Cisco Nexus 3548 Switch NX-OS Security Configuration Guide, Release 10.5(x)

First Published: 2024-07-26

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

CONTENTS

Preface

The preface contains the following sections:

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 3000 Series Switches

The entire Cisco Nexus 3000 Series switch documentation set is available at the following URL:

https://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/tsd-products-support-series-home.html

Documentation Feedback

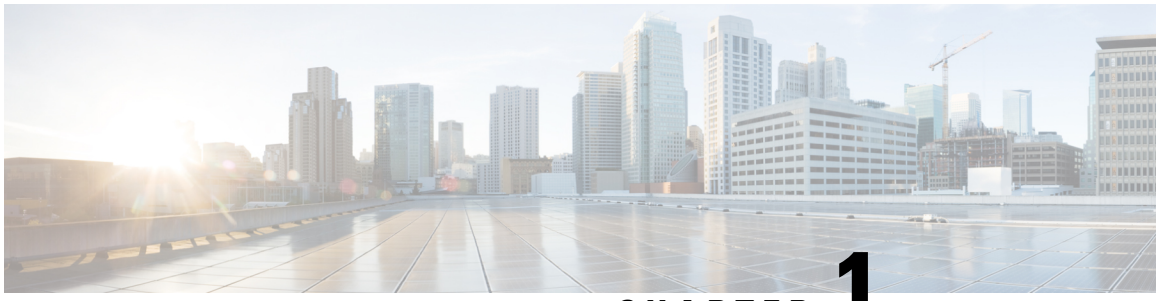
To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus3k-docfeedback@cisco.com. We appreciate your feedback.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

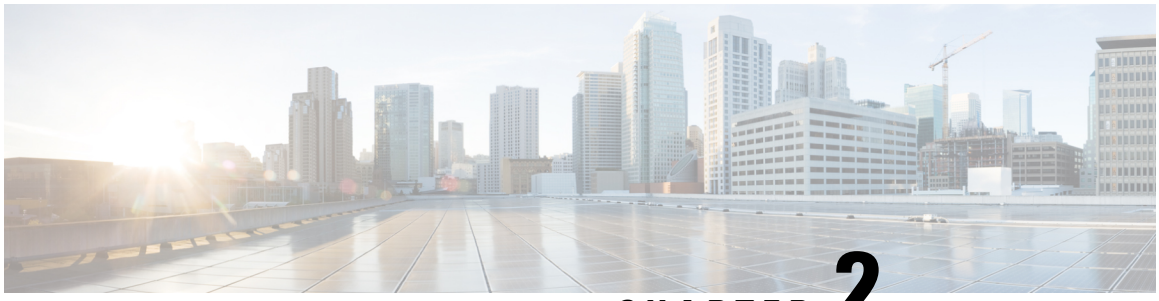
New and Changed Information

New and Changed Information

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
NA	No feature updates for this release.	10.5(1)F	NA

New and Changed Information



CHAPTER 2

Overview

This chapter contains the following sections:

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide* and the *Cisco NX-OS Licensing Options Guide*.

Supported Platforms

Starting with Cisco NX-OS release 7.0(3)I7(1), use the Nexus Switch Platform Support Matrix to know from which Cisco NX-OS releases various Cisco Nexus 9000 and 3000 switches support a selected feature.

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 way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods and then applying that list to various interfaces.

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

Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared with the information contained in a database for a given user, and the result is returned to AAA to determine the user's actual capabilities and restrictions.

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

RADIUS and TACACS+ Security Protocols

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

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

RADIUS
A distributed client/server system implemented through AAA that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

TACACS+
A security application implemented through AAA that provides 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.

SSH and Telnet

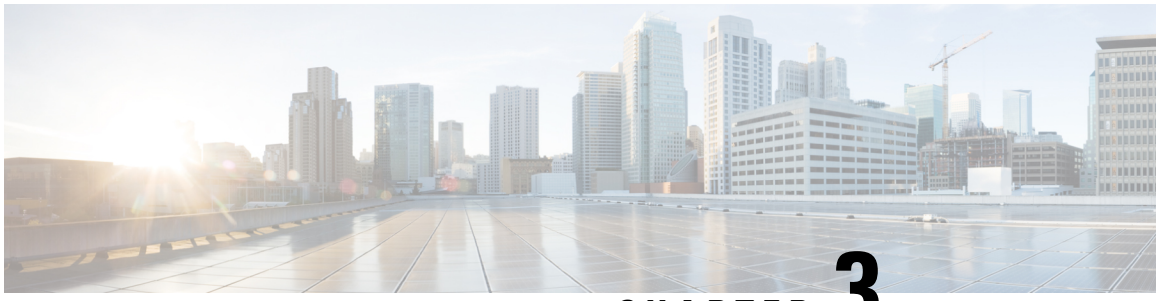
You can use the Secure Shell (SSH) server to enable an SSH client to make a secure, encrypted connection to a Cisco NX-OS device. SSH uses strong encryption for authentication. The SSH server in the Cisco NX-OS software can interoperate with publicly and commercially available SSH clients.

The SSH client in the Cisco NX-OS software works with publicly and commercially available SSH 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.

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 the Cisco NX-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, the Cisco NX-OS software applies the applicable default rule. The Cisco NX-OS software continues processing packets that are permitted and drops packets that are denied.



CHAPTER 3

Configuring Authentication, Authorization, and Accounting

This chapter contains the following sections:

Information About AAA

AAA Security Services

The authentication, authorization, and accounting (AAA) features allows you to verify the identity of, grant access to, and track the actions of users who manage Cisco Nexus devices. The Cisco Nexus device 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 that you provide, the switches perform local authentication or authorization using the local database or remote authentication or authorization using one or more AAA servers. A preshared secret key provides security for communication between the switch 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, encryption depending on the security protocol that you select.
- Authorization—Provides access control.

  Authorization to access a Cisco Nexus device 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, and sending the information to the AAA server for billing, auditing, and reporting.

Note The Cisco NX-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+
- Multiple backup devices

Remote AAA Services

Remote AAA services provided through RADIUS and TACACS+ protocols have the following advantages over local AAA services:

- User password lists for each switch in the fabric are easier to manage.
- AAA servers are already deployed widely across enterprises and can be easily used for AAA services.
- The accounting log for all switches in the fabric can be centrally managed.
- User attributes for each switch in the fabric are easier to manage than using the local databases on the switches.

AAA Server Groups

You can specify remote AAA servers for authentication, authorization, and accounting using server groups. A server group is a set of remote AAA servers that implement the same AAA protocol. A server group provides for failover servers if a remote AAA server fails to respond. If the first remote server in the group fails to respond, the next remote server in the group is tried until one of the servers sends a response. If all the AAA servers in the server group fail to respond, that server group option is considered a failure. If required, you can specify multiple server groups. If a switch encounters errors from the servers in the first group, it tries the servers in the next server group.

AAA Service Configuration Options

On Cisco Nexus devices, you can have separate AAA configurations for the following services:

- User Telnet or Secure Shell (SSH) login authentication
- Console login authentication

- User management session accounting

The following table lists the CLI commands for each AAA service configuration option.

Table 2: 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 method is for all RADIUS servers, instead of a specific server group, the Cisco Nexus devices choose the RADIUS server from the global pool of configured RADIUS servers in the order of configuration. Servers from this global pool are the servers that can be selectively configured in a RADIUS server group on the Cisco Nexus devices.

The following table describes the AAA authentication methods that you can configure for the AAA services.

Table 3: 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, user login authentication, and user management session accounting, the Cisco Nexus devices try each option in the order specified. The local option is the default method when other configured options fail.

Authentication and Authorization Process for User Logins

The authentication and authorization process for user login is as occurs:

- When you log in to the required Cisco Nexus device, you can use the Telnet, SSH, Fabric Manager or Device Manager, or console login options.
- When you have configured the AAA server groups using the server group authentication method, the Cisco Nexus device sends an authentication request to the first AAA server in the group as follows:

  If the AAA server fails to respond, then the next AAA server is tried and so on until the remote server responds to the authentication request.

  If all AAA servers in the server group fail to respond, the servers in the next server group are tried.

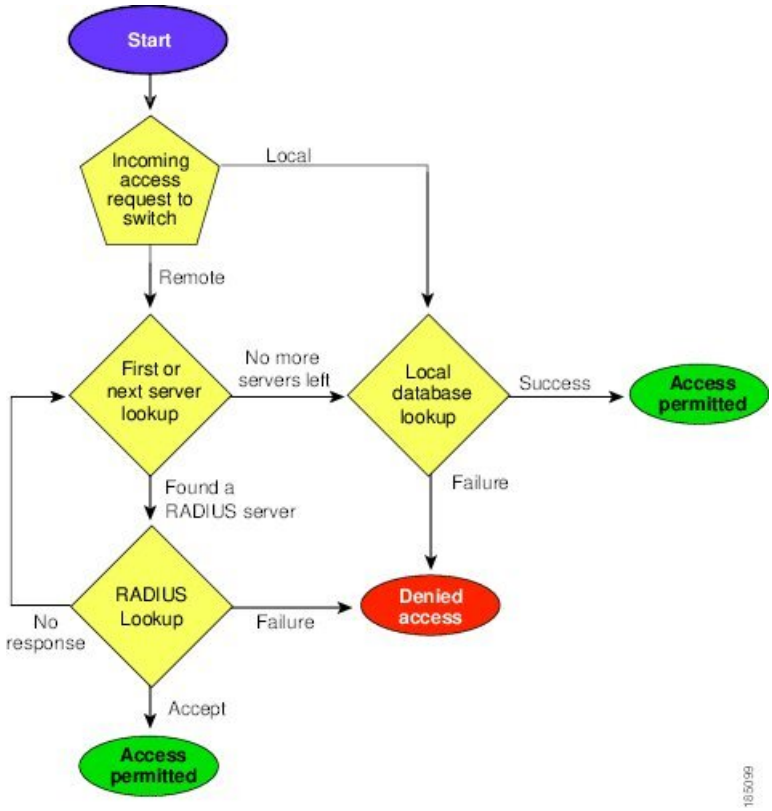
  If all configured methods fail, the local database is used for authentication.
- If a Cisco Nexus device successfully authenticates you through a remote AAA server, the following conditions apply:

  If the AAA server protocol is RADIUS, user roles specified in the cisco-av-pair attribute are downloaded with an authentication response.

  If the AAA server protocol is TACACS+, another request is sent to the same server to get the user roles specified as custom attributes for the shell.
- If your username and password are successfully authenticated locally, the Cisco Nexus device logs you in and assigns you the roles configured in the local database.

The following figure shows a flowchart of the authentication and authorization process.

Figure 1: Authentication and Authorization Flow for User Login



In the figure, "No more servers left" means that there is no response from any server within this server group.

Prerequisites for Remote AAA

Remote AAA servers have the following prerequisites:

- At least one RADIUS or TACACS+ server must be IP reachable.
- The Cisco Nexus device is configured as a client of the AAA servers.
- The preshared secret key is configured on the Cisco Nexus device and on the remote AAA servers.
- The remote server responds to AAA requests from the Cisco Nexus device.

Guidelines and Limitations for AAA

The Cisco Nexus devices do not support all numeric usernames, whether created with TACACS+ or RADIUS, or created locally. If an all numeric username exists on an AAA server and is entered during a login, the Cisco Nexus device still logs in the user.

Caution You should not create user accounts with usernames that are all numeric.

Beginning with Cisco NX-OS release 10.4(3)F, support for SSH based authorization of X.509 certificates using TACACS+ server is being provided on the Cisco Nexus 3548 Series platform switches. This feature can be enabled using **aaa authorization ssh-certificate default group** *tac-group-name* command. For more information, see Configuring AAA SSH-Cert-Authorization on TACACS Servers, on page 18.

Configuring AAA

Configuring Console 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 Nexus device.
- Username only **none**

The default method is local.

Note The **group radius** and **group** *server-name* forms of the **aaa authentication** command are used for a set of previously defined RADIUS servers. Use the **radius server-host** command to configure the host servers. Use the **aaa group server radius** command to create a named group of servers.

Before you configure console login authentication methods, configure RADIUS or TACACS+ server groups as needed.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**}
3. switch(config)# **exit**
4. (Optional) switch# **show aaa authentication**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**}	Configures login authentication methods for the console. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius** —Uses the global pool of RADIUS servers for authentication. • *named-group* —Uses a named subset of TACACS+ or RADIUS servers for authentication. The **local** method uses the local database for authentication. The **none** method uses the username only. The default console login method is **local**, which is used when no methods are configured or when all of the configured methods fail to respond.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show aaa authentication**	Displays the configuration of the console login authentication methods.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure authentication methods for the console login:

```
switch# configure terminal
switch(config)# aaa authentication login console group radius
switch(config)# exit
switch# show aaa authentication
switch# copy running-config startup-config
```

Configuring Default Login Authentication Methods

The default method is local.

Before you configure default login authentication methods, configure RADIUS or TACACS+ server groups as needed.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **aaa authentication login default** {**group** *group-list* [**none**] | **local** | **none**}
3. switch(config)# **exit**
4. (Optional) switch# **show aaa authentication**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login default** {**group** *group-list* [**none**] | **local** | **none**}	Configures the default authentication methods. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius** —Uses the global pool of RADIUS servers for authentication. • *named-group* —Uses a named subset of TACACS+ or RADIUS servers for authentication. The **local** method uses the local database for authentication. The **none** method uses the username only. The default login method is **local** , which is used when no methods are configured or when all of the configured methods do not respond.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show aaa authentication**	Displays the configuration of the default login authentication methods.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Enabling Login Authentication Failure Messages

When you log in, the login is processed by the local user database if the remote AAA servers do not respond. If you have enabled the displaying of login failure messages, the following message is displayed:

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **aaa authentication login error-enable**
3. switch(config)# **exit**
4. (Optional) switch# **show aaa authentication**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login error-enable**	Enables login authentication failure messages. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show aaa authentication**	Displays the login failure message configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring AAA Command Authorization

When a TACACS+ server authorization method is configured, you can authorize every command that a user executes with the TACACS+ server which includes all EXEC mode commands and all configuration mode commands.

The authorization methods include the following:

- Group—TACACS+ server group
- Local—Local role-based authorization
- None—No authorization is performed

The default method is Local.

Note Authorization on the console session is not supported on the Cisco Nexus 5000 platform. It is supported on the Cisco Nexus 5500 platform, release 6.x onwards.

Before you begin

You must enable TACACS+ before configuring AAA command authorization.

SUMMARY STEPS

1. **configure terminal**

2. **aaa authorization** {**commands** | **config-commands**} {**default**} {{[**group** *group-name*] | [**local**]} | {[**group** *group-name*] | [**none**]}}

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authorization** {**commands**	**config-commands**} {**default**} {{[**group** *group-name*]	[**local**]}	{[**group** *group-name*]	[**none**]}}
					Example:
					`switch(config)# aaa authorization config-commands default group tac1`
					Example:
					`switch# aaa authorization commands default group tac1`
						Configures authorization parameters.
						Use the **commands**keyword to authorize EXEC mode commands.
						Use the **config-commands** keyword to authorize configuration mode commands.
						Use the **group**, **local**, or **none** keywords to identify the authorization method.

Example

The following example shows how to authorize EXEC mode commands with TACACS+ server group *tac1*:

```
switch# aaa authorization commands default group tac1
```

The following example shows how to authorize configuration mode commands with TACACS+ server group *tac1*:

```
switch(config)# aaa authorization config-commands default group tac1
```

The following example shows how to authorize configuration mode commands with TACACS+ server group *tac1*:

- If the server is reachable, the command is allowed or not allowed based on the server response.
- If there is an error reaching the server, the command is authorized based on the user's *local* role.

```
switch(config)# aaa authorization config-commands default group tac1 local
```

The following example shows how to authorize configuration mode commands with TACACS+ server group *tac1*:

- If the server is reachable, the command is allowed or not allowed based on the server response.
- If there is an error reaching the server, allow the command regardless of the local role.

```
switch# aaa authorization commands default group tac1 none
```

The following example shows how to authorize EXEC mode commands regardless of the local role:

```
switch# aaa authorization commands default none
```

The following example shows how to authorize EXEC mode commands using the local role for authorization:

```
switch# aaa authorization commands default local
```

Enabling MSCHAP Authentication

Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is the Microsoft version of CHAP. You can use MSCHAP for user logins to a Cisco Nexus device through a remote authentication server (RADIUS or TACACS+).

By default, the Cisco Nexus device uses Password Authentication Protocol (PAP) authentication between the switch and the remote server. If you enable MSCHAP, you must configure your RADIUS server to recognize the MSCHAP vendor-specific attributes (VSAs).

The following table describes the RADIUS VSAs required for MSCHAP.

Table 4: MSCHAP RADIUS VSAs

Vendor-ID Number	Vendor-Type Number	VSA	Description
311	11	MSCHAP-Challenge	Contains the challenge sent by an AAA server to an MSCHAP user. It can be used in both Access-Request and Access-Challenge packets.
211	11	MSCHAP-Response	Contains the response value provided by an MSCHAP user in response to the challenge. It is only used in Access-Request packets.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **aaa authentication login mschap enable**
3. switch(config)# **exit**
4. (Optional) switch# **show aaa authentication login mschap**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa authentication login mschap enable**	Enables MS-CHAP authentication. The default is disabled.

	Command or Action	Purpose
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show aaa authentication login mschap**	Displays the MS-CHAP configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring AAA Authorization on TACACS+ Servers

You can configure the default AAA authorization method for TACACS+ servers.

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authorization ssh-certificate default** {**group** *group-list* [**none**] | **local** | **none**}
3. **exit**
4. (Optional) **show aaa authorization** [**all**]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authorization ssh-certificate default** {**group** *group-list* [**none**]	**local**	**none**}
			Example:
			`switch(config)# aaa authorization ssh-certificate`
			`default group TACACSServer1 TACACSServer2`
				Configures the default AAA authorization method for the TACACS+ servers.
				The **ssh-certificate** keyword configures TACACS+ or local authorization with certificate authentication. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role.
				The *group-list* argument consists of a space-delimited list of TACACS+ server group names. Servers belonging to this group are contacted for AAA authorization. The **local** method uses the local database for authorization, and the **none** method specifies that no AAA authorization be used.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

	Command or Action	Purpose
Step 4	(Optional) **show aaa authorization** [**all**] **Example:** `switch# show aaa authorization`	Displays the AAA authorization configuration. The **all** keyword displays the default values.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring AAA SSH-Cert-Authorization on TACACS Servers

To configure AAA SSH-Cert-Authorization on TACACS Servers, follow these steps:

SUMMARY STEPS

1. **configure terminal**
2. **aaa authorization ssh-certificate default** {**group** *group-list* [**none**] | **local** | **none**}
3. **exit**
4. (Optional) **show aaa authorization** [**all**]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authorization ssh-certificate default** {**group** *group-list* [**none**]	**local**	**none**} **Example:** `switch(config)# aaa authorization ssh-certificate default group tac1`	Configures the default AAA authorization-method for SSH request having X509 certificate as TACACS server-group(s). The **ssh-certificate** keyword configures TACACS or local authorization with certificate authentication. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role. The *group-list* argument consists of a space-delimited list of TACACS server group names. Servers belonging to this group are contacted for AAA authorization. The **local** method uses the local database for authorization, and the **none** method specifies that no AAA authorization be used.
Step 3	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`switch(config)# exit` `switch#`	
Step 4	(Optional) **show aaa authorization** [**all**] **Example:** `switch# show aaa authorization`	Displays the AAA authorization configuration. The **all** keyword displays the default values.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring AAA Accounting Default Methods

The Cisco Nexus device supports TACACS+ and RADIUS methods for accounting. The switches report user activity to TACACS+ or RADIUS security servers in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the AAA server.

When you activate AAA accounting, the Cisco Nexus device reports these attributes as accounting records, which are then stored in an accounting log on the security server.

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

- RADIUS server group—Uses the 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 for accounting.

Note If you have configured server groups and the server groups do not respond, by default, the local database is used for authentication.

Before you begin

Before you configure AAA accounting default methods, configure RADIUS or TACACS+ server groups as needed.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **aaa accounting default** {**group** *group-list* | **local**}
3. switch(config)# **exit**
4. (Optional) switch# **show aaa accounting**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **aaa accounting default** {**group** *group-list* | **local**}	Configures the default accounting method. One or more server group names can be specified in a space-separated list. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius** —Uses the global pool of RADIUS servers for accounting. • *named-group* —Uses a named subset of TACACS+ or RADIUS servers for accounting. The **local** method uses the local database for accounting. The default method is **local**, which is used when no server groups are configured or when all the configured server group do not respond.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show aaa accounting**	Displays the configuration AAA accounting default methods.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

About No Service Password-Recovery

The No Service Password-Recovery feature enables anyone with console access, the ability to access the router and its network.

Enabling No Service Password-Recovery

If the no service password-recovery feature is enabled, then none except the administrator with network privileges will be able to modify the administrator password.

Before you begin

If you plan to enter the no service password-recovery command, Cisco recommends that you save a copy of the system configuration file in a location away from the device.

SUMMARY STEPS

1. **configure terminal**
2. **no service password-recovery**
3. (Optional) **copy running-config startup-config**

4. **Reload**
5. **exit**
6. (Optional) **show user-account**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no service password-recovery**
	Example:
	`switch(config)# no service password-recovery`
	`WARNING: Executing this command will disable the password recovery mechanism. Do not execute this command without another plan for password recovery. Are you sure you want to continue? (y/n) : [y] y`
	`switch(config)# copy run start`
	`[########################################] 100%`
	`Copy complete, now saving to disk (please wait)...`
	`Copy complete.`
		Disables the password recovery mechanism.

Step 3	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Step 4	**Reload**
	Example:
	`switch(config)# Reload`
	`This command will reboot the system. (y/n)?  [n] y`
	`2018 Jun 26 16:23:19 BAR %$ VDC-1 %$ %PLATFORM-2-PFM_SYSTEM_RESET: Manual system restart from Command Line Interface`
	
	`CISCO SWITCH Ver 8.34`
	
	`CISCO SWITCH Ver 8.34`
	`Manual system restart from Command Line Interface`
	`writing reset reason 9,`
	`..`
	`..`
	
	`switch(boot)# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(boot)(config)# admin-password Abcd!123$`
	`ERROR: service password-recovery disabled. Cannot change password!`
	`switch(boot)(config)#`	

	Command or Action	Purpose

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 6	(Optional) **show user-account** **Example:** `switch# show user-account`	Displays the role configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Using AAA Server VSAs

VSAs

You can use vendor-specific attributes (VSAs) to specify the Cisco Nexus device user roles and SNMPv3 parameters on AAA servers.

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating VSAs between the network access server and the RADIUS server. The IETF uses attribute 26. 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 seperator 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 RADIUS servers for authentication on a Cisco Nexus device, 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.

VSA Format

The following VSA protocol options are supported by the Cisco Nexus device:

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

The following attributes are supported by the Cisco Nexus device:

- 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.

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

Specifying Switch User Roles and SNMPv3 Parameters on AAA Servers

You can use the VSA cisco-av-pair on AAA servers to specify user role mapping for the Cisco Nexus device 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.

Note For information on Cisco Unified Wireless Network TACACS+ configurations and to change the user roles, see Cisco Unified Wireless Network TACACS+ Configuration.

You can also specify your SNMPv3 authentication and privacy protocol attributes as follows:

```
shell:roles="roleA roleB..." snmpv3:auth=SHA priv=AES-128
```

The SNMPv3 authentication protocol options are SHA and MD5. The privacy protocol options are AES-128 and DES. If you do not specify these options in the cisco-av-pair attribute, MD5 and DES are the default authentication protocols.

For additional information, see the Configuring User Accounts and RBAC chapter in the System Management Configuration Guide for your Cisco Nexus device.

Monitoring and Clearing the Local AAA Accounting Log

The Cisco Nexus device maintains a local log for the AAA accounting activity.

SUMMARY STEPS

1. switch# **show accounting log** [*size*] [**start-time** *year month day hh* **:** *mm* **:** *ss*]
2. (Optional) switch# **clear accounting log**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show accounting log** [*size*] [**start-time** *year month day hh* **:** *mm* **:** *ss*]	Displays the accounting log contents. By default, the command output contains up to 250,000 bytes of the accounting log. You can use the size argument to limit command output. The range is from 0 to 250000 bytes. You can also specify a start time for the log output.
Step 2	(Optional) switch# **clear accounting log**	Clears the accounting log contents.

Verifying the AAA Configuration

To display AAA configuration information, perform one of the following tasks:

Command	Purpose
show aaa accounting	Displays AAA accounting configuration.
show aaa authentication [**login** {**error-enable** | **mschap**}]	Displays AAA authentication information.
show aaa authorization	Displays AAA authorization information.
show aaa groups	Displays the AAA server group configuration.
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.

Configuration Examples for AAA

The following example shows how to configure AAA:

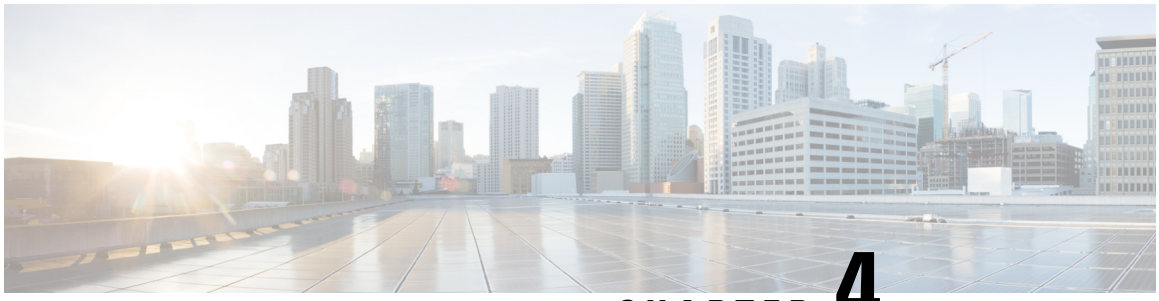
```
switch(config)# aaa authentication login default group radius
switch(config)# aaa authentication login console group radius
switch(config)# aaa accounting default group radius
```

Default AAA Settings

The following table lists the default settings for AAA parameters.

Table 5: Default AAA Parameters

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



CHAPTER 4

Configuring 802.1X

This chapter describes how to configure IEEE 802.1X port-based authentication on Cisco NX-OS devices and includes the following sections:

About 802.1X

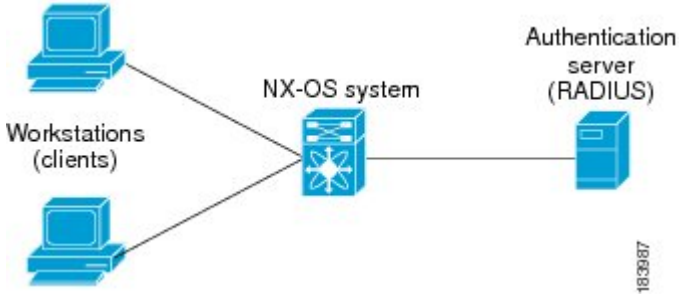
802.1X defines a client-server based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. The authentication server authenticates each client connected to a Cisco NX-OS device port.

Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

Device Roles

With 802.1X port-based authentication, the devices in the network have specific roles.

Figure 2: 802.1X Device Roles



The specific roles are as follows:

Supplicant

The client device that requests access to the LAN and Cisco NX-OS device services and responds to requests from the Cisco NX-OS device. The workstation must be running 802.1X-compliant client software such as that offered in the Microsoft Windows XP operating device.

Authentication server

The authentication server performs the actual authentication of the supplicant. The authentication server validates the identity of the supplicant and notifies the Cisco NX-OS device regarding whether the supplicant is authorized to access the LAN and Cisco NX-OS device services. Because the Cisco NX-OS device acts as the proxy, the authentication service is transparent to the supplicant. The Remote Authentication Dial-In User Service (RADIUS) security device with Extensible Authentication Protocol (EAP) extensions is the only supported authentication server; it is available in Cisco Secure Access Control Server, version 3.0. RADIUS uses a supplicant-server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.

Authenticator

The authenticator controls the physical access to the network based on the authentication status of the supplicant. The authenticator acts as an intermediary (proxy) between the supplicant and the authentication server, requesting identity information from the supplicant, verifying the requested identity information with the authentication server, and relaying a response to the supplicant. The authenticator includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

When the authenticator receives EAPOL frames and relays them to the authentication server, the authenticator strips off the Ethernet header and encapsulates the remaining EAP frame in the RADIUS format. This encapsulation process does not modify or examine the EAP frames, and the authentication server must support EAP within the native frame format. When the authenticator receives frames from the authentication server, the authenticator removes the server's frame header, leaving the EAP frame, which the authenticator then encapsulates for Ethernet and sends to the supplicant.

The Cisco NX-OS device can only be an 802.1X authenticator.

Authentication Initiation and Message Exchange

Either the authenticator (Cisco NX-OS device) or the supplicant (client) can initiate authentication. If you enable authentication on a port, the authenticator must initiate authentication when it determines that the port link state transitions from down to up. The authenticator then sends an EAP-request/identity frame to the supplicant to request its identity (typically, the authenticator sends an initial identity/request frame followed

by one or more requests for authentication information). When the supplicant receives the frame, it responds with an EAP-response/identity frame.

