGrid pkcs12 bootflash:adminid.p12 nbv123
router_cgr01(config)# copy bootflash:adminid.p12 tftp:adminid.p12
```

Deleting Certificates from the CA Configuration

You can delete the CA certificates and identity certificates that are configured in a trustpoint. You must first delete the CA certificates, followed by the identity certificate. After deleting the identity certificate, you can disassociate the RSA key-pair from a trustpoint. You must delete certificates to remove expired or revoked certificates, certificates that have compromised (or suspected to be compromised) RSA key-pairs, or CAs that are no longer trusted.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**crypto ca trustpoint** *trustpoint-label*	Specifies a trustpoint CA and enters trustpoint configuration mode.
Step 3	**delete ca-certificates**	Deletes the CA certificate or certificate chain.

	Command	Purpose
Step 4	**delete certificate** [**force**]	Deletes the identity certificate. You must use the **force** option if the identity certificate you want to delete is the last certificate in a certificate chain or only identity certificate in the device. This requirement ensures that you do not mistakenly delete the last certificate in a certificate chain or only identity certificate and leave the applications without a certificate to use.
Step 5	**show crypto ca certificates** [*trustpoint-label*]	(Optional) Displays the CA certificate information.
Step 6	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to delete CA certificates and identity certificate from the Cisco CG-OS router.

```
router_cgr01# configure terminal
router_cgr01(config)# crypto ca trustpoint admin-ca
router_cgr01(config-trustpoint)# delete ca-certificate
router_cgr01(config-trustpoint)# delete certificate
router_cgr01(config-trustpoint)# copy running-config startup-config
```

Deleting RSA Key-Pairs from the Cisco CG-OS Router

You can delete the RSA key-pairs on the Cisco CG-OS router when you believe the integrity of the RSA key-pairs are compromised or should no longer be used.

Note After you delete RSA key-pairs from the Cisco CG-OS router, ask the CA administrator to revoke the certificates of the Cisco CG-OS router at the CA. You must supply the challenge password that was created when the certificates were originally created. (See Generating Certificate Requests, page 6-22.)

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**crypto key zeroize rsa** *label*	Deletes the RSA key-pair.

	Command	Purpose
Step 3	**show crypto key mypubkey rsa**	(Optional) Displays the RSA key-pair configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to delete RSA key-pairs on the Cisco CG-OS router.

```
router_cgr01# configure terminal
router_cgr01(config)# crypto key zeroize rsa MyKey
router_cgr01(config)# copy running-config startup-config
```

Verifying the Configuration

To verify the PKI and CA configurations, use the following commands:

Command	Purpose
show crypto key mypubkey rsa	Displays information about the RSA public keys generated on the Cisco CG-OS router.
show crypto pki certificates	Displays information about CA and identity certificates.
show crypto ca trustpoints	Displays information about CA trustpoints.

Configuring User Accounts and RBAC

This chapter describes how to configure user accounts and role-based access control (RBAC) on the Cisco 1000 Series Connected Grid Routers (hereafter referred to as Cisco CG-OS router).

This chapter includes the following sections:

Information About User Accounts and RBAC

You can create and manage user accounts and assign roles that limit access to operations on the Cisco CG-OS router. RBAC allows you to define the rules for and assign roles that restrict the authorization that the user has to access management operations.

This section includes the following topics:

About User Accounts

You can configure up to a maximum of 256 user accounts. Cisco CG-OS provides one default user account: *admin*.

By default, the user account does not expire unless you explicitly configure it to expire. The expire option determines the date when Cisco CG-OS disables the user account.

Caution User accounts can only be defined on the default VDC. The Cisco CG-OS router does not support multiple configuration on multiple VDCs.

Caution Cisco CG-OS does not support all numeric usernames, whether created with TACACS+ or RADIUS, or created locally on the CG-OS router. When an all numeric user name exists on an AAA server and is entered during login, the user is not logged in.

Tip The following words are reserved and cannot be used to configure users: bin, daemon, adm, lp, sync, shutdown, halt, mail, news, uucp, operator, games, gopher, ftp, nobody, nscd, mailnull, root, rpc, rpcuser, xfs, gdm, mtsuser, ftpuser, man, and sys.

Characteristics of Strong Passwords

A strong password has the following characteristics:

- At least eight characters long
- Does not contain many consecutive characters (such as “abcd”)
- Does not contain many repeating characters (such as “aaabbb”)
- Does not contain dictionary words
- Does not contain proper names
- Contains both uppercase and lowercase characters
- Contains numbers

The following are examples of strong passwords:

- If2CoM18
- 2004AsdfLkj30
- Cb1955S21

Note Please note these important notes about passwords on the Cisco CG-OS router:

- User passwords are not displayed in the configuration files.
- Clear text passwords cannot contain dollar signs ($) or spaces anywhere in the password. Also, they cannot include these special characters at the beginning of the password: quotation marks (" or '), vertical bars (|), or right angle brackets (>).
- When a password is trivial (such as a short, easy-to-decipher password), Cisco CG-OS rejects the password configuration when password-strength checking is enabled. (See Enabling Password-Strength Checking, page 7-5.) Be sure to configure a strong password as shown in the sample configuration. Passwords are case sensitive.

About User Roles

User roles contain rules that define the operations allowed for the user who is assigned the role. Each user role can contain multiple rules and each user can have multiple roles. For example, if role1 allows access only to configuration operations, and role2 allows access only to debug operations, then users who belong to both role1 and role2 can access configuration and debug operations. You can also change a user role interface policy to limit the interfaces that the user can access.

Cisco CG-OS provides four default user roles within the default VDC:

- network-admin—Complete read-and-write access to the entire Cisco CG-OS router
- network-operator—Complete read access to the entire Cisco CG-OS router
- vdc-admin—Read-and-write access limited to a VDC (in this case, the default VDC)
- vdc-operator—Read access limited to a VDC (in this case, the default VDC)

Note You cannot change the default user roles.

You can create custom roles within the default VDC. By default, the user roles that you create allow access only to the **show**, **exit**, **end**, and **configure terminal** commands.You must add rules to allow users to display or configure features.

Note When you belong to multiple roles, you can execute a combination of all the commands permitted by these roles. Access to a command takes priority over being denied access to a command. For example, suppose a user has RoleA, which denied access to the configuration commands. However, the user also has RoleB, which has access to the configuration commands. In this case, the user has access to the configuration commands.

About User Role Rules

The rule is the basic element of a role. A rule defines what operations the role allows the user to perform. You can apply rules for the following parameters:

- Command—A command or group of commands defined in a regular expression
- Feature—Commands that apply to a function provided by Cisco CG-OS
- Feature group—Default or user-defined group of features

These parameters create a hierarchical relationship. The most basic control parameter is the command. The next control parameter is the feature, which represents all commands associated with the feature. The last control parameter is the feature group. The feature group combines related features and allows you to easily manage the rules. Cisco CG-OS also supports the predefined feature group L3 that you can use.

You can configure up to 256 rules for each role.

The user-specified rule number determines the order in which the rules are applied. Rules are applied in descending order. For example, when a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1.

Guidelines and Limitations

Configuring User Accounts

You can configure up to 256 user accounts on the Cisco CG-OS router.

If you have a user account configured on the local Cisco CG-OS router that has the same name as a remote user account on an AAA server, Cisco CG-OS applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.

You cannot delete the default admin user account.

You cannot remove the default user roles from the default admin user account.

Note A user account must have at least one user role.

Configuring Roles

You can create up to 64 user-defined roles on the Cisco CG-OS router in addition to the four default user roles (network-admin, vdc-admin, network-operator, and vdc-operator).You cannot change the default user roles.

You can add up to 256 rules to a user role.

You can assign a maximum of 64 user roles to a user account.

Creating Feature Groups

You can add up to 64 user-defined feature groups on the Cisco CG-OS router in addition to the default feature group, L3.

Default Settings

Table 7-1 lists the default settings for user accounts and RBAC parameters.

Table 7-1 Default User Accounts and RBAC Parameters

Parameters	Default
User account password	Undefined
Password strength checking	Enabled
User account expiry date	None
User account role in the default VDC	Network-operator if the creating user has the network-admin role, or vdc-operator if the creating user has the vdc-admin role
Default user roles in the default VDC	Network-admin, network-operator, vdc-admin, and vdc-operator
Interface policy	All interfaces are accessible
Feature group	L3: Enables Layer 3 on the Cisco CG-OS router

Enabling Password-Strength Checking

You can enable password-strength checking, which prevents you from creating weak passwords for user accounts. For information about strong passwords, see Characteristics of Strong Passwords, page 7-2.

By default, this option is enabled on the Cisco CG-OS router.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**password strength-check**	Enables password-strength checking. By default, this option is enabled. You can disable password-strength checking by using the **no** form of this command.
Step 3	**show password strength-check**	(Optional) Displays the password-strength check configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to enable password-strength checking.

```
router_cgr01# configure terminal
router_cgr01(config)# password strength-check
router_cgr01(config)# copy running-config startup-config
```

Configuring User Accounts

You can create a maximum of 256 user accounts on a Cisco CG-OS router. User accounts have the following attributes:

- Username
- Password
- Expiry date
- User roles

You can enter the password in clear text format or encrypted format for the Cisco CG-OS router. The password encrypts clear text passwords before saving them to the running configuration. Encrypted format passwords are saved to the running configuration without further encryption.

User accounts can have a maximum of 64 user roles. For more information on user roles, see Configuring Roles, page 7-7.

Note The Cisco CG-OS router only supports the default VDC. The Cisco CG-OS router does not support multiple VDCs.

Note Changes to user account attributes do not take effect until the user logs in and creates a new session.

Note You cannot delete the default admin user account. You can create another account with the network-admin or vdc-admin role.

BEFORE YOU BEGIN

Determine which roles to assign to which user accounts.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**show role**	(Optional) Displays the user roles available. You can configure other user roles, if necessary. (See Creating User Roles and Rules, page 7-7.)
Step 3	**username** *user-id* [**password** [**0** | **5**] *password*] [**expire** *date*] [**role** *role-name*]	Configures a user account. User accounts can have a maximum of 64 user roles. *user-id*–Case-sensitive, alphanumeric character string with a maximum length of 28 characters. *password*–Default password is undefined. The **0** option indicates that the password is clear text and the **5** option indicates that the password is encrypted. The default is **0** (clear text). **Note** If you do not specify a password, the user might not be able to log in to the Cisco CG-OS router. For information about using SSH public keys instead of passwords, see Specifying the SSHv2 Public Keys for User Accounts, page 5-4. *date*–Format is YYYY-MM-DD. The default is no expiry date.
Step 4	**show user-account**	(Optional) Displays the role configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to configure a user account.

```
router_cgr01# configure terminal
router_cgr01(config)# show role
router_cgr01(config)# username NewUser password 5 4Ty18Rnt expire 2013-01-15 jsmith
router_cgr01(config)# copy running-config startup-config
```

Configuring Roles

This section includes the following topics:

Creating User Roles and Rules

You can configure up to 64 user roles. Each user role can have up to 256 rules. You can assign a user role to more that one user account.

The rule number that you specify determines the order in which the rules are applied. Cisco CG-OS applies the rules in descending order. For example, when a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1.

BEFORE YOU BEGIN

Identify which roles and rules must be assigned to each user account.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**role name** *role-name*	Specifies a user role and enters role configuration mode. The *role-name* argument is a case-sensitive, alphanumeric character string with a maximum length of 16 characters.

	Command	Purpose
Step 3	**rule** *number* {**deny** | **permit**} **command** *command-string*	Configures a command rule. The *command-string* argument can contain spaces and regular expressions. For example, "interface ethernet *" includes all Ethernet interfaces. Repeat this command for as many rules as needed.
	rule *number* {**deny** | **permit**} {**read** | **read-write**}	Configures a read-only or read-and-write rule for all operations.
	rule *number* {**deny** | **permit**} {**read** | **read-write**} **feature** *feature-name*	Configures a read-only or read-and-write rule for a feature. Use the **show role feature** command to display a list of features. Repeat this command for as many rules as needed.
	rule *number* {**deny** | **permit**} {**read** | **read-write**} **feature-group** *group-name*	Configures a read-only or read-and-write rule for a feature group. Use the **show role feature-group** command to display a list of feature groups. Repeat this command for as many rules as needed.
Step 4	**description** *text*	(Optional) Configures the role description. You can include spaces in the description.
Step 5	**show role**	(Optional) Displays the user role configuration.
Step 6	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to create the user role UserA and define rules for that user.

```
router_cgr01# configure terminal
router_cgr01(config)# role name operator_noc
router_cgr01(config-role)# rule 1 deny command clear users
router_cgr01(config-role)# rule 2 deny read-write
router_cgr01(config-role)# rule 3 permit read feature ospf
router_cgr01(config-role)# rule 4 deny read-write L3
router_cgr01(config-role)# copy running-config startup-config
```

Creating Feature Groups

You can create custom feature groups to add to the default list of features provided by Cisco CG-OS. These groups contain one or more of the features. You can create up to 64 feature groups on the Cisco CG-OS router.

Note You cannot change the default feature group L3.

BEFORE YOU BEGIN

Identify any specific features that must be specified in the Cisco CG-OS router.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**role feature-group** *group-name*	Specifies a user role feature group and enters role feature group configuration mode. The *group-name* argument is a case-sensitive, alphanumeric character string with a maximum length of 32 characters.
Step 3	**feature** *feature-name*	Specifies a feature for the feature group. Repeat this command for as many features as needed. **Note** Use the **show role component** command to display a list of features.
Step 4	**show role feature-group**	(Optional) Displays the role feature group configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to create the custom feature group GroupA to expand the default list of features provided by Cisco CG-OS.

```
router_cgr01# configure terminal
router_cgr01(config)# role feature GroupA
router_cgr01(config-role-featuregrp)# feature abc
router_cgr01(config-role-featuregrp)# copy running-config startup-config
```

Changing User Role Interface Policies

You can change a user role interface policy to limit the interfaces that the user can access. By default, a user role (unless specifically configured otherwise) allows a user to have access to all interfaces on the Cisco CG-OS router.

Note You cannot change the default roles network-admin, network-operator, vdc-admin, and vdc-operator.

BEFORE YOU BEGIN

Create one or more user roles. (See Creating User Roles and Rules, page 7-7.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**role name** *role-name*	Specifies a user role and enters role configuration mode.
Step 3	**interface policy deny**	Enters role interface policy configuration mode.
Step 4	**permit interface {ethernet | cellular | wimax}** *slot/port*	Specifies a list of interfaces that the role can access. Repeat this command for each interface to which you want the user to have access.
Step 5	**show role**	(Optional) Displays the role configuration.
Step 6	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to change a user role interface policy to limit the user to be able to access and configure only specific Ethernet interfaces on the Cisco CG-OS router.