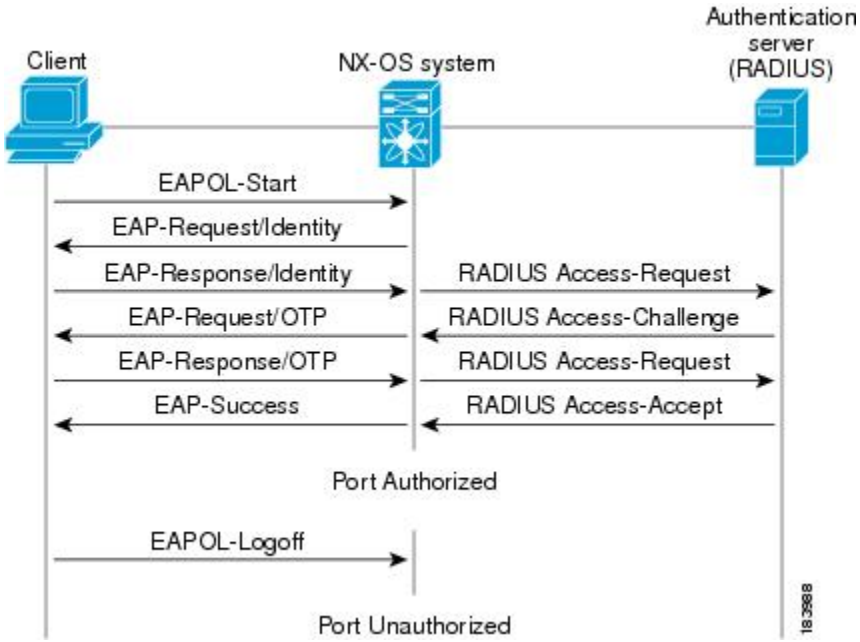
If the supplicant does not receive an EAP-request/identity frame from the authenticator during bootup, the supplicant can initiate authentication by sending an EAPOL-start frame, which prompts the authenticator to request the supplicant's identity.

If 802.1X is not enabled or supported on the network access device, the Cisco NX-OS device drops any EAPOL frames from the supplicant. If the supplicant does not receive an EAP-request/identity frame after three attempts to start authentication, the supplicant transmits data as if the port is in the authorized state. A port in the authorized state means that the supplicant has been successfully authenticated.

When the supplicant supplies its identity, the authenticator begins its role as the intermediary, passing EAP frames between the supplicant and the authentication server until authentication succeeds or fails. If the authentication succeeds, the authenticator port becomes authorized.

The specific exchange of EAP frames depends on the authentication method being used.

Figure 3: Message Exchange



The user's secret pass-phrase never crosses the network at any time such as during authentication or during pass-phrase changes.

Authenticator PAE Status for Interfaces

When you enable 802.1X on an interface, the Cisco NX-OS software creates an authenticator port access entity (PAE) instance. An authenticator PAE is a protocol entity that supports authentication on the interface. When you disable 802.1X on the interface, the Cisco NX-OS software does not automatically clear the authenticator PAE instances. You can explicitly remove the authenticator PAE from the interface and then reapply it, as needed.

Ports in Authorized and Unauthorized States

The authenticator port state determines if the supplicant is granted access to the network. The port starts in the unauthorized state. In this state, the port disallows all ingress and egress traffic except for 802.1X protocol packets. When a supplicant is successfully authenticated, the port transitions to the authorized state, allowing all traffic for the supplicant to flow normally.

If a client that does not support 802.1X is connected to an unauthorized 802.1X port, the authenticator requests the client's identity. In this situation, the client does not respond to the request, the port remains in the unauthorized state, and the client is not granted access to the network.

In contrast, when an 802.1X-enabled client connects to a port that is not running the 802.1X protocol, the client initiates the authentication process by sending the EAPOL-start frame. When no response is received, the client sends the request for a fixed number of times. Because no response is received, the client begins sending frames as if the port is in the authorized state.

Ports can have the following authorization states:

Force authorized
: Disables 802.1X port-based authentication and transitions to the authorized state without requiring any authentication exchange. The port transmits and receives normal traffic without 802.1X-based authentication of the client. This authorization state is the default.

Force unauthorized
: Causes the port to remain in the unauthorized state, ignoring all attempts by the client to authenticate. The authenticator cannot provide authentication services to the client through the interface.

Auto
: Enables 802.1X port-based authentication and causes the port to begin in the unauthorized state, allowing only EAPOL frames to be sent and received through the port. The authentication process begins when the link state of the port transitions from down to up or when an EAPOL-start frame is received from the supplicant. The authenticator requests the identity of the client and begins relaying authentication messages between the client and the authentication server. Each supplicant that attempts to access the network is uniquely identified by the authenticator by using the supplicant's MAC address.

If the supplicant is successfully authenticated (receives an Accept frame from the authentication server), the port state changes to authorized, and all frames from the authenticated supplicant are allowed through the port. If the authentication fails, the port remains in the unauthorized state, but authentication can be retried. If the authentication server cannot be reached, the authenticator can retransmit the request. If no response is received from the server after the specified number of attempts, authentication fails, and the supplicant is not granted network access.

When a supplicant logs off, it sends an EAPOL-logoff message, which causes the authenticator port to transition to the unauthorized state.

If the link state of a port transitions from up to down, or if an EAPOL-logoff frame is received, the port returns to the unauthorized state.

MAC Authentication Bypass

You can configure the Cisco NX-OS device to authorize a supplicant based on the supplicant MAC address by using the MAC authentication bypass feature. For example, you can enable this feature on interfaces configured for 802.1X that are connected to devices such as printers.

If 802.1X authentication times out while waiting for an EAPOL response from the supplicant, the Cisco NX-OS device tries to authorize the client by using MAC authentication bypass.

When you enable the MAC authentication bypass feature on an interface, the Cisco NX-OS device uses the MAC address as the supplicant identity. The authentication server has a database of supplicant MAC addresses that are allowed network access. After detecting a client on the interface, the Cisco NX-OS device waits for an Ethernet packet from the client. The Cisco NX-OS device sends the authentication server a RADIUS-access/request frame with a username and password based on the MAC address. If authorization succeeds, the Cisco NX-OS device grants the client access to the network.

If an EAPOL packet is detected on the interface during the lifetime of the link, the Cisco NX-OS device determines that the device connected to that interface is an 802.1X-capable supplicant and uses 802.1X authentication (not MAC authentication bypass) to authorize the interface. EAPOL history is cleared if the interface link status goes down.

If the Cisco NX-OS device already authorized an interface by using MAC authentication bypass and detects an 802.1X supplicant, the Cisco NX-OS device does not unauthorize the client connected to the interface. When reauthentication occurs, the Cisco NX-OS device uses 802.1X authentication as the preferred reauthentication process.

Clients that were authorized with MAC authentication bypass can be reauthenticated. The reauthentication process is the same as that for clients that were authenticated with 802.1X. During reauthentication, the port remains in the previously assigned VLAN. If reauthentication is successful, the switch keeps the port in the same VLAN.

If reauthentication is based on the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]) and if the Termination-Action RADIUS attribute (Attribute [29]) action is Initialize (the attribute value is DEFAULT), the MAC authentication bypass session ends, and connectivity is lost during reauthentication. If MAC authentication bypass is enabled and the 802.1X authentication times out, the switch uses the MAC authentication bypass feature to initiate reauthorization. For more information about these AV pairs, see RFC 3580, IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines.

MAC authentication bypass interacts with the following features:

802.1X authentication—You can enable MAC authentication bypass only if 802.1X authentication is enabled on the port.

Port security—This feature is not supported on the Nexus 3548 platform switches.

Network admission control (NAC) Layer 2 IP validation—This feature takes effect after an 802.1X port is authenticated with MAC authentication bypass, including hosts in the exception list.

Dynamic VLAN Assignment based on MAC-Based Authentication (MAB)

The Cisco Nexus 3548 Series switches supports dynamic VLAN assignment. After the 802.1x authentication or MAB is completed, before bringing up the port, you may want to (as part of authorization) allow the peer/host to be placed into a particular VLAN based as a result of the authentication. The RADIUS server typically indicates the desired VLAN by including tunnel attributes within the Access-Accept message. This procedure of getting the VLAN an binding it to the port constitutes to Dynamic VLAN assignment.

VLAN Assignment from RADIUS

After authentication is completed either through dot1x or MAB, the response from the RADIUS server can have dynamic VLAN information, which can be assigned to a port. This information is present in response from RADIUS server in Accept-Access message in the form of tunnel attributes. For use in VLAN assignment, the following tunnel attributes are sent:

Tunnel-type=VLAN(13)

Tunnel-Medium-Type=802

Tunnel-Private-Group-ID=VLANID

All the three parameters must be received for configuring access VLAN.

Single Host and Multiple Host Support

The 802.1X feature can restrict traffic on a port to only one endpoint device (single-host mode) or allow traffic from multiple endpoint devices on a port (multi-host mode).

Single-host mode allows traffic from only one endpoint device on the 802.1X port. Once the endpoint device is authenticated, the Cisco NX-OS device puts the port in the authorized state. When the endpoint device leaves the port, the Cisco NX-OS device put the port back into the unauthorized state. A security violation in 802.1X is defined as a detection of frames sourced from any MAC address other than the single MAC address authorized as a result of successful authentication. In this case, the interface on which this security association violation is detected (EAPOL frame from the other MAC address) will be disabled. Single host mode is applicable only for host-to-switch topology and when a single host is connected to the Layer 2 (Ethernet access port) or Layer 3 port (routed port) of the Cisco NX-OS device.

Only the first host has to be authenticated on the 802.1X port configured with multiple host mode. The port is moved to the authorized state after the successful authorization of the first host. Subsequent hosts are not required to be authorized to gain network access once the port is in the authorized state. If the port becomes unauthorized when reauthentication fails or an EAPOL logoff message is received, all attached hosts are denied access to the network. The capability of the interface to shut down upon security association violation is disabled in multiple host mode. This mode is applicable for both switch-to-switch and host-to-switch topologies.

Supported Topology

The 802.1X port-based authentication supports point-to-point topology.

In this configuration, only one supplicant (client) can connect to the 802.1X-enabled authenticator (Cisco NX-OS device) port. The authenticator detects the supplicant when the port link state changes to the up state. If a supplicant leaves or is replaced with another supplicant, the authenticator changes the port link state to down, and the port returns to the unauthorized state.

Licensing Requirements for 802.1x

The following table shows the licensing requirements for this feature:

Table 6: Licensing Requirements

Product	License Requirement
Cisco NX-OS	802.1X requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you.

Guidelines and Limitations for 802.1x

802.1X port-based authentication has the following configuration guidelines and limitations:

- Multi-authentication mode is enabled on an 802.1X port. VLAN assignment occurs successfully for the first authenticated host. Subsequent authorized (based on user credentials) data hosts are considered successfully authenticated, provided either they have no VLAN assignment or have a VLAN assignment matching the first successfully authenticated host on the port. This ensures that all successfully authenticated hosts on a port are members of the same VLAN. Flexibility of VLAN assignment is only provided to the first authenticated host.
- Cisco Nexus Series switches do not support 802.1X on the following:
  - 40G interfaces
  - Transit topology set ups
  - VPC ports
  - PVLAN ports
  - L3 (routed) ports
  - Port security
  - Ports that are enabled with CTS and MACsec
  - Dot1x with LACP port-channels
  - Disable 802.1X on VPC ports and all unsupported features
- The Cisco NX-OS software supports 802.1X authentication only on physical ports.
- The Cisco NX-OS software does not support 802.1X authentication on port channels or subinterfaces.
- The Cisco NX-OS software supports 802.1X authentication on member ports of a port channel but not on the port channel itself.
- When the members are configured for 802.1X, Cisco NX-OS software does not support configuring single-host mode on port channel members. Only multi-host mode is supported on the member ports.
- Member ports with and without a 802.1X configuration can coexist in a port channel. However, you must ensure the identical 802.1X configuration on all the member ports in order for channeling to operate with 802.1X.
- When you enable 802.1X authentication, supplicants are authenticated before any other Layer 2 or Layer 3 features are enabled on an Ethernet interface.
- On a 802.1X enabled port, the STP BPDUs are permitted only after a successful authentication. We recommend that you enable the 802.1X functionality only on the STP edge ports to avoid STP disputes.
- The Cisco NX-OS software supports 802.1X authentication only on Ethernet interfaces that are in a port channel, a trunk, or an access port.
- The Cisco NX-OS software does not work with the CTS or the MACsec features. Global "mac-learn disable" and dot1x feature are mutually exclusive and cannot be configured together.

- Dot1x is mutually exclusive with the IP Source Guard and URPF features and cannot be configured together. When you upgrade the Cisco Nexus Series switch to Cisco NX-OS Release 9.3(3), you must disable one of these features.
- The Cisco NX-OS software does not support single host mode on trunk interfaces or member interfaces in a port channel.
- The Cisco NX-OS software does not support MAC address authentication bypass on a port channel. The multi-host mode is the only supported mode on the port-channels.
- The Cisco NX-OS software does not support Dot1x on vPC ports and MCT.
- During a switch reload, Dot1x does not generate RADIUS accounting stops.
- The Cisco NX-OS software does not support the following 802.1X protocol enhancements:
  - One-to-many logical VLAN name to ID mapping
  - Web authorization
  - Dynamic domain bridge assignment
  - IP telephony
- In order to prevent reauthentication of inactive sessions, use the authentication timer inactivity command to set the inactivity timer to an interval shorter than the reauthentication interval set with the authentication timer reauthenticate command.
- A security violation occurs when the same MAC is learned on a different VLAN with dot1x enabled on the interface.
- Configuring mac learn disable with dot1x enabled on a DME enabled platform does not display the error messages.
- Tagged EAPOL frames are processed although the VLAN is not configured on the interface and the authentication is successful on the interface for the client.
- Secure MAC learned on the orphan port is not synced on the vPC peer.
- The Cisco Nexus 3500 series switches do not support MAC address authentication bypass on a port channel and trunk interfaces.
- Beginning with Cisco NX-OS Release 10.4(3)F, EAP-TLS supports Transport Layer Security version 1.3 and 1.2 on Cisco Nexus switches.

Note If the RADIUS server is not capable of TLS v1.3, then TLS v1.2 is used, as it is the minimum supported version.

Default Settings for 802.1x

Table 7: Default 802.1x Parameters

Parameters	Default
802.1X feature	Disabled
AAA 802.1X authentication method	Not configured
Per-interface 802.1X protocol enable state	Disabled (force-authorized) The port transmits and receives normal traffic without 802.1X-based authentication of the supplicant.
Periodic reauthentication	Disabled
Number of seconds between reauthentication attempts	3,600 seconds
Quiet timeout period	60 seconds (number of seconds the Cisco NX-OS device remains in the quiet state following a failed authentication exchange with the supplicant)
Retransmission timeout period	30 seconds (number of seconds the Cisco NX-OS device waits for a response to an EAP request/identity frame from the supplicant before retransmitting the request)
Maximum retransmission number	Two times (number of times the Cisco NX-OS device sends an EAP-request/identity frame before restarting the authentication process)
Host mode	Single host
Supplicant timeout period	30 seconds (time that the Cisco NX-OS device waits for a response before retransmitting the request to the supplicant when relaying a request from the authentication server to the supplicant)
Authentication server timeout period	30 seconds (time that the Cisco NX-OS device waits for a reply before retransmitting the response to the server when relaying a response from the supplicant to the authentication server)

Configuring 802.1X

Process for Configuring 802.1X

This section describes the process for configuring 802.1X.

SUMMARY STEPS

1. Enable the 802.1X feature.
2. Configure the connection to the remote RADIUS server.
3. Enable 802.1X feature on the Ethernet interfaces.

DETAILED STEPS

Step 1 Enable the 802.1X feature.

Step 2 Configure the connection to the remote RADIUS server.

Step 3 Enable 802.1X feature on the Ethernet interfaces.

Enabling 802.1X

You must enable the 802.1X feature on the Cisco NX-OS device before authenticating any supplicant devices.

SUMMARY STEPS

1. **configure terminal**
2. **feature dot1x**
3. **exit**
4. **show dot1x**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature dot1x** **Example:** `switch(config)# feature dot1x`	Enables the 802.1X feature. The default is disabled.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	**show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X feature status.

	Command or Action	Purpose
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring AAA Authentication Methods for 802.1X

You can use remote RADIUS servers for 802.1X authentication. You must configure RADIUS servers and RADIUS server groups and specify the default AAA authentication method before the Cisco NX-OS device can perform 802.1X authentication.

Before you begin

Obtain the names or addresses for the remote RADIUS server groups.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authentication dot1x default group**
3. **exit**
4. **show radius-server**
5. **show radius-server group**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authentication dot1x default group** **Example:** `switch(config)# aaa authentication dot1x default group rad2`	Specifies the RADIUS server groups to use for 802.1X authentication. The group-list argument consists of a space-delimited list of group names. The group names are the following: • radius—Uses the global pool of RADIUS servers for authentication. • named group—Uses the global pool of RADIUS servers for authentication.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

	Command or Action	Purpose
Step 4	**show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 5	**show radius-server group** **Example:** `switch# show radius-server group rad2`	Displays the RADIUS server group configuration.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Controlling 802.1x Authentication on an Interface

You can control the 802.1X authentication performed on an interface. An interface can have the following 802.1X authentication states:

Auto
Enables 802.1X authentication on the interface.

Force-authorized
Disables 802.1X authentication on the interface and allows all traffic on the interface without authentication. This state is the default.

Force-unauthorized
Disallows all traffic on the interface.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot* | *port*
3. **dot1x port-control** {**auto** | **force-authorized** | **force-unauthorised**}
4. **exit**
5. **show dot1x all**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**interface ethernet** *slot* | *port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x port-control** {**auto** | **force-authorized** | **force-unauthorised**} **Example:** `switch(config-if)# dot1x port-control auto`	Changes the 802.1X authentication state on the interface. The default is force-authorized.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	**show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating or Removing an Authenticator PAE on an Interface

You can create or remove the 802.1X authenticator port access entity (PAE) instance on an interface.

Note By default, the Cisco NX-OS software creates the authenticator PAE instance on the interface when you enable 802.1X on an interface.

Before you begin

Enable the 802.1X feature.

SUMMARY STEPS

1. **configure terminal**
2. **show dot1x interface ethernet** *slot* | *port*
3. **interface ethernet** *slot* | *port*
4. [**no**] **dot1x pae authenticator**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**show dot1x interface ethernet** *slot* | *port* **Example:** `switch# show dot1x interface ethernet 2/1`	Displays the 802.1X configuration on the interface.

Step 3	**interface ethernet** *slot* | *port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 4	[**no**] **dot1x pae authenticator** **Example:** `switch(config-if)# dot1x pae authenticator`	Creates an authenticator PAE instance on the interface. Use the **no** form to remove the PAE instance from the interface. **Note** Creates an authenticator PAE instance on the interface. Use the **no** form to remove the PAE instance from the interface.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling Periodic Reauthentication for an Interface

You can enable periodic 802.1X reauthentication on an interface and specify how often it occurs. If you do not specify a time period before enabling reauthentication, the number of seconds between reauthentication defaults to the global value.

Note During the reauthentication process, the status of an already authenticated supplicant is not disrupted.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot* / port
3. dot1x re-authentication
4. **dot1x timeout re-authperiod**

5. **exit**
6. **show dot1x all**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot* / port
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	dot1x re-authentication **Example:** `switch(config-if)# dot1x re-authentication`	Enables periodic reauthentication of the supplicants connected to the interface. By default, periodic authentication is disabled.
Step 4	**dot1x timeout re-authperiod** **Example:** `switch(config-if)# dot1x timeout re-authperiod 3300`	Sets the number of seconds between reauthentication attempts. The default is 3600 seconds. The range is from 1 to 65535. **Note** This command affects the behavior of the Cisco NX-OS device only if you enable periodic reauthentication on the interface.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 6	**show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 7	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Manually Reauthenticating Supplicants

You can manually reauthenticate the supplicants for the entire Cisco NX-OS device or for an interface.

Note During the reauthentication process, the status of an already authenticated supplicant isn't disrupted.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

SUMMARY STEPS

1. **dot1x re-authenticate** [**interface** *slot* | **port**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**dot1x re-authenticate** [**interface** *slot* | **port**] **Example:** `switch# dot1x re-authenticate interface 2/1`	Reauthenticates the supplicants on the Cisco NX-OS device or on an interface.

Changing 802.1X Authentication Timers for an Interface

You can change the following 802.1X authentication timers on the Cisco NX-OS device interfaces:

Quiet-period timer
When the Cisco NX-OS device can't authenticate the supplicant, the switch remains idle for a set period and then tries again. The quiet-period timer value determines the idle period. An authentication failure might occur because the supplicant provided an invalid password. You can provide a faster response time to the user by entering a smaller number than the default. The default is the value of the global quiet period timer. The range is from 1 to 65535 seconds.

Rate-limit timer
The rate-limit period throttles EAPOL-Start packets from supplicants that are sending too many EAPOL-Start packets. The authenticator ignores EAPOL-Start packets from supplicants that have successfully authenticated for the rate-limit period duration. The default value is 0 seconds and the authenticator processes all EAPOL-Start packets. The range is from 1 to 65535 seconds.

Switch-to-authentication-server retransmission timer for Layer 4 packets
The authentication server notifies the switch each time that it receives a Layer 4 packet. If the switch does not receive a notification after sending a packet, the Cisco NX-OS device waits a set period and then retransmits the packet. The default is 30 seconds. The range is from 1 to 65535 seconds.

Switch-to-supplicant retransmission timer for EAP response frames
The supplicant responds to the EAP-request/identity frame from the Cisco NX-OS device with an EAP-response/identity frame. If the Cisco NX-OS device does not receive this response, it waits a set period (known as the retransmission time) and then retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.
Switch-to-supplicant retransmission timer for EAP request frames

Note Change the default values only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain supplicants and authentication servers.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

SUMMARY STEPS

1. **configure terminal**
2. **configure interface ethernet 2/1**
3. **dot1x timeout quiet-period** *seconds*
4. **dot1x timeout ratelimit-period** *seconds*
5. **dot1x timeout server-timeout** *seconds*
6. **dot1x timeout supp-timeout** *seconds*
7. **dot1x timeout tx-period** *seconds*
8. **dot1x timeout inactivity-period** *seconds*
9. **exit**
10. **show dot1x all**
11. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**configure interface ethernet 2/1**
	Example:
	`switch# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x timeout quiet-period** *seconds* **Example:** `switch(config-if)# dot1x timeout quiet-period 25`	Sets the number of seconds that the authenticator waits for a response to an EAP-request/identity frame from the supplicant before retransmitting the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.
Step 4	**dot1x timeout ratelimit-period** *seconds* **Example:** `switch(config-if)# dot1x timeout ratelimit-period 10`	Sets the number of seconds that the authenticator ignores EAPOL-Start packets from supplicants that have successfully authenticated. The default value is 0 seconds. The range is from 1 to 65535 seconds.
Step 5	**dot1x timeout server-timeout** *seconds* **Example:** `switch(config-if)# dot1x timeout server-timeout 60`	Sets the number of seconds that the Cisco NX-OS device waits before retransmitting a packet to the authentication server. The default is 30 seconds. The range is from 1 to 65535 seconds.
Step 6	**dot1x timeout supp-timeout** *seconds* **Example:**	Sets the number of seconds that the Cisco NX-OS device waits for the supplicant to respond to an EAP request frame

	Command or Action	Purpose
	`switch(config-if)# dot1x timeout supp-timeout 20`	before the Cisco NX-OS device retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.
Step 7	**dot1x timeout tx-period** *seconds* **Example:** `switch(config-if)# dot1x timeout tx-period 40`	Sets the number of seconds between the retransmission of EAP request frames when the supplicant does not send notification that it received the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.
Step 8	**dot1x timeout inactivity-period** *seconds* **Example:** `switch(config-if)# dot1x timeout inactivity-period 1800`	Sets the number of seconds the switch can remain inactive. The recommended minimum value is 1800 seconds.

Step 9	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 10	**show dot1x all** **Example:** `switch# show dot1x all`	Displays the 802.1X configuration.
Step 11	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling MAC Authentication Bypass

You can enable MAC authentication bypass on an interface that has no supplicant connected.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot* | *port*
3. **dot1x mac-auth-bypass** [**eap**]
4. **exit**
5. **show dot1x all**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot* | *port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x mac-auth-bypass** [**eap**] **Example:** `switch(config-if)# dot1x mac-auth-bypass`	Enables MAC authentication bypass. The default is bypass disabled. Use the **eap** keyword to configure the Cisco NX-OS device to use EAP for authorization.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	**show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling Single Host or Multiple Hosts Mode

You can enable single host or multiple hosts mode on an interface.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot* | *port*
3. **dot1x host-mode** { **multi-host** | **single-host**}
4. **dot1x host-mode multi-auth**
5. **exit**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuraiton mode.

Step 2	**interface ethernet** *slot* | *port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x host-mode** { **multi-host** | **single-host**} **Example:** `switch(config-if)# dot1x host-mode multi-host`	Configures the host mode. The default is single-host. **Note** Make sure that the dot1x port-control interface configuration command is set to auto for the specified interface.
Step 4	**dot1x host-mode multi-auth** **Example:** `switch(config-if)# dot1x host-mode multi-auth`	Configures the multiple authentication mode. The port is authorized only on a successful authentication of either EAP or MAB or a combination of both. Failure to authenticate will restrict network access.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 6	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Disabling the 802.1X feature

You can disable the 802.1X feature on the Cisco NX-OS device.

When you disable 802.1X, all related configurations are automatically discarded. The Cisco NX-OS software creates an automatic checkpoint that you can use if you reenable 802.1X and want to recover the configuration. For more information, see the *Cisco NX-OS System Management Configuration Guide* for your platform.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

SUMMARY STEPS

1. **configure terminal**
2. **no feature dot1x**
3. **exit**

4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no feature dot1x** **Example:** `no feature dot1x`	Disables 802.1X. **Note** Disabling the 802.1X feature removes all 802.1X configuration.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Resetting the 802.1X Interface Configuration to the Default Values

You can reset the 802.1X configuration for an interface to the default values.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slots port*
3. **dot1x default**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**interface ethernet** *slots port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x default** **Example:** `switch(config-if)# dot1x default`	Reverts to the 802.1X configuration default values for the interface.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Setting the Maximum Authenticator-to-Supplicant Frame for an Interface

You can set the maximum number of times that the Cisco NX-OS device retransmits authentication requests to the supplicant on an interface before the session times out. The default is 2 times and the range is from 1 to 10.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slots* | *port*
3. **dot1x max-req** *count*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slots*	*port*
		Example:
		`switch(config)# interface ethernet 2/1`
		`switch(config-if)#`
			Selects the interface to configure and enters interface configuration mode.

	Command or Action	Purpose
Step 3	**dot1x max-req** *count* **Example:** `switch(config-if)# dot1x max-req 3`	Changes the maximum authorization request retry count. The default is 2 times and the range is from 1 to 10. **Note** Make sure that the dot1x port-control interface configuration command is set to auto for the specified interface.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Setting the Maximum Reauthentication Retry Count on an Interface

You can set the maximum number of times that the Cisco NX-OS device retransmits reauthentication requests to the supplicant on an interface before the session times out. The default is 2 times and the range is from 1 to 10.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slots* | *port*
3. **dot1x max-reauth-req** *retry-count*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slots*	*port*
		Example:
		`switch(config)# interface ethernet 2/1`
		`switch(config-if)#`
			Selects the interface to configure and enters interface configuration mode.

	Command or Action	Purpose
Step 3	**dot1x max-reauth-req** *retry-count* **Example:** `switch(config-if)# dot1x max-reauth-req 3`	Changes the maximum reauthentication request retry count. The default is 2 times and the range is from 1 to 10.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the 802.1X configuration

To display 802.1X information, perform one of the following tasks:

Command	Purpose
show dot1x	Displays the 802.1X feature status.
show dot1x all [**details** | **statistics** | **summary**]	Displays all 802.1X feature status and configuration information.
show dot1x interface ethernet *slot*/*port* [**details** | **statistics** | **summary**]	Displays the 802.1X feature status and configuration information for an Ethernet interface.
show running-config dot1x [**all**]	Displays the 802.1X feature configuration in the running configuration.
show startup-config dot1x	Displays the 802.1X feature configuration in the startup configuration.

For detailed information about the fields in the output from these commands, see the *Cisco NX-OS Security Command Reference* for your platform.

Monitoring 802.1X

You can display the statistics that the Cisco NX-OS device maintains for the 802.1X activity.

Before you begin

Enable the 802.1X feature on the Cisco NX-OS device.

SUMMARY STEPS

1. **show dot1x** {**all** | **interface ethernet** *slot* | *port*} **statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show dot1x** {**all**	**interface ethernet** *slot*	*port*} **statistics** **Example:** `switch# show dot1x all statistics`	Displays the 802.1X statistics.

Configuration Example for 802.1X

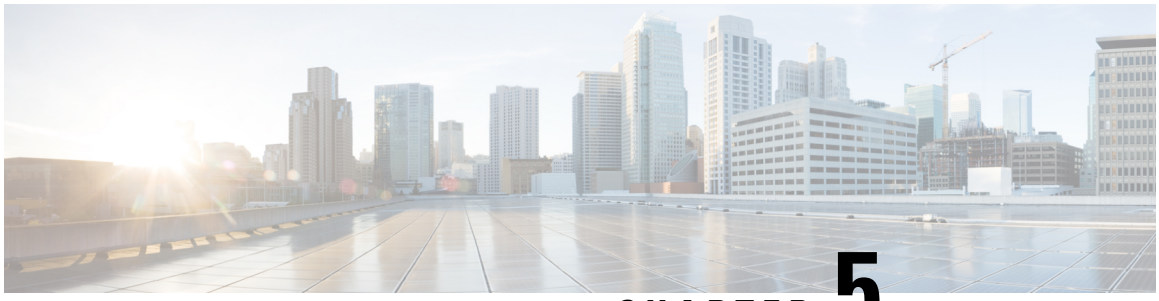
The following example shows how to configure 802.1X for an access port:

```
feature dot1x
aaa authentication dot1x default group rad2
interface Ethernet2/1
dot1x pae-authenticator
dot1x port-control auto
```

The following example shows how to configure 802.1X for a trunk port:

```
feature dot1x
aaa authentication dot1x default group rad2
interface Ethernet2/1
dot1x pae-authenticator
dot1x port-control auto
dot1x host-mode multi-host
```

Note Repeat the **dot1x pae authenticator** and **dot1x port-control auto** commands for all interfaces that require 802.1X authentication.