```
router_cgr01# configure terminal
router_cgr01(config)# role name EthOnly1to4
router_cgr01(config-role)# interface policy deny
router_cgr01(config-role-interface)# permit interface ethernet 2/1-4
router_cgr01(config-role-interface)# copy running-config startup-config
```

Verifying Configuration

To display user account and RBAC configuration information, enter any or all of the following commands:

Command	Purpose
show role	Displays the user role configuration.
show role feature	Displays the feature list.
show role feature-group	Displays the feature group configuration.
show startup-config security	Displays the user account configuration in the startup configuration.
show running-config security [**all**]	Displays the user account configuration in the running configuration. The **all** keyword displays the default values for the user accounts.
show user-account	Displays user account information.

Configuration Example

The following example shows how to configure a user role:

```
role name UserA
    rule 3 permit interface ethernet 2/1-4
    rule 2 permit read feature ospf
    rule 1 deny command clear *
```

The following example shows how to configure a user role feature group:

```
role feature-group name Security-features
    feature radius
    feature aaa
    feature access-list
```

Configuring IKEv2 and IPSec

This chapter describes how to configure Internet Key Exchange version 2 (IKEv2) and IP Security (IPSec) on the Cisco 1000 Series Connected Grid Routers (hereafter referred to as Cisco CG-OS router) to support secure communications between a source (Cisco CG-OS router) and destination router over a virtual tunnel.

This chapter includes the following sections:

Information About IKEv2 and IPSec

Internet Key Exchange Version 2 (IKEv2) is a key management protocol standard that is used in conjunction with the IPsec standard. IPSec is a security protocol that provides data security by tunnel and transport mode.

Virtual Tunnels

In the tunnel mode, IPSec protects peer-to-peer communication between two end nodes by establishing a virtual tunnel between those two endpoints. On the Cisco CG-OS router, this virtual tunnel is built between itself (source) and the destination router such as the Cisco ASR 1000 Series Aggregation Services Routers (Cisco ASR), which serve as a head-end router.

The virtual tunnel does not manage or modify any packets that are sent over the physical interfaces of the Cisco CG-OS router. Therefore, the Cisco CG-OS router can interoperate with most IPSec implementations (operating with IKEv2) that support IPSec Encapsulating Secure Payload (ESP)

operating in tunnel mode. (See limitations in Guidelines and Limitations for IKEv2 and IPSec, page 8-3.)

IKEv2 Authentication

The Cisco CG-OS router employs IKEv2 to authenticate to the destination router by using either a pre-shared key (PSK) or by using RSA signatures with a Public Key Infrastructure (PKI). IKEv2 must be configured on the source and destination router (peers) and both routers must employ the same authentication method.

- PSK authenticates each router (peer) by requiring proof of possession of a shared secret. Each router (peer) must have the same shared secret configured.
- RSA signatures employ a PKI-based method of authentication. (See Configuring PKI, page 6-1.) IKEv2 interacts with PKI to obtain the identity certificates and to validate the peer (such as Cisco CG-OS router and head-end router) certificates.

IPSec Tunnel Encryption and De-encryption

Encryption Flow

When a packet arrives at the router through an interface, the Cisco CG-OS router applies any configured Policies to that interface such as ingress IP access control lists (IP ACLs) or QoS policies. During IP routing, the Cisco CG-OS router identifies any traffic destined for the virtual tunnel. Before forwarding that traffic to the virtual tunnel interface (VTI), the Cisco CG-OS router attaches any egress policies defined for the VTI. At the VTI, IPSec encrypts the original packet and then encapsulates it within another packet. The encapsulated packet has the Differentiated Services Code Point (DSCP) field of the original packet and its outer address has the source (Cisco CG-OS router) and destination (head-end router) addresses of the VTI.

After encapsulation, IPSec resubmits the packet to the routing function for forwarding to an interface for transmission to the head-end router. The Cisco CG-OS router applies any configured egress IP ACL or QoS policies configured for the interface, before the packet exits the interface.

De-encryption Flow

When the encapsulated packet (with an IP protocol field of ESP) arrives at the destination router (head-end router), the Cisco CG-OS router applies any ingress IP ACL and QoS policies configured for the ingress interface to the packet. The encapsulated packet is then forwarded for processing by IPSec (before any route lookup occurs) for de-encryption. After de-encryption, IPSec forwards the original packet back into the routing function where the Cisco CG-OS router applies egress IP ACL and QoS policies configured for the VTI.

Policies

IKEv2 employs policies to negotiate handshakes between the two peers. These policies, which are configured on each peer, are a combination of the various security parameters listed below:

- Encryption method (3DES, AES)
- Hash algorithm (SHA)
- Diffie-Hellman (DH) group (768-bit, 1024-bit or 1536-bit DH).

Each policy has a unique priority number assigned to it.

The peers must share at least one common policy to allow for successful secure communication.

During the IKEv2 Security Association (SA) negotiation, IKEv2 searches for a policy that is the same for both peers. The peer that initiates the negotiation (handshake) sends all its supported policies to the remote peer.

- If a match is found by the remote peer, then the peers employ that security policy for all future communications.
- If no policy match exists between the two peers, then IKEv2 terminates the negotiation.

After successful IKEv2 SA negotiation between the peers, IPSec SA negotiation occurs by exchanging profiles (known as transform-sets) between the two peers.

Application

The primary application of IPSec and IKEv2 is to allow the configuration of tunnels between the Cisco CG-OS router and the head-end router to securely encapsulate and de-encapsulate traffic sent and received over a WAN interface from an insecure backhaul.

IKEv2 negotiates the secure communication channel and IPSec encrypts and de-encrypts the traffic received from an insecure backhaul to provide data confidentiality, data integrity, and authentication. IPSec also provides support for the anti-replay protocol that provides IP packet-level security to prevent interception and modification of message packets that are being sent between a source and destination system.

IPv4 packets can be transported within the virtual tunnel. The Cisco CG-OS router supports up to 25 simultaneous IPSec virtual tunnels.

Prerequisites

A connection must exist between the Cisco CG-OS router and the head-end router before you can configure a virtual tunnel interface between the two systems.

Guidelines and Limitations for IKEv2 and IPSec

IKEv2

IKEv2 must be configured on the source (Cisco CG-OS router) and destination (head-end) routers.

IPSec

IPSec only supports key negotiation using IKEv2 and does not support connection to firewalls configured on the Cisco ASA 5500 Series Adaptive Security Appliance and other VPN concentrator products.

Default Settings

Table 8-1 lists the default settings for IKEv2 policy parameters.

Table 8-2 lists the default settings for IPSec profile parameters.

Table 8-1 Default Settings for IKEv2 Policy Parameters

Parameter	Default
Encryption algorithm	128-bit AES
Hash algorithm	SHA-1
Diffie-Hellman (DH) group	Group 2–1024-bit DH
Authentication method	RSA signatures
lifetime seconds value	86400 seconds

Table 8-2 Default Settings for IPSec Profile Parameters

Parameter	Default
set pfs group	Disabled
set security-association lifetime duration	4608000 kilobytes and 3600 seconds

Configuring IKEv2 and IPSec

BEFORE YOU BEGIN

Contact the system administrator to confirm the authentication method (PSK or RSA) to configure on the Cisco CG-OS router.

DETAILED STEPS

	Command	Purpose
Step 1	**feature crypto ike**	Enables IKEv2 on the Cisco CG-OS router. **Note** To prevent loss of IKEv2 configuration, do not disable IKEv2 when IPSec is enabled on the Cisco CG-OS router.
Step 2	**crypto ike domain ipsec**	Configures the IKEv2 domain and enters the IKEv2 configuration submode.
Step 3	**policy** *value*	Defines IKEv2 priority policy and enters the policy configuration submode. The lower the number, the higher the priority.
Step 4	**authentication** *method*	Specifies the IKEv2 authentication method. Method options are PSK (pre-share) and RSA signature (rsa-sig) authentication. Default setting for the Cisco CG-OS router is rsa-sig.
Step 5	**encryption** *enc_algo*	Specifies the encryption algorithm for the policy. Options are: 3des–168-bit DES (3DES) aes–AES-CBC Default setting for the Cisco CG-OS router is aes.

	Command	Purpose
Step 6	**hash** *hash_algo*	Configures the hash algorithm for the IKE policy. Options are: sha–HMAC-SHA1 md5–HMAC-MD5 Default setting for the Cisco CG-OS router is sha.
Step 7	**group** *DH_group*	Configures the Diffie-Hellman group for the policy. Options are: 1–768-bit Diffie-Hellman group 2–1024-bit Diffie-Hellman group 5–1536-bit Diffie-Hellman group Default setting for the Cisco CG-OS router is 2.
Step 8	**lifetime seconds** *value*	Specifies the IKE SA lifetime for the policy. Value is a range from 600 to 86400 seconds. Default setting is 86400 seconds.
Step 9	**exit**	Exits the policy mode.
Step 10	**keepalive** *value*	Configures the frequency of keep alive messages sent between peers in the tunnel. Keep alive messages validate the ability of peers to send and receive traffic. Value can be any number between 120 and 86400 seconds. The default value is 3600 seconds. (IKE global parameter)
Step 11	**identity hostname**	Configures the identity of the IKE protocol. By default, Cisco CG-OS employs the IP address of the Cisco CG-OS router as the identity for IKE protocol. This command must be set when using RSA. **Note** This command is optional when using PSK.
Step 12	**exit**	Exits to the configuration mode.
Step 13	**feature tunnel**	Enables tunneling on the Cisco CG-OS router.
Step 14	**feature crypto ipsec virtual-tunnel**	Enables IPSec tunnelling on the Cisco CG-OS router and creates a virtual tunnel interface.
Step 15	**crypto ipsec transform-set** *tx-form-name* {*txform*| *encr_txform auth_txform}*	Configures a single transform set that is included within the IPSec protection profile. Options for *txform* are: • esp-gcm 128–128-bit AES-GCM authenticated encryption • esp-gcm 256–256-bit AES-GCM authenticated encryption Options for *encr_txform auth_txform:* • *encr_txform* options: esp-aes 128 *or* esp-aes 256 AES-CBC encryption • *auth_txform* options: esp-sha1-hmac *or* esp-sha256-hmac HMAC-SHA authentication **Note** The transform-set name (tx-form-name) defined here must match that transform-set name associated with the IPSec profile in Step 20.
Step 16	**crypto ip sec profile** *profile-name*	Configures an IPSec profile for attachment to the virtual tunnel interface.

	Command	Purpose
Step 17	**description** *text*	(Optional) Allows the user to provide a description for the profile. The character limit is 64 characters.
Step 18	**set pfs** *group*	Configures the Diffie-Hillman group for perfect forward secrecy for the IPSec tunnel. Options for group are as follows: group1–768-bit mode Diffie-Hillman group 4–2048-bit mode Diffie-Hillman group2–1024-bit mode Diffie-Hillman group5–1536-bit mode Diffie-Hillman **Note** By default, PFS is disabled.
Step 19	**set security-association lifetime** *[seconds] [kilobytes]*	Specifies the lifetime of the IPSec security association. When the configured lifetime value expires, a new security association is negotiated. Lifetime can be expressed in both time (seconds, 120 to 86400) and data volume (kilobytes, 2560 to 4292967295). The default seconds value is 3600 seconds. The default data volume is 4608000 kilobytes.
Step 20	**set transform-set** *tx-form-name*	Associates the transformation set to the currently configured IPSec profile.
Step 21	**exit**	Exits the profile mode.
Step 22	**interface tunnel** *number*	Creates a virtual tunnel. *number*–Any value from 0 to 4095.
Step 23	**ip address** *ip address*	Assigns an IP address for the interface tunnel.
Step 24	**tunnel mode ipsec {ipv4}**	Configures the encapsulation mode for the tunnel. **Note** When the tunnel is configured to operate in IPSec mode, the keepalive parameter must be disabled. By default, keepalive is disabled.
Step 25	**tunnel source** {*ip-address* | *interface-type slot-port*}	Configures the source endpoint for the tunnel.
Step 26	**tunnel destination** {*ip-address* | *host-name*}	Configures the destination endpoint for the tunnel.
Step 27	**description** *text*	(Optional) Allows the user to provide a description for the profile. The character limit is 64 characters.
Step 28	**tunnel protection ipsec profile** *profile-name*	Binds the IPSec protection profile to the tunnel interfaces. **Note** The *profile-name* defined in this step must match the profile name assigned to the virtual tunnel interface in Step 16 by using the **crypto ip sec profile** *profile-name* command.
Step 29	**no shutdown**	Brings the interface up, administratively.

EXAMPLE

Example 1: RSA Authentication

This example shows how to enable IKEv2 and then create a virtual IPSec tunnel when employing RSA authentication for both the Cisco CG-OS router and the head-end router.

This example configuration employs a Cisco ASR 1000 Series as the head-end router.

RSA mode is the system default setting for the Cisco CG-OS router.

Cisco CG-OS Router Configuration

Note When you use the system default for a parameter there is no need to enter the associated command. In the configuration below, the Cisco CG-OS router uses the default settings for authentication, encryption, hash algorithm, group, and lifetime seconds (Step 4 to Step 8).

These commands show how to enable and configure IKEv2 on the Cisco CG-OS router.

```
router# configure terminal
router(config)# feature crypto ike
router(config)# crypto ike domain ipsec
router(config-ike-IPSec)# policy 10
router(config-ike-ipsec-policy)# exit
router(config-ike-IPSec)# identity hostname
router(config-ike-IPSec)# exit
```

These commands show how to enable tunnelling on the router and then create a virtual IPSec tunnel (Tunnel 0) and then define profiles for that tunnel.

Note In the configuration below, the connected grid router uses the default settings for the **set security-association lifetime** *seconds kilobytes* parameter (Step 19).

```
router(config)# feature tunnel
router(config)# feature crypto ipsec virtual-tunnel
router(config)# crypto ipsec transform-set domain AES128SHA1 esp-aes 128 esp-sha1-hmac
router(config)# crypto ipsec profile MyIPSecProfile
router(config-ipsec-profile)# description IPSec profile for Tunnel 0
router(config-ipsec-profile)# set transform-set AES128SHA1
router(config-ipsec-profile)# exit
router(config)# interface Tunnel 0
router(config-if)# ip address 192.168.170.10/24
router(config-if)# tunnel mode ipsec ipv4
router(config-if)# tunnel source ethernet 2/7
router(config-if)# tunnel destination 172.27.170.23
router(config-if)# description IPSec to HER_01
router(config-if)# tunnel protection ipsec profile MyIPSecProfile
router(config-if)# no shutdown
router(config-if)# exit

Head-End Router Configuration (Cisco ASR 1000 Series with Cisco IOS)

crypto ikev2 proposal MyIke2Proposal
encryption aes-cbc-128
integrity sha1
group 2
```

```
crypto ikev2 policy MyIKEPolicy
proposal MyIke2Proposal

crypto ikev2 profile MyIke2Profile_cgr
match fvrf any
match identity remote fqdn cgr01
identity local fqdn her01
authentication local rsa-sig
authentication remote rsa-sig
pki trustpoint jamesRA_MSCA2008

crypto ipsec transform-set AES128SHA1 esp-aes esp-sha-hmac
```

Note Any Cisco IOS router configured as the head-end router must be configured as **responder-only** as shown in the configuration section below.

Note Cisco recommends the **set security-association lifetime kilobytes** and **seconds** values set in the procedure below to protect against connection tear-downs.

```
crypto ipsec profile IPSecProfile_altamont
set security-association lifetime kilobytes 4294967295
set security-association lifetime seconds 86400
set transform-set AES128SHA1
set ike-profile MyIke2Profile_altamont
responder-only

interface Tunnel 1
description IPSec to head_end_rtr01
ip address 192.168.170.20 255.255.255.0
tunnel source GigabitEthernet0/0
tunnel mode ipsec ipv4
tunnel destination 172.27.170.20
```

Example 2: PSK Authentication

This example shows how to enable IKEv2 and then create a virtual IPSec tunnel employing pre-shared key (PSK) for authentication between the Cisco CG-OS router and the head-end router. This example configuration employs a Cisco ASR 1000 Series router as the head-end router.

Connected Grid Router Configuration

These commands show how to enable and configure IKEv2 on the Cisco CG-OS router.

```
router# configure terminal
router(config)# feature crypto ike
router(config)# crypto ike domain ipsec
router(config-ike-IPSec)# policy 10
router(config-ike-ipsec-policy)# authentication pre-share
router(config-ike-ipsec-policy)# lifetime seconds 600
router(config-ike-IPSec)# keepalive 120
router(config-ike-IPSec)# identity hostname
router(config-ike-IPSec)# key $ecreT1254 hostname brklyn
router(config-ike-IPSec)# key $ecreT1254 address 192.168.150.20
router(config-ike-IPSec)# exit

router(config)# feature tunnel
router(config)# feature crypto ipsec virtual-tunnel
```

```
router(config)# crypto ipsec transform-set AES128SHA1 esp-aes 128 esp-sha1-hmac
router(config)# crypto ipsec profile MyProfile
router(config-ipsec-profile)# set transform-set AES128SHA1
router(config-ipsec-profile)# exit

router(config)# interface Tunnel 0
router(config-if)# ip address 192.168.40.10/24
router(config-if)# tunnel mode ipsec ipv4
router(config-if)# tunnel source ethernet 2/1
router(config-if)# tunnel destination 192.168.150.20
router(config-if)# tunnel protection ipsec profile MyProfile
router(config-if)# no shutdown
router(config-if)# exit
router(config)#
```

Head-End Router Configuration (Cisco ASR 1000 Series with Cisco IOS)

```
crypto ikev2 proposal MyIke2Proposal
encryption aes-cbc-128
integrity sha1
group 2
!
crypto ike policy MyIKEPolicy
proposal MyIke2Proposal

crypto ikev2 keyring MyIke2KeyRing
peer cgr2-Milan3
 address 192.168.191.30
 pre-shared-key Cisco123

crypto ikev2 profile MyIke2Profile
match fvrf any
match identity remote fqdn HER_2
identity local fqdn IOL100
authentication local pre-share
authentication remote pre-share
keyring MyIke2KeyRing

crypto ipsec transform-set AES128SHA1 esp-aes esp-sha-hmac

crypto ipsec profile IPSecProfile
set security-association lifetime kilobytes 4294967295
set security-association lifetime seconds 86400
set transform-set AES128SHA1
set ike-profile MyIke2Profile
responder-only

interface Tunnel0
ip address 192.168.40.20 255.255.255.0
tunnel source Ethernet0/0
tunnel mode ipsec ipv4
tunnel destination 192.168.191.30
tunnel protection ipsec profile IPSecProfile
```

Verifying the Configuration

To display IKEv2 and IPSec configurations, enter any or all of the following commands.

Command	Purpose
show crypto ike domain ipsec	Displays the current IKEv2 configuration.
show crypto ike domain ipsec policy	Displays all configured IKEv2 policies.
show crypto ipsec profile [*profile name*]	Displays all configured IPSec profiles or a specific IPSec profile.
show crypto ipsec security-association	Displays all configured IPSec security associations.
show crypto ipsec transform-set	Displays all configured IPSec transform-sets.

Clear Commands

To clear the IKE security associations, enter the following command.

Command	Purpose
clear crypto ike domain ipsec sa	Clears all IKEv2 security associations. **Caution** Entering this command brings down all IPSec tunnels.

Monitoring Statistics

To display IKEv2 and IPSec statistics, refer to the commands summarized in Verifying the Configuration.

Debug Commands

To troubleshoot IKEv2 and IPSec configurations, you can use the following commands.

Command	Purpose
debug ike event	Enables debugging for IKE event generation.
debug ike protocol	Enables debugging for IKE protocol.
debug ike message	Enables debugging for IKE messages.
debug ipsec_tun trace	Enables debugging for IPSec tunnel traces.
debug ipsec_tun packet	Enables debugging for IPSec tunnel packets.

The following is example output for IKEv2 debug commands:

```
cgr1000(config-if)# debug ike event
cgr1000(config-if)# debug ike protocol
cgr1000(config-if)# debug ike message
cgr1000(config-if)# no shut
```

```
2014 Jun 27 17:00:44.328029 ike: Processing PF_KEY message
2014 Jun 27 17:00:44.328264 ike: mts_handle_pfkey: initiating a IKEV2 tunnel
2014 Jun 27 17:00:44.328316 ike: ike_pfkey_handler: get pfkey ACQUIRE message
2014 Jun 27 17:00:44.328364 ike: ike_pk_recvacquire: Creating new IKE_SA as a result of an
acquire message for seqnum 26.
2014 Jun 27 17:00:44.328405 ike: create_ike_sa: for seq_num 26 tunnel_id 2
2014 Jun 27 17:00:44.328467 ike: create_ike_sa: ike_sa successfully created for seq_num 26
for doi 0
2014 Jun 27 17:00:44.328517 ike: ike_state_init: State initialized to IKE_STATE_INIT.
2014 Jun 27 17:00:44.328662 ike: ike_state_change: State changed from IKE_STATE_INIT to
IKE_STATE_DOI_SA_REQ_RCVD.
2014 Jun 27 17:00:44.328763 ike: ** Dumping ike_info **
2014 Jun 27 17:00:44.328801 ike: { ike_fsm_type: IKE_FSM_IKE_SA_INITIATOR state:
IKE_STATE_DOI_SA_REQ_RCVD seq_num: 26 tmp_tx_id: 0 }
2014 Jun 27 17:00:44.328885 ike: ** Dumping ike_sa_info **
2014 Jun 27 17:00:44.328924 ike: { doi_val: 0 ike_tunnel_id: 2  direction: IKE_INITIATOR
status: IKE_SA_STATUS_NONE  local_addr: 172.27.126.42[500]  remote_addr:
172.27.126.172[500] i_spi: 0000000000000000 r_spi: 0000000000000000 }
2014 Jun 27 17:00:44.328962 ike: { my_curr_req_msg_id: 0  my_next_req_msg_id: 0
peer_curr_req_msg_id: 0 peer_next_req_msg_id: 0 num_tries: 0 }
2014 Jun 27 17:00:44.329012 ike: ike_state_change: State changed from
IKE_STATE_DOI_SA_REQ_RCVD to IKE_STATE_INIT_REQ_PREP_WAIT.
2014 Jun 27 17:00:44.329101 ike: ** Dumping ike_info **
2014 Jun 27 17:00:44.329137 ike: { ike_fsm_type: IKE_FSM_IKE_SA_INITIATOR state:
IKE_STATE_INIT_REQ_PREP_WAIT seq_num: 26 tmp_tx_id: 0 }
2014 Jun 27 17:00:44.329174 ike: ** Dumping ike_sa_info **
2014 Jun 27 17:00:44.329206 ike: { doi_val: 0 ike_tunnel_id: 2  direction: IKE_INITIATOR
status: IKE_SA_STATUS_NONE  local_addr: 172.27.126.42[500]  remote_addr:
172.27.126.172[500] i_spi: 0000000000000000 r_spi: 0000000000000000 }
2014 Jun 27 17:00:44.329245 ike: { my_curr_req_msg_id: 0  my_next_req_msg_id: 0
peer_curr_req_msg_id: 0 peer_next_req_msg_id: 0 num_tries: 0 }
2014 Jun 27 17:00:44.329297 ike: ike_set_dh_keys:
2014 Jun 27 17:00:44.391644 ike: ike_state_change: State changed from
IKE_STATE_INIT_REQ_PREP_WAIT to IKE_STATE_INIT_RSP_WAIT.
2014 Jun 27 17:00:44.391783 ike: ** Dumping ike_info **
2014 Jun 27 17:00:44.391821 ike: { ike_fsm_type: IKE_FSM_IKE_SA_INITIATOR state:
IKE_STATE_INIT_RSP_WAIT seq_num: 26 tmp_tx_id: 0 }
2014 Jun 27 17:00:44.391858 ike: ** Dumping ike_sa_info **
2014 Jun 27 17:00:44.391892 ike: { doi_val: 0 ike_tunnel_id: 2  direction: IKE_INITIATOR
status: IKE_SA_STATUS_NONE  local_addr: 172.27.126.42[500]  remote_addr:
172.27.126.172[500] i_spi: 0000000000000000 r_spi: 0000000000000000 }
2014 Jun 27 17:00:44.391932 ike: { my_curr_req_msg_id: 0  my_next_req_msg_id: 0
peer_curr_req_msg_id: 0 peer_next_req_msg_id: 0 num_tries: 0 }
2014 Jun 27 17:00:44.392038 ike: ike_msg_add_sa: Invoked
2014 Jun 27 17:00:44.392080 ike: ike_msg_add_prop: proto IKE, prop_no 1
2014 Jun 27 17:00:44.392119 ike: ike_msg_prop_add_encr: add AES-CBC
2014 Jun 27 17:00:44.392154 ike: ike_msg_prop_add_encr: add key len 16 (bytes) for AES-CBC
2014 Jun 27 17:00:44.392189 ike: ike_msg_prop_add_prf: add HMAC-SHA1
2014 Jun 27 17:00:44.392223 ike: ike_msg_prop_add_auth: add HMAC-SHA1-96
2014 Jun 27 17:00:44.392258 ike: ike_msg_prop_add_dhg: add dhg MODP-1024
2014 Jun 27 17:00:44.392291 ike: ike_msg_add_prop: num_trans 4
2014 Jun 27 17:00:44.392326 ike: ike_msg_add_ke:
2014 Jun 27 17:00:44.392405 ike: ike_msg_add_nonce:
2014 Jun 27 17:00:44.392450 ike: --------------- IKE packet info (START) -------------
2014 Jun 27 17:00:44.392504 ike: i_spi: 4ca3c52580808d70 , r_spi: 0000000000000000  np:
SA, version: 32, etype: IKE_SA_INIT r_bit: 0, v_bit: 0, i_bit: 1 msg_id: 0, len:232
2014 Jun 27 17:00:44.392548 ike:  PAYLOAD: SA  np: KE, critical: 1, len: 48
2014 Jun 27 17:00:44.392587 ike:     np: NONE, len: 44, prop_no: 1, proto_id: 1,
spi_size: 0, num_trans: 4
2014 Jun 27 17:00:44.392627 ike:       np: TRANS, len: 12, type: Encryption Algorithm, id:
AES-CBC
2014 Jun 27 17:00:44.392664 ike:       np: TRANS, len: 8, type: Pseudo-random Function,
id: HMAC-SHA1
```

```
2014 Jun 27 17:00:44.392700 ike:       np: TRANS, len: 8, type: Integrity Algorithm, id:
HMAC-SHA1-96
2014 Jun 27 17:00:44.392736 ike:       np: NONE, len: 8, type: Diffie-Hellman Group, id:
MODP-1024
2014 Jun 27 17:00:44.392773 ike:  PAYLOAD: KE  np: NONCE, critical: 1, len: 136
2014 Jun 27 17:00:44.392927 ike:       dhg_id: 2 with key as follow:
51d9166bd30faa2a 815cd8bf8cdaa022 22f5dabd122a4c64 cd123cdf684abb04 2618808b5338d7a4
cdd137e407f6d6de a457d9934a6e33c5 3cd746ea94b5df58 17dd182ff069ecd3 a6d4319cc98ae87a
6c9034a50b36ff38 3e952d04df67c9ae b8cb8e89a21c2f1a e1a0fb087f142149
2014 Jun 27 17:00:44.392966 ike:  PAYLOAD: NONCE  np: NONE, critical: 1, len: 20
2014 Jun 27 17:00:44.393018 ike:    nonce: 917480a28b191e7a 8fbb2a1e892d56ac
2014 Jun 27 17:00:44.393051 ike: --------------- IKE packet info (END) --------------
2014 Jun 27 17:00:44.394786 ike: Send message (232 requested) of 260 bytes from
172.27.126.42:500 to 172.27.126.172:500
2014 Jun 27 17:00:44.394929 ike: ** Dumping ike_info **
2014 Jun 27 17:00:44.394974 ike: { ike_fsm_type: IKE_FSM_IKE_SA_INITIATOR state:
IKE_STATE_INIT_RSP_WAIT seq_num: 26 tmp_tx_id: 0 }
2014 Jun 27 17:00:44.395013 ike: ** Dumping ike_sa_info **
2014 Jun 27 17:00:44.395047 ike: { doi_val: 0 ike_tunnel_id: 2  direction: IKE_INITIATOR
status: IKE_SA_STATUS_NONE  local_addr: 172.27.126.42[500]  remote_addr:
172.27.126.172[500] i_spi: 4ca3c52580808d70 r_spi: 0000000000000000 }
2014 Jun 27 17:00:44.395087 ike: { my_curr_req_msg_id: 0  my_next_req_msg_id: 0
peer_curr_req_msg_id: 0 peer_next_req_msg_id: 0 num_tries: 1 }
2014 Jun 27 17:00:44.395129 ike: fsm_action_send_init_req: waiting for IKE_SA_INIT
response.
2014 Jun 27 17:00:44.395256 ike: Processing PF_KEY message
2014 Jun 27 17:00:44.395356 ike: mts_handle_pfkey: initiating a IKEV2 tunnel
2014 Jun 27 17:00:44.395401 ike: ike_pfkey_handler: get pfkey ACQUIRE message
2014 Jun 27 17:00:44.400966 ike: Recv message of 341 bytes from 172.27.126.172:500 to
172.27.126.42:500
2014 Jun 27 17:00:44.401085 ike: ike_parse_msg_pl: passed IKE sa
2014 Jun 27 17:00:44.401142 ike: ike_parse_msg_pl: recv message:
2014 Jun 27 17:00:44.401244 ike:         0-      31: 4ca3c52580808d70 4f58e388623f1727
2120222000000000 0000013922000030
2014 Jun 27 17:00:44.401351 ike:        32-      63: 0000002c01010004 0300000c0100000c
800e008003000008 0200000203000008
2014 Jun 27 17:00:44.401453 ike:        64-      95: 0300000200000008 0400000228000088
0002000086c83994 fc962bdb265c9ca6
2014 Jun 27 17:00:44.401552 ike:        96-     127: ecc1b1a21d60ab65 6f4da753b017a6ba
9355d12fa8c6e440 a55f715059c5d6ae
2014 Jun 27 17:00:44.401653 ike:       128-     159: af7859b73dd0dc98 b922913de4469903
2579c86677fcd03d e1ee0740c3a65cef
2014 Jun 27 17:00:44.401753 ike:       160-     191: c87d6ee5545cacd2 17622f1ed2d1d115
00ec9ae998be090e 38a188028b884ae7
2014 Jun 27 17:00:44.401848 ike:       192-     223: 6351409fb0ddb793 2147eb993c1000c0
9e6780262b000018 9a4bec5923892dbb
2014 Jun 27 17:00:44.401943 ike:       224-     255: ead5fc095441e467 c3aca6262b000017
434953434f2d4445 4c4554452d524541
2014 Jun 27 17:00:44.402039 ike:       256-     287: 534f4e2600001546 4c455856504e2d53
5550504f52544544 290000190c70ce9b
2014 Jun 27 17:00:44.402123 ike:          -     313: c23f4b229445a86e b299f2d02bdadd18
ca00000008010040 08
2014 Jun 27 17:00:44.402175 ike: ike_parse_msg_pl: un-encrypted payload
2014 Jun 27 17:00:44.402229 ike: --------------- IKE packet info (START) -------------
2014 Jun 27 17:00:44.402305 ike: i_spi: 4ca3c52580808d70 , r_spi: 4f58e388623f1727  np:
SA, version: 32, etype: IKE_SA_INIT r_bit: 1, v_bit: 0, i_bit: 0 msg_id: 0, len:313
2014 Jun 27 17:00:44.402370 ike:  PAYLOAD: SA  np: KE, critical: 0, len: 48
2014 Jun 27 17:00:44.402432 ike:     np: NONE, len: 44, prop_no: 1, proto_id: 1,
spi_size: 0, num_trans: 4
2014 Jun 27 17:00:44.402491 ike:       np: TRANS, len: 12, type: Encryption Algorithm, id:
AES-CBC
2014 Jun 27 17:00:44.402545 ike:       np: TRANS, len: 8, type: Pseudo-random Function,
id: HMAC-SHA1
```

```
2014 Jun 27 17:00:44.402596 ike:       np: TRANS, len: 8, type: Integrity Algorithm, id:
HMAC-SHA1-96
2014 Jun 27 17:00:44.403814 ike:       np: NONE, len: 8, type: Diffie-Hellman Group, id:
MODP-1024
2014 Jun 27 17:00:44.403902 ike:  PAYLOAD: KE  np: NONCE, critical: 0, len: 136
2014 Jun 27 17:00:44.404139 ike:       dhg_id: 2 with key as follow:
86c83994fc962bdb 265c9ca6ecc1b1a2 1d60ab656f4da753 b017a6ba9355d12f a8c6e440a55f7150
59c5d6aeaf7859b7 3dd0dc98b922913d e44699032579c866 77fcd03de1ee0740 c3a65cefc87d6ee5
545cacd217622f1e d2d1d11500ec9ae9 98be090e38a18802 8b884ae76351409f
2014 Jun 27 17:00:44.404202 ike:  PAYLOAD: NONCE  np: VENDOR-ID, critical: 0, len: 24
2014 Jun 27 17:00:44.404290 ike:    nonce: 9a4bec5923892dbb ead5fc095441e467 c3aca626
2014 Jun 27 17:00:44.404344 ike:    PAYLOAD: VENDOR-ID   np: VENDOR-ID, critical: 0, len:
23
2014 Jun 27 17:00:44.404397 ike:    skip np: VENDOR-ID
2014 Jun 27 17:00:44.404439 ike:  PAYLOAD: VENDOR-ID  np: CERTREQ, critical: 0, len: 21
2014 Jun 27 17:00:44.404475 ike:    skip np: VENDOR-ID
2014 Jun 27 17:00:44.404509 ike:  PAYLOAD: CERTREQ  np: NOTIF, critical: 0, len: 25
2014 Jun 27 17:00:44.404544 ike:    skip np: CERTREQ
2014 Jun 27 17:00:44.404576 ike:  PAYLOAD: NOTIF  np: NONE, critical: 0, len: 8
2014 Jun 27 17:00:44.404613 ike:       proto_id=IKE, spi_size=0, spi=,
type=HTTP-CERT-LOOKUP-SUPPORTED
2014 Jun 27 17:00:44.404649 ike: ---------------- IKE packet info (END) --------------
2014 Jun 27 17:00:44.404684 ike: ike_process_pl: SA
2014 Jun 27 17:00:44.404722 ike: ike_parse_pl_sa: new prop, 1
2014 Jun 27 17:00:44.404771 ike: ike_parse_pl_trans: AES-CBC
2014 Jun 27 17:00:44.404808 ike: ike_parse_pl_trans: key len 16 bytes
2014 Jun 27 17:00:44.404849 ike: ike_parse_pl_trans: HMAC-SHA1
2014 Jun 27 17:00:44.404889 ike: ike_parse_pl_trans: HMAC-SHA1-96
2014 Jun 27 17:00:44.404928 ike: ike_parse_pl_trans: MODP-1024
2014 Jun 27 17:00:44.404968 ike: ike_process_pl: KE
2014 Jun 27 17:00:44.405013 ike: ike_process_pl: NONCE
2014 Jun 27 17:00:44.405055 ike: ike_process_pl: VENDOR-ID
2014 Jun 27 17:00:44.405092 ike: ike_process_pl: VENDOR-ID
2014 Jun 27 17:00:44.405125 ike: ike_process_pl: CERTREQ
2014 Jun 27 17:00:44.405163 ike: ike_process_pl: NOTIF
2014 Jun 27 17:00:44.405199 ike: ike_compose_notif_info: proto_id 1, spi_size 0,
notif_type 16392, data_len 0
2014 Jun 27 17:00:44.405243 ike: ike_process_notif_list: for IKE sa
2014 Jun 27 17:00:44.405279 ike: ike_process_notif: process notif --
HTTP-CERT-LOOKUP-SUPPORTED
2014 Jun 27 17:00:44.405318 ike: process_ike_sa_init_rsp: IKE_SA_INIT response okay, start
preparing AUTH_REQ
2014 Jun 27 17:00:44.405353 ike: ike_save_sainfo: ignore update request since ike_sa
status is IKE_SA_STATUS_NONE
2014 Jun 27 17:00:44.405412 ike: ike_state_change: State changed from
IKE_STATE_INIT_RSP_WAIT to IKE_STATE_AUTH_REQ_PREP_WAIT.
2014 Jun 27 17:00:44.405510 ike: ** Dumping ike_info **
2014 Jun 27 17:00:44.405546 ike: { ike_fsm_type: IKE_FSM_IKE_SA_INITIATOR state:
IKE_STATE_AUTH_REQ_PREP_WAIT seq_num: 26 tmp_tx_id: 0 }
2014 Jun 27 17:00:44.405581 ike: ** Dumping ike_sa_info **
2014 Jun 27 17:00:44.405614 ike: { doi_val: 0 ike_tunnel_id: 2  direction: IKE_INITIATOR
status: IKE_SA_STATUS_NONE  local_addr: 172.27.126.42[500]  remote_addr:
172.27.126.172[500] i_spi: 4ca3c52580808d70 r_spi: 4f58e388623f1727 }
2014 Jun 27 17:00:44.405651 ike: { my_curr_req_msg_id: 0  my_next_req_msg_id: 1
peer_curr_req_msg_id: 0 peer_next_req_msg_id: 0 num_tries: 1 }
2014 Jun 27 17:00:44.405698 ike: ike_generate_ike_keys:
2014 Jun 27 17:00:44.405757 ike: ike_set_dh_shared_keys:
2014 Jun 27 17:00:44.483289 ike: ike_generate_keymat: spi_i: 4ca3c52580808d70
2014 Jun 27 17:00:44.483366 ike: ike_generate_keymat: spi_r: 4f58e388623f1727
2014 Jun 27 17:00:44.484067 ike: fqdn_2_id_info: domainname:
2014 Jun 27 17:00:44.484222 ike: fqdn_2_id_info: hostname: cgr1000
2014 Jun 27 17:00:44.484271 ike: ike_auth_message: RSA signatures
2014 Jun 27 17:00:44.615705 ike: Waiting for GETSPD response from DOI(0) for seq_num 26
2014 Jun 27 17:00:44.615794 ike: ike_handle_msg: message ref saved, cannot be freed
```

```
2014 Jun 27 17:00:44.616276 ike: Processing PF_KEY message
2014 Jun 27 17:00:44.616593 ike: get pfkey X_SPDGET2 message
2014 Jun 27 17:00:44.616799 ike: Couldn't find ph2 handle matching spdget2.
2014 Jun 27 17:00:44.616854 ike: ike_pfkey_handler: get pfkey SADB_X_SPDGET2 message
2014 Jun 27 17:00:44.616915 ike: construct_prop_list: Adding proposal: proto(3)
encr_id(12) encr_key_len(16 Bytes) auth_id(2) auth_key_len(20 Bytes)
2014 Jun 27 17:00:44.616959 ike: call pfkey_send_getspi for proto_id 3 for doi 0
2014 Jun 27 17:00:44.617138 ike: GETSPI sent: IPSEC-ESP 172.27.126.42->172.27.126.172
2014 Jun 27 17:00:44.617178 ike: ike_post_getspd_ex: waiting for getspi response
2014 Jun 27 17:00:44.617294 ike: Processing PF_KEY message
2014 Jun 27 17:00:44.617515 ike: get pfkey X_SPDGET2 message
2014 Jun 27 17:00:44.617730 ike: ike_pfkey_handler: get pfkey SADB_X_SPDGET2 message
2014 Jun 27 17:00:44.618322 ike: Processing PF_KEY message
2014 Jun 27 17:00:44.618567 ike: get pfkey GETSPI message
2014 Jun 27 17:00:44.618767 ike: seq 26 of GETSPI message not interesting.
2014 Jun 27 17:00:44.618817 ike: ike_pfkey_handler: get pfkey GETSPI message
2014 Jun 27 17:00:44.618910 ike: pfkey GETSPI succeeded: IPSEC-ESP
172.27.126.42->172.27.126.172 spi=3187237707(0xbdf9634b)
2014 Jun 27 17:00:44.618973 ike: ike_state_change: State changed from
IKE_STATE_AUTH_REQ_PREP_WAIT to IKE_STATE_AUTH_RSP_WAIT.
2014 Jun 27 17:00:44.619106 ike: ** Dumping ike_info **
2014 Jun 27 17:00:44.619155 ike: { ike_fsm_type: IKE_FSM_IKE_SA_INITIATOR state:
IKE_STATE_AUTH_RSP_WAIT seq_num: 26 tmp_tx_id: 0 }
2014 Jun 27 17:00:44.619207 ike: ** Dumping ike_sa_info **
2014 Jun 27 17:00:44.619255 ike: { doi_val: 0 ike_tunnel_id: 2  direction: IKE_INITIATOR
status: IKE_SA_STATUS_UNAUTH  local_addr: 172.27.126.42[500]  remote_addr:
172.27.126.172[500] i_spi: 4ca3c52580808d70 r_spi: 4f58e388623f1727 }
2014 Jun 27 17:00:44.619315 ike: { my_curr_req_msg_id: 0  my_next_req_msg_id: 1
peer_curr_req_msg_id: 0 peer_next_req_msg_id: 0 num_tries: 1 }
2014 Jun 27 17:00:44.619387 ike: ike_msg_add_idi:
2014 Jun 27 17:00:44.619440 ike: ike_msg_add_cert:
2014 Jun 27 17:00:44.619489 ike: ike_enlarge_buf_if_needed: increase the buffer size
2014 Jun 27 17:00:44.619553 ike: ike_msg_add_certreq:
2014 Jun 27 17:00:44.619601 ike: ike_msg_add_auth:
2014 Jun 27 17:00:44.619652 ike: ike_msg_add_sa: Invoked
2014 Jun 27 17:00:44.619702 ike: ike_enlarge_buf_if_needed: increase the buffer size
2014 Jun 27 17:00:44.619757 ike: ike_msg_add_prop: proto IPSEC-ESP, prop_no 1
2014 Jun 27 17:00:44.619809 ike: ike_msg_prop_add_encr: add AES-CBC
2014 Jun 27 17:00:44.619859 ike: ike_msg_prop_add_encr: add key len 16 (bytes) for AES-CBC
2014 Jun 27 17:00:44.619909 ike: ike_msg_prop_add_auth: add HMAC-SHA1-96
2014 Jun 27 17:00:44.619963 ike: ike_msg_prop_add_esn: add esn 0
2014 Jun 27 17:00:44.620010 ike: ike_msg_add_prop: num_trans 3
2014 Jun 27 17:00:44.620071 ike: ike_msg_add_ts: TSi (0.0.0.0:0 -- 255.255.255.255:65535)
2014 Jun 27 17:00:44.620134 ike: ike_msg_add_ts: TSr (0.0.0.0:0 -- 255.255.255.255:65535)
2014 Jun 27 17:00:44.620188 ike: ike_msg_add_IC_notif:
2014 Jun 27 17:00:44.620238 ike: ike_msg_add_notif: proto_id 1, spi_size 0, notif_type
INITIAL-CONTACT, data_len 0
2014 Jun 27 17:00:44.620296 ike: ike_msg_encrypt:
2014 Jun 27 17:00:44.620345 ike: NOT PRINTED: size (1309), maximum (1000)
2014 Jun 27 17:00:44.620396 ike: ---------------- IKE packet info (START) -------------
2014 Jun 27 17:00:44.620471 ike: i_spi: 4ca3c52580808d70 , r_spi: 4f58e388623f1727  np:
IDi, version: 32, etype: IKE_AUTH r_bit: 0, v_bit: 0, i_bit: 1 msg_id: 1, len:1309
2014 Jun 27 17:00:44.620531 ike:  PAYLOAD: IDi  np: CERT, critical: 1, len: 19
2014 Jun 27 17:00:44.620582 ike:      type: FQDN
2014 Jun 27 17:00:44.620653 ike:  PAYLOAD: CERT  np: CERTREQ, critical: 1, len: 873
2014 Jun 27 17:00:44.620699 ike:    skip np: CERT
2014 Jun 27 17:00:44.620733 ike:  PAYLOAD: CERTREQ  np: AUTH, critical: 1, len: 25
2014 Jun 27 17:00:44.620769 ike:    skip np: CERTREQ
2014 Jun 27 17:00:44.620803 ike:  PAYLOAD: AUTH  np: SA, critical: 1, len: 264
2014 Jun 27 17:00:44.620838 ike:     type: RSA signatures
2014 Jun 27 17:00:44.620871 ike:  PAYLOAD: SA  np: TSi, critical: 1, len: 44
2014 Jun 27 17:00:44.620909 ike:      np: NONE, len: 40, prop_no: 1, proto_id: 3,
spi_size: 4, num_trans: 3
```

```
2014 Jun 27 17:00:44.620949 ike:       np: TRANS, len: 12, type: Encryption Algorithm, id:
AES-CBC
2014 Jun 27 17:00:44.620986 ike:       np: TRANS, len: 8, type: Integrity Algorithm, id:
HMAC-SHA1-96
2014 Jun 27 17:00:44.621025 ike:       np: NONE, len: 8, type: Extended Sequence Numbers,
id: 0
2014 Jun 27 17:00:44.621062 ike:  PAYLOAD: TSi  np: TSr, critical: 1, len: 24
2014 Jun 27 17:00:44.621097 ike:     num_ts: 1
2014 Jun 27 17:00:44.621138 ike:         TS[1]: type=IPv4_addr_range, proto_id=0, len=16
start_port=0, end_port=65535, start_ip=0.0.0.0 end_ip=255.255.255.255
2014 Jun 27 17:00:44.621178 ike:  PAYLOAD: TSr  np: NOTIF, critical: 1, len: 24
2014 Jun 27 17:00:44.621214 ike:     num_ts: 1
2014 Jun 27 17:00:44.621253 ike:         TS[1]: type=IPv4_addr_range, proto_id=0, len=16
start_port=0, end_port=65535, start_ip=0.0.0.0 end_ip=255.255.255.255
2014 Jun 27 17:00:44.621293 ike:  PAYLOAD: NOTIF  np: NONE, critical: 1, len: 8
2014 Jun 27 17:00:44.621331 ike:       proto_id=IKE, spi_size=0, spi=,
type=INITIAL-CONTACT
2014 Jun 27 17:00:44.621367 ike: --------------- IKE packet info (END) --------------
2014 Jun 27 17:00:44.621401 ike: ike_msg_encrypt: AES-CBC with key_len of 16 bytes
2014 Jun 27 17:00:44.621524 ike: ike_msg_encrypt: padding length 14, total 1296 bytes to
be encrypted
2014 Jun 27 17:00:44.621701 ike: NOT PRINTED: size (1296), maximum (1000)
2014 Jun 27 17:00:44.621739 ike: ike_msg_encrypt: encryption output:
2014 Jun 27 17:00:44.621773 ike: NOT PRINTED: size (1296), maximum (1000)
2014 Jun 27 17:00:44.621807 ike: ike_msg_encrypt: encr_pl_len 1328, iv_len 16,
encry_out_len 1296, md_len 12
2014 Jun 27 17:00:44.621843 ike: ike_msg_encrypt: encr_msg->data_len 1356, encr_pl_len
1328
2014 Jun 27 17:00:44.621878 ike: ike_msg_encrypt: data (ike message) for checksum:
2014 Jun 27 17:00:44.621911 ike: NOT PRINTED: size (1344), maximum (1000)
2014 Jun 27 17:00:44.621945 ike: ike_integ_checksum:
2014 Jun 27 17:00:44.622098 ike: ike_msg_encrypt, checksum: e58a36a1d3a85aeb 2515cd4c
2014 Jun 27 17:00:44.622136 ike: ike_msg_encrypt: encrypted ike message follows
2014 Jun 27 17:00:44.622180 ike: NOT PRINTED: size (1356), maximum (1000)
2014 Jun 27 17:00:44.622247 ike: --------------- IKE packet info (START) -------------
2014 Jun 27 17:00:44.623472 ike: i_spi: 4ca3c52580808d70 , r_spi: 4f58e388623f1727  np:
ENCR, version: 32, etype: IKE_AUTH r_bit: 0, v_bit: 0, i_bit: 1 msg_id: 1, len:1356
2014 Jun 27 17:00:44.623517 ike:  PAYLOAD: ENCR  np: IDi, critical: 1, len: 1328
2014 Jun 27 17:00:44.623554 ike:    skip np: ENCR
2014 Jun 27 17:00:44.623586 ike: --------------- IKE packet info (END) --------------
2014 Jun 27 17:00:44.624855 ike: Send message (1356 requested) of 1384 bytes from
172.27.126.42:500 to 172.27.126.172:500
2014 Jun 27 17:00:44.624916 ike: fsm_action_send_auth_req: waiting for IKE_SA_AUTH
response.
2014 Jun 27 17:00:44.625052 ike: Processing PF_KEY message
2014 Jun 27 17:00:44.625331 ike: get pfkey GETSPI message
2014 Jun 27 17:00:44.625548 ike: ike_pfkey_handler: get pfkey GETSPI message
2014 Jun 27 17:00:44.639278 ike: Recv message of 1336 bytes from 172.27.126.172:500 to
172.27.126.42:500
2014 Jun 27 17:00:44.639368 ike: ike_parse_msg_pl: parsing for child sa (inc IKE-AUTH)
2014 Jun 27 17:00:44.639410 ike: ike_parse_msg_pl: recv message:
2014 Jun 27 17:00:44.639446 ike: NOT PRINTED: size (1308), maximum (1000)
2014 Jun 27 17:00:44.639479 ike: ike_parse_msg_pl: encrypted payload
2014 Jun 27 17:00:44.639513 ike: ike_msg_decrypt: AES-CBC with key_len of 16 bytes
2014 Jun 27 17:00:44.639565 ike: ike_msg_decrypt: data (ike message) for checksum follows
cgr1000(config-if)# 2014 Jun 27 17:00:44.639604 ike: NOT PRINTED: size (1296) too big to
print
2014 Jun 27 17:00:44.639637 ike: ike_integ_checksum:
2014 Jun 27 17:00:44.639826 ike: ike_msg_decrypt: checksum: dbfe24c9ae86af32 1707490c
2014 Jun 27 17:00:44.640003 ike: ike_msg_decrypt: decrypted payload:
2014 Jun 27 17:00:44.640041 ike: NOT PRINTED: size (1248), maximum (1000)
2014 Jun 27 17:00:44.640076 ike: ike_msg_decrypt: decrypted data (1248 bytes) with pad_len
5
```

```
2014 Jun 27 17:00:44.640111 ike: ike_msg_decrypt: orig_pl_len 1242 vs cal_len 1242,
decrypt_out_len 1248, pad_len 5
2014 Jun 27 17:00:44.640151 ike: ike_parse_msg_pl: decrypted message:
2014 Jun 27 17:00:44.640184 ike: NOT PRINTED: size (1270), maximum (1000)
2014 Jun 27 17:00:44.640219 ike: --------------- IKE packet info (START) -------------
2014 Jun 27 17:00:44.640271 ike: i_spi: 4ca3c52580808d70 , r_spi: 4f58e388623f1727  np:
VENDOR-ID, version: 32, etype: IKE_AUTH r_bit: 1, v_bit: 0, i_bit: 0 msg_id: 1, len:1270
2014 Jun 27 17:00:44.640314 ike:  PAYLOAD: VENDOR-ID  np: IDr, critical: 0, len: 20
2014 Jun 27 17:00:44.640349 ike:    skip np: VENDOR-ID
2014 Jun 27 17:00:44.640383 ike:  PAYLOAD: IDr  np: CERT, critical: 0, len: 12
2014 Jun 27 17:00:44.640428 ike:    type: IPv4-address, id: 172.27.126.172
2014 Jun 27 17:00:44.640464 ike:  PAYLOAD: CERT  np: AUTH, critical: 0, len: 826
2014 Jun 27 17:00:44.640500 ike:    skip np: CERT
2014 Jun 27 17:00:44.640534 ike:  PAYLOAD: AUTH  np: SA, critical: 0, len: 264
2014 Jun 27 17:00:44.640571 ike:     type: RSA signatures
2014 Jun 27 17:00:44.640605 ike:  PAYLOAD: SA  np: TSi, critical: 0, len: 44
2014 Jun 27 17:00:44.640643 ike:     np: NONE, len: 40, prop_no: 1, proto_id: 3,
spi_size: 4, num_trans: 3
2014 Jun 27 17:00:44.640684 ike:       np: TRANS, len: 12, type: Encryption Algorithm, id:
AES-CBC
2014 Jun 27 17:00:44.640721 ike:       np: TRANS, len: 8, type: Integrity Algorithm, id:
HMAC-SHA1-96
2014 Jun 27 17:00:44.640758 ike:       np: NONE, len: 8, type: Extended Sequence Numbers,
id: 0
2014 Jun 27 17:00:44.640795 ike:  PAYLOAD: TSi  np: TSr, critical: 0, len: 24
2014 Jun 27 17:00:44.640831 ike:     num_ts: 1
2014 Jun 27 17:00:44.640872 ike:         TS[1]: type=IPv4_addr_range, proto_id=0, len=16
start_port=0, end_port=65535, start_ip=0.0.0.0 end_ip=255.255.255.255
2014 Jun 27 17:00:44.640913 ike:  PAYLOAD: TSr  np: NOTIF, critical: 0, len: 24
2014 Jun 27 17:00:44.640948 ike:     num_ts: 1
2014 Jun 27 17:00:44.640988 ike:         TS[1]: type=IPv4_addr_range, proto_id=0, len=16
start_port=0, end_port=65535, start_ip=0.0.0.0 end_ip=255.255.255.255
2014 Jun 27 17:00:44.641027 ike:  PAYLOAD: NOTIF  np: NOTIF, critical: 0, len: 12
2014 Jun 27 17:00:44.641064 ike:       proto_id=IKE, spi_size=0, spi=,
type=SET-WINDOW-SIZE
2014 Jun 27 17:00:44.641100 ike:  PAYLOAD: NOTIF  np: NOTIF, critical: 0, len: 8
2014 Jun 27 17:00:44.641136 ike:       proto_id=IKE, spi_size=0, spi=,
type=ESP-TFC-PADDING-NOT-SUPPORTED
2014 Jun 27 17:00:44.641172 ike:  PAYLOAD: NOTIF  np: NONE, critical: 0, len: 8
2014 Jun 27 17:00:44.641209 ike:    proto_id=IKE, spi_size=0, spi=, type=16395
2014 Jun 27 17:00:44.641244 ike: --------------- IKE packet info (END) --------------
2014 Jun 27 17:00:44.641279 ike: ike_process_pl: VENDOR-ID
2014 Jun 27 17:00:44.641315 ike: ike_process_pl: IDr
2014 Jun 27 17:00:44.641356 ike: ike_process_pl: CERT
2014 Jun 27 17:00:44.641401 ike: ike_process_pl: AUTH
2014 Jun 27 17:00:44.641445 ike: ike_process_pl: SA
2014 Jun 27 17:00:44.641492 ike: ike_parse_pl_sa: new prop, 1
2014 Jun 27 17:00:44.641537 ike: ike_parse_pl_trans: AES-CBC
2014 Jun 27 17:00:44.641572 ike: ike_parse_pl_trans: key len 16 bytes
2014 Jun 27 17:00:44.641611 ike: ike_parse_pl_trans: HMAC-SHA1-96
2014 Jun 27 17:00:44.641651 ike: ike_parse_pl_trans: 0
2014 Jun 27 17:00:44.641691 ike: ike_process_pl: TSi
2014 Jun 27 17:00:44.641730 ike: ike_process_pl: TSr
2014 Jun 27 17:00:44.641769 ike: ike_process_pl: NOTIF
2014 Jun 27 17:00:44.641805 ike: ike_compose_notif_info: proto_id 1, spi_size 0,
notif_type 16385, data_len 4
2014 Jun 27 17:00:44.641849 ike: ike_process_pl: NOTIF
2014 Jun 27 17:00:44.641884 ike: ike_compose_notif_info: proto_id 1, spi_size 0,
notif_type 16394, data_len 0
2014 Jun 27 17:00:44.641923 ike: ike_process_pl: NOTIF
2014 Jun 27 17:00:44.641957 ike: ike_compose_notif_info: proto_id 1, spi_size 0,
notif_type 16395, data_len 0
2014 Jun 27 17:00:44.642002 ike: ike_process_notif_list: for child sa (inc IKE-AUTH)
2014 Jun 27 17:00:44.642041 ike: ike_process_notif: process notif -- 16395
```