CHAPTER 5

Configuring RADIUS

This chapter contains the following sections:

Configuring RADIUS

Information About RADIUS

The Remote Access Dial-In User Service (RADIUS) distributed client/server system allows you to secure networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco Nexus devices and send authentication and accounting requests to a central RADIUS server that contains all user authentication and network service access information.

RADIUS Network Environments

RADIUS can be 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 already using RADIUS.

  You can add a Cisco Nexus device with RADIUS to the network. This action might be the first step when you make a transition to an AAA server.

- 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 AAA authentication and set up per-user profiles. Per-user profiles enable the Cisco Nexus device to manage ports using their existing RADIUS solutions and to efficiently manage shared resources to offer different service-level agreements.

Information About RADIUS Operations

When a user attempts to log in and authenticate to a Cisco Nexus device using RADIUS, the following process occurs:

1. The user is prompted for and enters a username and password.
2. The username and encrypted password are sent over the network to the RADIUS server.
3. The user receives one of the following responses from the RADIUS server:
   - ACCEPT—The user is authenticated.
   - REJECT—The user is not authenticated and is prompted to reenter the username and password, or access is denied.
   - CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
   - CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new password.

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

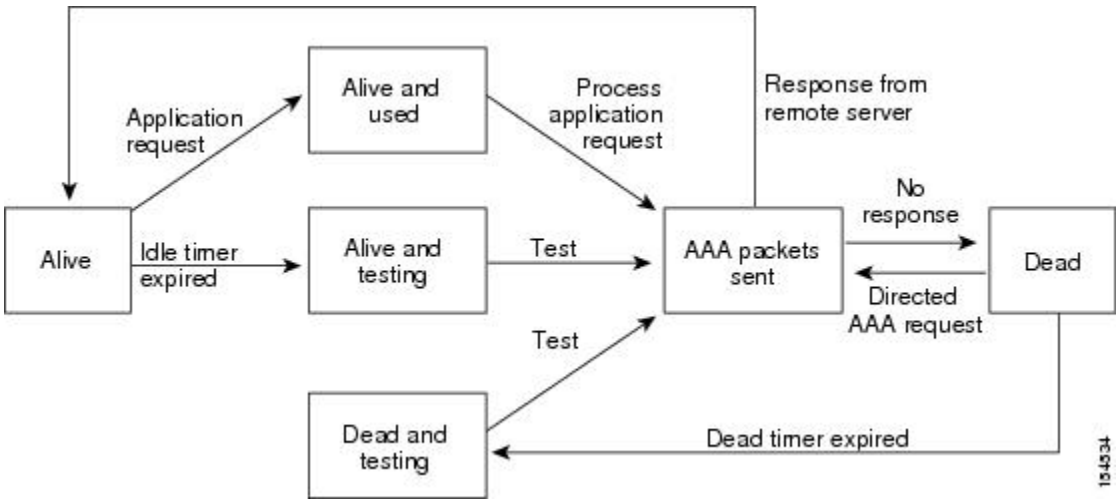
- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 address, access list, and user timeouts.

RADIUS Server Monitoring

An unresponsive RADIUS server can cause delay in processing of AAA requests. You can configure the switch to periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. The switch marks unresponsive RADIUS servers as dead and does not send AAA requests to any dead RADIUS servers. The switch periodically monitors the dead RADIUS servers and brings them to the alive state once they respond. This process verifies that a RADIUS server is in a working state before real AAA requests are sent to the server. Whenever a RADIUS server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the switch displays an error message that a failure is taking place.

The following figure shows the different RADIUS server states:

Figure 4: RADIUS Server States



Note The monitoring interval for alive servers and dead servers are different and can be configured by the user. The RADIUS server monitoring is performed by sending a test authentication request to the RADIUS 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 RADIUS server. The IETF uses attribute 26. 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 Nexus device, 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.

The following VSA protocol options are supported by the Cisco Nexus device:

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

The Cisco Nexus device 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 spaces.

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

Prerequisites for RADIUS

RADIUS has the following prerequisites:

- You must obtain IPv4 addresses or hostnames for the RADIUS servers.
- You must obtain preshared keys from the RADIUS servers.
- Ensure that the Cisco Nexus device is configured as a RADIUS client of the AAA servers.

Guidelines and Limitations for RADIUS

RADIUS has the following configuration guidelines and limitations:

- You can configure a maximum of 64 RADIUS servers on the Cisco Nexus device.
- ASCII (PAP) Authentication is not supported on RADIUS servers.

Guidelines and Limitations for RadSec

RadSec has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.3(1)F, the RADIUS Secure (RadSec) support is provided on Cisco Nexus switches to secure the communication between RADIUS/TCP peers at the transport layer.
- RadSec must be enabled/disabled at the switch level, as the combination of servers having different transport protocols (i.e. UDP and TCP-with-TLS) is not possible.
- **radius-server directed-request** command is not supported along with the RadSec feature.
- **test aaa server radius** command is not supported for the RadSec servers, only **test aaa group** command is supported with the RadSec.
- Dot1x is not officially supported with RadSec.
- RADIUS server monitoring is not supported along with the RadSec servers.
- RADIUS server re-transmit and timeout are applicable to UDP based RADIUS mode and not supported for RadSec servers.
- Beginning with Cisco NX-OS Release 10.4(3)F, TLS version 1.3 and 1.2 is supported on Cisco Nexus switches. TLS v1.1 is deprecated.

Configuring RADIUS Servers

This section describes how to configure RADIUS servers.

SUMMARY STEPS

1. Establish the RADIUS server connections to the Cisco Nexus device.
2. Configure the preshared secret keys for the RADIUS servers.
3. If needed, configure RADIUS server groups with subsets of the RADIUS servers for AAA authentication methods.
4. If needed, configure any of the following optional parameters:
5. If needed, configure periodic RADIUS server monitoring.

DETAILED STEPS

Step 1 Establish the RADIUS server connections to the Cisco Nexus device.

Step 2 Configure the preshared secret keys for the RADIUS servers.

Step 3 If needed, configure RADIUS server groups with subsets of the RADIUS servers for AAA authentication methods.

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

- Dead-time interval.
- Allow specification of a RADIUS server at login.
- Transmission retry count and timeout interval.
- Accounting and authentication attributes.

Step 5 If needed, configure periodic RADIUS server monitoring.

Configuring RADIUS Server Hosts

You must configure the IPv4 address or the hostname for each RADIUS server that you want to use for authentication. All RADIUS server hosts are added to the default RADIUS server group. You can configure up to 64 RADIUS servers.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server host** {*ipv4-address* | *host-name*}
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *host-name*}	Specifies the IPv4 address or hostname for a RADIUS server.

	Command or Action	Purpose
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure host 10.10.1.1 as a RADIUS server:

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

Configuring RADIUS Global Preshared Keys

You can configure preshared keys at the global level for all servers used by the Cisco Nexus device. A preshared key is a shared secret text string between the switch and the RADIUS server hosts.

Before you begin

Obtain the preshared key values for the remote RADIUS servers

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server key** [**0** | **7**] *key-value*
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server key** [**0** | **7**] *key-value*	Specifies a preshared key for all RADIUS servers. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. By default, no preshared key is configured.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.

	Command or Action	Purpose
		Note The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure preshared keys at the global level for all servers used by the device:

```
switch# configure terminal
switch(config)# radius-server key 0 QsEfThUkO
switch(config)# exit
switch# copy running-config startup-config
```

Configuring RADIUS Server Preshared Keys

A preshared key is a shared secret text string between the Cisco Nexus device and the RADIUS server host.

Before you begin

Obtain the preshared key values for the remote RADIUS servers.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *key-value*
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *key-value*	Specifies a preshared key for a specific RADIUS server. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. This preshared key is used instead of the global preshared key.

	Command or Action	Purpose
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration. **Note** The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure RADIUS preshared keys:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1 key 0 PlIjUhYg
switch(config)# exit
switch# show radius-server
switch# 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 belong to the RADIUS protocol. The servers are tried in the same order in which you configure them.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch (config)# **aaa group server radius** *group-name*
3. switch (config-radius)# **server** {*ipv4-address* |*server-name*}
4. (Optional) switch (config-radius)# **deadtime** *minutes*
5. (Optional) switch(config-radius)# **source-interface** *interface*
6. switch(config-radius)# **exit**
7. (Optional) switch(config)# **show radius-server group** [*group-name*]
8. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch (config)# **aaa group server radius** *group-name*	Creates a RADIUS server group and enters the RADIUS server group configuration submode for that group.

	Command or Action	Purpose
		The *group-name* argument is a case-sensitive, alphanumeric string with a maximum of 127 characters.
Step 3	switch (config-radius)# **server** {*ipv4-address* |*server-name*}	Configures the RADIUS server as a member of the RADIUS server group. If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.
Step 4	(Optional) switch (config-radius)# **deadtime** *minutes*	Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440. **Note** If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	(Optional) switch(config-radius)# **source-interface** *interface*	Assigns a source interface for a specific RADIUS server group. The supported interface types are management and VLAN. **Note** Use the **source-interface** command to override the global source interface assigned by the **ip radius source-interface** command.
Step 6	switch(config-radius)# **exit**	Exits configuration mode.
Step 7	(Optional) switch(config)# **show radius-server group** [*group-name*]	Displays the RADIUS server group configuration.
Step 8	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure a RADIUS server group:

```
switch# configure terminal
switch (config)# aaa group server radius RadServer
switch (config-radius)# server 10.10.1.1
switch (config-radius)# deadtime 30
switch (config-radius)# use-vrf management
switch (config-radius)# exit
switch (config)# show radius-server group
switch (config)# copy running-config startup-config
```

What to do next

Apply the RADIUS server groups to an AAA service.

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. You can also configure a different source interface for a specific RADIUS server group.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip radius source-interface** *interface*
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server**
5. (Optional) switch# **copy running-config startup config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip radius source-interface** *interface*	Configures the global source interface for all RADIUS server groups configured on the device. The source interface can be the management or the VLAN interface.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration information.
Step 5	(Optional) switch# **copy running-config startup config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the mgmt 0 interface as the global source interface for RADIUS server groups:

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

Allowing Users to Specify a RADIUS Server at Login

You can allow users to specify a RADIUS server at login.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server directed-request**
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server directed-request**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server directed-request**	Allows users to specify a RADIUS server to send the authentication request when logging in. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server directed-request**	Displays the directed request configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to allow users to select a RADIUS server when logging in to a network:

```
switch# configure terminal
switch(config)# radius-server directed-request
switch# exit
switch# copy running-config startup-config
```

Configuring RadSec

RadSec is a protocol for transporting RADIUS datagrams over TLS.

This procedure describes how to enable/disable the RadSec on a switch.

Before you begin

- Ensure that the client identity certificate and CA certificate of the server are installed on the switch.
- Ensure that the subject name in the server certificate is matching with the server host name/IP address that is configured on the switch.
- Before configuring AAA authentication and accounting to use RadSec servers, use **test aaa group** command and ensure RadSec authentication is success.
- Configure TLS idle-timeout to maximum value on RadSec server to avoid frequent TLS sessions retries from switch.

SUMMARY STEPS

1. **configure terminal**
2. **radius-server secure tls**
3. **radius-server host t** {*ipv4-address* | *ipv6-address*| *hostname*} **key** {*key*} **auth-port 2083 acct-port 2083 authentication accounting**
4. **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **tls client-trustpoint** *trustpoint*
5. **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **tls idle-timeout** *value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters configuration mode.
Step 2	**radius-server secure tls** **Example:** `switch# radius-server secure tls`	Enables the RadSec at global level. **Note** This CLI will not change or affect the port numbers that is used for RadSec.
Step 3	**radius-server host t** {*ipv4-address* | *ipv6-address*| *hostname*} **key** {*key*} **auth-port 2083 acct-port 2083 authentication accounting** **Example:** `switch# radius-server host 10.105.222.161 key radsec auth-port 2083 acct-port 2083 authentication accounting`	Configures the RadSec server with shared secret key along with the authentication and accounting ports. **Note** For server, the default RadSec port for authentication and accounting is "2083" and the key is “radsec”. For switch, there is no default configuration for RadSec port and key, please add this configuration explicitly as defined on server.
Step 4	**radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **tls client-trustpoint** *trustpoint* **Example:** `switch# radius-server host 10.105.222.161 tls client-trustpoint rad1`	Configures the TLS client trustpoint where the client identity certificate is installed.
Step 5	**radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **tls idle-timeout** *value* **Example:** `switch# radius-server host 10.105.222.161 tls idle-timeout 80`	Configures the TLS idle-timeout. The default value is 600 seconds. **Note** If there are no transactions from the RadSec client, server can close the connection based on its timeout value. The TLS idle-timeout on the client is not supported in this release. Client does not close connections on its own.

Note When remote user logs-in, you can notice delay in login for approximately 20 seconds i.e when TLS session establishment is happening for the first time between switch and RadSec server, Once TLS sessions are up no delay will be seen for consecutive remote log-ins.

Note When a RadSec client is facing certificate related issues such as no certificate or invalid certificates are being exchanged with the server, you may experience delay in `show run` commands.

About RadSec with DTLS

From Cisco NX-OS Release 10.4(1)F, RadSec with DTLS protocol is introduced. This protocol is for transporting RADIUS datagrams over a secure channel using UDP.

RadSec with DTLS provides secure communication between RADIUS peers at the transport layer. This protocol helps secure RADIUS packets transfer through different administrative domains and suspicious, and unsafe networks.

Configuring RadSec with DTLS

Before you begin

- Ensure that you create client identity certificate with subject and alternative name same as the IP address/DNS hostname of the switch. Install the client identity certificate on the switch using a trustpoint.
- Enusre that the server certificate of ISE server used for DTLS/RADIUS is installed on the switch.
- Make sure that the CA certificate used to sign client identity certificate is installed in trusted certificate store of ISE server.
- Ensure that the subject name in the server certificate is same as the server hostname/IP address that is configured on the switch.
- Before configuring AAA authentication and accounting groups to use RadSec servers, check with test aaa group command and ensure that the RadSec authentication is successful.
- You must enable RadSec with DTLS protocol at the switch level.
- Configuring combination of RadSec servers to use different transports protocols such as DTLS and TLS is not supported. You can configure one protocol at an instant.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**radius-server secure dtls** **Example:** `switch(config)# radius-server secure dtls`	Enables the RadSec with DTLS protocol on the switch.
Step 3	**radius-server host** {*ipv4-address* | *ipv6-address*| *hostname*} **key** {*radius/dtls*} **auth-port 2083 acct-port 2083 authentication accounting**	Configures the RadSec server with shared secret key along with the authentication and accounting ports.

	Command or Action	Purpose
	Example: `switch(config)# radius-server host 10.105.222.161 key radius/dtls auth-port 2083 acct-port 2083 authentication accounting`	**Note** The default destination DTLS port for authentication and accounting is **UDP/2083**. There is no default server key for DTLS as per RFC. Ensure that you add this configuration explicitly as defined on server. The ISE server must be pre-set with the "radius/dtls" key at that instant. Check and add the key on the Nexus switch while configuring DTLS with an ISE server.
Step 4	**radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **dtls client-trustpoint** *trustpoint* **Example:** `switch(config)# radius-server host 10.105.222.161 dtls client-trustpoint rad1`	Configures the DTLS client-trustpoint parameter with a trustpoint where the switch identity certificate is installed. The *rad1* is a trustpoint on the switch which must have the client identity certificate.
Step 5	**radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **dtls idle-timeout** *value* **Example:** `switch# radius-server host 10.105.222.161 dtls idle-timeout 80`	Configures the DTLS idle-timeout. The default value is 600 seconds. **Note** If there are no transactions from the RadSec client, server can close the connection as per defined timeout value. The DTLS idle-timeout on the client is not supported in this release. Client does not close connections on its own.

Note When remote user logs-in, you can notice delay in login for approximately 20 seconds i.e when TLS session establishment is happening for the first time between switch and RadSec server, Once TLS sessions are up no delay will be seen for consecutive remote log-ins.

Note When a RadSec client is facing certificate related issues such as no certificate or invalid certificates are being exchanged with the server, we may experience delay in `show run` commands.

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 switch 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 Nexus device waits for responses from RADIUS servers before declaring a timeout failure.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server retransmit** *count*
3. switch(config)# **radius-server timeout** *seconds*

4. switch(config)# **exit**
5. (Optional) switch# **show radius-server**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server retransmit** *count*	Specifies the retransmission count for all RADIUS servers. The default retransmission count is 1 and the range is from 0 to 5.
Step 3	switch(config)# **radius-server timeout** *seconds*	Specifies the transmission timeout interval for RADIUS servers. The default timeout interval is 5 seconds and the range is from 1 to 60 seconds.
Step 4	switch(config)# **exit**	Exits global configuration mode.
Step 5	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to set the retry count to 3 and the transmission timeout interval to 5 seconds for RADIUS servers:

```
switch# configure terminal
switch(config)# radius-server retransmit 3
switch(config)# radius-server timeout 5
switch(config)# exit
switch# copy running-config startup-config
```

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

By default, a Cisco Nexus switch 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. You can also set a timeout interval that the switch waits for responses from RADIUS servers before declaring a timeout failure.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **retransmit** *count*
3. switch(config)#**radius-server host** {*ipv4-address* | *host-name*} **timeout** *seconds*
4. switch(config)# **exit**
5. (Optional) switch# **show radius-server**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-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 RADIUS server overrides the count specified for all RADIUS servers.
Step 3	switch(config)#**radius-server host** {*ipv4-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 RADIUS server overrides the interval value specified for all RADIUS servers.
Step 4	switch(config)# **exit**	Exits global configuration mode.
Step 5	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to set the RADIUS transmission retry count to 3 and the timeout interval to 10 seconds on RADIUS host server server1:

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

Configuring Accounting and Authentication Attributes for RADIUS Servers

You can specify that a RADIUS server is to be used only for accounting purposes or only for authentication purposes. By default, RADIUS servers are used for both accounting and authentication. You can also specify the destination UDP port numbers where RADIUS accounting and authentication messages should be sent.

SUMMARY STEPS

1. switch# **configure terminal**
2. (Optional) switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **acct-port** *udp-port*
3. (Optional) switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **accounting**
4. (Optional) switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **auth-port** *udp-port*
5. (Optional) switch(config)# **radius-server host** {*ipv4-address* |*host-name*} **authentication**

6. switch(config)# **exit**
7. (Optional) switch(config)# **show radius-server**
8. switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **acct-port** *udp-port*	Specifies a UDP port to use for RADIUS accounting messages. The default UDP port is 1812. The range is from 0 to 65535.
Step 3	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **accounting**	Specifies that the specified RADIUS server is to be used only for accounting purposes. The default is both accounting and authentication.
Step 4	(Optional) switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **auth-port** *udp-port*	Specifies a UDP port to use for RADIUS authentication messages. The default UDP port is 1812. The range is from 0 to 65535.
Step 5	(Optional) switch(config)# **radius-server host** {*ipv4-address* |*host-name*} **authentication**	Specifies that the specified RADIUS server only be used for authentication purposes. The default is both accounting and authentication.
Step 6	switch(config)# **exit**	Exits configuration mode.
Step 7	(Optional) switch(config)# **show radius-server**	Displays the RADIUS server configuration.
Step 8	switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure accounting and authentication attributes for a RADIUS server:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1 acct-port 2004
switch(config)# radius-server host 10.10.1.1 accounting
switch(config)# radius-server host 10.10.2.2 auth-port 2005
switch(config)# radius-server host 10.10.2.2 authentication
switch # exit
switch # copy running-config startup-config
switch #
```

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 switch sends out a test packet. You can configure this option to test servers periodically.

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

The test idle timer specifies the interval during which a RADIUS server receives no requests before the switch sends out a test packet.

The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, the switch does not perform periodic RADIUS server monitoring.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server host** {*ipv4-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}
3. switch(config)# **radius-server deadtime** *minutes*
4. switch(config)# **exit**
5. (Optional) switch# **show radius-server**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server host** {*ipv4-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 default value for the idle timer is 0 minutes. The valid range is from 0 to 1440 minutes. **Note** For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.
Step 3	switch(config)# **radius-server deadtime** *minutes*	Specifies the number of minutes before the switch checks a RADIUS server that was previously unresponsive. The default value is 0 minutes. The valid range is 1 to 1440 minutes.
Step 4	switch(config)# **exit**	Exits configuration mode.
Step 5	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure RADIUS server host 10.10.1.1 with a username (user1) and password (Ur2Gd2BH) and with an idle timer of 3 minutes and a deadtime of 5 minutes:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1 test username user1 password Ur2Gd2BH idle-time
 3
switch(config)# radius-server deadtime 5
switch(config)# exit
switch# 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 Nexus device waits after declaring a RADIUS server is 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.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **radius-server deadtime**
3. switch(config)# **exit**
4. (Optional) switch# **show radius-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **radius-server deadtime**	Configures the dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show radius-server**	Displays the RADIUS server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure a deadtime of 5 minutes for a radius server:

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

Manually Monitoring RADIUS Servers or Groups

SUMMARY STEPS

1. switch# **test aaa server radius** {*ipv4-address* | *server-name*} [**vrf** *vrf-name*] *username password* **test aaa server radius** {*ipv4-address* | *server-name*} [**vrf** *vrf-name*] *username password*
2. switch# **test aaa group** *group-name username password*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **test aaa server radius** {*ipv4-address* | *server-name*} [**vrf** *vrf-name*] *username password* **test aaa server radius** {*ipv4-address* | *server-name*} [**vrf** *vrf-name*] *username password*	Sends a test message to a RADIUS server to confirm availability.
Step 2	switch# **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 send a test message to the RADIUS server and server group to confirm availability:

```
switch# test aaa server radius 10.10.1.1 user 1 Ur2Gd2BH
switch# test aaa group RadGroup user2 As3He3CI
```

Displaying RADIUS Server Statistics

SUMMARY STEPS

1. switch# **show radius-server statistics** {*hostname* | *ipv4-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show radius-server statistics** {*hostname* | *ipv4-address*}	Displays the RADIUS statistics.

Clearing RADIUS Server Statistics

You can display the statistics that the Cisco NX-OS device maintains for RADIUS server activity.

Before you begin

Configure RADIUS servers on the Cisco NX-OS device.

SUMMARY STEPS

1. (Optional) switch# **show radius-server statistics** {*hostname* | *ipv4-address*}
2. switch# **clear radius-server statistics** {*hostname* | *ipv4-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) switch# **show radius-server statistics** {*hostname* | *ipv4-address*}	Displays the RADIUS server statistics on the Cisco NX-OS device.
Step 2	switch# **clear radius-server statistics** {*hostname* | *ipv4-address*}	Clears the RADIUS server statistics.

Configuration Examples for RADIUS

The following example shows how to configure RADIUS:

```
switch# configure terminal
switch(config)# radius-server key 7 "ToIkLhPpG"
switch(config)# radius-server host 10.10.1.1 key 7 "ShMoMhTl" authentication accounting
switch(config)# aaa group server radius RadServer
switch(config-radius)# server 10.10.1.1
switch(config-radius)# exit
switch(config-radius)# use-vrf management
```

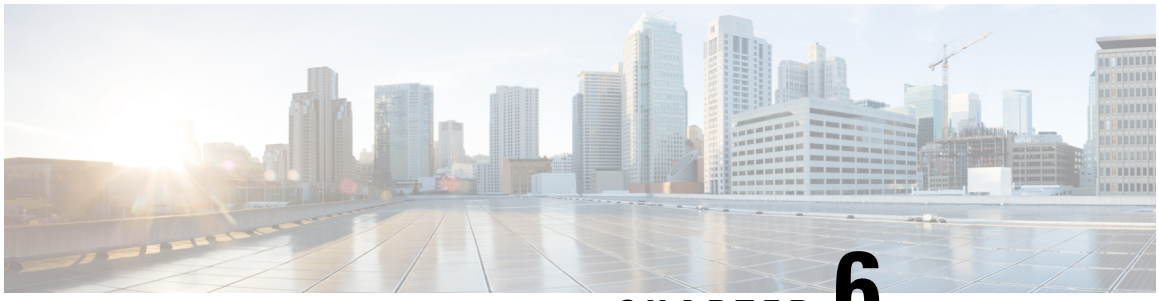
Default Settings for RADIUS

The following table lists the default settings for RADIUS parameters.

Table 8: Default RADIUS Parameters

Parameters	Default
Server roles	Authentication and accounting
Dead timer interval	0 minutes
Retransmission count	1

Parameters	Default
Retransmission timer interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test



CHAPTER 6

Configuring TACACS+

This chapter contains the following sections:

About Configuring TACACS+

Information About Configuring TACACS+

The Terminal Access Controller Access Control System Plus (TACACS+) security protocol provides centralized validation of users attempting to gain access to a Cisco Nexus device. TACACS+ services are maintained in a database on a TACACS+ daemon typically running on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your Cisco Nexus device are available.

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 is associated with 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 Nexus device provides centralized authentication using the TACACS+ protocol.

Beginning with Cisco NX-OS release 10.4(3)F, SSH based authorization of X.509 certificates using TACACS+ server can be done using the **aaa authorization ssh-certificate default group** command on the Cisco Nexus switches. For configuration details, see Configuring X.509 Certificate-Based SSH Authorization Using TACACS Server, on page 87.

TACACS+ Advantages

TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the Cisco Nexus device 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 switch and the AAA server to ensure higher data confidentiality. The RADIUS protocol only encrypts passwords.

User Login with TACACS+

When a user attempts a Password Authentication Protocol (PAP) login to a Cisco Nexus device using TACACS+, the following actions occur:

1. When the Cisco Nexus device 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 user's mother's maiden name.

2. The Cisco Nexus device receives one of the following responses from the TACACS+ daemon:
   - ACCEPT—User authentication succeeds and service begins. If the Cisco Nexus device requires user authorization, authorization begins.
   - REJECT—User authentication failed. The TACACS+ daemon either denies further access to the user or prompts the user to retry the login sequence.
   - ERROR—An error occurred at some time during authentication dither at the daemon or in the network connection between the daemon and the Cisco Nexus device. If the Cisco Nexus deviceh receives an ERROR response, the switch tries to use an alternative method for authenticating the user.

   The user also undergoes an additional authorization phase, if authorization has been enabled on the Cisco Nexus device. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.
3. If TACACS+ authorization is required, the Cisco Nexus device 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), access list, and user timeouts

Default TACACS+ Server Encryption Type and Preshared Key

You must configure the TACACS+ that is preshared key to authenticate the switch to the TACACS+ server. A preshared key is a secret text string shared between the Cisco Nexus device 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 preshared secret key for all TACACS+ server configurations on the Cisco Nexus deviceh to use.

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

Command Authorization Support for TACACS+ Servers

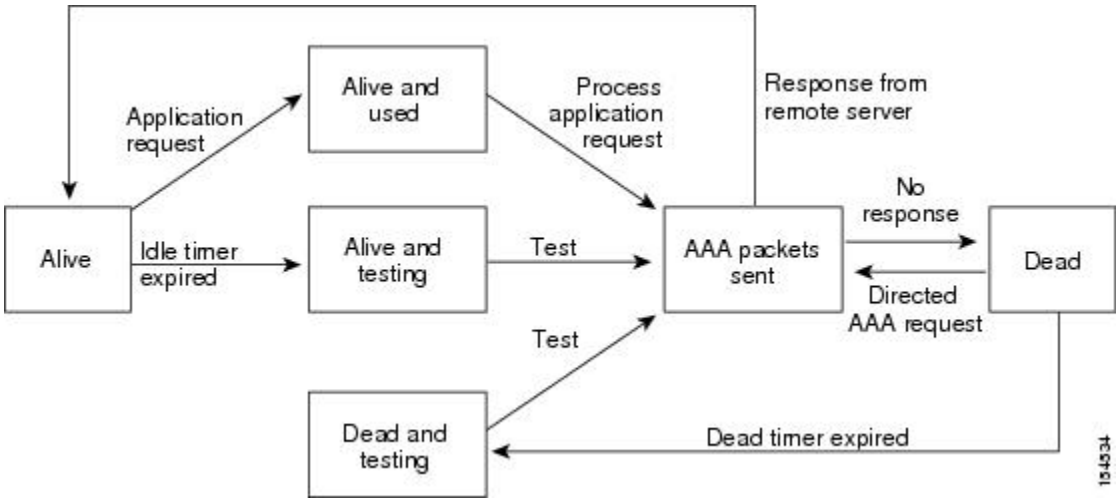
By default, command authorization is done against a local database in the Cisco NX-OS software when an authenticated user enters a command at the command-line interface (CLI). You can also verify authorized commands for authenticated users using TACACS+.

TACACS+ Server Monitoring

An unresponsive TACACS+ server can delay the processing of AAA requests. A Cisco Nexus device can periodically monitor an TACACS+ server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco Nexus device marks unresponsive TACACS+ servers as dead and does not send AAA requests to any dead TACACS+ servers. The Cisco Nexus device periodically monitors dead TACACS+ servers and brings them to the alive state once they are responding. This process verifies that a TACACS+ server is in a working state before real AAA requests are sent to the server. Whenever an TACACS+ server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the Cisco Nexus device displays an error message that a failure is taking place before it can impact performance.

The following figure shows the different TACACS+ server states:

Figure 5: TACACS+ Server States



Note The monitoring interval for alive servers and dead servers are different and can be configured by the user. The TACACS+ server monitoring is performed by sending a test authentication request to the TACACS+ server.

Prerequisites for TACACS+

TACACS+ has the following prerequisites:

- You must obtain the IPv4 addresses or hostnames for the TACACS+ servers.
- You must obtain the preshared keys from the TACACS+ servers, if any.
- Ensure that the Cisco Nexus device is configured as a TACACS+ client of the AAA servers.

Guidelines and Limitations for TACACS+

TACACS+ has the following configuration guidelines and limitations:

- You can configure a maximum of 64 TACACS+ servers on the Cisco Nexus device.
- You may get the following error message sporadically after you have configured a TACACS+ server host followed by the AAA configuration to actually use the host:

  ```
  %TACACS-3-TACACS_ERROR_MESSAGE: All servers failed to respond
  ```

  This is a known issue and there is no workaround. If the remote authentication works properly without any TACACS server connectivity issue, you can ignore the message and continue with your further configuration.
- Beginning with Cisco NX-OS release 10.4(3)F, SSH based authorization of X.509 certificates using TACACS+ server can be done using the aaa authorization ssh-certificate default group command on the Cisco Nexus switches.

Configuring TACACS+

TACACS+ Server Configuration Process

This section describes how to configure TACACS+ servers.

SUMMARY STEPS

1. Enable TACACS+.
2. Establish the TACACS+ server connections to the Cisco Nexus device.
3. Configure the preshared secret keys for the TACACS+ servers.
4. If needed, configure TACACS+ server groups with subsets of the TACACS+ servers for AAA authentication methods.
5. If needed, configure any of the following optional parameters:
6. If needed, configure periodic TACACS+ server monitoring.

DETAILED STEPS

Step 1 Enable TACACS+.

Step 2 Establish the TACACS+ server connections to the Cisco Nexus device.

Step 3 Configure the preshared secret keys for the TACACS+ servers.

Step 4 If needed, configure TACACS+ server groups with subsets of the TACACS+ servers for AAA authentication methods.

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

- Dead-time interval
- Allow TACACS+ server specification at login
- Timeout interval
- TCP port

Step 6 If needed, configure periodic TACACS+ server monitoring.

Enabling TACACS+

Although by default, the TACACS+ feature is disabled on the Cisco Nexus device. You can enable the TACACS+ feature to access the configuration and verification commands for authentication.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature tacacs+**
3. switch(config)# **exit**
4. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature tacacs+**	Enables TACACS+.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring TACACS+ Server Hosts

To access a remote TACACS+ server, you must configure the IPv4 address or the hostname for the TACACS+ server on the Cisco Nexus device. All TACACS+ server hosts are added to the default TACACS+ server group.You can configure up to 64 TACACS+ servers.

If a preshared key is not configured for a configured TACACS+ server, a warning message is issued if a global key is not configured. If a TACACS+ server key is not configured, the global key (if configured) is used for that server.

Before you configure TACACS+ server hosts, you should do the following:

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*}
3. switch(config)# **exit**
4. (Optional) switch# **show tacacs-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*}	Specifies the IPv4 address or hostname for a TACACS+ server.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

You can delete a TACACS+ server host from a server group.

Configuring TACACS+ Global Preshared Keys

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

Before you configure preshared keys, you should do the following:

- Enable TACACS+.
- Obtain the preshared key values for the remote TACACS+ servers.

SUMMARY STEPS

1. switch# **configure terminal**
2. **tacacs-server key** [**0** | **6** | **7**] *key-value*
3. switch(config)# **exit**
4. (Optional) switch# **show tacacs-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**tacacs-server key** [**0** | **6** | **7**] *key-value*	Specifies a TACACS+ key for all TACACS+ server. You can specify that the *key-value* is in clear text format (**0**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The default format is clear text. The maximum length is 63 characters. By default, no preshared key is configured.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.

	Command or Action	Purpose
		Note The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure global preshared keys:

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

Configuring TACACS+ Server Preshared Keys

You can configure preshared keys for a TACACS+ server. A preshared key is a shared secret text string between the Cisco Nexus device and the TACACS+ server host.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *key-value*
3. switch(config)# **exit**
4. (Optional) switch# **show tacacs-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *key-value*	Specifies a preshared key for a specific TACACS+ server. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. This preshared key is used instead of the global preshared key.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.

	Command or Action	Purpose
		Note The preshared keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted preshared keys.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure the TACACS+ preshared keys:

```
switch# configure terminal
switch(config)# tacacs-server host 10.10.1.1 key 0 PlIjUhYg
switch(config)# exit
switch# show tacacs-server
switch# 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 belong to the TACACS+ protocol. The servers are tried 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.

Before you begin

You must use the **feature tacacs+** command to enable TACACS+ before you configure TACACS+.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **aaa group server tacacs+** *group-name*
3. switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *key-value*
4. (Optional) switch(config-tacacs+)# **deadtime** *minutes*
5. (Optional) switch(config-tacacs+)# **source-interface** *interface*
6. switch(config-tacacs+)# **exit**
7. (Optional) switch(config)# **show tacacs-server groups**
8. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **aaa group server tacacs+** *group-name*	Creates a TACACS+ server group and enters the TACACS+ server group configuration mode for that group.
Step 3	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **key** [**0** | **7**] *key-value*	Specifies a preshared key for a specific TACACS+ server. You can specify a clear text (**0**) or encrypted (**7**) preshared key. The default format is clear text. The maximum length is 63 characters. This preshared key is used instead of the global preshared key.
Step 4	(Optional) switch(config-tacacs+)# **deadtime** *minutes*	Configures the monitoring dead time. The default is 0 minutes. The range is from 0 through 1440. **Note** If the dead-time interval for a TACACS+ server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	(Optional) switch(config-tacacs+)# **source-interface** *interface*	Assigns a source interface for a specific TACACS+ server group. The supported interface types are management and VLAN. **Note** Use the **source-interface** command to override the global source interface assigned by the **ip tacacs source-interface** command.
Step 6	switch(config-tacacs+)# **exit**	Exits configuration mode.
Step 7	(Optional) switch(config)# **show tacacs-server groups**	Displays the TACACS+ server group configuration.
Step 8	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure a TACACS+ server group:

```
switch# configure terminal
switch(config)# aaa group server tacacs+ TacServer
switch(config-tacacs+)# server 10.10.2.2
switch(config-tacacs+)# deadtime 30
switch(config-tacacs+)# exit
switch(config)# show tacacs-server groups
switch(config)# 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. You can also configure a different source interface for a specific TACACS+ server group.

SUMMARY STEPS

1. **configure terminal**
2. **ip tacacs source-interface** *interface*
3. **exit**
4. (Optional) **show tacacs-server**
5. (Optional) **copy running-config startup config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip tacacs source-interface** *interface* **Example:** `switch(config)# ip tacacs source-interface mgmt 0`	Configures the global source interface for all TACACS+ server groups configured on the device. The source interface can be the management or the VLAN interface.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show tacacs-server** **Example:** `switch# show tacacs-server`	Displays the TACACS+ server configuration information.
Step 5	(Optional) **copy running-config startup config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Specifying a TACACS+ Server at Login

You can configure the switch to allow the user to specify which TACACS+ server to send the authenticate request by enabling the directed-request option. By default, a Cisco Nexus device forwards an authentication request based on the default AAA authentication method. If you enable this option, the user can log in as *username@hostname*, where *hostname* is the name of a configured RADIUS server.

Note User specified logins are only supported for Telnet sessions.

SUMMARY STEPS

1. switch# **configure terminal**

2. switch(config)# **tacacs-server directed-request**
3. switch(config)# **exit**
4. (Optional) switch# **show tacacs-server directed-request**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server directed-request**	Allows users to specify a TACACS+ server to send the authentication request when logging in. The default is disabled.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server directed-request**	Displays the TACACS+ directed request configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring Command Authorization on TACACS+ Servers

You can configure authorization for commands on TACACS+ servers. Command authorization disables user role-based authorization control (RBAC), including the default roles.

Before you begin

Enable TACACS+.

Configure TACACS host and server groups before configuring AAA command authorization.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authorization** {**commands** | **config-commands**} **default** [**group** *group-list* [**local**] | **local**]
3. **exit**
4. (Optional) **show aaa authorization** [**all**]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authorization** {**commands**	**config-commands**} **default** [**group** *group-list* [**local**]	**local**]	Configures the default authorization method for commands for all roles.

	Command or Action	Purpose
	Example: `switch(config)# aaa authorization commands default group TacGroup`	The **commands** keyword configures authorization sources for all EXEC commands, and the **config-commands** keyword configures authorization sources for all configuration commands. The default authorization for all commands is local authorization, which is the list of authorized commands for the user's assigned role. The *group-list* argument consists of a space-delimited list of TACACS+ server group names. Servers that belong to this group are contacted for command authorization. The **local** method uses the local role-based database for authorization. The **local** method is used only if all the configured server groups fail to respond and you have configured **local** as the fallback method. The default method is **local**. If you have not configured a fallback method after the TACACS+ server group method, authorization fails if all server groups fail to respond.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show aaa authorization** [**all**] **Example:** `switch(config)# show aaa authorization`	Displays the AAA authorization configuration. The **all** keyword displays the default values.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Testing Command Authorization on TACACS+ Servers

You can test the command authorization for a user on the TACACS+ servers.

Note You must send correct commands for authorization or the results might not be reliable.

Before you begin

Enable TACACS+.

Ensure that you have configured command authorization for the TACACS+ servers.

SUMMARY STEPS

1. **test aaa authorization command-type** {**commands** | **config-commands**} **user** *username* **command** *command-string*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**test aaa authorization command-type** {**commands** | **config-commands**} **user** *username* **command** *command-string*
	Example:
	`switch# test aaa authorization command-type commands`
	`user TestUser command reload`
		Tests a user's authorization for a command on the TACACS+ servers.
		The **commands** keyword specifies only EXEC commands and the **config-commands** keyword specifies only configuration commands.
		Note Put double quotes (") before and after the *command-string* argument if it contains spaces.

Enabling and Disabling Command Authorization Verification

You can enable and disable command authorization verificaiton on the command-line interface (CLI) for the default user session or for another username.

Note The commands do not execute when you enable authorization verification.

SUMMARY STEPS

1. **terminal verify-only** [**username** *username*]
2. **terminal no verify-only** [**username** *username*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**terminal verify-only** [**username** *username*] **Example:** `switch# terminal verify-only`	Enables command authorization verification. After you enter this command, the Cisco NX-OS software indicates whether the commands you enter are authorized or not.
Step 2	**terminal no verify-only** [**username** *username*] **Example:** `switch# terminal no verify-only`	Disables command authorization verification.

Configuring Privilege Level Support for Authorization on TACACS+ Servers

You can configure privilege level support for authorization on TACACS+ servers.

Unlike Cisco IOS devices, which use privilege levels to determine authorization, Cisco NX-OS devices use role-based access control (RBAC). To enable both types of devices to be administered by the same TACACS+

servers, you can map the privilege levels configured on TACACS+ servers to user roles configured on Cisco NX-OS devices.

When a user authenticates with a TACACS+ server, the privilege level is obtained and used to form a local user role name of the format "priv-*n*," where *n* is the privilege level. The user assumes the permissions of this local role. Sixteen privilege levels, which map directly to corresponding user roles, are available. The following table shows the user role permissions that correspond to each privilege level.

Privilege Level	User Role Permissions
15	network-admin permissions
13 - 1	• Standalone role permissions, if the **feature privilege** command is disabled. • Same permissions as privilege level 0 with cumulative privileges for roles, if the **feature privilege** command is enabled.
0	Permission to execute **show** commands and **exec** commands (such as **ping**, **trace**, and **ssh**).

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **feature privilege**
3. [**no**] **enable secret** [**0** | **5**] *password* [**priv-lvl** *priv-lvl* | **all**]
4. [**no**] **username** *username* **priv-lvl** *n*
5. (Optional) **show privilege**
6. (Optional) **copy running-config startup-config**
7. **exit**
8. **enable** *level*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature privilege** **Example:** `switch(config)# feature privilege`	Enables or disables the cumulative privilege of roles. Users can see the **enable** command only if this feature is enabled. The default is disabled.
Step 3	[**no**] **enable secret** [**0**	**5**] *password* [**priv-lvl** *priv-lvl*	**all**] **Example:** `switch(config)# enable secret 5 def456 priv-lvl 15`	Enables or disables a secret password for a specific privilege level. Users are prompted to enter the correct password upon each privilege level escalation. The default is disabled. You can enter **0** to specify that the password is in clear text or **5** to specify that the password is in encrypted format.

	Command or Action	Purpose
		The *password* argument can be up to 64 alphanumeric characters. The *priv-lvl* argument is from 1 to 15. **Note** To enable the secret password, you must have enabled the cumulative privilege of roles by entering the **feature privilege** command.
Step 4	[**no**] **username** *username* **priv-lvl** *n* **Example:** `switch(config)# username user2 priv-lvl 15`	Enables or disables a user to use privilege levels for authorization. The default is disabled. The **priv-lvl** keyword specifies the privilege level to which the user is assigned. There is no default privilege level. Privilege levels 0 to 15 (priv-lvl 0 to priv-lvl 15) map to user roles priv-0 to priv-15.
Step 5	(Optional) **show privilege** **Example:** `switch(config)# show privilege`	Displays the username, current privilege level, and status of cumulative privilege support.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Step 7	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 8	**enable** *level* **Example:** `switch# enable 15`	Enables a user to move to a higher privilege level. This command prompts for the secret password. The *level* argument specifies the privilege level to which the user is granted access. The only available level is 15.

Configuring X.509 Certificate-Based SSH Authorization Using TACACS Server

Beginning with Cisco NX-OS release 10.4(3)F, you can configure SSH-based authorization of x509v3-certificates using a TACAC+ server on the Cisco Nexus switches.

To configure X.509 certificate-based SSH-authorization using a TACAC+ server, follow these steps:

SUMMARY STEPS

1. **configure terminal**
2. **aaa authorization ssh-certificate default group** *tacacs-group-name*
3. **exit**
4. (Optional) **show aaa authorization** [**all**]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authorization ssh-certificate default group** *tacacs-group-name* **Example:** `switch(config)# aaa authorization ssh-certificate default group tac`	Configures the default AAA authorization method for the TACAC+ servers. The **ssh-certificate** keyword configures TACACS or local authorization with certificate authentication. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role. **Note** • Ensure that the *tacacs-group-name* is configured under the TACACS-server configuration using the **aaa group server tacacs+** *tacacs-group-name* command. • To support SSH certificate-based authentication, configure a crypto trustpoint and install the root CA. For more details, see the Configuring PKI, on page 131 section.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show aaa authorization** [**all**] **Example:** `switch# show aaa authorization`	Displays the AAA authorization configuration. The **all** keyword displays the default values.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Permitting or Denying Commands for Users of Privilege Roles

As a network administrator, you can modify the privilege roles to permit users to execute specific commands or to prevent users from running those commands.

You must follow these guidelines when changing the rules of privilege roles:

- You cannot modify the priv-14 and priv-15 roles.
- You can add deny rules only to the priv-0 role.

- These commands are always permitted for the priv-0 role: **configure**, **copy**, **dir**, **enable**, **ping**, **show**, **ssh**, **telnet**, **terminal**, **traceroute**, **end**, and **exit**.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **role name priv-***n*
3. **rule** *number* {**deny** | **permit**} **command** *command-string*
4. **exit**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **role name priv-***n*
	Example:
	`switch(config)# role name priv-5`
	`switch(config-role)#`
		Enables or disables a privilege role and enters role configuration mode. The *n* argument specifies the privilege level and is a number between 0 and 13.

Step 3	**rule** *number* {**deny**	**permit**} **command** *command-string* **Example:** `switch(config-role)# rule 2 permit command pwd`	Configures a command rule for users of privilege roles. These rules permit or deny users to execute specific commands. You can configure up to 256 rules for each role. The rule number determines the order in which the rules are applied. Rules are applied in descending order. For example, if a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1. The *command-string* argument can contain spaces. **Note** Repeat this command for 256 rules.

Step 4	**exit**
	Example:
	`switch(config-role)# exit`
	`switch(config)#`
		Exits role configuration mode.

Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Global TACACS+ Timeout Interval

You can set a global timeout interval that the Cisco Nexus device waits for responses from all TACACS+ servers before declaring a timeout failure. The timeout interval determines how long the switch waits for responses from TACACS+ servers before declaring a timeout failure.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server timeout** *seconds*
3. switch(config)# **exit**
4. (Optional) switch# **show tacacs-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

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

Configuring the Timeout Interval for a Server

You can set a timeout interval that the Cisco Nexus device waits for responses from a TACACS+ server before declaring a timeout failure. The timeout interval determines how long the switch waits for responses from a TACACS+ server before declaring a timeout failure.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **timeout** *seconds*
3. switch(config)# **exit**
4. (Optional) switch# **show tacacs-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **timeout** *seconds*	Specifies the timeout interval for a specific server. The default is the global value.

	Command or Action	Purpose
		Note The timeout interval value specified for a TACACS+ server overrides the global timeout interval value specified for all TACACS+ servers.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring TCP Ports

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **port** *tcp-port*
3. switch(config)# **exit**
4. (Optional) switch# **show tacacs-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **port** *tcp-port*	Specifies the UDP port to use for TACACS+ accounting messages.The default TCP port is 49. The range is from 1 to 65535.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure TCP ports:

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

Configuring Periodic TACACS+ Server Monitoring

You can monitor the availability of TACACS+ servers. These 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 Nexus device 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, we recommend that you use a username that is not the same as an existing username in the TACACS+ database.

The test idle timer specifies the interval in which a TACACS+ server receives no requests before the Cisco Nexus device sends out a test packet.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic TACACS+ server monitoring is not performed.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server host** {*ipv4-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}
3. switch(config)# **tacacs-server dead-time** *minutes*
4. switch(config)# **exit**
5. (Optional) switch# **show tacacs-server**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **tacacs-server host** {*ipv4-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 default value for the idle timer is 0 minutes and the valid range is from 0 to 1440 minutes. **Note** For periodic TACACS+ server monitoring, the idle timer value must be greater than 0.
Step 3	switch(config)# **tacacs-server dead-time** *minutes*	Specifies the number minutes before the Cisco Nexus device checks a TACACS+ server that was previously unresponsive. The default value is 0 minutes and the valid range is 0 to 1440 minutes.
Step 4	switch(config)# **exit**	Exits configuration mode.
Step 5	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.

	Command or Action	Purpose
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure periodic TACACS+ server monitoring:

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

Configuring the Dead-Time Interval

You can configure the dead-time interval for all TACACS+ servers. The dead-time interval specifies the time that the Cisco Nexus device 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-time interval is 0 minutes, TACACS+ servers are not marked as dead even if they are not responding. You can configure the dead-time interval per group.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **tacacs-server deadtime** *minutes*
3. switch(config)# **exit**
4. (Optional) switch# **show tacacs-server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **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	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **show tacacs-server**	Displays the TACACS+ server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Manually Monitoring TACACS+ Servers or Groups

SUMMARY STEPS

1. switch# **test aaa server tacacs+** {*ipv4-address* | *host-name*} [**vrf** *vrf-name*] *username password*
2. switch# **test aaa group** *group-name username password*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **test aaa server tacacs+** {*ipv4-address* | *host-name*} [**vrf** *vrf-name*] *username password*	Sends a test message to a TACACS+ server to confirm availability.
Step 2	switch# **test aaa group** *group-name username password*	Sends a test message to a TACACS+ server group to confirm availability.

Example

The following example shows how to manually issue a test message:

```
switch# test aaa server tacacs+ 10.10.1.1 user1 Ur2Gd2BH
switch# test aaa group TacGroup user2 As3He3CI
```

Disabling TACACS+

You can disable TACACS+.

Caution When you disable TACACS+, all related configurations are automatically discarded.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no feature tacacs+**
3. switch(config)# **exit**
4. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature tacacs+**	Disables TACACS+.
Step 3	switch(config)# **exit**	Exits configuration mode.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Displaying TACACS+ Statistics

To display the statistics, the switch maintains for TACACS+ activity, perform this task:

SUMMARY STEPS

1. switch# **show tacacs-server statistics** {*hostname* | *ipv4-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show tacacs-server statistics** {*hostname* | *ipv4-address*}	Displays the TACACS+ statistics.

Example

For detailed information about the fields in the output from this command, see the *Command Reference* for your Nexus switch.

Verifying the TACACS+ Configuration

To display TACACS+ configuration information, perform one of the following tasks:

SUMMARY STEPS

1. switch# **show tacacs+** {**status** | **pending** | **pending-diff**}
2. switch# **show running-config tacacs** [**all**]
3. switch# **show startup-config tacacs**
4. switch# **show tacacs-serve** [*host-name* | *ipv4-address*] [**directed-request** | **groups** | **sorted** | **statistics**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show tacacs+** {**status** | **pending** | **pending-diff**}	Displays the TACACS+ Cisco Fabric Services distribution status and other details.
Step 2	switch# **show running-config tacacs** [**all**]	Displays the TACACS+ configuration in the running configuration.
Step 3	switch# **show startup-config tacacs**	Displays the TACACS+ configuration in the startup configuration.
Step 4	switch# **show tacacs-serve** [*host-name* | *ipv4-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured TACACS+ server parameters.

Configuration Examples for TACACS+

This example shows how to configure TACACS+:

```
switch# configure terminal
switch(config)# feature tacacs+
switch(config)# tacacs-server key 7 "ToIkLhPpG"
switch(config)# tacacs-server host 10.10.2.2 key 7 "ShMoMhTl"
switch(config)# aaa group server tacacs+ TacServer
switch(config-tacacs+)# server 10.10.2.2
switch(config-tacacs+)# use-vrf management
```

This example shows how to enable tacacs+ and how to configure the tacacs+ server preshared keys to specify remote AAA servers to authenticate server group TacServer1:

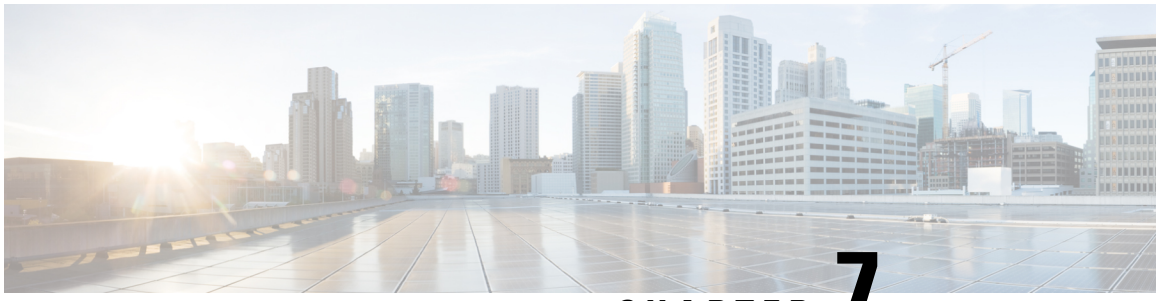
```
switch# configure terminal
switch(config)# feature tacacs+
switch(config)# tacacs-server key 7 "ikvhw10"
switch(config)# tacacs-server host 1.1.1.1
switch(config)# tacacs-server host 1.1.1.2