```
2014 Jun 27 17:00:44.642076 ike: ike_process_notif: process notif --
ESP-TFC-PADDING-NOT-SUPPORTED
2014 Jun 27 17:00:44.642117 ike: ike_process_notif: process notif -- SET-WINDOW-SIZE
2014 Jun 27 17:00:44.642154 ike: ike_save_sainfo: ignore update request since ike_sa
status is IKE_SA_STATUS_UNAUTH
2014 Jun 27 17:00:44.642189 ike: ike_save_sainfo: ignore update request since ike_sa
status is IKE_SA_STATUS_UNAUTH
2014 Jun 27 17:00:44.642222 ike: process_ike_auth_rsp: IKE_AUTH response okay, start
processing AUTH rsp
2014 Jun 27 17:00:44.642267 ike: ike_state_change: State changed from
IKE_STATE_AUTH_RSP_WAIT to IKE_STATE_AUTH_RSP_PROC_WAIT.
2014 Jun 27 17:00:44.642360 ike: ** Dumping ike_info **
2014 Jun 27 17:00:44.642396 ike: { ike_fsm_type: IKE_FSM_IKE_SA_INITIATOR state:
IKE_STATE_AUTH_RSP_PROC_WAIT seq_num: 26 tmp_tx_id: 0 }
2014 Jun 27 17:00:44.642431 ike: ** Dumping ike_sa_info **
2014 Jun 27 17:00:44.642464 ike: { doi_val: 0 ike_tunnel_id: 2  direction: IKE_INITIATOR
status: IKE_SA_STATUS_UNAUTH  local_addr: 172.27.126.42[500]  remote_addr:
172.27.126.172[500] i_spi: 4ca3c52580808d70 r_spi: 4f58e388623f1727 }
2014 Jun 27 17:00:44.642503 ike: { my_curr_req_msg_id: 1  my_next_req_msg_id: 2
peer_curr_req_msg_id: 0 peer_next_req_msg_id: 1 num_tries: 1 }
2014 Jun 27 17:00:44.646214 ike: ike_handle_msg: message ref saved, cannot be freed
2014 Jun 27 17:00:44.739192 ike: mds_cert_rcb_process_response called
2014 Jun 27 17:00:44.739268 ike: response for a cert verify request
2014 Jun 27 17:00:44.739328 ike: ikev2_process_verify_cert_result: IKEv2 CA reqid matches,
processing. (1041917).
2014 Jun 27 17:00:44.739384 ike: ike_verify_auth: RSA signatures
2014 Jun 27 17:00:44.747497 ike: ike_verify_auth: verify success
2014 Jun 27 17:00:44.747608 ike: ike_generate_child_keys:
2014 Jun 27 17:00:44.747662 ike: ike_init_child_keys: auth HMAC-SHA1-96 key_len 20
2014 Jun 27 17:00:44.747722 ike: ike_init_child_keys: encr AES-CBC key_len 16
2014 Jun 27 17:00:44.747986 ike: ike_state_change: State changed from
IKE_STATE_AUTH_RSP_PROC_WAIT to IKE_STATE_NEW_IKESA_UPDATE_INITIATOR_SA_WAIT.
2014 Jun 27 17:00:44.748130 ike: ** Dumping ike_info **
2014 Jun 27 17:00:44.748181 ike: { ike_fsm_type: IKE_FSM_IKE_SA_INITIATOR state:
IKE_STATE_NEW_IKESA_UPDATE_INITIATOR_SA_WAIT seq_num: 26 tmp_tx_id: 0 }
2014 Jun 27 17:00:44.748232 ike: ** Dumping ike_sa_info **
2014 Jun 27 17:00:44.748282 ike: { doi_val: 0 ike_tunnel_id: 2  direction: IKE_INITIATOR
status: IKE_SA_STATUS_UNAUTH  local_addr: 172.27.126.42[500]  remote_addr:
172.27.126.172[500]  i_spi: 4ca3c52580808d70  r_spi: 4f58e388623f1727 }
2014 Jun 27 17:00:44.748334 ike: { my_curr_req_msg_id: 1  my_next_req_msg_id: 2
peer_curr_req_msg_id: 0  peer_next_req_msg_id: 1  num_tries: 1 }
2014 Jun 27 17:00:44.748401 ike: call pfkey_send_update for proto_id 3 for doi 0
2014 Jun 27 17:00:44.749716 ike: UPDATE sent: IPSEC-ESP 172.27.126.42->172.27.126.172
2014 Jun 27 17:00:44.749788 ike: ike_pk_sendupdate: waiting for update response
2014 Jun 27 17:00:44.749960 ike: Processing PF_KEY message
2014 Jun 27 17:00:44.750266 ike: get pfkey UPDATE message
2014 Jun 27 17:00:44.750532 ike: seq 26 of UPDATE message not interesting.
2014 Jun 27 17:00:44.750600 ike: ike_pfkey_handler: get pfkey UPDATE message
2014 Jun 27 17:00:44.750697 ike: pfkey UPDATE succeeded: IPSEC-ESP
172.27.126.172->172.27.126.42 spi=3187237707(0xbdf9634b)
2014 Jun 27 17:00:44.750756 ike: call pfkey_send_add for proto_id 3 for doi 0
2014 Jun 27 17:00:44.751952 ike: ADD sent: IPSEC-ESP 172.27.126.42->172.27.126.172
2014 Jun 27 17:00:44.752029 ike: ike_pk_sendadd: waiting for add response
2014 Jun 27 17:00:44.752200 ike: Processing PF_KEY message
2014 Jun 27 17:00:44.752513 ike: get pfkey UPDATE message
2014 Jun 27 17:00:44.752787 ike: seq 26 of UPDATE message not interesting.
2014 Jun 27 17:00:44.752855 ike: ike_pfkey_handler: get pfkey UPDATE message
2014 Jun 27 17:00:44.753058 ike: Processing PF_KEY message
2014 Jun 27 17:00:44.753335 ike: get pfkey ADD message
2014 Jun 27 17:00:44.753598 ike: seq 26 of ADD message not interesting.
2014 Jun 27 17:00:44.753667 ike: ike_pfkey_handler: get pfkey ADD message
2014 Jun 27 17:00:44.753770 ike: pfkey ADD succeeded: IPSEC-ESP
172.27.126.172->172.27.126.42 spi=3868651307(0xe696ef2b)
```

```
2014 Jun 27 17:00:44.753842 ike: ike_timer_init: for state(IKE_STATE_ESTABLISHED), state
timeout set to 81176 sec
2014 Jun 27 17:00:44.753902 ike: ike_state_change: State changed from
IKE_STATE_NEW_IKESA_UPDATE_INITIATOR_SA_WAIT to IKE_STATE_ESTABLISHED.
2014 Jun 27 17:00:44.754041 ike: ** Dumping ike_info **
2014 Jun 27 17:00:44.754094 ike: { ike_fsm_type: IKE_FSM_IKE_SA_INITIATOR state:
IKE_STATE_ESTABLISHED seq_num: 26 tmp_tx_id: 0 }
2014 Jun 27 17:00:44.754147 ike: ** Dumping ike_sa_info **
2014 Jun 27 17:00:44.754196 ike: { doi_val: 0 ike_tunnel_id: 2  direction: IKE_INITIATOR
status: IKE_SA_STATUS_UNAUTH  local_addr: 172.27.126.42[500]  remote_addr:
172.27.126.172[500]  i_spi: 4ca3c52580808d70  r_spi: 4f58e388623f1727 }
2014 Jun 27 17:00:44.754253 ike: { my_curr_req_msg_id: 1  my_next_req_msg_id: 2
peer_curr_req_msg_id: 0  peer_next_req_msg_id: 1  num_tries: 1 }
2014 Jun 27 17:00:44.754315 ike: ike_save_sainfo: Starting save SA INfo in PSS (Update 0)
2014 Jun 27 17:00:44.754779 ike: ike_save_sainfo: Done save SA INfo in PSS: Status 0x0
2014 Jun 27 17:00:44.759425 ike: ike_save_sainfo: Starting save SA INfo in PSS (Update 1)
2014 Jun 27 17:00:44.759837 ike: ike_save_sainfo: Done save SA INfo in PSS: Status 0x0
2014 Jun 27 17:00:44.759915 ike: start_substate_timer: started the substate_timer to 3600
sec
2014 Jun 27 17:00:44.760107 ike: Processing PF_KEY message
2014 Jun 27 17:00:44.760452 ike: get pfkey ADD message
2014 Jun 27 17:00:44.760735 ike: seq 26 of ADD message not interesting.
2014 Jun 27 17:00:44.760805 ike: ike_pfkey_handler: get pfkey ADD message
2014 Jun 27 17:00:44.761003 ike: Processing PF_KEY message
2014 Jun 27 17:00:44.761291 ike: get pfkey X_COMMIT message
2014 Jun 27 17:00:44.761583 ike: ike_pfkey_handler: get pfkey SADB_X_COMMIT message
```

This example output from the **debug ipsec_tun trace** command shows a successful handshake:

```
cgr1000# debug ipsec_tun trace
cgr1000# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
cgr1000(config)# int t100
cgr1000(config-if)# no shut
2014 Jun 23 23:03:59.689448 ipsec_tun:
ipsec_tun_handle_profile_check(../routing-sw/routing/ipsec_tun/server/ipsec_tun_mts.c:1278
): Invoked!
2014 Jun 23 23:03:59.715701 ipsec_tun: ipsec_tun_handle_acquire_sa_cmd(): SA request from
Tunnel100 (iod = 12)
2014 Jun 23 23:03:59.715879 ipsec_tun: ipsec_tun_create_connection(): Profile
MyIPSecProfile selected.
2014 Jun 23 23:03:59.716153 ipsec_tun: ipsec_tun_create_connection(): Connection 0x812f0cc
for Tunnel100 created.
2014 Jun 23 23:03:59.716216 ipsec_tun: ipsec_tun_store_to_runtime_info_pss():
PSS_TYPE_RUNTIME_CONNECTION
2014 Jun 23 23:03:59.716317 ipsec_tun: ipsec_tun_handle_acquire_sa_cmd(): Connection data
structure (0x812f0cc) created
2014 Jun 23 23:03:59.719979 ipsec_tun: ipsec_register_with_led(): Register with LED,
ret_val: 0x0
2014 Jun 23 23:03:59.720063 ipsec_tun: ipsec_tun_store_to_runtime_info_pss():
PSS_TYPE_RUNTIME_CONNECTION
2014 Jun 23 23:03:59.720417 ipsec_tun: ipsec_set_status_with_led(): Updated LED status --
STARTING : 0x0
2014 Jun 23 23:03:59.720498 ipsec_tun: ipsec_tun_fsm_ac_ike_send_acquire(): Initiate IKE
handshake local=172.27.126.42 peer=172.27.126.172
2014 Jun 23 23:03:59.720548 ipsec_tun: ipsec_tun_initiate_ike_handshake(): Connection
0x812f0cc
2014 Jun 23 23:03:59.720597 ipsec_tun: ipsec_tun_new_ike_txn(): IKE transaction 16
assigned
2014 Jun 23 23:03:59.720768 ipsec_tun: ipsec_tun_start_timer(): Starting handshake timer
for 15 seconds
2014 Jun 23 23:03:59.972535 ipsec_tun: ikecb_getspd_ex(): Received X_GETSPD_EX seqno 0x10
for src 172.27.126.42 dst 172.27.126.172
```

```
2014 Jun 23 23:03:59.972667 ipsec_tun: ikecb_getspd_ex(): Tunnel100 selected.
2014 Jun 23 23:03:59.972708 ipsec_tun: ikecb_getspd_ex(): Profile MyIPSecProfile selected.
2014 Jun 23 23:03:59.972747 ipsec_tun: ikecb_getspd_ex(): Transform set AES128SHA1
selected.
2014 Jun 23 23:03:59.972908 ipsec_tun: ikecb_getspd_ex(): Responded with policy enc
AES-CBC len 16, auth SHA1, len 20 group 0, proto IPSEC_PROTO_ANY
2014 Jun 23 23:03:59.973793 ipsec_tun: ikecb_getspi(): Received GETSPI, seqno 0x10 for src
172.27.126.172 dst 172.27.126.42
2014 Jun 23 23:03:59.973947 ipsec_tun: ikecb_getspi(): Tunnel100 selected.
2014 Jun 23 23:03:59.974033 ipsec_tun: ikecb_getspi(): Responded with SPI 0x62b2e60
2014 Jun 23 23:04:00.070883 ipsec_tun: ikecb_update(): Received UPDATE (ingress), seqno
0x10, for src 172.27.126.42 dst 172.27.126.172
2014 Jun 23 23:04:00.070972 ipsec_tun: ikecb_update():                  tunnel 2, spi
0x62b2e60, enc AES-CBC len 16 bytes, auth SHA1 len 20 bytes
2014 Jun 23 23:04:00.071111 ipsec_tun: ikecb_update(): Tunnel100 selected.
2014 Jun 23 23:04:00.072776 ipsec_tun: ikecb_add(): Received ADD (egress), seqno 0x10, for
src 172.27.126.42 dst 172.27.126.172
2014 Jun 23 23:04:00.072863 ipsec_tun: ikecb_add():               tunnel 2, spi
0x3b141ec4, enc AES-CBC len 16 bytes, auth SHA1 len 20 bytes
2014 Jun 23 23:04:00.072997 ipsec_tun: ikecb_add(): Tunnel100 selected.
2014 Jun 23 23:04:00.074438 ipsec_tun: ikecb_commit(): Received X_COMMIT, seqno 0x10
2014 Jun 23 23:04:00.074601 ipsec_tun: ipsec_tun_handle_pfkey_cmd(): X_COMMIT: Tunnel100
selected for SPI 0x062b2e60/0x3b141ec4.
2014 Jun 23 23:04:00.074843 ipsec_tun: ipsec_tun_stop_timer(): Stopping handshake timer
2014 Jun 23 23:04:00.074961 ipsec_tun: ipsec_tun_store_to_runtime_info_pss():
PSS_TYPE_RUNTIME_CONNECTION
2014 Jun 23 23:04:00.075074 ipsec_tun: ipsec_tun_free_ike_txn(): IKE transaction 16 freed
2014 Jun 23 23:04:00.079320 ipsec_tun: ipsec_tun_fsm_ac_process_sa_committed():
MTS_OPC_IPSEC_TUN_SA_UPDATE notify Tunnel100 with spi 0x00000000/0x00000000
2014 Jun 23 23:04:00.079526 ipsec_tun: ipsec_tun_start_timer(): Starting rekey timer for
3240 seconds
2014 Jun 23 23:04:00.080214 ipsec_tun: ipsec_set_status_with_led(): Updated LED status --
OK : 0x0
cgr1000(config-if)# no debug all
```

The following is output of the **debug ipsec_tun packet** command while sending a ping packet:

```
cgr1000# debug ipsec_tun packet
cgr1000# ping 192.168.100.1 count 1
PING 192.168.100.1 (192.168.100.1): 56 data bytes
64 bytes from 192.168.100.1: icmp_seq=0 ttl=254 time=6.826 ms

--- 192.168.100.1 ping statistics ---
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 6.826/6.825/6.826 ms
2014 Jun 23 22:36:50.737405 ipsec_tun: ipsec_tun_handle_encap_cmd:    ESP encap process
starts
2014 Jun 23 22:36:50.737469 ipsec_tun: ipsec_tun_encap_processing:     SA lookup
2014 Jun 23 22:36:50.737580 ipsec_tun: ipsec_tun_encap_processing:      Prepare outer IP
header
2014 Jun 23 22:36:50.737708 ipsec_tun: ipsec_tun_encap_processing:      Inner packet
length 84
2014 Jun 23 22:36:50.737750 ipsec_tun: ipsec_tun_encap_processing:         iv_size=16,
pad_len=10, mac_size=12, final_pkt_size=152
2014 Jun 23 22:36:50.737786 ipsec_tun: ipsec_tun_encap_processing:      Sending to FPGA
for encryption
2014 Jun 23 22:36:50.737966 ipsec_tun: ipsec_tun_post_engine_encrypt:   Encrytped packet
from FPGA
2014 Jun 23 22:36:50.738036 ipsec_tun: ipsec_tun_post_engine_encrypt:   Final encap of
packet
2014 Jun 23 22:36:50.738097 ipsec_tun: ipsec_tun_post_engine_encrypt:   Post encap'ed
packet to netstack
2014 Jun 23 22:36:50.738204 ipsec_tun: ipsec_tun_post_engine_encrypt: ESP encap process
completed
```

```
2014 Jun 23 22:36:50.740815 ipsec_tun: ipsec_tun_handle_decap_cmd:    ESP decap process
starts
2014 Jun 23 22:36:50.740876 ipsec_tun: ipsec_tun_decap_processing:   SA lookup
2014 Jun 23 22:36:50.740914 ipsec_tun: ipsec_tun_decap_processing:  Replay detection
2014 Jun 23 22:36:50.740948 ipsec_tun: ipsec_tun_decap_processing:      Remove outer IP
header
2014 Jun 23 22:36:50.741086 ipsec_tun: ipsec_tun_decap_processing:      Sending to FPGA
for decryption
2014 Jun 23 22:36:50.741245 ipsec_tun: ipsec_tun_post_engine_decrypt:   Decrypted packet
from FPGA
2014 Jun 23 22:36:50.741301 ipsec_tun: ipsec_tun_post_engine_decrypt: Remove padding
2014 Jun 23 22:36:50.741346 ipsec_tun: ipsec_tun_post_engine_decrypt:      pad_len=10,
inner_pkt_len=84
2014 Jun 23 22:36:50.741398 ipsec_tun: ipsec_tun_post_engine_decrypt:   Post decap'ed
packet to netstack
2014 Jun 23 22:36:50.741584 ipsec_tun: ipsec_tun_post_engine_decrypt: ESP decap process
completed
cgr1000# no debug all
```

Configuration Example

See Example 1: RSA Authentication and Example 2: PSK Authentication.

Configuring IP ACLs

This chapter describes how to configure IP access control lists (ACLs) on the Cisco 1000 Series Connected Grid Routers (hereafter referred to as the Cisco CG-OS router).

Unless otherwise specified, the term IP ACL refers to IPv4 and IPv6 ACLs.

This chapter includes the following sections:

Information About ACLs

An ACL is an ordered set of rules that you can use to filter traffic. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the Cisco CG-OS router determines that an ACL applies to a packet, it tests the packet against the conditions of all rules. The first matching rule determines whether the packet is permitted or denied. If there is no match, the Cisco CG-OS router applies the applicable default rule. The Cisco CG-OS router continues processing packets that are permitted and drops packets that are denied. For more information, see Implicit Rules, page 9-3.

You can use ACLs to protect networks and specific hosts from unnecessary or unwanted traffic. For example, you could use ACLs to disallow HTTP traffic from a high-security network to the Internet. You could also use ACLs to allow HTTP traffic but only to specific sites, using the IP address of the site to identify it in an IP ACL.

This section includes the following topics:

ACL Types and Applications

The Cisco CG-OS router supports the following types of ACLs for security traffic filtering:

- IPv4 ACLs—The Cisco CG-OS router applies IPv4 ACLs only to IPv4 traffic.
- IPv6 ACLs— The Cisco CG-OS router applies IPv6 ACLs only to IPv6 traffic.

IP ACLs supports the following Router ACL application, which filters Layer 3 traffic.

Table 9-1 summarizes the applications for security ACLs.

Table 9-1 Security ACL Applications

Application	Supported Interfaces	Types of ACLs Supported
Router ACL	Physical Layer 3 interfaces	IPv4 ACLs
	Tunnels	IPv6 ACLs

Order of ACL Application

When the Cisco CG-OS router processes a packet, it determines the forwarding path of the packet. The path determines which ACLs that the Cisco CG-OS router applies to the traffic. The Cisco CG-OS router applies the ACLs in the following order:

1. Ingress router ACL
2. Egress router ACL

About Rules

Rules are what you create, modify, and remove when you configure how an ACL filters network traffic. Rules appear in the running configuration. When you apply an ACL to an interface or change a rule within an ACL that is already applied to an interface, the Cisco CG-OS router creates ACL entries from the rules in the running configuration and sends those ACL entries to the applicable interface. Depending on how you configure the ACL, there might be more ACL entries than rules, especially if you use object groups when you configure rules. For more information, see Policy-Based ACLs, page 9-7.

You can create rules in access-list configuration mode by using the **permit** or **deny** command. The Cisco CG-OS router allows traffic that matches the criteria in a permit rule and blocks traffic that matches the criteria in a deny rule. You have many options for configuring the criteria that traffic must meet in order to match the rule.

This section describes some of the options that you can use when you configure a rule.

This section includes the following topics:

Protocols

IPv4 and IPv6 ACLs allow you to identify traffic by protocol. For your convenience, you can specify some protocols by name. For example, in an IPv4 or IPv6 ACL, you can specify ICMP by name.

You can specify any protocol by number. In IPv4 and IPv6 ACLs, you can specify protocols by the integer that represents the Internet protocol number.

Source and Destination

In each rule, you specify the source and the destination of the traffic that matches the rule. You can specify both the source and destination as a specific host, a network or group of hosts, or any host. How you specify the source and destination depends on whether you are configuring IPv4 and IPv6 ACLs. For information about specifying source and destination, see the applicable **permit** and **deny** commands.

Implicit Rules

IP ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the Cisco CG-OS router applies them to traffic when no other rules in an ACL match. When you configure the Cisco CG-OS router to maintain per-rule statistics for an ACL, the Cisco CG-OS router does not maintain statistics for implicit rules.

IPv4 Implicit Rules

All IPv4 ACLs include the following implicit rule:

```
deny ip any any
```

This implicit rule ensures that the Cisco CG-OS router denies unmatched IP traffic.

IPv6 Implicit Rules

All IPv6 ACLs include the following implicit rules:

```
permit icmp any any nd-na
permit icmp any any nd-ns
permit icmp any any router-advertisement
permit icmp any any router-solicitation
deny ipv6 any any
```

Unless you configure an IPv6 ACL with a rule that denies ICMPv6 neighbor discovery messages, the first four rules ensure that the Cisco CG-OS router permits neighbor discovery advertisement and solicitation messages. The fifth rule ensures that the Cisco CG-OS router denies unmatched IPv6 traffic.

Note If you explicitly configure an IPv6 ACL with a **deny ipv6 any any** rule, the implicit permit rules can never permit traffic. If you explicitly configure a **deny ipv6 any any** rule but want to permit ICMPv6 neighbor discovery messages, explicitly configure a rule for all five implicit IPv6 ACL rules.

Additional Filtering Options

You can identify traffic by using additional options. These options differ by ACL type. The following list includes most but not all additional filtering options:

- IPv4 ACLs support the following additional filtering options:
  - Authentication Header Protocol
  - Encapsulating Security Payload
  - KA9Q NOS-compatible IP-over-IP tunneling
  - Open Shortest Path First (OSPF versions 2 and 3)
  - Payload Compression Protocol
  - TCP and UDP ports
  - ICMP types and codes
  - IGMP types
  - Precedence level
  - Differentiated Services Code Point (DSCP) value
  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
  - Established TCP connections
  - Packet length
- IPv6 ACLs support the following additional filtering options:
  - Layer 4 protocol
  - Authentication Header Protocol
  - Encapsulating Security Payload
  - Payload Compression Protocol
  - Stream Control Transmission Protocol (SCTP)
  - TCP and UDP ports
  - ICMP types and codes
  - IGMP types
  - Flow label
  - DSCP value
  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
  - Established TCP connections
  - Packet length

For information about all filtering options available in rules, see the applicable **permit** and **deny** commands in the Command Lookup Tool on Cisco.com.

Sequence Numbers

The Cisco CG-OS router supports sequence numbers for rules. Every rule that you enter receives a sequence number, either assigned by you or assigned automatically by the Cisco CG-OS router. Sequence numbers simplify the following ACL tasks:

- Adding new rules between existing rules—By specifying the sequence number, you specify where in the ACL a new rule should be positioned. For example, if you need to insert a rule between rules numbered 100 and 110, you could assign a sequence number of 105 to the new rule.
- Removing a rule—Without using a sequence number, removing a rule requires that you enter the whole rule, as follows:

  ```
  router(config-acl)# no permit tcp 10.0.0.0/8 any
  ```

  However, if the same rule had a sequence number of 101, removing the rule requires only the following command:

  ```
  router(config-acl)# no 101
  ```

- Moving a rule—With sequence numbers, if you need to move a rule to a different position within an ACL, you can add a second instance of the rule using the sequence number that positions it correctly, and then you can remove the original instance of the rule. This action allows you to move the rule without disrupting traffic.

When you enter a rule without a sequence number, the Cisco CG-OS router adds the rule to the end of the ACL and assigns a sequence number that is 10 greater than the sequence number of the preceding rule to the rule. For example, if the last rule in an ACL has a sequence number of 225 and you add a rule without a sequence number, the Cisco CG-OS router assigns the sequence number 235 to the new rule.

In addition, Cisco CG-OS allows you to reassign sequence numbers to rules in an ACL. Resequencing is useful when an ACL has rules numbered contiguously, such as 100 and 101, and you need to insert one or more rules between those rules.

Logical Operators and Logical Operation Units

IP ACL rules for TCP and UDP traffic can use logical operators to filter traffic based on port numbers. The Cisco CG-OS router stores operator-operand couples in registers called logical operator units (LOUs). The Cisco CG-OS router supports 104 LOUs.

The LOU usage for each type of operator is as follows:

- eq—Is never stored in an LOU
- gt—Uses 1/2 LOU
- lt—Uses 1/2 LOU
- neq—Uses 1/2 LOU
- range—Uses 1 LOU

The following guidelines determine when the Cisco CG-OS router store operator-operand couples in LOUs:

- If the operator or operand differs from other operator-operand couples that are used in other rules, the couple is stored in an LOU.

  For example, the operator-operand couples “gt 10” and “gt 11” would be stored separately in half an LOU each. The couples “gt 10” and “lt 10” would also be stored separately.

Logging

You can enable the Cisco CG-OS router to create an informational log message for packets that match a rule.

Note ACL logging supports ACL processing that occurs on interfaces only. For more information about ACL processing, see Guidelines and Limitations, page 9-8.

The log message contains the following information about the packet:

- Protocol
- Status of whether the packet is a TCP, UDP, or ICMP packet, or if the packet is only a numbered packet
- Source and destination address

Time Ranges

You can use time ranges to control when an ACL rule is in effect. For example, if the Cisco CG-OS router determines that a particular ACL applies to traffic arriving on an interface, and a rule in the ACL uses a time range that is not in effect, the Cisco CG-OS router does not compare the traffic to that rule. The Cisco CG-OS router evaluates time ranges based on its clock.

When you apply an ACL that uses time ranges, the Cisco CG-OS router updates the affected interface whenever a time range referenced in the ACL starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. When the Cisco CG-OS router is especially busy when a time range causes an update, the Cisco CG-OS router might delay the update by up to a few seconds.

IPv4 and IPv6 support time ranges. When the Cisco CG-OS router applies an ACL to traffic, the rules in effect are as follows:

- All rules without a time range specified.
- Rules with a time range that includes the second when the Cisco CG-OS router applies the ACL to traffic.

The Cisco CG-OS router supports named, reusable time ranges, which allows you to configure a time range once and specify it by name when you configure many ACL rules. Time range names have a maximum length of 64 alphanumeric characters.

A time range contains one or more rules. The two types of rules are as follows:

- Absolute—A rule with a specific start date and time, specific end date and time, both, or neither. The following items describe how the presence or absence of a start or end date and time affect whether an absolute time range rule is active:
  - Start and end date and time both specified—The time range rule is active when the current time is later than the start date and time and earlier than the end date and time.
  - Start date and time specified with no end date and time—The time range rule is active when the current time is later than the start date and time.
  - No start date and time with end date and time specified—The time range rule is active when the current time is earlier than the end date and time.
  - No start or end date and time specified—The time range rule is always active.

For example, you could prepare your network to allow access to a new subnet by specifying a time range that allows access beginning at midnight of the day that you plan to place the subnet online. You can use that time range in ACL rules that apply to the subnet. After the start time and date have passed, the Cisco CG-OS router automatically begins applying the rules that use this time range when it applies the ACLs that contain the rules.

- Periodic—A rule that is active one or more times per week. For example, you could use a periodic time range to allow access to a lab subnet only during work hours on weekdays. The Cisco CG-OS router automatically applies ACL rules that use this time range only when the range is active and when it applies the ACLs that contain the rules.

Note The order of rules in a time range does not affect how a Cisco CG-OS router evaluates whether a time range is active. Cisco CG-OS includes sequence numbers in time ranges to make editing the time range easier.

Time ranges also allow you to include remarks, which you can use to insert comments into a time range. Remarks have a maximum length of 100 alphanumeric characters.

The Cisco CG-OS router determines whether a time range is active as follows:

- The time range contains one or more absolute rules—The time range is active if the current time is within one or more absolute rules.
- The time range contains one or more periodic rules—The time range is active if the current time is within one or more periodic rules.
- The time range contains both absolute and periodic rules—The time range is active if the current time is within one or more absolute rules and within one or more periodic rules.

When a time range contains both absolute and periodic rules, the periodic rules can only be active when at least one absolute rule is active.

Policy-Based ACLs

The Cisco CG-OS router supports policy-based ACLs (PBACLs), which allow you to apply access control policies across object groups. An object group is a group of IP addresses or a group of TCP or UDP ports. When you create a rule, you specify the object groups rather than specifying IP addresses.

Using object groups when you configure IPv4 or IPv6 ACLs can help reduce the complexity of updating ACLs when you need to add or remove addresses from the source or destination of rules. For example, if three rules reference the same IP address group object, you can add an IP address to the object instead of changing all three rules.

PBACLs do not reduce the resources required by an ACL when you apply it to an interface. When you apply a PBACL or update a PBACL that is already applied, the Cisco CG-OS router expands each rule that refers to object groups into one ACL entry per object within the group. If a rule specifies the source and destination both with object groups, the number of ACL entries created on the interface when you apply the PBACL is equal to the number of objects in the source group multiplied by the number of objects in the destination group.

The following object group types apply to the Cisco CG-OS router:

- IPv4 address object groups—Can be used with IPv4 ACL rules to specify source or destination addresses. When you use the **permit** or **deny** command to configure a rule, the **addrgroup** keyword allows you to specify an object group for the source or destination.

- IPv6 address object groups—Can be used with IPv6 ACL rules to specify source or destination addresses. When you use the **permit** or **deny** command to configure a rule, the **addrgroup** keyword allows you to specify an object group for the source or destination.

Statistics

The Cisco CG-OS router can maintain global statistics for each rule that you configure in IPv4 and IPv6 ACLs. If an ACL is applied to multiple interfaces, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that ACL is applied.

Note

- The Cisco CG-OS router does not support interface-level ACL statistics.
- ACL statistics are not supported if the DHCP snooping feature is enabled.

For each ACL that you configure, you can specify whether the Cisco CG-OS router maintains statistics for that ACL, which allows you to turn ACL statistics on or off as needed to monitor traffic filtered by an ACL or to help troubleshoot the configuration of an ACL.

The Cisco CG-OS router does not maintain statistics for implicit rules in an ACL. For example, the Cisco CG-OS router does not maintain a count of packets that match the implicit **deny ip any any** rule at the end of all IPv4 ACLs. If you want to maintain statistics for implicit rules, then you must explicitly configure the ACL with rules that are identical to the implicit rules. For more information, see Implicit Rules, page 9-3.

For information about displaying IP ACL statistics, see Monitoring and Clearing IP ACL Statistics, page 9-14.

Session Manager Support for IP ACLs

Session Manager supports the configuration of IP ACLs. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration.

Prerequisites

You must be familiar with IP addressing and protocols to configure IP ACLs.

You must be familiar with the interface types that you want to configure with ACLs.

Guidelines and Limitations

Cisco recommends that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. This is especially useful for ACLs that include more than about 1000 rules.

In most cases, ACL processing for IP packets occurs on the interfaces, which use hardware that accelerates ACL processing. Management interface traffic is always processed on the main board of the Cisco CG-OS router as are IP packets (in any of the following categories) that are exiting a Layer 3 interface:

- Packets that fail the Layer 3 maximum transmission unit check and therefore require fragmenting.
- IPv4 packets that have IP options (additional IP packet header fields following the destination address field).
- IPv6 packets that have extended IPv6 header fields.
- When you apply an ACL that uses time ranges, the Cisco CG-OS router updates the ACL entries on the affected interfaces whenever a time range referenced in an ACL entry starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. When the Cisco CG-OS router is especially busy when a time range causes an update, the Cisco CG-OS router may delay the update by up to a few seconds.

Default Settings

Table 9-2 lists the default settings for IP ACL parameters.

Table 9-2 Default IP ACL Parameters

Parameters	Default
IP ACLs	No IP ACLs exist by default.
ACL rules	Implicit rules apply to all ACLs. (See Implicit Rules, page 9-3.)
Object groups	No object groups exist by default.
Time ranges	No time ranges exist by default.

Configuring IP ACLs

This section includes the following topics:

Creating an IP ACL

You can create an IPv4 ACL or IPv6 ACL on the Cisco CG-OS router and add rules to it.

BEFORE YOU BEGIN

Cisco recommends that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	{**ip** | **ipv6**} **access-list** *name*	Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.
Step 3	[*sequence-number*] {**permit** | **deny**} *protocol source destination*	Creates a rule in the IP ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic.
Step 4	**statistics per-entry**	(Optional) Specifies that the Cisco CG-OS router maintains global statistics for packets that match the rules in the ACL.
Step 5	**show ip access-lists** *name*	(Optional) Displays the IP ACL configuration.
Step 6	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to create an IP ACL.