switch(config)# aaa group server tacacs+ TacServer1
switch(config-tacacs+)#   server 1.1.1.1
switch(config-tacacs+)#   server 1.1.1.2
```

Default Settings for TACACS+

The following table lists the default settings for TACACS+ parameters.

Table 9: Default TACACS+ Parameters

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



CHAPTER 7

Configuring LDAP

This chapter describes how to configure the Lightweight Directory Access Protocol (LDAP) on Cisco NX-OS devices and includes the following sections:

About LDAP

The Lightweight Directory Access Protocol (LDAP) provides centralized validation of users attempting to gain access to a Cisco NX-OS device. LDAP services are maintained in a database on an LDAP daemon running typically on a UNIX or Windows NT workstation. You must have access to and must configure an LDAP server before the configured LDAP features on your Cisco NX-OS device are available.

LDAP provides for separate authentication and authorization facilities. LDAP allows for a single access control server (the LDAP daemon) to provide each service authentication and authorization 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 LDAP client/server protocol uses TCP (port 389) for transport requirements. Cisco NX-OS devices provide centralized authentication using the LDAP protocol.

LDAP Authentication and Authorization

Clients establish a TCP connection and authentication session with an LDAP server through a simple bind (username and password). As part of the authorization process, the LDAP server searches its database to retrieve the user profile and other information.

You can configure the bind operation to first bind and then search, where authentication is performed first and authorization next, or to first search and then bind. The default method is to first search and then bind.

The advantage of searching first and binding later is that the distinguished name (DN) received in the search result can be used as the user DN during binding rather than forming a DN by prepending the username (cn attribute) with the baseDN. This method is especially helpful when the user DN is different from the username plus the baseDN. For the user bind, the bindDN is constructed as baseDN + append-with-baseDN, where append-with-baseDN has a default value of cn=$userid.

Note As an alternative to the bind method, you can establish LDAP authentication using the compare method, which compares the attribute values of a user entry at the server. For example, the user password attribute can be compared for authentication. The default password attribute type is userPassword.

LDAP Operation for User Login

When a user attempts a Password Authentication Protocol (PAP) login to a Cisco NX-OS device using LDAP, the following actions occur:

1. When the Cisco NX-OS device establishes a connection, it contacts the LDAP daemon to obtain the username and password.
2. The Cisco NX-OS device eventually receives one of the following responses from the LDAP daemon:
   - ACCEPT—User authentication succeeds and service begins. If the Cisco NX-OS device requires user authorization, authorization begins.
   - REJECT—User authentication fails. The LDAP daemon either denies further access to the user or prompts the user to retry the login sequence.
   - ERROR—An error occurs at some time during authentication either at the daemon or in the network connection between the daemon and the Cisco NX-OS device. If the Cisco NX-OS device receives an ERROR response, the Cisco NX-OS device tries to use an alternative method for authenticating the user.

   After authentication, the user also undergoes an additional authorization phase if authorization has been enabled on the Cisco NX-OS device. Users must first successfully complete LDAP authentication before proceeding to LDAP authorization.
3. If LDAP authorization is required, the Cisco NX-OS device again contacts the LDAP 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

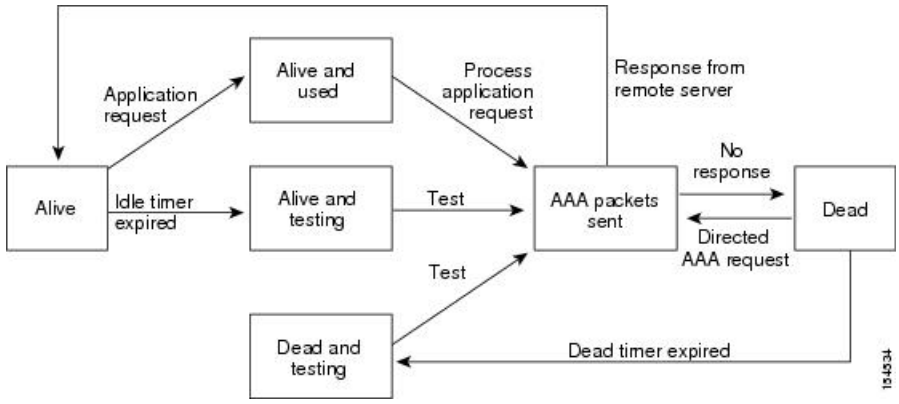
Note LDAP allows an arbitrary conversation between the daemon and the user until the daemon receives enough information to authenticate the user. This action is usually done by prompting for a username and password combination but may include prompts for other items.

Note In LDAP, authorization can occur before authentication.

LDAP Server Monitoring

An unresponsive LDAP server can delay the processing of AAA requests. A Cisco NX-OS device can periodically monitor an LDAP server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco NX-OS device marks unresponsive LDAP servers as dead and does not send AAA requests to any dead LDAP servers. A Cisco NX-OS device periodically monitors dead LDAP servers and brings them to the alive state once they are responding. This process verifies that an LDAP server is in a working state before real AAA requests are sent its way. Whenever an LDAP server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated, and the Cisco NX-OS device displays an error message that a failure is taking place before it can impact performance. The following figure shows the server states for LDAP server monitoring.

Figure 6: LDAP Server States



Note The monitoring interval for alive servers and dead servers is different and can be configured by the user. The LDAP server monitoring is performed by sending a test authentication request to the LDAP server.

Vendor-Specific Attributes for LDAP

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

Cisco VSA Format for LDAP

The Cisco LDAP 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, the separator is an = (equal sign) for mandatory attributes, and an * (asterisk) indicates optional attributes. When you use LDAP servers for authentication on a Cisco NX-OS device, LDAP directs the LDAP server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs. The following VSA protocol option is supported by the Cisco NX-OS software:

- Shell—Protocol used in access-accept packets to provide user profile information.

The Cisco NX-OS software supports the following attribute:

- 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.

Virtualization Support for LDAP

The Cisco NX-OS device uses virtual routing and forwarding instances (VRFs) to access the LDAP servers. For more information on VRFs, see the *Cisco Nexus 3548 Switch NX-OS Unicast Routing Configuration Guide*

Prerequisites for LDAP

LDAP has the following prerequisites:

- Obtain the IPv4 or IPv6 addresses or hostnames for the LDAP servers.
- Ensure that the Cisco NX-OS device is configured as an LDAP client of the AAA servers.

Guidelines and Limitations for LDAP

LDAP has the following guidelines and limitations:

- You can configure a maximum of 64 LDAP servers on the Cisco NX-OS device.
- Cisco NX-OS supports only LDAP version 3.
- Cisco NX-OS supports only these LDAP servers:
  - OpenLDAP
  - Microsoft Active Directory
- LDAP over Secure Sockets Layer (SSL) supports only SSL version 3 and Transport Layer Security (TLS) version 1.2.
- Beginning with Cisco NX-OS Release 10.4(3)F, LDAP over Secure Sockets Layer (SSL) supports TLS version 1.3 and 1.2 on Cisco Nexus switches. TLS v1.1 is deprecated.
- For LDAP over SSL, the LDAP client configuration must include the hostname as a subject in the LDAP server certificate.
- If you have a user account configured on the local Cisco NX-OS device that has the same name as a remote user account on a AAA server, the Cisco NX-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.

- Beginning with Cisco NX-OS Release 10.4(1)F, LDAP is supported on the Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.

Default Settings for LDAP

This table lists the default settings for LDAP parameters.

Parameters	Default
LDAP	Disabled
LDAP authentication method	First search and then bind
LDAP authentication mechanism	Plain
Dead-time interval	0 minutes
Timeout interval	5 seconds
Idle timer interval	60 minutes
Periodic server monitoring username	test
Periodic server monitoring password	Cisco

Configuring LDAP

This section describes how to configure LDAP on a Cisco NX-OS device.

LDAP Server Configuration Process

You can configure LDAP servers by following this configuration process.

1. Enable LDAP.
2. Establish the LDAP server connections to the Cisco NX-OS device.
3. If needed, configure LDAP server groups with subsets of the LDAP servers for AAA authentication methods.
4. (Optional) Configure the TCP port.
5. (Optional) Configure the default AAA authorization method for the LDAP server.
6. (Optional) Configure an LDAP search map.
7. (Optional) If needed, configure periodic LDAP server monitoring.

Related Topics

Enabling or Disabling LDAP, on page 102
Configuring LDAP Server Hosts, on page 103
Configuring the RootDN for an LDAP Server, on page 104
Configuring LDAP Server Groups, on page 105
Configuring TCP Ports, on page 109

Configuring LDAP Search Maps, on page 110
Configuring Periodic LDAP Server Monitoring, on page 111

Enabling or Disabling LDAP

By default, the LDAP feature is disabled on the Cisco NX-OS device. You must explicitly enable the LDAP feature to access the configuration and verification commands for authentication.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **feature ldap**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Required: [**no**] **feature ldap** **Example:** `switch(config)# feature ldap`	Enables LDAP. Use the **no** form of this command to disable LDAP. **Note** When you disable LDAP, all related configurations are automatically discarded.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

LDAP Server Configuration Process, on page 101
Configuring LDAP Server Hosts, on page 103
Configuring the RootDN for an LDAP Server, on page 104
Configuring LDAP Server Groups, on page 105
Configuring the Global LDAP Timeout Interval, on page 107
Configuring the Timeout Interval for an LDAP Server, on page 108
Configuring TCP Ports, on page 109
Configuring LDAP Search Maps, on page 110
Configuring Periodic LDAP Server Monitoring, on page 111
Configuring the LDAP Dead-Time Interval, on page 112
Configuring AAA Authorization on LDAP Servers, on page 113

Configuring LDAP Server Hosts

To access a remote LDAP server, you must configure the IP address or the hostname for the LDAP server on the Cisco NX-OS device. You can configure up to 64 LDAP servers.

Note By default, when you configure an LDAP server IP address or hostname on the Cisco NX-OS device, the LDAP server is added to the default LDAP server group. You can also add the LDAP server to another LDAP server group.

Before you begin

Enable LDAP.

Obtain the IPv4 or IPv6 addresses or the hostnames for the remote LDAP servers.

If you plan to enable the Secure Sockets Layer (SSL) protocol, make sure that the LDAP server certificate is manually configured on the Cisco NX-OS device.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *host-name*} [**enable-ssl**] [**referral-disable**]
3. (Optional) **show ldap-server**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *host-name*} [**enable-ssl**] [**referral-disable**] **Example:** `switch(config)# ldap-server host 10.10.2.2 enable-ssl`	Specifies the IPv4 or IPv6 address or hostname for an LDAP server. The **enable-ssl** keyword ensures the integrity and confidentiality of the transferred data by causing the LDAP client to establish an SSL session prior to sending the bind or search request. The **referral-disable** keyword disables the unwanted referral links.
Step 3	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server configuration.

	Command or Action	Purpose
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

LDAP Server Configuration Process, on page 101
Enabling or Disabling LDAP, on page 102
Configuring LDAP Server Groups, on page 105
Configuring the RootDN for an LDAP Server, on page 104
Configuring LDAP Server Groups, on page 105
Configuring Periodic LDAP Server Monitoring, on page 111
Monitoring LDAP Servers, on page 116
Clearing LDAP Server Statistics, on page 116

Configuring the RootDN for an LDAP Server

You can configure the root designated name (DN) for the LDAP server database. The rootDN is used to bind to the LDAP server to verify its state.

Before you begin

Enable LDAP.

Obtain the IPv4 or IPv6 addresses or the hostnames for the remote LDAP servers.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **rootDN** *root-name* [**password** *password* [**port** *tcp-port* [**timeout** *seconds*] | **timeout** *seconds*]]
3. (Optional) **show ldap-server**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **rootDN** *root-name* [**password** *password* [**port** *tcp-port* [**timeout** *seconds*] | **timeout** *seconds*]] **Example:**	Specifies the rootDN for the LDAP server database and the bind password for the root. Optionally specifies the TCP port to use for LDAP messages to the server. The range is from 1 to 65535, and the default TCP port is the global value or 389 if a global value is not

	Command or Action	Purpose
	`switch(config)# ldap-server host 10.10.1.1 rootDN cn=manager,dc=acme,dc=com password Ur2Gd2BH timeout 60`	configured. Also specifies the timeout interval for the server. The range is from 1 to 60 seconds, and the default timeout is the global value or 5 seconds if a global value is not configured.
Step 3	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

LDAP Server Configuration Process, on page 101
Enabling or Disabling LDAP, on page 102
Configuring LDAP Server Hosts, on page 103

Configuring LDAP Server Groups

You can specify one or more remote AAA servers to authenticate users using server groups. All members of a group must be configured to use LDAP. The servers are tried in the same order in which you configure them.

You can configure these server groups at any time, but they take effect only when you apply them to an AAA service.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **aaa group server ldap** *group-name*
3. [**no**] **server** {*ipv4-address* | *ipv6-address* | *host-name*}
4. (Optional) [**no**] **authentication** {**bind-first** [**append-with-baseDN** *DNstring*] | **compare** [**password-attribute** *password*]}
5. (Optional) [**no**] **enable user-server-group**
6. (Optional) [**no**] **enable Cert-DN-match**
7. (Optional) [**no**] **use-vrf** *vrf-name*
8. **exit**
9. (Optional) **show ldap-server groups**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **aaa group server ldap** *group-name*
	Example:
	`switch(config)# aaa group server ldap LDAPServer1`
	`switch(config-ldap)#`
		Creates an LDAP server group and enters the LDAP server group configuration mode for that group.

Step 3	[**no**] **server** {*ipv4-address* | *ipv6-address* | *host-name*} **Example:** `switch(config-ldap)# server 10.10.2.2`	Configures the LDAP server as a member of the LDAP server group. If the specified LDAP server is not found, configure it using the **ldap-server host** command and retry this command.
Step 4	(Optional) [**no**] **authentication** {**bind-first** [**append-with-baseDN** *DNstring*] | **compare** [**password-attribute** *password*]} **Example:** `switch(config-ldap)# authentication compare password-attribute TyuL8r`	Performs LDAP authentication using the bind or compare method. The default LDAP authentication method is the bind method using first search and then bind.
Step 5	(Optional) [**no**] **enable user-server-group** **Example:** `switch(config-ldap)# enable user-server-group`	Enables group validation. The group name should be configured in the LDAP server. Users can login through public-key authentication only if the username is listed as a member of this configured group in the LDAP server.
Step 6	(Optional) [**no**] **enable Cert-DN-match** **Example:** `switch(config-ldap)# enable Cert-DN-match`	Enables users to login only if the user profile lists the subject-DN of the user certificate as authorized for login.
Step 7	(Optional) [**no**] **use-vrf** *vrf-name* **Example:** `switch(config-ldap)# use-vrf vrf1`	Specifies the VRF to use to contact the servers in the server group.

Step 8	**exit**
	Example:
	`switch(config-ldap)# exit`
	`switch(config)#`
		Exits LDAP server group configuration mode.

Step 9	(Optional) **show ldap-server groups** **Example:** `switch(config)# show ldap-server groups`	Displays the LDAP server group configuration.

	Command or Action	Purpose
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

LDAP Server Configuration Process, on page 101
Configuring LDAP Server Hosts, on page 103
Enabling or Disabling LDAP, on page 102
Configuring LDAP Server Hosts, on page 103

Configuring the Global LDAP Timeout Interval

You can set a global timeout interval that determines how long the Cisco NX-OS device waits for responses from all LDAP servers before declaring a timeout failure.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server timeout** *seconds*
3. (Optional) **show ldap-server**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server timeout** *seconds* **Example:** `switch(config)# ldap-server timeout 10`	Specifies the timeout interval for LDAP servers. The default timeout interval is 5 seconds. The range is from 1 to 60 seconds.
Step 3	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling or Disabling LDAP, on page 102
Configuring the Timeout Interval for an LDAP Server, on page 108
Configuring the Timeout Interval for an LDAP Server, on page 108

Configuring the Timeout Interval for an LDAP Server

You can set a timeout interval that determines how long the Cisco NX-OS device waits for responses from an LDAP server before declaring a timeout failure.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **timeout** *seconds*
3. (Optional) **show ldap-server**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **timeout** *seconds* **Example:** `switch(config)# ldap-server host server1 timeout 10`	Specifies the timeout interval for a specific server. The default is the global value. **Note** The timeout interval value specified for an LDAP server overrides the global timeout interval value specified for all LDAP servers.
Step 3	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring the Global LDAP Timeout Interval, on page 107
Enabling or Disabling LDAP, on page 102
Configuring the Global LDAP Timeout Interval, on page 107

Configuring TCP Ports

You can configure another TCP port for the LDAP servers if there are conflicts with another application. By default, Cisco NX-OS devices use port 389 for all LDAP requests.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **port** *tcp-port* [**timeout** *seconds*]
3. (Optional) **show ldap-server**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **port** *tcp-port* [**timeout** *seconds*] **Example:** `switch(config)# ldap-server host 10.10.1.1 port 200 timeout 5`	Specifies the TCP port to use for LDAP messages to the server. The default TCP port is 389. The range is from 1 to 65535. Optionally specifies the timeout interval for the server. The range is from 1 to 60 seconds, and the default timeout is the global value or 5 seconds if a global value is not configured. **Note** The timeout interval value specified for an LDAP server overrides the global timeout interval value specified for all LDAP servers.
Step 3	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

LDAP Server Configuration Process, on page 101
Enabling or Disabling LDAP, on page 102

Configuring LDAP Search Maps

You can configure LDAP search maps to send a search query to the LDAP server. The server searches its database for data meeting the criteria specified in the search map.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. **ldap search-map** *map-name*
3. (Optional) [**userprofile** | **trustedCert** | **CRLLookup** | **user-certdn-match** | **user-pubkey-match** | **user-switch-bind**] **attribute-name** *attribute-name* **search-filter** *filter* **base-DN** *base-DN-name*
4. (Optional) **exit**
5. (Optional) **show ldap-search-map**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**ldap search-map** *map-name*
	Example:
	`switch(config)# ldap search-map map1`
	`switch(config-ldap-search-map)#`
		Configures an LDAP search map.

Step 3	(Optional) [**userprofile** | **trustedCert** | **CRLLookup** | **user-certdn-match** | **user-pubkey-match** | **user-switch-bind**] **attribute-name** *attribute-name* **search-filter** *filter* **base-DN** *base-DN-name* **Example:** `switch(config-ldap-search-map)# userprofile attribute-name att-name search-filter (&(objectClass=inetOrgPerson)(cn=$userid)) base-DN dc=acme,dc=com`	Configures the attribute name, search filter, and base-DN for the user profile, trusted certificate, CRL, certificate DN match, public key match, or user-switchgroup lookup search operation. These values are used to send a search query to the LDAP server. The *attribute-name* argument is the name of the attribute in the LDAP server that contains the Nexus role definition.

Step 4	(Optional) **exit**
	Example:
	`switch(config-ldap-search-map)# exit`
	`switch(config)#`
		Exits LDAP search map configuration mode.

Step 5	(Optional) **show ldap-search-map** **Example:**	Displays the configured LDAP search maps.

	Command or Action	Purpose
	`switch(config)# show ldap-search-map`	
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

LDAP Server Configuration Process, on page 101
Enabling or Disabling LDAP, on page 102

Configuring Periodic LDAP Server Monitoring

You can monitor the availability of LDAP servers. The configuration parameters include the username and password to use for the server, the rootDN to bind to the server to verify its state, and an idle timer. The idle timer specifies the interval in which an LDAP server receives no requests before the Cisco NX-OS device 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, we recommend that you use a username that is not the same as an existing username in the LDAP database.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **test rootDN** *root-name* [**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]]
3. [**no**] **ldap-server deadtime** *minutes*
4. (Optional) **show ldap-server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Required: [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **test rootDN** *root-name* [**idle-time**	Specifies the parameters for server monitoring. The default username is test, and the default password is Cisco. The

	Command or Action	Purpose
	minutes | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]] **Example:** `switch(config)# ldap-server host 10.10.1.1 test rootDN root1 username user1 password Ur2Gd2BH idle-time 3`	default value for the idle timer is 60 minutes, and the valid range is from 1 to 1440 minutes. **Note** We recommend that the user not be an existing user in the LDAP server database.
Step 3	[**no**] **ldap-server deadtime** *minutes* **Example:** `switch(config)# ldap-server deadtime 5`	Specifies the number of minutes before the Cisco NX-OS device checks an LDAP server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 1 to 60 minutes.
Step 4	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

LDAP Server Configuration Process, on page 101
Enabling or Disabling LDAP, on page 102
Configuring LDAP Server Hosts, on page 103

Configuring the LDAP Dead-Time Interval

You can configure the dead-time interval for all LDAP servers. The dead-time interval specifies the time that the Cisco NX-OS device waits, after declaring that an LDAP server is dead, before sending out a test packet to determine if the server is now alive.

Note When the dead-time interval is 0 minutes, LDAP servers are not marked as dead even if they are not responding. You can configure the dead-time interval per group.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server deadtime** *minutes*
3. (Optional) **show ldap-server**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server deadtime** *minutes* **Example:** `switch(config)# ldap-server deadtime 5`	Configures the global dead-time interval. The default value is 0 minutes. The range is from 1 to 60 minutes.
Step 3	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling or Disabling LDAP, on page 102

Configuring AAA Authorization on LDAP Servers

You can configure the default AAA authorization method for LDAP servers.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authorization** {**ssh-certificate** | **ssh-publickey**} **default** {**group** *group-list* | **local**}
3. (Optional) **show aaa authorization** [**all**]
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**aaa authorization** {**ssh-certificate** | **ssh-publickey**} **default** {**group** *group-list* | **local**} **Example:** `switch(config)# aaa authorization ssh-certificate default group LDAPServer1 LDAPServer2`	Configures the default AAA authorization method for the LDAP servers. The **ssh-certificate** keyword configures LDAP or local authorization with certificate authentication, and the **ssh-publickey** keyword configures LDAP or local authorization with the SSH public key. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role. The *group-list* argument consists of a space-delimited list of LDAP server group names. Servers that belong to this group are contacted for AAA authorization. The **local** method uses the local database for authorization.
Step 3	(Optional) **show aaa authorization** [**all**] **Example:** `switch(config)# show aaa authorization`	Displays the AAA authorization configuration. The **all** keyword displays the default values.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling or Disabling LDAP, on page 102

Configuring LDAP SSH Public Key Authorization

The AAA authorization is performed through LDAP servers with the public key of the user which is saved in the user entry of the LDAP server.

Before configuring LDAP SSH public key authorization, ensure that the following are taken care of:

- Save the public key of the user as a user attribute in the LDAP server.
- Sign-in using the private key from the SSH client.

Note The private key that is presented during SSH sign-in is verified with the public key which is saved in the LDAP server.

The following example shows the sample LDAP client configuration.

In the following example, the public key of the user is saved in the LDAP server under the attribute mentioned in **user-pubkey-match** configuration, ie, **sshPublicKeys** attribute in the below case:

```
ldap-server host fully qualified domain name.com rootDN
"CN=ucsadmin1,CN=Users,DC=PI-Sec-DT,DC=com"  password 7 password1
ldap search-map Map1
  userprofile attribute-name "description" search-filter "(cn=$userid)" base-DN
```

```
"DC=PI-Sec-DT,DC=com"
  user-pubkey-match attribute-name "sshPublicKeys" search-filter "(cn=$userid)" base-DN
"DC=PI-Sec-DT,DC=com"
aaa group server ldap ldap1
  server fully qualified domain name.com
  use-vrf management
  ldap-search-map Map1

aaa authorization ssh-publickey default group ldap1
```

In the following example, the SSH client private key of the user is used to sign in to the switch management IP address:

```
ssh ldapuser@10.0.0.1 -i ldap_pub_key_test
```

Configuring LDAP SSH Certificate Authorization

AAA authorization is performed through an LDAP server with a certificate and the DN of the certificate which is saved in the user attribute of the LDAP server.

During LDAP SSH certificate authorization, following things are taken care of:

- Validation of the user certificate presented through the SSH client using the CA certificate installed in the switch.
- As the **enable cert-dn-match** configuration is enabled by default, the cert-DN-match with the DN stored in the LDAP server to validate the certificate is taken care automatically.

The following example shows the sample LDAP client configurations.

- The following example shows how to save the certificate DN in an LDAP server under any specific attribute that is mentioned in the **user-certdn-match** configuration.

  The format is "x509v3-sign-rsa DN /DC=com, DC=PI-Sec-DT, CN=Users, CN=username1".

  ```
  ldap-server host fully qualified domain name.com rootDN
  "CN=ucsadmin1,CN=Users,DC=PI-Sec-DT,DC=com"  password 7 password1
  ldap search-map Map24
    userprofile attribute-name "description" search-filter "(cn=$userid)" base-DN
  "DC=PI-Sec-DT,DC=com"
    user-certdn-match attribute-name <attribute> search-filter "(cn=$userid)" base-DN
  "DC=PI-Sec-DT,DC=com"
  aaa group server ldap ldap24
    server fully qualified domain name.com
    enable Cert-DN-match
    use-vrf management
    ldap-search-map Map24

  aaa authorization ssh-certificate default group ldap24
  ```

- The following show command shows the details of the rootCA certificate installed on the box:

  ```
  switch# show crypto ca certificates
  Trustpoint: ldap
  CA certificate 0:
  subject=C = IN, ST = KAR, L = BGL, O = Cisco, OU = DCBG-Cert, CN = RootCA
  issuer=C = IN, ST = KAR, L = BGL, O = Cisco, OU = DCBG-Cert, CN = RootCA
  serial=82EE7603BF7E74A9
  notBefore=May 29 07:12:30 2023 GMT
  notAfter=May 26 07:12:30 2033 GMT
  SHA1 Fingerprint=D5:AE:75:8E:A1:4F:79:1E:80:3E:5E:67:C5:42:44:10:13:C6:F7:1D
  purposes: sslserver sslclient
  ```

```
n7700-DE#
```

- The following example shows how user sign-in is performed from the SSH client:
  - In the SSH client, the input certificate contains both private key and user certificate concatenated in a single file '<user>.crt'.
  - The rootCA.crt is the rootCA certificate file.
  - The IP Address is the switch management IP address.

```
ssh username1@10.0.0.1 -i username1.crt -vvv -oCACertificateFile=rootCA.crt
```

Monitoring LDAP Servers

You can monitor the statistics that the Cisco NX-OS device maintains for LDAP server activity.

Before you begin

Configure LDAP servers on the Cisco NX-OS device.

SUMMARY STEPS

1. **show ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch# show ldap-server statistics 10.10.1.1`	Displays the LDAP server statistics.

Related Topics

Configuring LDAP Server Hosts, on page 103
Clearing LDAP Server Statistics, on page 116
Clearing LDAP Server Statistics, on page 116

Clearing LDAP Server Statistics

You can display the statistics that the Cisco NX-OS device maintains for LDAP server activity.

Before you begin

Configure LDAP servers on the Cisco NX-OS device.

SUMMARY STEPS

1. (Optional) **show ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}
2. **clear ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **show ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch# show ldap-server statistics 10.10.1.1`	Displays the LDAP server statistics.
Step 2	**clear ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch# clear ldap-server statistics 10.10.1.1`	Clears the LDAP server statistics.

Related Topics

Monitoring LDAP Servers, on page 116
Configuring LDAP Server Hosts, on page 103
Monitoring LDAP Servers, on page 116

Verifying the LDAP Configuration

To display LDAP configuration information, perform one of the following tasks.

Command	Purpose
show running-config ldap [**all**]	Displays the LDAP configuration in the running configuration.
show startup-config ldap	Displays the LDAP configuration in the startup configuration.
show ldap-server	Displays LDAP configuration information.
show ldap-server groups	Displays LDAP server group configuration information.
show ldap-server statistics {*hostname* | *ipv4-address* | *ipv6-address*}	Displays LDAP statistics.
show ldap-search-map	Displays information about the configured LDAP attribute maps.

Configuration Examples for LDAP

The following example shows how to configure an LDAP server host and server group:

```
feature ldap
ldap-server host 10.10.2.2 enable-ssl
aaa group server ldap LdapServer
server 10.10.2.2
exit
show ldap-server
show ldap-server groups
```

The following example shows how to configure an LDAP search map:

```
ldap search-map s0
userprofile attribute-name att-name search-filter "
(&(objectClass=Person)(sAMAccountName=$userid))" base-DN dc=acme,dc=com
exit
show ldap-search-map
```

The following example shows how to configure AAA authorization with certificate authentication for an LDAP server:

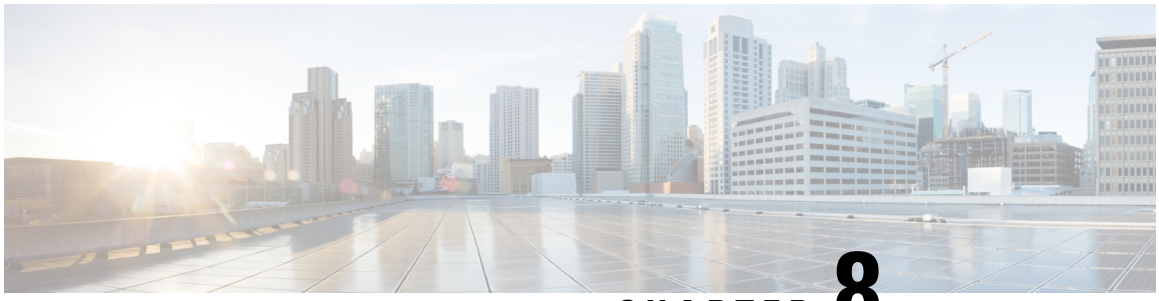
```
aaa authorization ssh-certificate default group LDAPServer1 LDAPServer2
exit
show aaa authorization
```

The following example shows how you can validate the authentication:

```
failing
test aaa group LdapServer user <user-password>
user has failed authentication

! working
test aaa group LdapServer user <user-password>
user has been authenticated
```

Where to Go Next

You can now configure AAA authentication methods to include the server groups.



CHAPTER 8

Configuring SSH and Telnet

This chapter contains the following sections:

Configuring SSH and Telnet

Information About SSH and Telnet

SSH Server

The Secure Shell Protocol (SSH) server feature enables a SSH client to make a secure, encrypted connection to a Cisco Nexus device. SSH uses strong encryption for authentication. The SSH server in the Cisco Nexus device switch interoperates with publicly and commercially available SSH clients.

The user authentication mechanisms supported for SSH are RADIUS, TACACS+, and the use of locally stored user names and passwords.

SSH Client

The SSH client feature is an application running over the SSH protocol to provide device authentication and encryption. The SSH client enables a switch to make a secure, encrypted connection to another Cisco Nexus device or to any other device running an SSH server. This connection provides an outbound connection that is encrypted. With authentication and encryption, the SSH client allows for a secure communication over an insecure network.

The SSH client in the Cisco Nexus device works with publicly and commercially available SSH servers.

SSH Server Keys

SSH requires server keys for secure communications to the Cisco Nexus device. You can use SSH keys for the following SSH options:

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

Be sure to have an SSH server key-pair with the appropriate version before enabling the SSH service. You can generate the SSH server key-pair according to the SSH client version used. The SSH service accepts three types of key-pairs for use by SSH version 2:

- The dsa option generates the DSA key-pair for the SSH version 2 protocol.
- The rsa option generates the RSA key-pair for the SSH version 2 protocol.

By default, the Cisco Nexus device generates an RSA key using 1024 bits.

SSH supports the following public key formats:

- OpenSSH
- IETF Secure Shell (SECSH)

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

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 system to the other. Telnet can accept either an IP address or a domain name as the remote system address.

The Telnet server is enabled by default on the Cisco Nexus device.

Guidelines and Limitations for SSH

SSH has the following configuration guidelines and limitations:

- The Cisco Nexus device supports only SSH version 2 (SSHv2).
- SSH public and private keys imported into user accounts that are remotely authenticated through a AAA protocol (such as RADIUS or TACACS+) for the purpose of SSH Passwordless File Copy will not persist when the Nexus device is reloaded unless a local user account with the same name as the remote user account is configured on the device before the SSH keys are imported.

Configuring SSH

Generating SSH Server Keys

You can generate an SSH server key based on your security requirements. The default SSH server key is an RSA key that is generated using 1024 bits.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}
3. switch(config)# **exit**
4. (Optional) switch# **show ssh key**

5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}	Generates the SSH server key. 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. Use the **force** keyword to replace an existing key.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show ssh key**	Displays the SSH server keys.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to generate an SSH server key:

```
switch# configure terminal
switch(config)# ssh key rsa 2048
switch(config)# exit
switch# show ssh key
switch# copy running-config startup-config
```

Specifying the SSH Public Keys for User Accounts

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

- Open SSH format
- IETF SECSH format
- Public Key Certificate in PEM format

Specifying the SSH Public Keys in Open SSH Format

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **username** *username* **sshkey** *ssh-key*
3. switch(config)# **exit**
4. (Optional) switch# **show user-account**

5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **username** *username* **sshkey** *ssh-key*	Configures the SSH public key in SSH format.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show user-account**	Displays the user account configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to specify an SSH public key in open SSH format:

```
switch# configure terminal
switch(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYz
CfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZcTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4Z
XIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5Ninn0McNinn0Mc=
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

Note The **username** command in the example above is a single line that has been broken for legibility.

Specifying the SSH Public Keys in IETF SECSH Format

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

SUMMARY STEPS

1. switch# **copy** *server-file* **bootflash:** *filename*
2. switch# **configure terminal**
3. switch(config)# **username** *username* **sshkey file** *filename*
4. switch(config)# **exit**
5. (Optional) switch# **show user-account**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **copy** *server-file* **bootflash:** *filename*	Downloads the file that contains the SSH key in IETF SECSH format from a server. The server can be FTP, SCP, SFTP, or TFTP.
Step 2	switch# **configure terminal**	Enters global configuration mode.
Step 3	switch(config)# **username** *username* **sshkey file** *filename*	Configures the SSH public key in SSH format.
Step 4	switch(config)# **exit**	Exits global configuration mode.
Step 5	(Optional) switch# **show user-account**	Displays the user account configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to specify the SSH public key in the IETF SECSH format:

```
switch#copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub
switch# configure terminal
switch(config)# username User1 sshkey file bootflash:secsh_file.pub
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

Specifying the SSH Public Keys in PEM-Formatted Public Key Certificate Form

You can specify the SSH public keys in PEM-formatted Public Key Certificate form for user accounts.

SUMMARY STEPS

1. switch# **copy** *server-file* **bootflash:** *filename*
2. switch# **configure terminal**
3. (Optional) switch# **show user-account**
4. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **copy** *server-file* **bootflash:** *filename*	Downloads the file that contains the SSH key in PEM-formatted Public Key Certificate form from a server. The server can be FTP, SCP, SFTP, or TFTP
Step 2	switch# **configure terminal**	Enters global configuration mode.
Step 3	(Optional) switch# **show user-account**	Displays the user account configuration.

	Command or Action	Purpose
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to specify the SSH public keys in PEM-formatted public key certificate form:

```
switch# copy tftp://10.10.1.1/cert.pem bootflash:cert.pem
switch# configure terminal
switch# show user-account
switch# copy running-config startup-config
```

Starting SSH Sessions to Remote Devices

You can start SSH sessions to connect to remote devices from your Cisco Nexus device.

SUMMARY STEPS

1. switch# **ssh** {*hostname* | *username@hostname*} [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **ssh** {*hostname* | *username@hostname*} [**vrf** *vrf-name*]	Creates an SSH session to a remote device. The *hostname* argument can be an IPv4 address or a hostname.

Clearing SSH Hosts

When you download a file from a server using SCP or SFTP, you establish a trusted SSH relationship with that server.

SUMMARY STEPS

1. switch# **clear ssh hosts**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **clear ssh hosts**	Clears the SSH host sessions.

Disabling the SSH Server

By default, the SSH server is enabled on the Cisco Nexus device.

SUMMARY STEPS

1. switch# **configure terminal**

2. switch(config)# **[no] feature ssh**
3. switch(config)# **exit**
4. (Optional) switch# **show ssh server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] feature ssh**	Enables/disables the SSH server. The default is enabled.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show ssh server**	Displays the SSH server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Deleting SSH Server Keys

You can delete SSH server keys after you disable the SSH server.

Note To reenable SSH, you must first generate an SSH server key.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no feature ssh**
3. switch(config)# **no ssh key** [**dsa** | **rsa**]
4. switch(config)# **exit**
5. (Optional) switch# **show ssh key**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature ssh**	Disables the SSH server.
Step 3	switch(config)# **no ssh key** [**dsa** | **rsa**]	Deletes the SSH server key. The default is to delete all the SSH keys.
Step 4	switch(config)# **exit**	Exits global configuration mode.
Step 5	(Optional) switch# **show ssh key**	Displays the SSH server configuration.

	Command or Action	Purpose
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Clearing SSH Sessions

You can clear SSH sessions from the Cisco Nexus device.

SUMMARY STEPS

1. switch# **show users**
2. switch# **clear line** *vty-line*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show users**	Displays user session information.
Step 2	switch# **clear line** *vty-line*	Clears a user SSH session.

Configuration Examples for SSH

The following example shows how to configure SSH:

SUMMARY STEPS

1. Generate an SSH server key.
2. Enable the SSH server.
3. Display the SSH server key.
4. Specify the SSH public key in Open SSH format.
5. Save the configuration.

DETAILED STEPS

Step 1 Generate an SSH server key.