```
router# configure terminal
router(config)# ip access-list acl-01
router(config-acl)# permit ip 192.168.2.0/24 any
router(config-acl)# statistics per-entry
router(config-acl)# copy running-config startup-config
```

Changing an IP ACL

You can add and remove rules in an existing IPv4 or IPv6 ACL. You cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers. For more information, see Changing Sequence Numbers in an IP ACL, page 9-11.

BEFORE YOU BEGIN

Cisco recommends that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	{**ip** | **ipv6**} **access-list** *name*	Enters IP ACL configuration mode for the ACL that you specify by name.
Step 3	[*sequence-number*] {**permit** | **deny**} *protocol source destination*	(Optional) Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic.
Step 4	**no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*}	(Optional) Removes the rule that you specified from the IP ACL. The **permit** and **deny** commands support many ways of identifying traffic.
Step 5	[**no**] **statistics per-entry**	(Optional) Specifies that the Cisco CG-OS router maintains global statistics for packets that match the rules in the ACL. The **no** option stops the Cisco CG-OS router from maintaining global statistics for the ACL.
Step 6	**show ip access-lists** *name*	(Optional) Displays the IP ACL configuration.
Step 7	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to change an IP ACL.

```
router# configure terminal
router(config)# ip access-list acl-01
router(config-acl)# 100 permit ip 192.168.2.0/24 any
router(config-acl)# no 80
router(config-acl)# statistics per-entry
router(config-acl)# copy running-config startup-config
```

Changing Sequence Numbers in an IP ACL

You can change all the sequence numbers assigned to the rules in an IP ACL.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**resequence** {**ip** | **ipv6**} **access-list** *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. The *starting-sequence-number* argument and the *increment* argument can be a whole number between 1 and 4294967295.
Step 3	**show ip access-lists** *name*	(Optional) Displays the IP ACL configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to change sequence numbers in an IP ACL.

```
router# configure terminal
router(config)# resequence access-list ip acl-01 100 10
router(config)# copy running-config startup-config
```

Removing an IP ACL

You can remove an IP ACL from the Cisco CG-OS router.

BEFORE YOU BEGIN

Ensure that you know whether the ACL is applied to an interface. The Cisco CG-OS router allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, the Cisco CG-OS router considers the removed ACL to be empty. Use the **show ip access-lists** command or the **show ipv6 access-lists** command with the **summary** keyword to find the interfaces that an IP ACL is configured on.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**no** {**ip** | **ipv6**} **access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.

	Command	Purpose
Step 3	**show** {**ip** | **ipv6**} **access-list** *name* **summary**	(Optional) Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to remove an ACL.

```
router# configure terminal
router(config)# no ip access-list acl-01
router(config)# copy running-config startup-config
```

Applying an IP ACL as a Router ACL

You can apply an IPv4 or IPv6 ACL to any of the following types of interfaces:

- Physical Layer 3 interfaces
- Tunnels

ACLs applied to these interface types are considered router ACLs.

BEFORE YOU BEGIN

Ensure that the ACL you want to apply exists and that it is configured to filter traffic in the manner that you need for this application. For more information, see Creating an IP ACL, page 9-9 or Changing an IP ACL, page 9-10.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface {ethernet | cellular | wimax}** *slot*//*number*]	Enters interface configuration mode for a Layer 3 physical interface.
	interface tunnel *tunnel-number*	Enters interface configuration mode for a tunnel.
Step 3	{**ip access-group** | **ipv6 traffic-filter**} *access-list* {**in** | **out**}	Applies an IPv4 or IPv6 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 4	**show running-config aclmgr**r	(Optional) Displays the ACL configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to apply an IPv4 ACL to a cellular interface.

```
router# configure terminal
router(config)# interface cellular 3/1
router(config-if)# ip access-group acl-20 out
router(config-if)# copy running-config startup-config
```

Verifying Configurations

To display IP ACL configuration information, use one of the following commands:

Command	Purpose
show running-config aclmgr	Displays the ACL configuration, including IP ACL configuration and interfaces that IP ACLs are applied to.
show ip access-lists	Displays the IPv4 ACL configuration.
show ipv6 access-lists	Displays the IPv6 ACL configuration.
show running-config interface	Displays the configuration of an interface to which you have applied an ACL.

For detailed information about the fields in the output from these commands, refer to the Command Lookup Tool on Cisco.com.

Monitoring and Clearing IP ACL Statistics

To display or clear IP ACL statistics, use one of the following commands:

Command	Purpose
show ip access-lists	Displays IPv4 ACL configuration. If the IPv4 ACL includes the **statistics per-entry** command, then the **show ip access-lists** command output includes the number of packets that have matched each rule.
show ipv6 access-lists	Displays IPv6 ACL configuration. If the IPv6 ACL includes the **statistics per-entry** command, then the **show ipv6 access-lists** command output includes the number of packets that have matched each rule.
clear ip access-list counters	Clears statistics for all IPv4 ACLs.
clear ipv6 access-list counters	Clears statistics for all IPv6 ACLs.

For detailed information about the fields in the output from these commands, refer to the Command Lookup Tool on Cisco.com.

Configuration Example

The following example shows how to create an IPv6 ACL named acl-120 and apply it as a router ACL to Ethernet interface 2/3, which is a Layer 3 interface:

```
ipv6 access-list acl-120
  permit tcp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
  permit udp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
  permit tcp 2001:0db8:69f2::/48 2001:0db8:be03:2112::/64
  permit udp 2001:0db8:69f2::/48 2001:0db8:be03:2112::/64
interface ethernet 2/3
  ipv6 traffic-filter acl-120 in
```

Configuring Object Groups

You can use object groups to specify source and destination addresses in IPv4 ACL and IPv6 ACL rules.

This section includes the following topics:

Session Manager Support for Object Groups

Session Manager supports the configuration of object groups. This feature allows you to create a configuration session and verify your object group configuration changes prior to committing them to the running configuration.

Creating and Changing an IPv4 Address Object Group

You can create and change an IPv4 address group object.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**object-group ip address** *name*	Creates the IPv4 address object group and enters IPv4 address object-group configuration mode. *name*–Maximum of 64 characters allowed.

	Command	Purpose
Step 3	[*sequence-number*] {**host** *IPv4-address* | *IPv4-address network-wildcard* | *IPv4-address/prefix-len*}	Creates an entry in the object group. For each entry that you want to create, use the **host** command and specify a single host or omit the **host** command to specify a network of hosts.
	no [*sequence-number* | **host** *IPv4-address* | *IPv4-address network-wildcard* | *IPv4-address/prefix-len*}	Removes an entry in the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.
Step 4	**show object-group** *name*	(Optional) Displays the object group configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to create and change an object group:

```
router# configure terminal
router (config)# object-group ip address ipv4-addr-group-13
router(config-ipaddr-ogroup)# host 10.99.32.6
router(config-ipaddr-ogroup)# copy running-config startup-config
```

Creating and Changing an IPv6 Address Object Group

You can create and change an IPv6 address group object.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**object-group ipv6 address** *name*	Creates the IPv6 address object group and enters IPv6 address object-group configuration mode. *name*–Maximum of 64 characters allowed.
Step 3	[*sequence-number*] {**host** *IPv6-address* | *IPv6-address/prefix-len*}	Creates an entry in the object group. For each entry that you want to create, use the **host** command and specify a single host or omit the **host** command specify a network of hosts.
	no [*sequence-number* | **host** *IPv6-address* | *IPv6-address/prefix-len*}	Removes an entry from the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.
Step 4	**show object-group** *name*	(Optional) Displays the object group configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to create and change an IPv6 address group object.

```
router# configure terminal
router (config)# object-group ipv6 address ipv6-addr-group-A7
router(config-ipv6addr-ogroup)# host 2001:db8:0:3ab0::1
router(config-ipv6addr-ogroup)# copy running-config startup-config
```

Removing an Object Group

You can remove an IPv4 address object group and an IPv6 address object group.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**no object-group** {**ip address** | **ipv6 address**} *name*	Removes the object group that you specified.
Step 3	**show object-group**	(Optional) Displays all object groups. The removed object group should not appear.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to remove an IPv6 object group.

```
router# configure terminal
router (config)# no object-group ipv6 address ipv6-addr-group-A7
router(config-ipv6addr-ogroup)# copy running-config startup-config
```

Verifying Object-Group Configurations

To display object-group configuration information, use one of the following commands:

Command	Purpose
show object-group	Displays the object-group configuration
show running-config aclmgr	Displays ACL configuration, including object groups.

For detailed information about the fields in the output from these commands, see the Command Lookup Tool on Cisco.com.

Configuring Time Ranges

This section includes the following topics:

Session Manager Support for Time Ranges

Session Manager supports the configuration of time ranges. This feature allows you to create a configuration session and verify your time-range configuration changes prior to committing them to the running configuration.

Creating a Time Range

You can create a time range on the Cisco CG-OS router and add rules to it.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**time-range** *name*	Creates the time range and enters time-range configuration mode.

	Command	Purpose
Step 3	[*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time*	Creates a periodic rule that is in effect for one or more contiguous days between and including the specified start and end days and times.
	[*sequence-number*] **periodic** *list-of-weekdays time* **to** *time*	Creates a periodic rule that is in effect on the days specified by the *list-of-weekdays* argument between and including the specified start and end times. The following keywords are also valid values for the *list-of-weekdays* argument: daily—All days of the week. weekdays—Monday through Friday. weekend—Saturday through Sunday.
	[*sequence-number*] **absolute start** *time date* [**end** *time date*]	Creates an absolute rule that is in effect beginning at the time and date specified after the **start** keyword. If you omit the **end** keyword, the rule is always in effect after the start time and date have passed.
	[*sequence-number*] **absolute** [**start** *time date*] **end** *time date*	Creates an absolute rule that is in effect until the time and date specified after the **end** keyword. If you omit the **start** keyword, the rule is always in effect until the end time and date have passed.
Step 4	**show time-range** *name*	(Optional) Displays the time-range configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to create a time range on the Cisco CG-OS router and add rules to it.

```
router# configure terminal
router (config)# time-range workday-daytime
router(config-time-range)# periodic monday 00:00:00 to friday 23:59:59
router(config-time-range)# copy running-config startup-config
```

Changing a Time Range

You can add and remove rules in an existing time range.

Note You cannot change existing rules. Instead, to change a rule, you can remove it using the no version of the command and recreate it with the desired changes.

When you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers. For more information, see the Changing Sequence Numbers in a Time Range, page 9-21.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**time-range** *name*	Enters time-range configuration mode for the specified time range.
Step 3	[*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time*	Creates a periodic rule that is in effect for one or more contiguous days between and including the specified start and end days and times.
	[*sequence-number*] **periodic** *list-of-weekdays time* **to** *time*	Creates a periodic rule that is in effect on the days specified by the *list-of-weekdays* argument between and including the specified start and end times. The following keywords are also valid values for the *list-of-weekdays* argument: • **daily**—All days of the week. • **weekdays**—Monday through Friday. • **weekend**—Saturday through Sunday.
	[*sequence-number*] **absolute start** *time date* [**end** *time date*]	Creates an absolute rule that is in effect beginning at the time and date specified after the **start** keyword. If you omit the **end** keyword, the rule is always in effect after the start time and date have passed.
	[*sequence-number*] **absolute** [**start** *time date*] **end** *time date*	Creates an absolute rule that is in effect until the time and date specified after the **end** keyword. If you omit the **start** keyword, the rule is always in effect until the end time and date have passed.
	no {*sequence-number* | **periodic** *arguments* . . . | **absolute** *arguments*. . .}	Removes the specified (existing) rule from the time range.
Step 4	**show time-range** *name*	(Optional) Displays the time-range configuration.
Step 5	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to change a time range.

```
router# configure terminal
router (config)# time-range workday-daytime
router(config-time-range)# no periodic monday 00:00:00 to friday 23:59:59
router(config-time-range)# periodic weekdays 05:00:00 to 22:00:00
```

```
router(config-time-range)# copy running-config startup-config
```

Removing a Time Range

You can remove a time range from the Cisco CG-OS router.

BEFORE YOU BEGIN

Ensure that you know whether the time range is used in any ACL rules. The Cisco CG-OS router allows you to remove time ranges that are used in ACL rules. Removing a time range that is in use in an ACL rule does not affect the configuration of interfaces where you have applied the ACL. Instead, the Cisco CG-OS router considers the ACL rule using the removed time range to be empty.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**no time-range** *name*	Removes the time range that you specified by name.
Step 3	**show time-range**	(Optional) Displays configuration for all time ranges. The removed time range should not appear.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to remove a time range.

```
router# configure terminal
router (config)# time-range workday-daytime
router(config-time-range)# no periodic monday 00:00:00 to friday 23:59:59
router(config-time-range)# copy running-config startup-config
```

Changing Sequence Numbers in a Time Range

You can change all the sequence numbers assigned to rules in a time range.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**resequence time-range** *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the time range, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify.
Step 3	**show time-range** *name*	(Optional) Displays the time-range configuration.
Step 4	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.

EXAMPLE

This example shows how to change a sequence number that is assigned to a rule in a time range.