```
switch(config)# ssh key rsa
generating rsa key(1024 bits).....
.
generated rsa key
```

Step 2 Enable the SSH server.

```
switch# configure terminal
switch(config)# feature ssh
```

Note This step should not be required because the SSH server is enabled by default.

Step 3 Display the SSH server key.

```
switch(config)# show ssh key
rsa Keys generated:Fri May  8 22:09:47 2009

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYzCfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZ/
cTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4ZXIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5/
Ninn0Mc=

bitcount:1024
fingerprint:
4b:4d:f6:b9:42:e9:d9:71:3c:bd:09:94:4a:93:ac:ca
***********************************
could not retrieve dsa key information
***********************************
```

Step 4 Specify the SSH public key in Open SSH format.

```
switch(config)# username User1 sshkey ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYz
CfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZcTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4Z
XIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5Ninn0McNinn0Mc=
```

Step 5 Save the configuration.

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

Configuring Telnet

Enabling the Telnet Server

By default, the Telnet server is enabled. You can disable the Telnet server on your Cisco Nexus device.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **[no] feature telnet**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] feature telnet**	Enables/disables the Telnet server. The default is enabled.

Reenabling the Telnet Server

If the Telnet server on your Cisco Nexus device has been disabled, you can reenable it.

SUMMARY STEPS

1. switch(config)# **[no] feature telnet**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config)# **[no] feature telnet**	Reenables the Telnet server.

Starting Telnet Sessions to Remote Devices

Before you start a Telnet session to connect to remote devices, you should do the following:

- Obtain the hostname for the remote device and, if needed, obtain the username on the remote device.
- Enable the Telnet server on the Cisco Nexus device.
- Enable the Telnet server on the remote device.

SUMMARY STEPS

1. switch# **telnet** *hostname*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **telnet** *hostname*	Creates a Telnet session to a remote device. The *hostname* argument can be an IPv4 address or a device name.

Example

The following example shows how to start a Telnet session to connect to a remote device:

```
switch# telnet 10.10.1.1
Trying 10.10.1.1...
Connected to 10.10.1.1.
Escape character is '^]'.
switch login:
```

Clearing Telnet Sessions

You can clear Telnet sessions from the Cisco Nexus device.

SUMMARY STEPS

1. switch# **show users**
2. switch# **clear line** *vty-line*

DETAILED STEPS

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

Verifying the SSH and Telnet Configuration

To display the SSH configuration information, perform one of the following tasks:

Procedure

- switch# **show ssh key** [**dsa** | **rsa**]

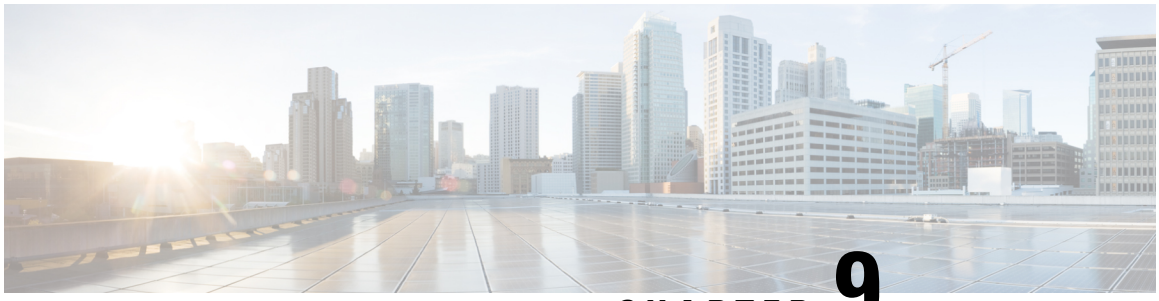
Command or Action	Purpose
switch# **show running-config security**[**all**]	Displays the SSH and user account configuration in the running configuration. The **all**keyword displays the default values for the SSH and user accounts.
switch# **show ssh server**	Displays the SSH server configuration.
switch# **show user-account**	Displays user account information

Default Settings for SSH

The following table lists the default settings for SSH parameters.

Table 10: Default SSH Parameters

Parameters	Default
SSH server	Enabled
SSH server key	RSA key generated with 1024 bits
RSA key bits for generation	1024
Telnet server	Enabled



CHAPTER 9

Configuring PKI

This chapter describes the Public Key Infrastructure (PKI) support on the Cisco NX-OS device. PKI allows the device to obtain and use digital certificates for secure communication in the network and provides manageability and scalability for Secure Shell (SSH).

This chapter includes the following sections:

Information About PKI

This section provides information about PKI.

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 only. However, the public key is known to everybody. Anything encrypted with one of the keys can be decrypted with the other. A signature is formed when data is encrypted with a sender's private key. The receiver verifies the signature by decrypting 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, Trust Points, 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 NX-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's 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 *trust point* and the CA itself is called a *trust point CA*. This information consists of a CA certificate (or certificate chain in case of a subordinate CA) and certificate revocation checking information.

The Cisco NX-OS device can also enroll with a trust point to obtain an identity certificate to associate with a key pair. This trust point is called an *identity CA*.

CA Certificate Hierarchy

For secure services, you typically have multiple trusted CAs. The CAs are usually installed in all the hosts as a bundle. The NX-OS PKI infrastructure does support importing certificate chain. However, with the current CLIs, one chain at a time can be installed. This procedure can be cumbersome when there are several CA chains to be installed. This requires a facility to download CA bundles that could include several intermediate and root CAs.

Importing CA Bundle

The **crypto CA trustpoint** command binds the CA certificates, CRLs, identity certificates and key pairs to a named label. All files corresponding to each of these entities are stored in the NX-OS certstore directory (/isan/etc/certstore) and tagged with the trustpoint label.

To access the CA certificates, an SSL app only needs to point to the standard NX-OS cert-store and specify that as the CA path during SSL initialization. It does not need to be aware of the trustpoint label under which CAs are installed.

If clients need to bind to an identity certificate, the trustpoint label needs to be used as the binding point.

The import pkcs command is enhanced to install the CA certificates under a trustpoint label. This can be further enhanced to install a CA bundle. The import command structure is modified to add pkcs7 option which is used for providing CA bundle file in pkcs7 format.

Once installed, there is no logical binding of all CA chains to a bundle.

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 trust point CA where the Cisco NX-OS device intends to enroll. The Cisco NX-OS device needs only one identity per CA, which consists of one key pair and one identity certificate per CA.

The Cisco NX-OS software allows you to generate RSA key pairs with a configurable key size (or modulus). The default key size is 512. 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 trust points, RSA key pairs, and identity certificates:

- A trust point corresponds to a specific CA that the Cisco NX-OS device trusts for peer certificate verification for any application (such as SSH).
- A Cisco NX-OS device can have many trust points and all applications on the device can trust a peer certificate issued by any of the trust point CAs.
- A trust point is not restricted to a specific application.
- A Cisco NX-OS device enrolls with the CA that corresponds to the trust point to obtain an identity certificate. You can enroll your device with multiple trust points which means that you can obtain a separate identity certificate from each trust point. The identity certificates are used by applications depending upon the 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 trust point, you must specify an RSA key pair to be certified. This key pair must be generated and associated to the trust point before generating the enrollment request. The association between the trust point, key pair, and identity certificate is valid until it is explicitly removed by deleting the certificate, key pair, or trust point.
- The subject name in the identity certificate is the fully qualified domain name for the Cisco NX-OS device.
- You can generate one or more RSA key pairs on a device and each can be associated to one or more trust points. But no more than one key pair can be associated to a trust point, which means only one identity certificate is allowed from a CA.
- If the Cisco NX-OS device obtains multiple identity certificates (each from a distinct CA), the certificate that an application selects to use in a security protocol exchange with a peer is application specific.
- You do not need to designate one or more trust points for an application. Any application can use any certificate issued by any trust point as long as the certificate purpose satisfies the application requirements.
- You do not need more than one identity certificate from a trust point or more than one key pair to be associated to a trust point. 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 trust point for the same CA, associate another key pair to it, and have it certified.

Multiple Trusted CA Support

The Cisco NX-OS device can trust multiple CAs by configuring multiple trust points 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 NX-OS device 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 like SSH. It occurs between the device that requests the certificate and the certificate authority.

The Cisco NX-OS device performs the following steps when performing the PKI enrollment process:

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

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

- Receives the issued certificate back from the CA, signed with the CA's private key.
- Writes the certificate into a nonvolatile storage area on the device (bootflash).

Manual Enrollment Using Cut-and-Paste

The Cisco NX-OS software supports certificate retrieval and enrollment using manual cut-and-paste. Cut-and-paste enrollment means that you must cut and paste the certificate requests and resulting certificates between the device and the CA.

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

- Create an enrollment certificate request, which the Cisco NX-OS device displays in base64-encoded text form.
- Cut and paste the encoded certificate request text in an e-mail or in a web form and send it to the CA.
- Receive the issued certificate (in base64-encoded text form) from the CA in an e-mail or in a web browser download.
- Cut and paste the issued certificate to the device using the certificate import facility.

Multiple RSA Key Pair and Identity CA Support

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

The multiple RSA key-pair feature allows the device 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 device can generate multiple RSA key pairs and associate each key pair with a distinct trust point. Thereafter, when enrolling with a trust point, the associated key pair is used to construct the certificate request.

Peer Certificate Verification

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

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

For revocation checking, the Cisco NX-OS software supports the certificate revocation list (CRL). A trust point CA can use this method to verify that the peer certificate has not been revoked.

Certificate Revocation Checking

The Cisco NX-OS software can check the revocation status of CA certificates. The applications can use the revocation checking mechanisms in the order that you specify. The choices are CRL, none, or a combination of these methods.

CRL Support

The CAs maintain certificate revocation lists (CRLs) to provide information about certificates revoked prior to their expiration dates. The CAs publish the CRLs in a repository and provide the download public URL in all issued certificates. A client verifying a peer's certificate can obtain the latest CRL from the issuing CA and use it to determine if the certificate has been revoked. A client can cache the CRLs of some or all of its trusted CAs locally and use them later if necessary until the CRLs expire.

The Cisco NX-OS software allows the manual configuration of predownloaded CRLs for the trust points, and then caches them in the device bootflash (cert-store). During the verification of a peer certificate, the Cisco NX-OS software checks the CRL from the issuing CA only if the CRL has already been cached locally and the revocation checking is configured to use the CRL. Otherwise, the Cisco NX-OS software does not perform CRL checking and considers the certificate to be not revoked unless you have configured other revocation checking methods.

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 trust point 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).

Guidelines and Limitations for PKI

PKI has the following configuration guidelines and limitations:

- The maximum number of key pairs you can configure on a Cisco NX-OS device is 16.
- The maximum number of trust points you can declare on a Cisco NX-OS device is 16.
- The maximum number of identify certificates that you can configure on a Cisco NX-OS device are 16.
- The maximum number of certificates in a CA certificate chain is 10.
- The maximum number of trust points you can authenticate to a specific CA is 10.
- Configuration rollbacks do not support the PKI configuration.
- Beginning with Cisco NX-OS Release 10.3(3)F, Elliptic Curve Cyptography (ECC) key pair support is provided to generate and import the certificate on Cisco Nexus switches.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings for PKI

This table lists the default settings for PKI parameters.

Table 11: Default PKI Parameters

Parameters	Default
Trust point	None
RSA key pair	None
RSA key-pair label	Device FQDN
RSA key-pair modulus	512
RSA key-pair exportable	Enabled
Revocation check method	CRL

Configuring CAs and Digital Certificates

This section describes the tasks that you must perform to allow CAs and digital certificates on your Cisco NX-OS device to interoperate.

Configuring the Hostname and IP Domain Name

You must configure the hostname and IP domain name of the device if you have not yet configured them because the Cisco NX-OS software uses the fully qualified domain name (FQDN) of the device as the subject in the identity certificate. Also, the Cisco NX-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.

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

SUMMARY STEPS

1. **configure terminal**
2. **hostname** *hostname*
3. **ip domain-name** *name* [**use-vrf** *vrf-name*]
4. **exit**
5. (Optional) **show hosts**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**hostname** *hostname* **Example:** `switch(config)# hostname DeviceA`	Configures the hostname of the device.
Step 3	**ip domain-name** *name* [**use-vrf** *vrf-name*] **Example:** `DeviceA(config)# ip domain-name example.com`	Configures the IP domain name of the device. If you do not specify a VRF name, the command uses the default VRF.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show hosts** **Example:** `switch# show hosts`	Displays the IP domain name.
Step 6	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch# copy running-config startup-config`	

Generating an RSA Key Pair

You can generate an RSA key pairs to sign and/or encrypt and decrypt the security payload during security protocol exchanges for applications. You must generate the RSA key pair before you can obtain a certificate for your device.

Beginning Cisco NX-OS Release 9.3(3), you must explicitly generate RSA key pairs before you associate the Cisco NX-OS device with a trust point CA. Prior to Cisco NX-OS Releases 9.3(3), if unavailable, the RSA key pairs would be auto generated.

SUMMARY STEPS

1. **configure terminal**
2. **crypto key generate rsa** [**label** *label-string*] [**exportable**] [**modulus** *size*]
3. **exit**
4. (Optional) **show crypto key mypubkey rsa**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto key generate rsa** [**label** *label-string*] [**exportable**] [**modulus** *size*] **Example:** `switch(config)# crypto key generate rsa exportable`	Generates an RSA key pair. The maximum number of key pairs on a device is 16. The label string is alphanumeric, case sensitive, and has a maximum length of 64 characters. The default label string is the hostname and the FQDN separated by a period character (.). Valid modulus values are 512, 768, 1024, 1536, and 2048. The default modulus size is 512. **Note** The security policy on the Cisco NX-OS device and on the CA (where enrollment is planned) should 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.

	Command or Action	Purpose

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show crypto key mypubkey rsa** **Example:** `switch# show crypto key mypubkey rsa`	Displays the generated key.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Generating an ECC Key Pair

You can generate an ECC key pair to sign and/or encrypt and decrypt the security payload during security protocol exchanges for applications. You must generate the ECC key pair before you can obtain a certificate for your device. The ECC keys are stronger compared to RSA keys for a given length.

Beginning Cisco NX-OS Release 10.3(3)F, you can generate an ECC key pair to associate the Cisco NX-OS device with a trust point CA.

SUMMARY STEPS

1. **configure terminal**
2. **crypto key generate ecc** [**label** *ecc-key-label*] [**exportable**] [**modulus** *size*]
3. **no crypto key generate ecc** [**label** *ecc-key-label*]
4. **exit**
5. (Optional) **show crypto key mypubkey ecc**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto key generate ecc** [**label** *ecc-key-label*] [**exportable**] [**modulus** *size*] **Example:** `switch(config)# crypto key generate ecc exportable modulus 224`	Generates an RSA key pair. The maximum number of key pairs on a device is 16. The label string is alphanumeric, case sensitive, and has maximum length of 64 characters. The default label string is the hostname and the FQDN separated by a period character (.).

	Command or Action	Purpose
		Valid modulus values are 224, 384, and 521. The default modulus size is 224. **Note** The security policy on the Cisco NX-OS device and on the CA (where enrollment is planned) should 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	**no crypto key generate ecc** [**label** *ecc-key-label*] **Example:** `switch(config)# no crypto key generate ecc label label-name`	Deletes the ECC key.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show crypto key mypubkey ecc** **Example:** `switch# show crypto key mypubkey ecc`	Displays the generated ECC key.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating a Trust Point CA Association

You must associate the Cisco NX-OS device with a trust point CA.

Before you begin

Generate the RSA key pair.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca trustpoint** *name*
3. **enrollment terminal**
4. **rsakeypair** *label*
5. **exit**
6. (Optional) **show crypto ca trustpoints**

7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca trustpoint** *name*
	Example:
	`switch(config)# crypto ca trustpoint admin-ca`
	`switch(config-trustpoint)#`
		Declares a trust point CA that the device should trust and enters trust point configuration mode.
		Note The maximum number of trustpoints that can be configured is 50.

Step 3	**enrollment terminal** **Example:** `switch(config-trustpoint)# enrollment terminal`	Enables manual cut-and-paste certificate enrollment. The default is enabled. **Note** The Cisco NX-OS software supports only the manual cut-and-paste method for certificate enrollment.
Step 4	**rsakeypair** *label* **Example:** `switch(config-trustpoint)# rsakeypair SwitchA`	Specifies the label of the RSA key pair to associate to this trust point for enrollment. **Note** You can specify only one RSA key pair per CA.

Step 5	**exit**
	Example:
	`switch(config-trustpoint)# exit`
	`switch(config)#`
		Exits trust point configuration mode.

Step 6	(Optional) **show crypto ca trustpoints** **Example:** `switch(config)# show crypto ca trustpoints`	Displays trust point information.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Generating an RSA Key Pair, on page 138

Configuring Certificate Mapping Filters

You can configure mapping filters to validate the CA certificates that are used for authentication. The mapping filters are used to match the CA certificate against a username.

Cisco NX-OS supports the following certificate mapping filters:

- %username%—Substitutes the user's login name.
- %hostname%—Substitutes the peer hostname.

Before you begin

Configure a cert-store for certificate authentication.

SUMMARY STEPS

1. **configure terminal**
2. **crypto certificatemap mapname** *map-name*
3. **filter** [**subject-name** *subject-name* | **altname-email** *e-mail-ID* | **altname-upn** *user-principal-name*]
4. **exit**
5. (Optional) **crypto cert ssh-authorize** [**default** | *issuer-CAname*] [**map** *map-name1* [*map-name2*]]
6. (Optional) **show crypto certificatemap**
7. (Optional) **show crypto ssh-auth-map**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto certificatemap mapname** *map-name* **Example:** `switch(config)# crypto certificatemap mapname filtermap1`	Creates a new filter map.

Step 3	**filter** [**subject-name** *subject-name* | **altname-email** *e-mail-ID* | **altname-upn** *user-principal-name*]
	Example:
	`switch(config-certmap-filter)# filter altname-upn %username%@cisco.com`
		Configures one or more certificate mapping filters within the filter map. These certificate field attributes are supported in the filters: The validation passes if the certificate passes all of the filters configured in the map.
		• subject-name—The required subject name in the LDAP distinguished name (DN) string format. For example:
		`filter subject-name CN=%username%`
		or
		`filter subject-name /C=IN/ST=KA/L=BLR/O=CISCO/OU=ABC/CN=%username%`
		• altname-email—The e-mail address that must be present in the certificate as a subject alternative name. For example:
		`filter altname-email %username%@cisco.com`

	Command or Action	Purpose
		• altname-upn—The principal name that must be present in the certificate as a subject alternative name. For example: `filter altname-upn %username%@%hostname%` The validation passes if the certificate passes all of the filters configured in the map.

Step 4	**exit**
	Example:
	`switch(config-certmap-filter)# exit`
	`switch(config)#`
		Exits certificate mapping filter configuration mode.

Step 5	(Optional) **crypto cert ssh-authorize** [**default** | *issuer-CAname*] [**map** *map-name1* [*map-name2*]] **Example:** `switch(config)# crypto cert ssh-authorize default map filtermap1`	Configures a certificate mapping filter for the Secure Shell (SSH) protocol. You can use the default filter map for SSH authorization or specify the issuer of the CA certificate. If you do not use the default map, you can specify one or two filter maps for authorization. If you specify the issuer of the CA certificate, the certificate bound to the user account is validated as successful if it passes one of the configured maps.
Step 6	(Optional) **show crypto certificatemap** **Example:** `switch(config)# show crypto certificatemap`	Displays the certificate mapping filters.
Step 7	(Optional) **show crypto ssh-auth-map** **Example:** `switch(config)# show crypto ssh-auth-map`	Displays the mapping filters configured for SSH authentication.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Authenticating the CA

The configuration process of trusting a CA is complete only when the CA is authenticated to the Cisco NX-OS device. You must authenticate your Cisco NX-OS device 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 manually authenticated by contacting the CA administrator to compare the fingerprint of the CA certificate.

Note The CA that you are authenticating is not a self-signed CA when it is a subordinate CA to another CA, which itself may be a subordinate to yet another CA, and so on, finally ending in a self-signed CA. This type of CA certificate is called the *CA certificate chain* of the CA being authenticated. In this case, you must input the full list of the CA certificates of all the CAs in the certification chain during the CA authentication. The maximum number of certificates in a CA certificate chain is 10.

Before you begin

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca authenticate** *name*
3. **exit**
4. (Optional) **show crypto ca trustpoints**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca authenticate** *name*
	Example:
	`switch(config)# crypto ca authenticate admin-ca`
	`input (cut & paste) CA certificate (chain) in PEM format;`
	`end the input with a line containing only END OF INPUT :`
	`-----BEGIN CERTIFICATE-----`
	`MIIC4jCCAoygAwIBAgIQEWDSiay0GZRPSRIljK0ZejANBgkqhkiG9w0BAQUFADCB`
	`kDEgMB4GCSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklO`
	`MRIwEAYDVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UE`
	`ChMFQ2lzY28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBD`
	`QIAeFw0wNTA1MDMyMjQ2MzdaFw0wNzA1MDMyMjU1MTdaMIGQMSAwHgYJKoZIhvcN`
	`AQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UEBhMCSU4xEjAQBgNVBAgTCUth`
	`cm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4wDAYDVQQKEwVDaXNjbzEIMBEG`
	`A1UECxMKamV0c3RvcmFnZTESMBAGA1UEAxMJQXBhcm5hIENBMFwwDQYJKoZIhvcN`
	`AQEBBQADSwAwSAJBAMW/7b3+DXJPANBsIHHzluNccNM87ypyzwuoSNZXOMpeRXXI`
	`OzyBAgiXT2ASFuUOwQ1iDM8rO/41jf8RxvYKvysCAwEAAaOBvzCBvDALBgNVHQ8E`
	`BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUUyjyRoMbrCNMRU2OyRhQ`
	`GgsWbHEwawYDVR0fBGQwYjAuoCygKoYoaHR0cDovL3NzZS0wOC9DZXJ0RW5yb2xs`
	`L0FwYXJuYSUyMENBLmNybDAwoC6gLIYqZmlsZTovL1xcc3NlLTA4XENlcnRFbnJv`
	`bGxcQXBhcm5hJTIwQ0EuY3JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEB`
	`BQUAA0EAHv6UQ+8nE399Tww+KaGr0g0NIJaqNgLh0AFcT0rEyuyt/WYGPzksF9Ea`
	`NBG7E0oN66zex0EOEfG1Vs6mXp1//w==`
		Prompts you to cut and paste the certificate of the CA. Use the same name that you used when declaring the CA.
		The maximum number of trust points that you can authenticate to a specific CA is 10.
		Note For subordinate CA authentication, the Cisco NX-OS software 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.

	Command or Action	Purpose

	`-----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`	

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show crypto ca trustpoints** **Example:** `switch# show crypto ca trustpoints`	Displays the trust point CA information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Creating a Trust Point CA Association, on page 140

Configuring Certificate Revocation Checking Methods

During security exchanges with a client (for example, an SSH user), the Cisco NX-OS device performs the certificate verification of the peer certificate sent by the client. The verification process may involve certificate revocation status checking.

You can configure the device to check the CRL downloaded from the CA. Downloading the CRL and checking locally does not generate traffic in your network. However, certificates can be revoked between downloads and your device would not be aware of the revocation.

Before you begin

Authenticate the CA.

Ensure that you have configured the CRL if you want to use CRL checking.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca trustpoint** *name*
3. **revocation-check** {**crl** [**none**] | **none**}
4. **exit**
5. (Optional) **show crypto ca trustpoints**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca trustpoint** *name*
	Example:
	`switch(config)# crypto ca trustpoint admin-ca`
	`switch(config-trustpoint)#`
		Specifies a trust point CA and enters trust point configuration mode.

Step 3	**revocation-check** {**crl** [**none**] | **none**} **Example:** `switch(config-trustpoint)# revocation-check none`	Configures the certificate revocation checking methods. The default method is **crl**. The Cisco NX-OS software uses the certificate revocation methods in the order that you specify.

Step 4	**exit**
	Example:
	`switch(config-trustpoint)# exit`
	`switch(config)#`
		Exits trust point configuration mode.

Step 5	(Optional) **show crypto ca trustpoints** **Example:** `switch(config)# show crypto ca trustpoints`	Displays the trust point CA information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Authenticating the CA, on page 143
Configuring a CRL, on page 152

Generating Certificate Requests

You must generate a request to obtain identity certificates from the associated trust point CA for each of your device's RSA key pairs. 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.

Obtain the CA certificate or CA certificate chain.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca enroll** *name*
3. **exit**
4. (Optional) **show crypto ca certificates**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca enroll** *name*
	Example:
	`switch(config)# crypto ca enroll admin-ca`
	`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: DeviceA.cisco.com`
	`Include the switch serial number in the subject name? [yes/no]: no`
	`Include an IP address in the subject name [yes/no]: yes`
	`ip address:172.22.31.162`
	`The certificate request will be displayed...`
	`-----BEGIN CERTIFICATE REQUEST-----`
	`MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ`
	`KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUJ1DVaSMqNIgJ2kt8rl4lKY`
	`0JC6ManNy4qxk8VeMXZSiLJ4JgTzKWdxbLDkTTysnjuCXGvjb+wj0hEhv/y51T9y`
	`P2NJJ8ornqShrvFZgC7ysN/PyMwKcgzhbVpj+rargZvHtGJ91XTq4WoVkSCzXv8S`
	`VqyH0vEvAgMBAAGgTzAVBgkqhkiG9w0BCQcxCBMGbmJ2MTIzMDYGCSqGSIb3DQEJ`
	`DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ`
	`KoZIhvcNAQEEBQADgYEAkT60KER6Qo8nj0sDXZVHSfJZh6K6JtDz3Gkd99GlFWgt`
	`PftrNcWUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktExiI6Ul88nTOjglXMjja8`
	`8a23bNDpNsM8rklwA6hWkrVL8NUZEFJxqbjfngPNIZacJCUS6ZqKCMetbKytUx0=`
	`-----END CERTIFICATE REQUEST-----`
		Generates a certificate request for an authenticated CA.
		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.

Step 3	**exit**
	Example:
	`switch(config-trustpoint)# exit`
	`switch(config)#`
		Exits trust point configuration mode.

Step 4	(Optional) **show crypto ca certificates** **Example:**	Displays the CA certificates.

	Command or Action	Purpose
	`switch(config)# show crypto ca certificates`	
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Creating a Trust Point CA Association, on page 140

Installing Identity Certificates

You can receive the identity certificate from the CA by e-mail 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

Create an association with the CA.

Obtain the CA certificate or CA certificate chain.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca import** *name* **certificate**
3. **exit**
4. (Optional) **show crypto ca certificates**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca import** *name* **certificate**
	Example:
	`switch(config)# crypto ca import admin-ca certificate`
	`input (cut & paste) certificate in PEM format:`
	`-----BEGIN CERTIFICATE-----`
	`MIIEADCCA6qgAwIBAgIKCjOOoQAAAAAAdDANBgkqhkiG9w0BAQUFADCBkDEgMB4G`
	`CSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD`
	`VQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UEChMFQ2lz`
	`Y28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBDQTAeFw0w`
	`NTExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTEu`
	`Y2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNVACdjQu41C`
	`dQ1WkjKjSICdpLfK5eJSmNCQujGpzcuKsZPFXjF2UoiyeCYE8ylncWyw5E08rJ47`
		Prompts you to cut and paste the identity certificate for the CA named admin-ca.
		The maximum number of identify certificates that you can configure on a device is 16.

	Command or Action	Purpose
	glxr42/sI9IRIb/8udU/cj9jSSfKK56koa7xWYAu8rDfz8jMCnIM4W1aY/q2q4Gb x7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw GYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwHQYDVR0OBBYEFKCLi+2sspWEfgrR bhWmlVyo9jngMIHMBgNVHSMEgcQwgcGAFCco8kaDG6wjTEVNjskYUBoLFmxxoYGW pIGIMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UE BhMCSU4xEjAQBgNVBAgTCUthcm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w DAYDVQQKEwVDaXNjbzETMBEGA1UECxMKamV0c3RvcmFnZTESMBAGA1UEAxMJQXBh cm5hIENBghAFYNKJrLQZlE9JEiWMrRl6MGsGA1UdHwRkMGIwLqAsoCqGKGh0dHA6 Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFybmElMjBDQS5jcmwwMKAuoCyGKmZpbGU6 Ly9cXHNzZS0wOFxDZXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQUH AQEEfjB8MDsGCCsGAQUFBzAChi9odHRwOi8vc3NlLTA4L0NlcnRFbnJvbGwvc3Nl LTA4X0FwYXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYxZmlsZTovL1xcc3NlLTA4 XENlcnRFbnJvbGxcc3NlLTA4X0FwYXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF AANBADbGBGsbe7GNLh9xeOTWBNbm24U69ZSuDDcOcUZUUTgrpnTqVpPyejtsyflw E36cIZu4WsExREqxbTk8ycx7V5o= -----END CERTIFICATE-----	
Step 3	**exit** **Example:** switch(config)# exit switch#	Exits configuration mode.
Step 4	(Optional) **show crypto ca certificates** **Example:** switch# show crypto ca certificates	Displays the CA certificates.
Step 5	(Optional) **copy running-config startup-config** **Example:** switch# copy running-config startup-config	Copies the running configuration to the startup configuration.

Related Topics

Creating a Trust Point CA Association, on page 140

Ensuring Trust Point Configurations Persist Across Reboots

You can ensure that the trustpoint configuration persists across Cisco NX-OS device reboots.

The trust point configuration is a normal Cisco NX-OS device configuration that persists across system reboots only if you copy it explicitly to the startup configuration. The certificates, key pairs, and CRL associated with a trust point are automatically persistent if you have already copied the trust point configuration in the startup configuration. Conversely, if the trust point configuration is not copied to the startup configuration, the certificates, key pairs, and CRL associated with it are not persistent since they require the corresponding trust point configuration after a reboot. Always copy the running configuration to the startup configuration to ensure that the configured certificates, key pairs, and CRLs are persistent. Also, save the running configuration after deleting a certificate or key pair to ensure that the deletions permanent.

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

We recommend that you create a password-protected backup of the identity certificates and save it to an external server.

Note Copying the configuration to an external server does include the certificates and key pairs.

Related Topics

Exporting Identity Information in PKCS 12 Format, on page 150

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 trust point 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 or when you replace the supervisor modules.

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

Before you begin

Authenticate the CA.

Install an identity certificate.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca export** *name* **pkcs12 bootflash:***filename password*
3. **exit**
4. **copy booflash:***filename scheme***://***server/* [*url /*]*filename*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca export** *name* **pkcs12 bootflash:***filename password* **Example:** `switch(config)# crypto ca export admin-ca pkcs12 bootflash:adminid.p12 nbv123`	Exports the identity certificate and associated key pair and CA certificates for a trust point CA. The password is alphanumeric, case sensitive, and has a maximum length of 128 characters.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

	Command or Action	Purpose
Step 4	**copy booflash:***filename scheme***://***server/* [*url /*]*filename* **Example:** `switch# copy bootflash:adminid.p12 tftp:adminid.p12`	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.

Related Topics

Generating an RSA Key Pair, on page 138
Authenticating the CA, on page 143
Installing Identity Certificates, on page 148

Importing Identity Information in PKCS 12 or PKCS 7 Format

You can import the certificate and RSA key pair to recover from a system crash on your device or when you replace the supervisor modules.

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

Before you begin

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

SUMMARY STEPS

1. **copy** *scheme***://** *server/*[*url /*]*filename* **bootflash:***filename*
2. **configure terminal**
3. **crypto ca import** *name* **[pksc12 | pkcs7] bootflash:***filename*
4. **exit**
5. (Optional) **show crypto ca certificates**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**copy** *scheme***://** *server/*[*url /*]*filename* **bootflash:***filename* **Example:** `switch# copy tftp:adminid.p12 bootflash:adminid.p12`	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.

	Command or Action	Purpose

Step 2	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 3	**crypto ca import** *name* **[pksc12 | pkcs7] bootflash:***filename* **Example:** `switch(config)# crypto ca import admin-ca pkcs12 bootflash:adminid.p12 nbv123`	Imports the identity certificate and associated key pair and CA certificates for trust point CA.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show crypto ca certificates** **Example:** `switch# show crypto ca certificates`	Displays the CA certificates.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a CRL

You can manually configure CRLs that you have downloaded from the trust points. The Cisco NX-OS software caches the CRLs in the device bootflash (cert-store). During the verification of a peer certificate, the Cisco NX-OS software checks the CRL from the issuing CA only if you have downloaded the CRL to the device and you have configured certificate revocation checking to use the CRL.

Before you begin

Ensure that you have enabled certificate revocation checking.

SUMMARY STEPS

1. **copy** *scheme*:[*//server/*[*url /*]]*filename* **bootflash:***filename*
2. **configure terminal**
3. **crypto ca crl request** *name* **bootflash:***filename*
4. **exit**
5. (Optional) **show crypto ca crl** *name*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**copy** *scheme*:[*//server/*[*url /*]]*filename* **bootflash:***filename* **Example:** `switch# copy tftp:adminca.crl bootflash:adminca.crl`	Downloads the CRL from 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.

Step 2	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 3	**crypto ca crl request** *name* **bootflash:***filename* **Example:** `switch(config)# crypto ca crl request admin-ca bootflash:adminca.crl`	Configures or replaces the current CRL with the one specified in the file.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show crypto ca crl** *name* **Example:** `switch# show crypto ca crl admin-ca`	Displays the CA CRL information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Deleting Certificates from the CA Configuration

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

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca trustpoint** *name*
3. **delete ca-certificate**
4. **delete certificate** [**force**]

5. **exit**
6. (Optional) **show crypto ca certificates** [*name*]
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca trustpoint** *name*
	Example:
	`switch(config)# crypto ca trustpoint admin-ca`
	`switch(config-trustpoint)#`
		Specifies a trust point CA and enters trust point configuration mode.

Step 3	**delete ca-certificate** **Example:** `switch(config-trustpoint)# delete ca-certificate`	Deletes the CA certificate or certificate chain.
Step 4	**delete certificate** [**force**] **Example:** `switch(config-trustpoint)# delete certificate`	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 the identity certificate and leave the applications (such as SSH) without a certificate to use.

Step 5	**exit**
	Example:
	`switch(config-trustpoint)# exit`
	`switch(config)#`
		Exits trust point configuration mode.

Step 6	(Optional) **show crypto ca certificates** [*name*] **Example:** `switch(config)# show crypto ca certificates admin-ca`	Displays the CA certificate information.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Deleting RSA Key Pairs from a Cisco NX-OS Device

You can delete the RSA key pairs from a Cisco NX-OS device if you believe the RSA key pairs were compromised in some way and should no longer be used.

Note After you delete RSA key pairs from a device, ask the CA administrator to revoke your device's certificates at the CA. You must supply the challenge password that you created when you originally requested the certificates.

SUMMARY STEPS

1. **configure terminal**
2. **crypto key zeroize rsa** *label*
3. **exit**
4. (Optional) **show crypto key mypubkey rsa**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto key zeroize rsa** *label* **Example:** `switch(config)# crypto key zeroize rsa MyKey`	Deletes the RSA key pair.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show crypto key mypubkey rsa** **Example:** `switch# show crypto key mypubkey rsa`	Displays the RSA key pair configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Generating Certificate Requests, on page 146

Verifying the PKI Configuration

To display PKI configuration information, perform one of the following tasks:

Command	Purpose
show crypto key mypubkey rsa	Displays information about the RSA public keys generated on the Cisco NX-OS device.
show crypto ca certificates	Displays information about CA and identity certificates.
show crypto ca crl	Displays information about CA CRLs.
show crypto ca trustpoints	Displays information about CA trust points.

Configuration Examples for PKI

This section shows examples of the tasks that you can use to configure certificates and CRLs on Cisco NX-OS devices using a Microsoft Windows Certificate server.

Note You can use any type of certificate server to generate digital certificates. You are not limited to using the Microsoft Windows Certificate server.

Configuring Certificates on a Cisco NX-OS Device

To configure certificates on a Cisco NX-OS device, follow these steps:

Step 1 Configure the device FQDN.

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# hostname Device-1
Device-1(config)#
```

Step 2 Configure the DNS domain name for the device.

```
Device-1(config)# ip domain-name cisco.com
```

Step 3 Create a trust point.

```
Device-1(config)# crypto ca trustpoint myCA
Device-1(config-trustpoint)# exit
Device-1(config)# show crypto ca trustpoints
trustpoint: myCA; key:
revokation methods:  crl
```

Step 4 Create an RSA key pair for the device.

```
Device-1(config)# crypto key generate rsa label myKey exportable modulus 1024
Device-1(config)# show crypto key mypubkey rsa
key label: myKey
key size: 1024
exportable: yes
```

Step 5 Associate the RSA key pair to the trust point.

```
Device-1(config)# crypto ca trustpoint myCA
Device-1(config-trustpoint)# rsakeypair myKey
Device-1(config-trustpoint)# exit
Device-1(config)# show crypto ca trustpoints
trustpoint: myCA; key: myKey
revokation methods:  crl
```

Step 6 Download the CA certificate from the Microsoft Certificate Service web interface.

Step 7 Authenticate the CA that you want to enroll to the trust point.

```
Device-1(config)# crypto ca authenticate myCA
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
cm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4wDAYDVQQKEwVDaXNjbzETMBEG
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]:y

Device-1(config)# show crypto ca certificates
Trustpoint: myCA
CA certificate 0:
subject= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/
L=Bangalore/O=Yourcompany/OU=netstorage/CN=Aparna CA
issuer= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/
L=Bangalore/O=Yourcompany/OU=netstorage/CN=Aparna CA
serial=0560D289ACB419944F4912258CAD197A
notBefore=May  3 22:46:37 2005 GMT
notAfter=May  3 22:55:17 2007 GMT
MD5 Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12
purposes: sslserver sslclient ike
```

Step 8 Generate a request certificate to use to enroll with a trust point.

```
Device-1(config)# crypto ca enroll myCA
 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: Device-1.cisco.com
 Include the switch serial number in the subject name? [yes/no]: no
 Include an IP address in the subject name [yes/no]: yes
ip address: 10.10.1.1
 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-----
```

Step 9 Request an identity certificate from the Microsoft Certificate Service web interface.

Step 10 Import the identity certificate.

```
Device-1(config)# crypto ca import myCA certificate
input (cut & paste) certificate in PEM format:
-----BEGIN CERTIFICATE-----
MIIEADCCA6qgAwIBAgIKCjOOoQAAAAAAdDANBgkqhkiG9w0BAQUFADCBkDEgMB4G
CSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD
VQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UEChMFQ2lz
Y28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBDQTAeFw0w
NTExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTEu
Y2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNVACdjQu41C
dQ1WkjKjSICdpLfK5eJSmNCQujGpzcuKsZPFXjF2UoiyeCYE8ylncWyw5E08rJ47
glxr42/sI9IRIb/8udU/cj9jSSfKK56koa7xWYAu8rDfz8jMCnIM4W1aY/q2q4Gb
x7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw
GYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwHQYDVR0OBBYEFKCLi+2sspWEfgrR
bhWmlVyo9jngMIHMBgNVHSMEgcQwgcGAFCco8kaDG6wjTEVNjskYUBoLFmxxoYGW
pIGTMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UE
BhMCSU4xEjAQBgNVBAgTCUthcm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w
DAYDVQQKEwVDaXNjbzETMBEGA1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBh
cm5hIENBghAFYNKJrLQZlE9JEiWMrRl6MGsGA1UdHwRkMGIwLqAsoCqGKGh0dHA6
Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFybmElMjBDQS5jcmwwMKAuoCyGKmZpbGU6
Ly9cXHNzZS0wOFxDZXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQUH
AQEEfjB8MDsGCCsGAQUFBzAChi9odHRwOi8vc3NlLTA4L0NlcnRfbnJvbGwvc3Nl
LTA4X0FwYXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYxZmlsZTovL1xcc3NlLTA4
XENlcnRfbnJvbGxcc3NlLTA4X0FwYXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF
AANBADbGBGsbe7GNLh9xeOTWBNbm24U69ZSuDDcOcUZUUTgrpnTqVpPyejtsyflw
E36cIZu4WsExREqxbTk8ycx7V5o=
-----END CERTIFICATE-----
Device-1(config)# exit
Device-1#
```

Step 11 Verify the certificate configuration.

Step 12 Save the certificate configuration to the startup configuration.

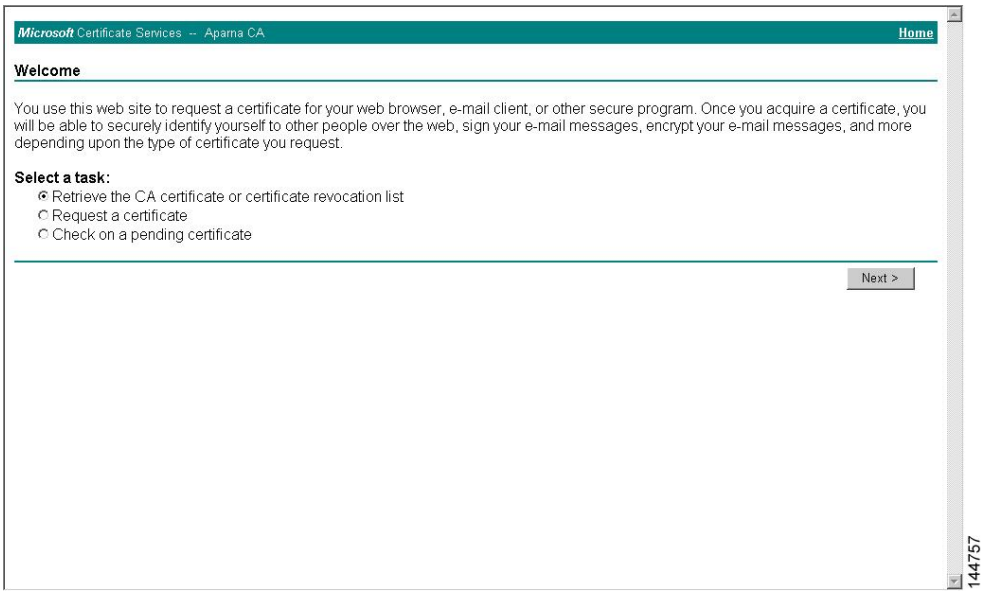
Related Topics

Downloading a CA Certificate, on page 159
Requesting an Identity Certificate, on page 162

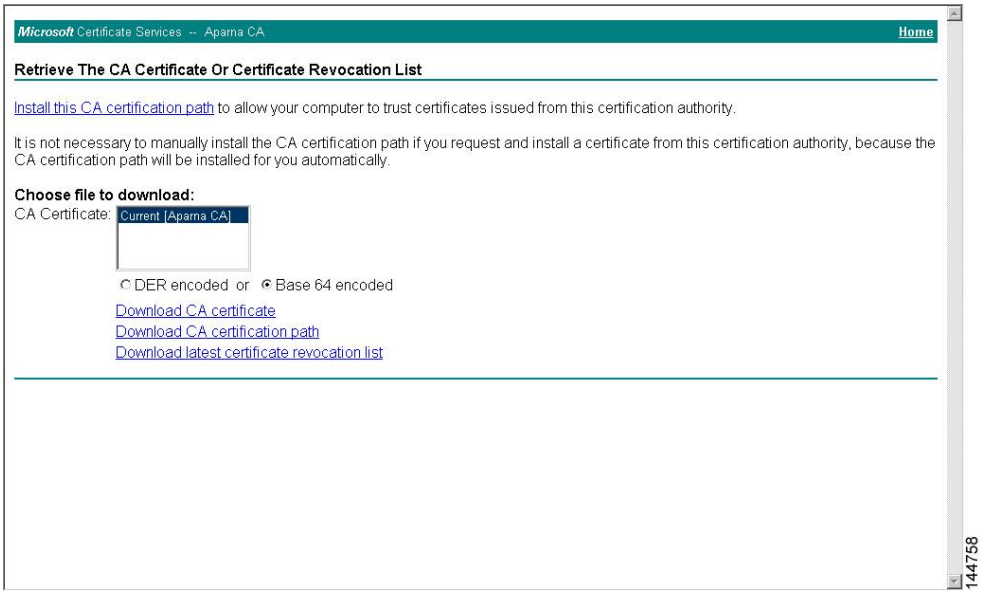
Downloading a CA Certificate

To download a CA certificate from the Microsoft Certificate Services web interface, follow these steps:

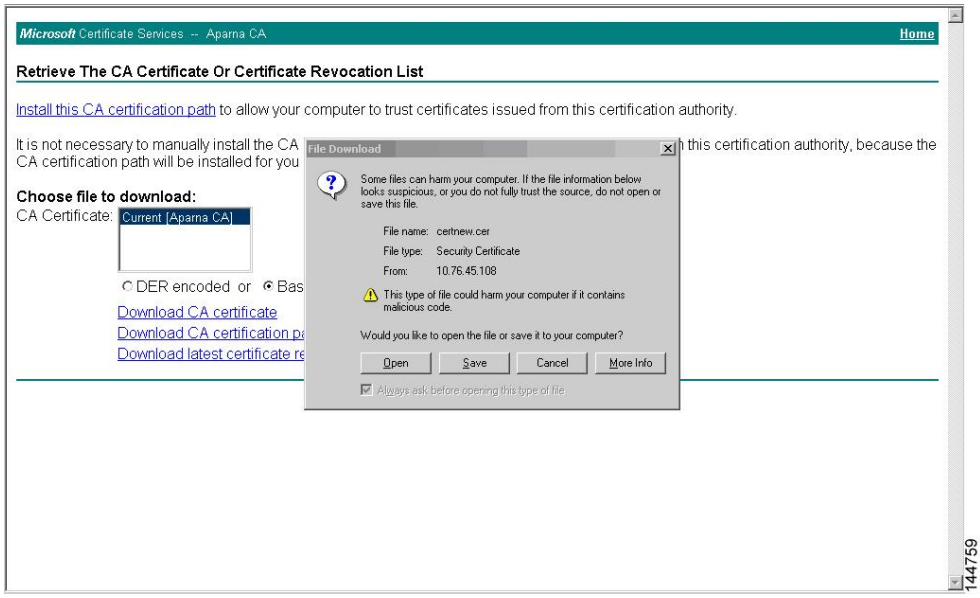
Step 1 From the Microsoft Certificate Services web interface, click **Retrieve the CA certificate or certificate revocation task** and click **Next**.



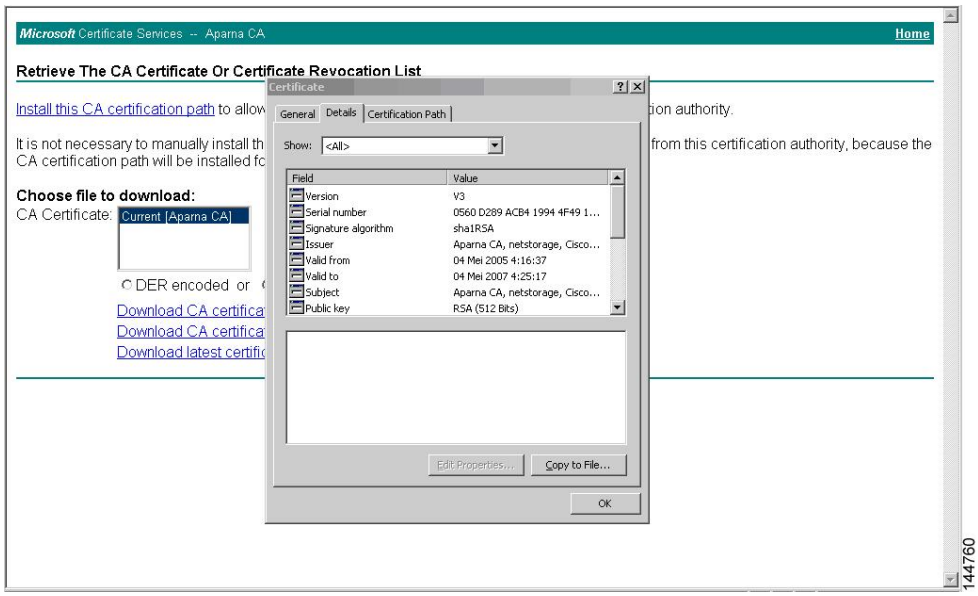
Step 2 From the display list, choose the CA certificate file to download from the displayed list. Then click **Base 64 encoded** and click **Download CA certificate**.



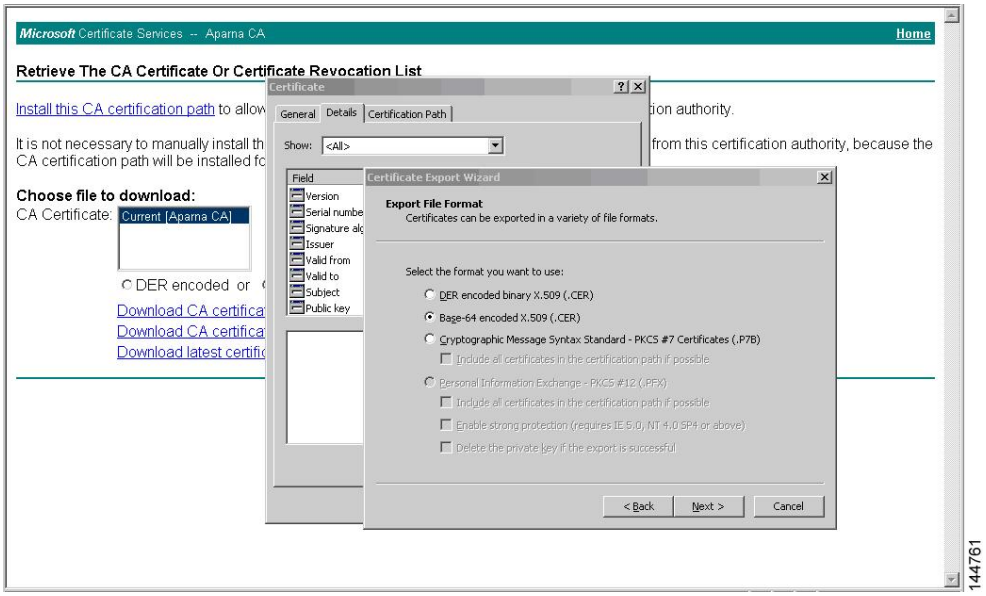
Step 3 Click **Open** in the File Download dialog box.



Step 4 In the Certificate dialog box, click **Copy to File** and click **OK**.



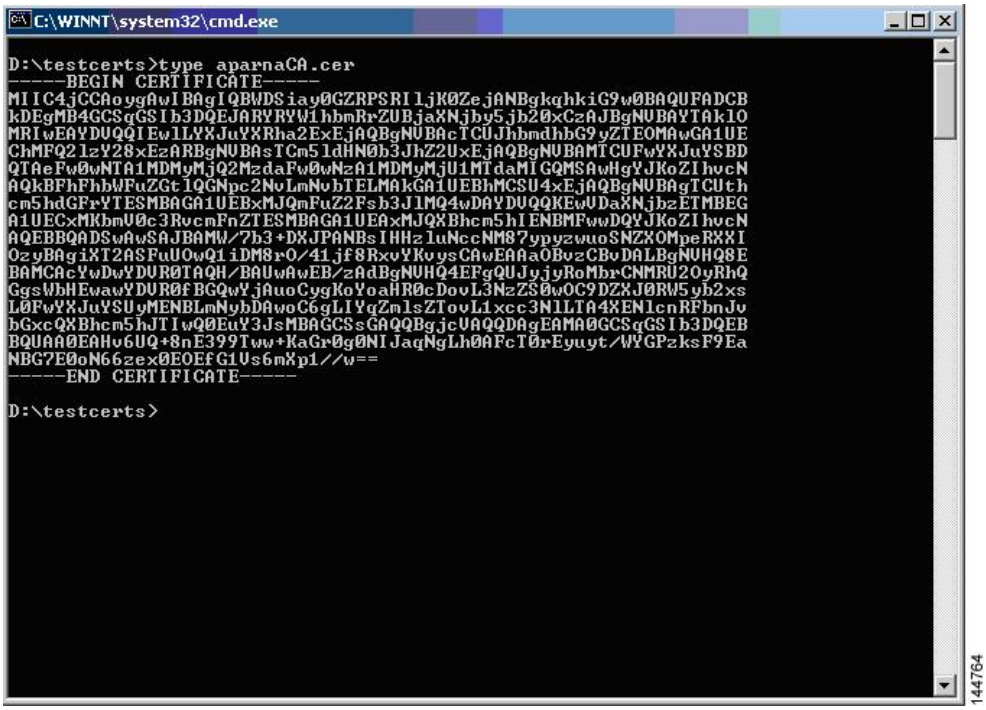
Step 5 From the Certificate Export Wizard dialog box, choose the **Base-64 encoded X.509 (CER)** and click **Next**.



Step 6 In the File name: text box on the Certificate Export Wizard dialog box, enter the destination file name and click **Next**.

Step 7 In the Certificate Export Wizard dialog box, click **Finish**.

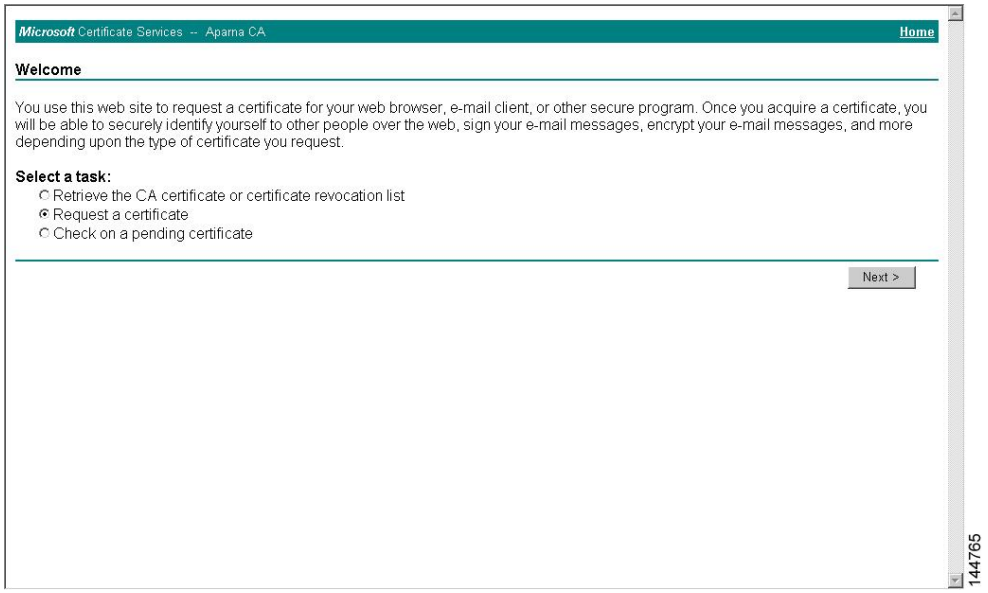
Step 8 Enter the Microsoft Windows **type** command to display the CA certificate stored in Base-64 (PEM) format.