```
router# configure terminal
router(config)# resequence time-range daily-workhours 100 10
router(config)# copy running-config startup-config
```

Verifying Time-Range Configurations

To display time-range configuration information, use one of the following commands:

Command	Purpose
show time-range	Displays the time-range configuration
show running-config aclmgr	Displays ACL configuration, including all time ranges.

For detailed information about the fields in the output from these commands, see the Command Lookup Tool on Cisco.com.

Configuring Control-Plane Policing

This chapter describes how to configure Control-Plane Policing (CoPP) on the Cisco 1000 Series Connected Grid Routers (hereafter referred to as the Cisco CG-OS router).

This chapter includes the following sections:

Information About CoPP

To prevent the Cisco CG-OS router from Denial of Service (DoS) attacks, the system employs control-plane policing (CoPP or CPP). CoPP increases security on the router by protecting the system from unnecessary or DoS traffic and gives priority to important control-plane and management traffic.

To protect the control plane against DoS attacks and to restrict specific flows, there should be a flexible way to police different classes of traffic destined to the CPU.

For information on deploying CoPP:

http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6642/prod_white_paper0900aecd804fa16a.html

For information on CoPP best practices:

http://www.cisco.com/web/about/security/intelligence/coppwp_gs.html

CoPP can protect the control and management planes and ensure routing stability, accessibility, and packet delivery. CoPP uses a dedicated control-plane configuration through Cisco Modular QoS CLI (MQC) to provide filtering and rate-limiting capabilities for control-plane packets. (See Using Modular CLI in the *Cisco 1000 Series Connected Grid Routers QoS Software Configuration Guide.*) CoPP policy can be used to protect the CPU from DoS attacks by restricting SYNC packets, FIN packets, and IP fragments.

CoPP manager (Coppmgr) is the part of CG-OS that processes control-plane configuration commands. Because CoPP uses MQC, it must interact with the Access Control List (ACL) manager for the ACLs, and the QoS manager for the class maps.

When a CoPP policy refers to a QoS class map, the QoS manager sends the changes in the class map to the clients that use the policy. Similarly, when an ACL, referenced by CoPP policy, changes, the CG-OS software sends that change to the client by employing the ACL manager.

Key Concepts

CoPP involves the following actions:

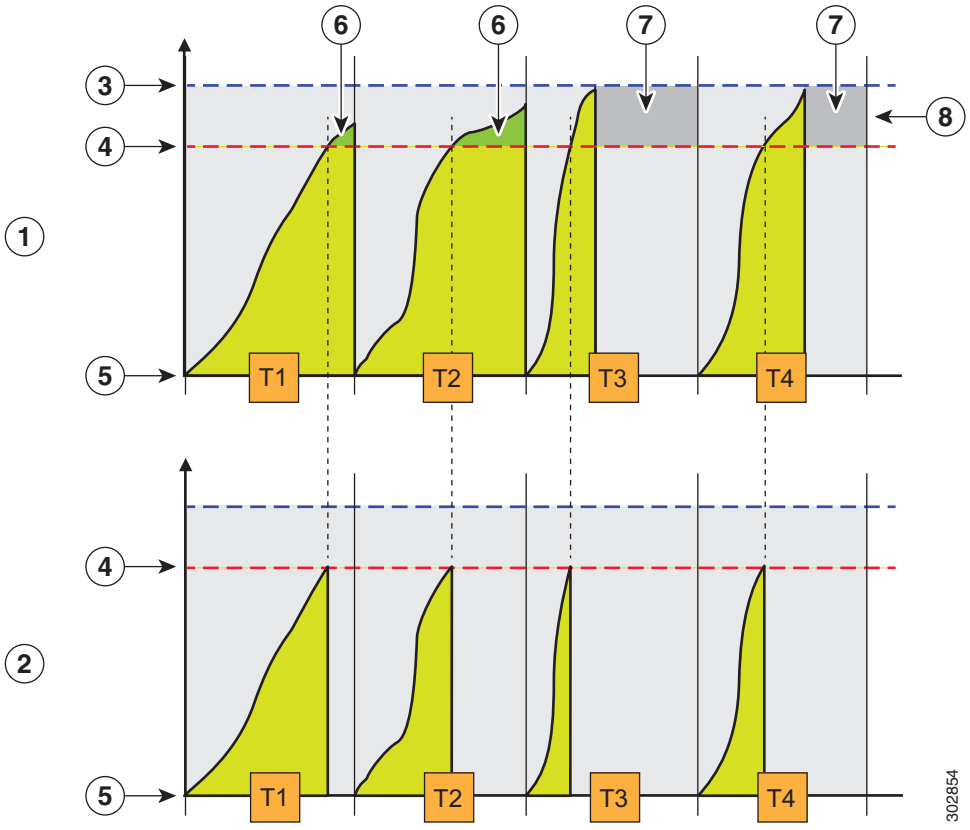
Rate—Defines the amount of traffic sent by the Cisco CG-OS router in a given interval.

Policing—The process of limiting traffic to a prescribed rate. Allows the definition of a rate and a burst. The router does not forward any further traffic for a given interval after the specified amount has passed through the interface.

Burst—Defines the amount of traffic that can be held in the queue for future transmission. Traffic in excess of the burst can be either dropped or have its priority setting reduced.

Figure 10-1 demonstrates that committed information rate (CIR) [4] and burst rate [3] are integral to policing. While the traffic allowed within the time window is at the rate of committed information rate, traffic is only dropped after the burst rate is reached.

Figure 10-1 QoS Policing



1	QoS with burst	**2**	QoS without burst (Cisco CG-OS router)
3	Burst rate (maximum bytes)	**4**	CIR (bytes)
5	Zero (bytes)	**6**	Actual burst
7	No traffic received	**8**	Burst rate
T	Sampling window		

The CG-OS router does not have a burst rate [8]. The sampling window duration [T*x*] is in seconds. The CIR [4] is in packets per second. The router drops packets that exceed the CIR setting [7]. The router does not support additional actions such as marking traffic.

In Figure 10-1, at 5-second intervals, the router allows for the committed number of packets [4] for the specified flow and drops additional packets. The committed number of packets [4] is calculated by multiplying by 5 the committed information rate provided in the input.

Prerequisites

Refer to the Before You Begin paragraph at the beginning of each section for prerequisites.

Guidelines and Limitations

The Cisco CG-OS router supports a limited set of the policing parameters for CoPP.

The router supports the following commands shown in bold:

```
router(config)# policy-map type control-plane copp_policy
router(config-pmap)# class copp_class
router(config-pmap-c)# police ?
<CR>
cir         Specify committed information rate
<CR>
router(config-pmap-c)# police cir ?
  <1-100000>  Committed Information Rate in pps
router(config-pmap-c)# police cir 50 ?
<CR>
pps            Packets per second
```

The CG-OS router does not support the following CoPP policing parameters when defining a policy map and class-map at the (config-pmap-c)# prompt:

- <1-512000000>–Defines the committed burst size in bytes
- bc–Specifies committed burst
- bps–Specifies bits per second
- conform–Specifies a conform action
- gbps–Specifies gigabits per second
- kbps–Specifies kilobits per second
- mbps–Specifies kilobits per second

- pir–Specifies a peak information rate

Default Settings

Table 10-1 Default Settings

Parameters	Default
class-map type control-plane	match-any

Configuring CoPP

This section includes the following topics:

Configuring an ACL

A CoPP policy protects the CPU from DoS attacks by restricting synchronization (sync) packets, finish (FIN) packets and IP fragments.

This section provides details on configuring an ACL for CoPP.

See Configuring IP ACLs in this guide for more information on configuring ACLs on the Cisco CG-OS router.

BEFORE YOU BEGIN

No prerequisites.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip access-list default_copp_acl**	Creates or accesses the IP ACL, named default_copp_acl, and enters IP ACL configuration mode.
Step 3	**permit tcp any any syn**	Defines the traffic match conditions that the router permits for synchronization.

	Command	Purpose
Step 4	**permit tcp any any fin**	Defines the traffic match conditions that the router permits for finish packets.
Step 5	**permit ip any any fragments**	Defines the traffic match conditions that the router permits for IP packets.

EXAMPLE

This example shows how to create the ACL, default_copp_acl, and define ACL permits.

```
router# configure terminal
router(config-acl)# ip access-list default_copp_acl
router(config-acl)# permit tcp any any syn
router(config-acl)# permit tcp any any fin
router(config-acl)# permit ip any any fragments
```

Configuring a Class Map

Create a class map for the control-plane and classify traffic based on the ACL.

See Configuring Priority Queuing in the *Cisco 1000 Series Connected Grid Routers QoS Software Configuration Guide* for more information on configuring class maps on the Cisco CG-OS router.

BEFORE YOU BEGIN

Configure an ACL. *See* Configuring an ACL.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**class-map type control-plane match-any default_copp_class**	Creates or accesses the class-map for the control-plane, and then enters class-map qos mode.
Step 3	**match access-group name default_copp_acl**	Creates or accesses the traffic class by matching packets based on the *acl-name*, default_copp_acl. The system ignores permit and deny ACL keywords in the matching.

EXAMPLE

This example shows how to create the class-map for the control-plane.

```
router# configure terminal
router(config)# class-map type control-plane match-any default_copp_class
router(config-cmap)# match access-group name default_copp_acl
```

Configuring a Policy Map

Configure a policy map for the control-plane and define a policing action within a subordinate class map.

See Configuring Priority Queuing in the Cisco 1000 Series Connected Grid Routers QoS Software Configuration Guide for more information on configuring policy maps on the Cisco CG-OS router.

BEFORE YOU BEGIN

See Guidelines and Limitations for a summary of supported policing commands.

Create a class map. (*See* Configuring a Class Map.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**policy-map type control-plane default_copp_policy**	Creates or accesses the policy map and then enters policy-map mode for the policy-map name that you specify. Policy-map names can contain alphabetic, hyphen, or underscore characters, are case sensitive, and can be up to 40 characters.
Step 3	**class copp_class**	Configures the class map and then enters the class-map qos mode.
Step 4	**police cir** *value* **pps**	Specifies the CIR policing rate in packets per second (pps). *value*–1 to 100000 **Note** The router drops packets that exceed the CIR setting.

EXAMPLE

This example shows how to define a policing action for the control-plane policy map.

```
router# configure terminal
router(config)# policy-map type control-plane default_copp_policy
router(config-pmap)# class copp_class
router(config-pmap-c)# police cir 50 pps
```

Configuring the Control-Plane

Apply the policy map created in Configuring a Policy Map to the control-plane.

BEFORE YOU BEGIN

Create a policy map. (See Configuring a Policy Map.)

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**control-plane**	Enters the control plane configuration mode.
Step 3	**service-policy input default_copp_policy**	Applies the defined policy to incoming packets on the control plane.

EXAMPLE

This example shows how to apply a policy map to the control-plane.

```
router# configure terminal
router(config)# control-plane
router(config)# service-policy input default_copp_policy
```

Verifying Configuration

To display information about the CoPP configuration, enter any or all of the following commands:

Command	Purpose
show ip traffic	Displays details on processed IP traffic. **Note** In the display, the COPP Drop field refers to the number of dropped packets due to control-plane policing.
show policy-map interface control-plane	Displays the configuration details for the policing policy defined on the control plane.

show commands

show ip traffic

To see whether CoPP has initiated policing to drop packets, enter the **show ip traffic** command.

```
router# show ip traffic

IP Software Processed Traffic Statistics
----------------------------------------
Transmission and reception:
  Packets received: 680962, sent: 26263, consumed: 457,
  Forwarded, unicast: 2027, multicast: 0, Label: 0
Opts:
  end: 0, nop: 0, basic security: 0, loose source route: 0
  timestamp: 0, record route: 0
  strict source route: 0, alert: 0,
  other: 0
Errors:
  Bad checksum: 0, packet too small: 0, bad version: 0,
  Bad header length: 0, bad packet length: 0, bad destination: 0,
  Bad ttl: 0, could not forward: 3826, no buffer dropped: 0,
  Bad encapsulation: 46045, no route: 0, non-existent protocol: 0
  Bad options: 0
  Stateful Restart Recovery: 0,  MBUF pull up fail: 0
  Bad context id: 0, rpf drops: 0
  Ingress Drop (ifmgr init): 0,
  Ingress Drop (invalid filter): 0
  Ingress Drop (Invalid L2 msg): 0
  ACL Filter Drops :
       Ingress - 0
       Egree -   0
       Directed Broadcast - 0
  COPP Drop : 90,               <-- CoPP drop packets
```

```
Fragmentation/reassembly:
  Fragments received: 10, fragments sent: 0, fragments created: 0,
  Fragments dropped: 9, packets with DF: 0, packets reassembled: 0,
```

show policy-map interface control-plane

To review configuration details for the policing policy defined on the control plane, enter the **show policy-map interface control-plane** command:

```
router# show policy-map interface control-plane
Control Plane
  service-policy  input: copp_policy
    class-map copp_class match-any
      match access-group name copp_acl
      police cir 2000 pps <-- Committed Information Rate (CIR)
```

Configuration Example

This example shows how to configure an IP ACL named default_copp_a, create a control-plane policy map with a class map that specifies policing as an action, and apply that policy map to the control-plane.

```
router# configure terminal
router(config-acl)# ip access-list default_copp_acl
router(config-acl)# permit tcp any any syn
router(config-acl)# permit tcp any any fin
router(config-acl)# permit ip any any fragments
router(config-acl)# exit
router(config)# class-map control-plane match-any default_copp_class
router(config-cmap)# match access-group name default_copp_acl
router(config-cmap)# exit
router(config)# policy-map type control-plane default_copp_policy
router(config-pmap)# class copp_class
router(config-pmap-c)# police cir 50 pps
router(config-pmap-c)# exit
router(config)# control-plane
router(config)# service-policy input default_copp_policy
router(config)# copy running-config startup-config
```

Feature History

Feature Name	Release	Feature Information
Control-Plane Policing	Cisco CG-OS Release CG2(1)	Initial support of the feature on the CGR 1000 Series Routers.

Zero Touch Deployment

This chapter provides an high-level description of Zero Touch Deployment.

Zero Touch Deployment is an ease-of-use feature that automatically registers (enrolls) and distributes X.509 certificates and provisioning information over secure connections within a connected grid network. (See Figure 11-1.)

Figure 11-1 Zero Touch Deployment within a Cisco Connected Grid Network

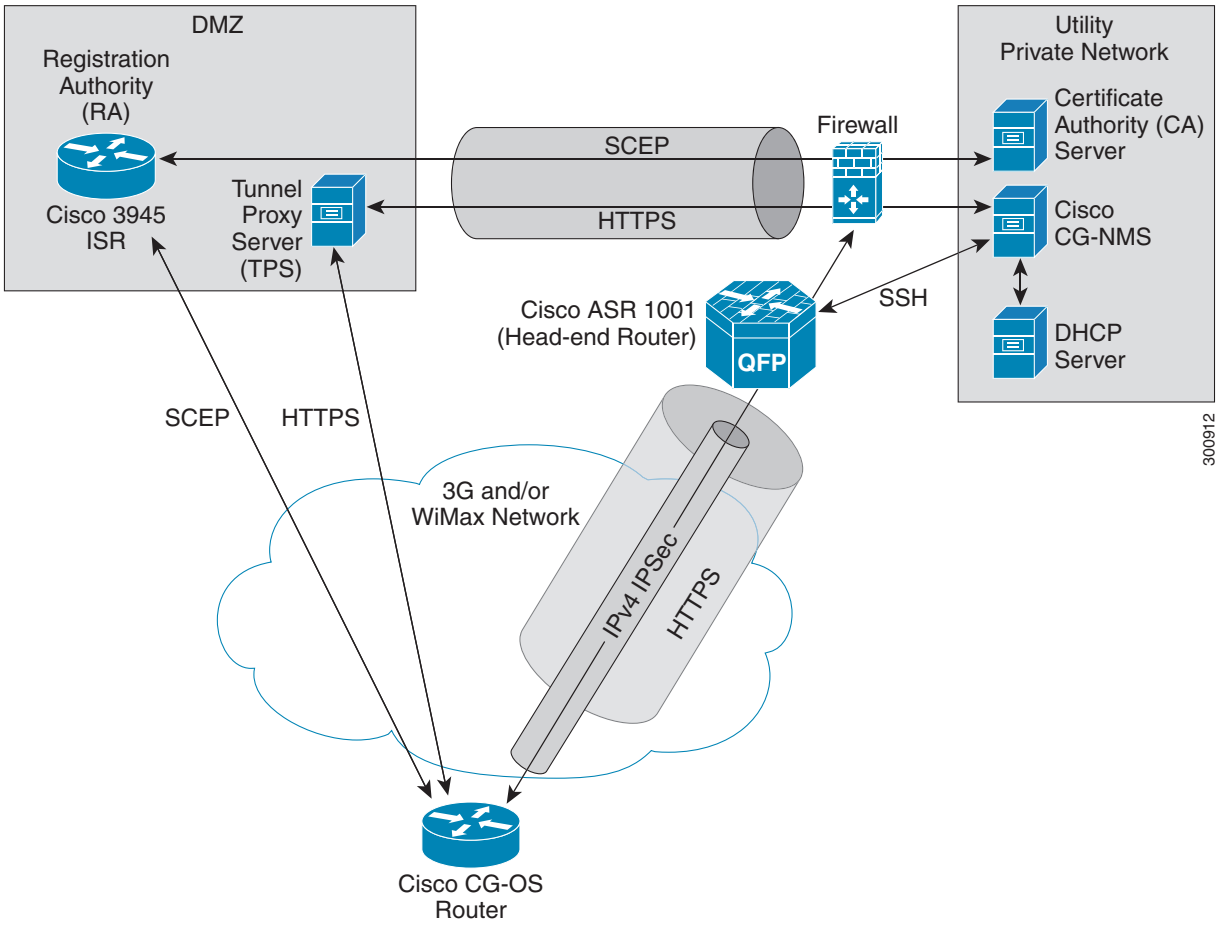


Figure 11-1 provides a high-level view of the systems and communication that might exist in a Utility connected grid network in which Zero Touch Deployment is in use.

In this example, the firewall provides separation between those items in the Utility public network (identified as DMZ) and its private network.

The Utility private network shows systems that might reside behind the firewall such as the Cisco CG-NMS, the DHCP server, and the Certificate Authority (CA). The Tunnel Proxy Server (TPS) and Registration Authority (RA) might be located in the public network (DMZ).

After installing and powering on the Cisco CG-OS router on a pole top, it becomes active in the network and registers its certificate with the RA by employing the Simple Certificate Enrollment Protocol (SCEP). The RA (Cisco 3945 ISR), functioning as a CA proxy, obtains certificates for the Cisco CG-OS router from the CA. Additionally, the Cisco CG-OS router establishes an HTTPS connection with the provisioning server (Cisco CG-NMS) through the TPS.

Cisco CG-NMS manages collection of all the information necessary to configure a tunnel between the Cisco CG-OS router and the head-end router (Cisco ASR 1001). The tunnel can be built to support IPv4 traffic over IPSec, which is encapsulated within an HTTPS tunnel.

When the tunnel is active, the Cisco CG-OS router (after configuration) connects to the Utility network like a Virtual Private Network (VPN).

Tip For details on implementing Zero Touch Deployment within your network, please contact your Cisco partner or Cisco system engineer.