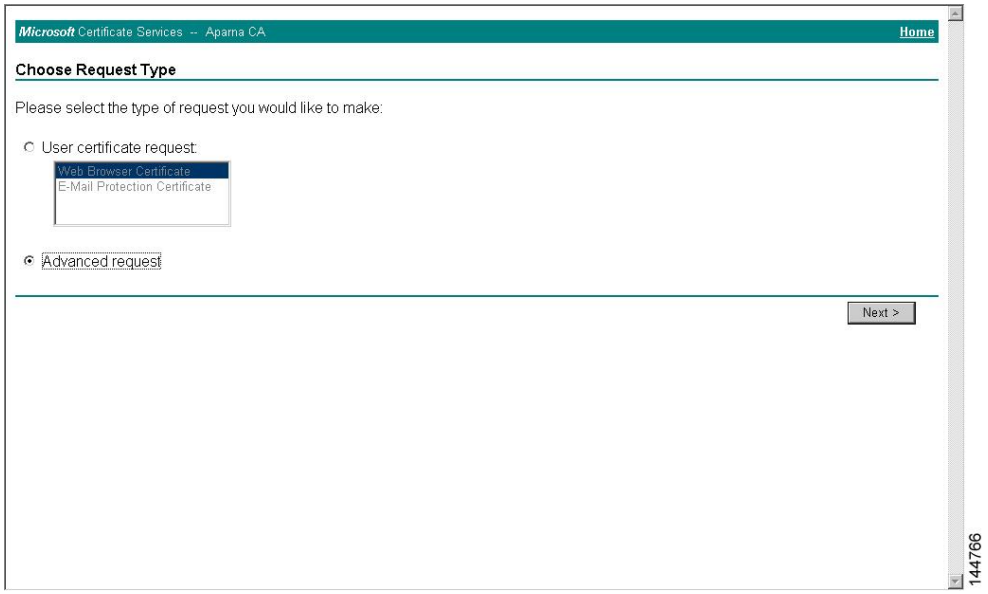
Requesting an Identity Certificate

To request an identify certificate from a Microsoft Certificate server using a PKCS#12 certificate signing request (CRS), follow these steps:

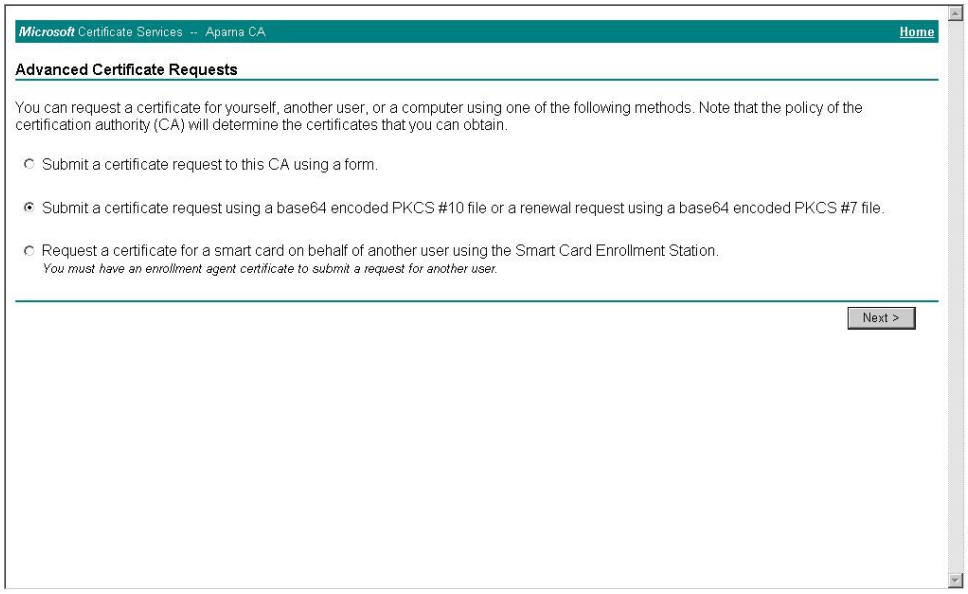
Step 1 From the Microsoft Certificate Services web interface, click **Request a certificate** and click **Next**.



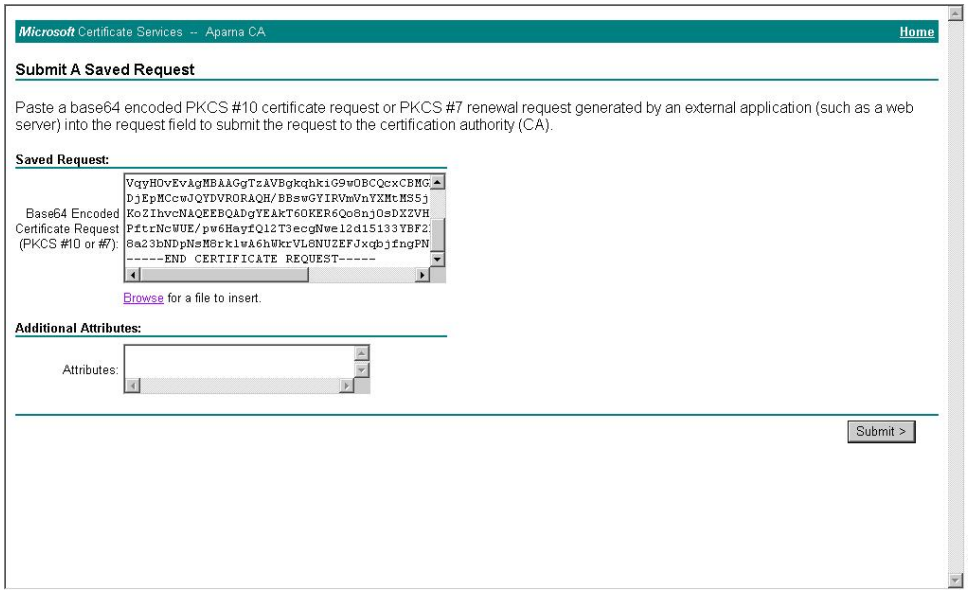
Step 2 Click **Advanced request** and click **Next**.



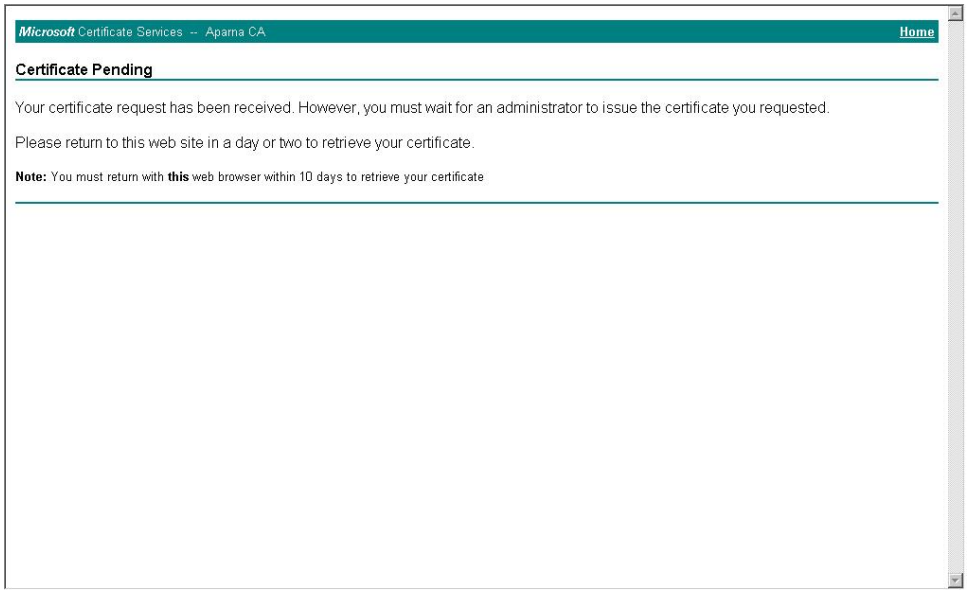
Step 3 Click **Submit a certificate request using a base64 encoded PKCS#10 file or a renewal request using a base64 encoded PKCS#7 file** and click **Next**.



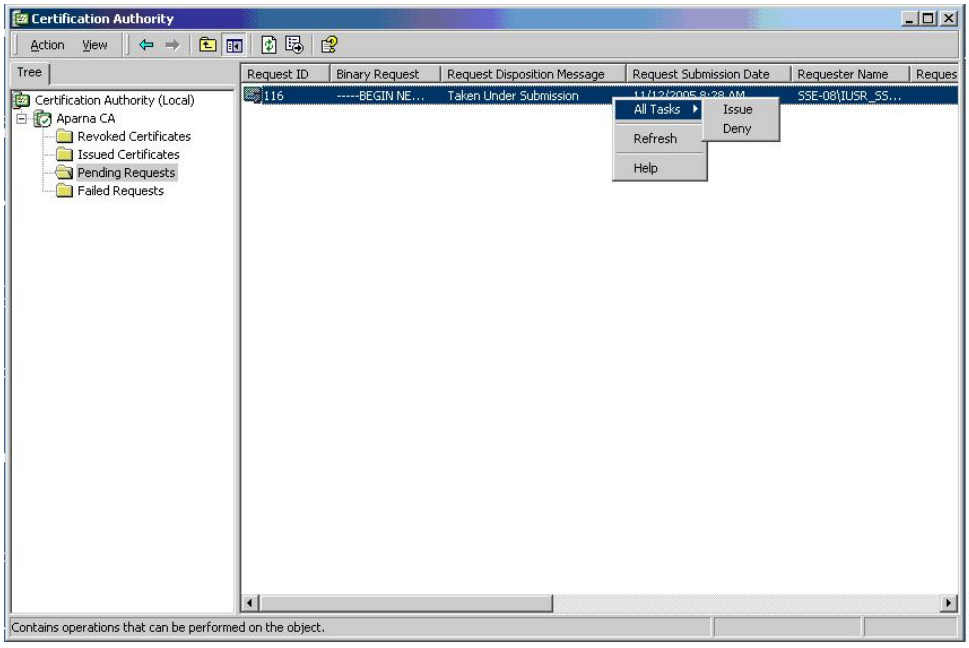
Step 4 In the Saved Request text box, paste the base64 PKCS#10 certificate request and click **Next**. The certificate request is copied from the Cisco NX-OS device console.



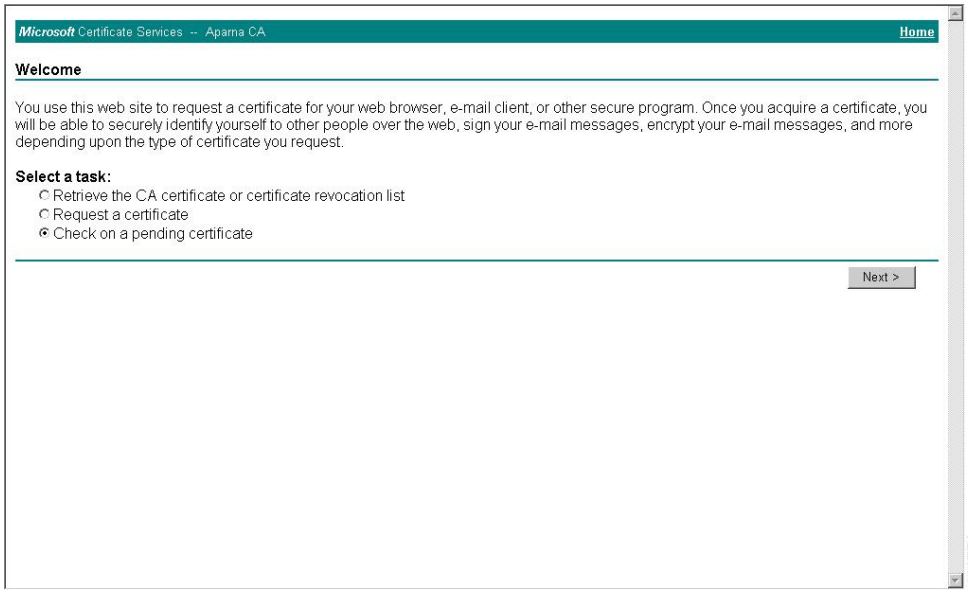
Step 5 Wait one or two days until the certificate is issued by the CA administrator.



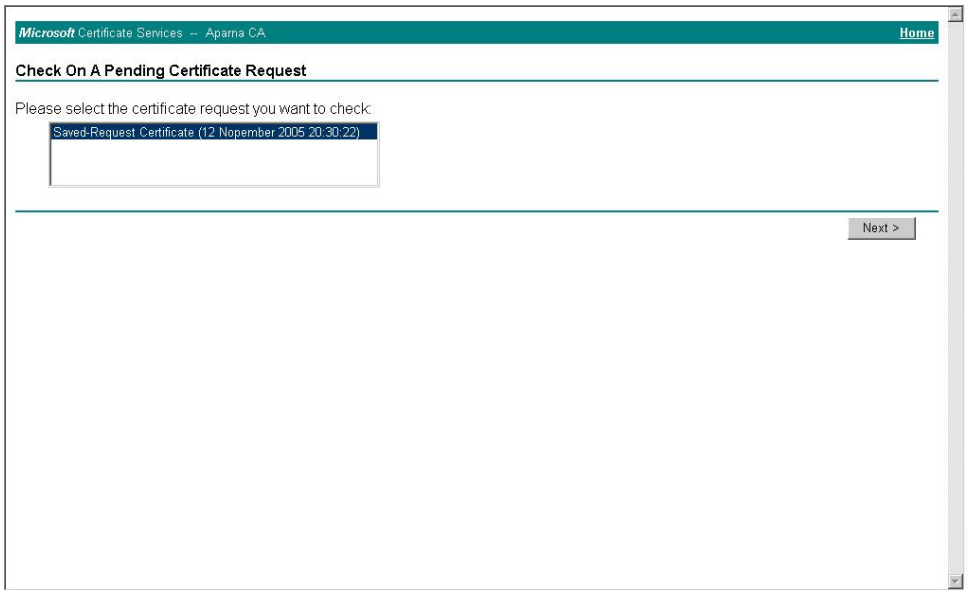
Step 6 Note that the CA administrator approves the certificate request.



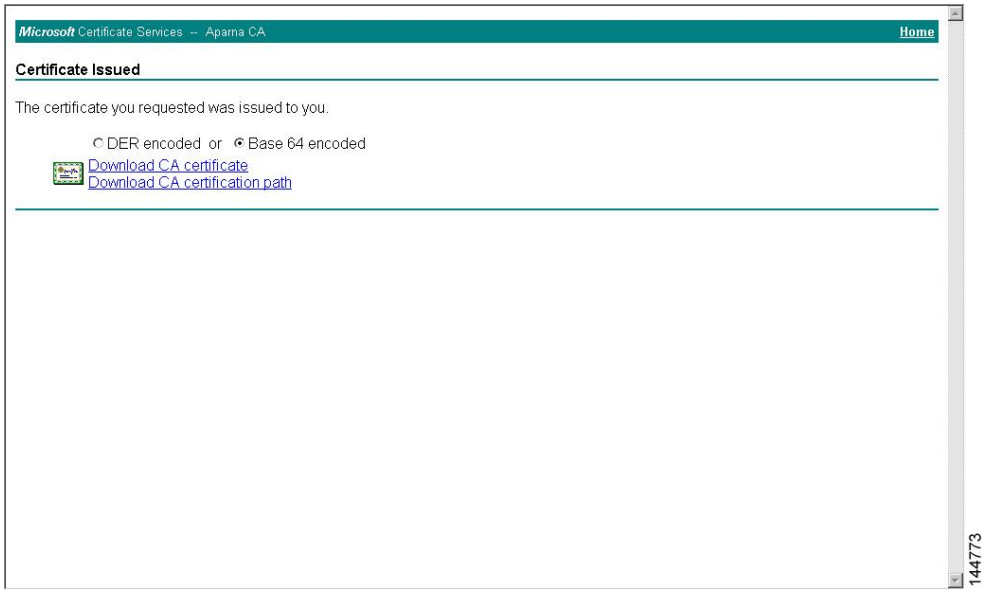
Step 7 From the Microsoft Certificate Services web interface, click **Check on a pending certificate** and click **Next**.



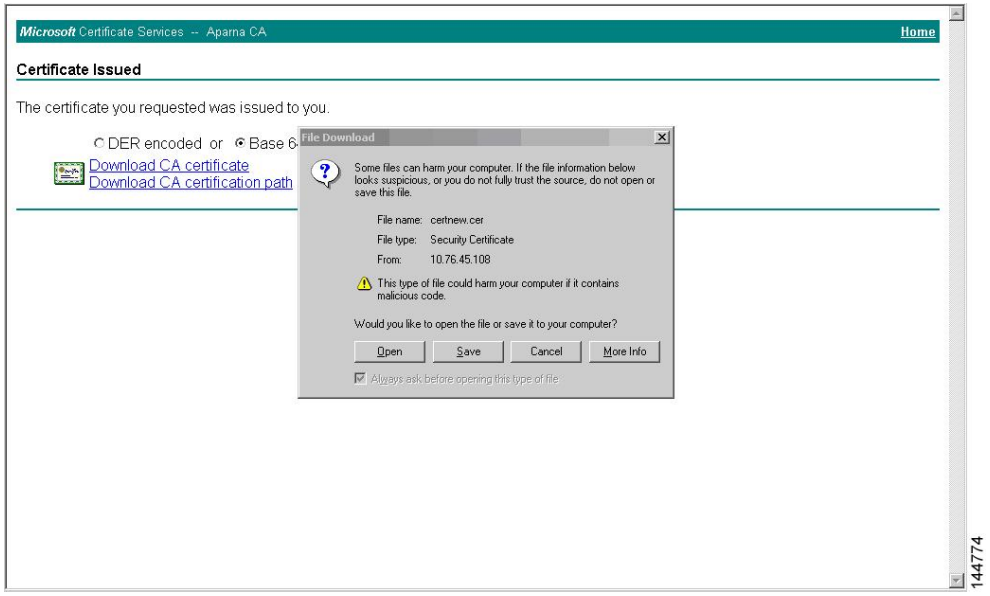
Step 8 Choose the certificate request that you want to check and click **Next**.



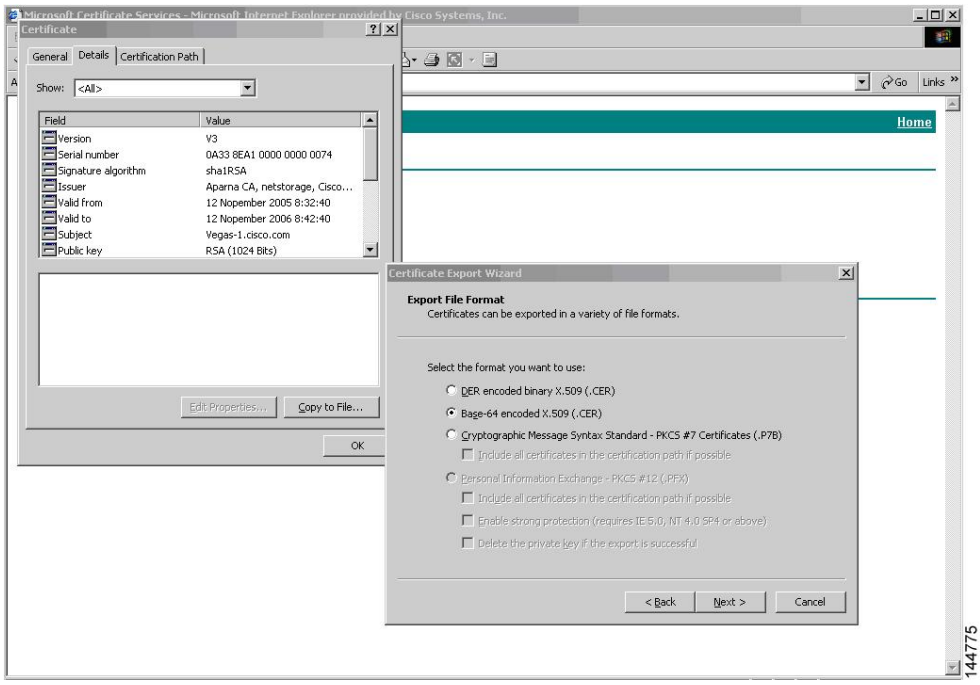
Step 9 Click **Base 64 encoded** and click **Download CA certificate**.



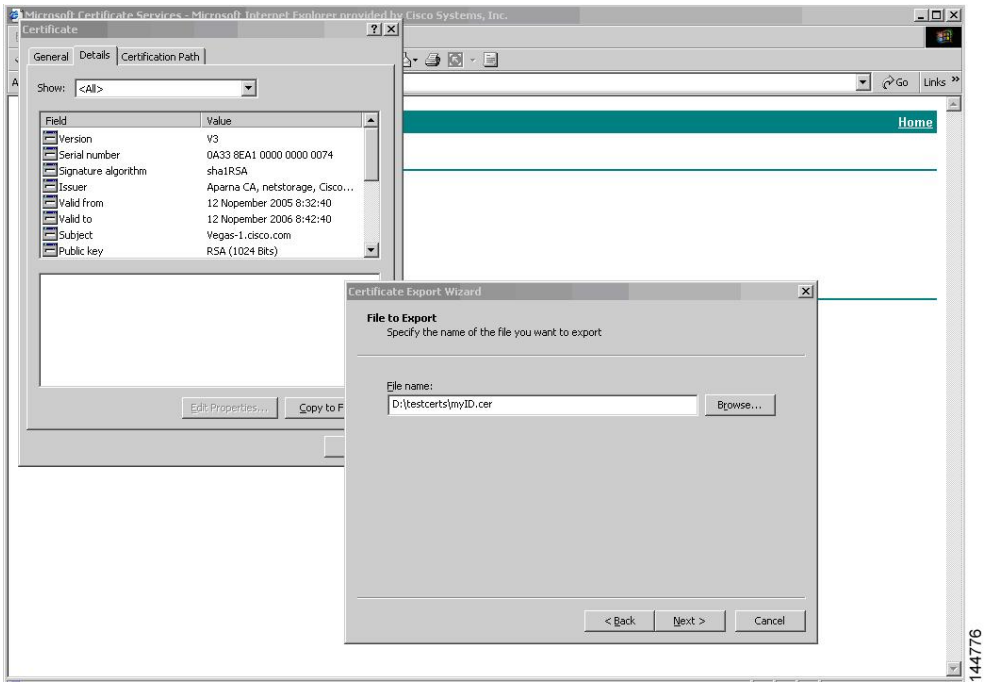
Step 10 In the File Download dialog box, click **Open**.

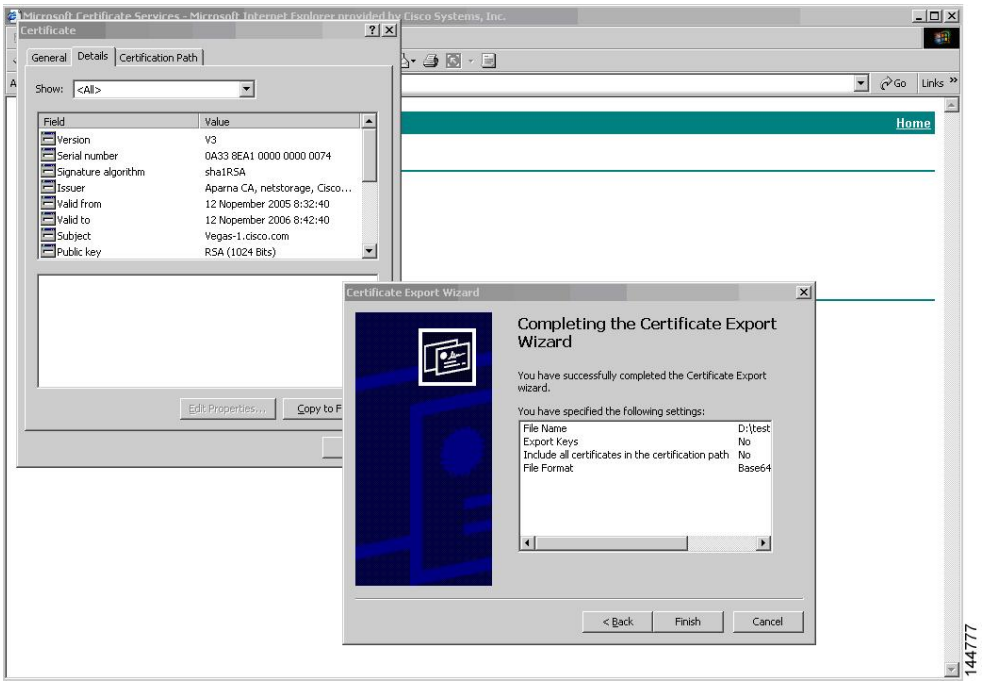


Step 11 In the Certificate box, click **Details** tab and click **Copy to File...**. In the Certificate Export Dialog box, click **Base-64 encoded X.509 (.CER)**, and click **Next**.



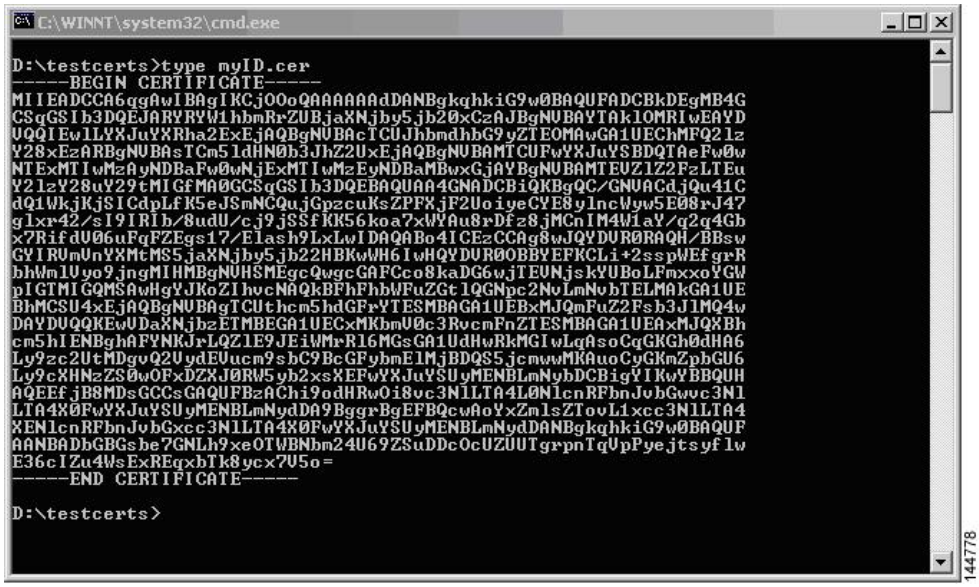
Step 12 In the File name: text box on the Certificate Export Wizard dialog box, enter the destination file name and click **Next**.





Step 13 Click **Finish**.

Step 14 Enter the Microsoft Windows **type** command to display the identity certificate in base64-encoded format.



Related Topics

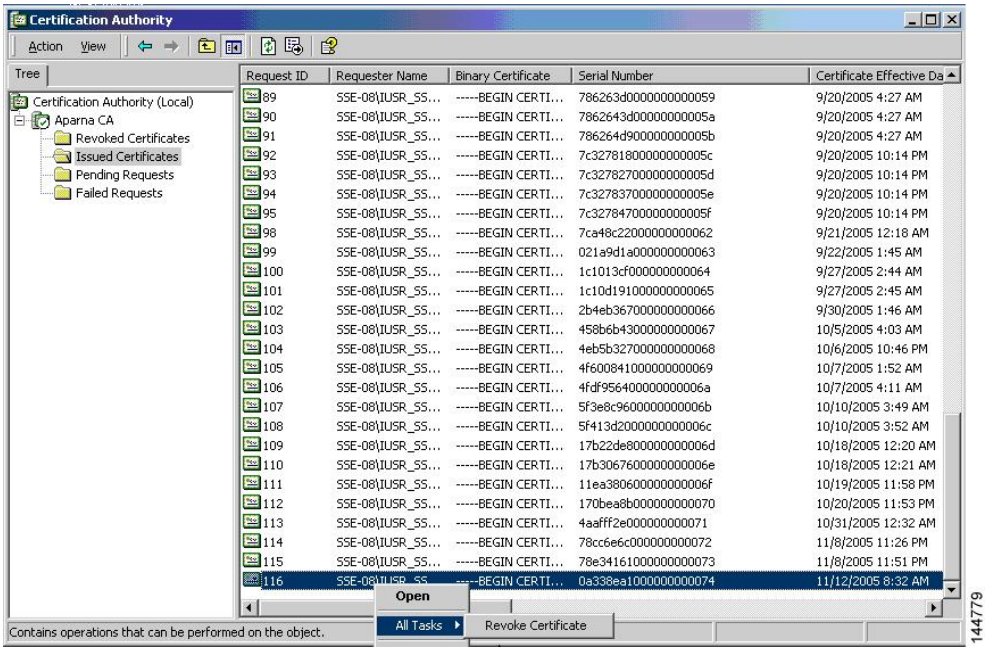
Generating Certificate Requests, on page 146
Configuring Certificates on a Cisco NX-OS Device, on page 156

Revoking a Certificate

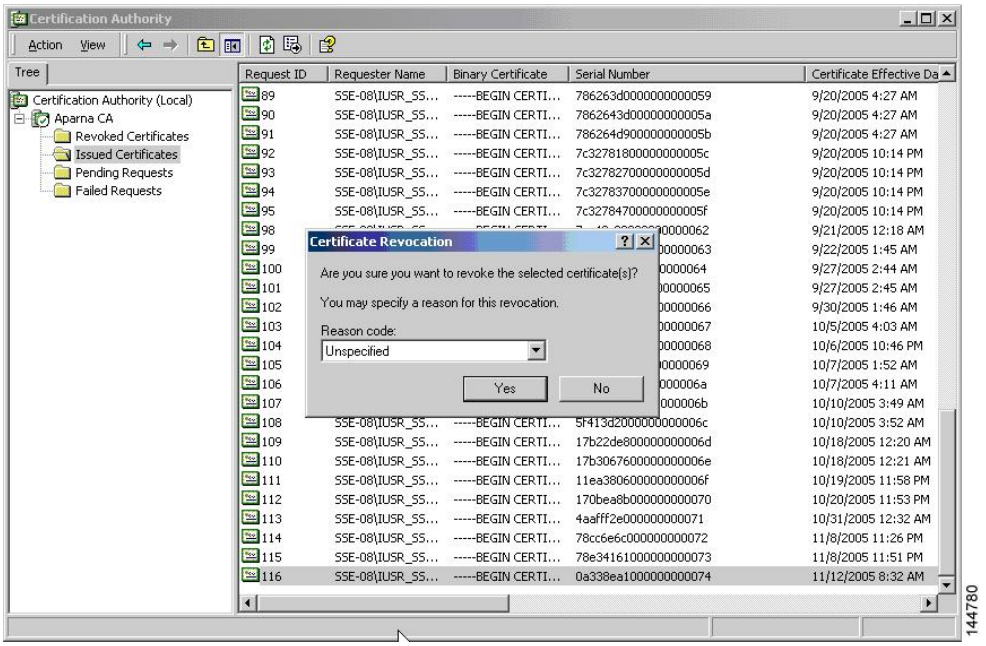
To revoke a certificate using the Microsoft CA administrator program, follow these steps:

Step 1 From the Certification Authority tree, click **Issued Certificates** folder. From the list, right-click the certificate that you want to revoke.

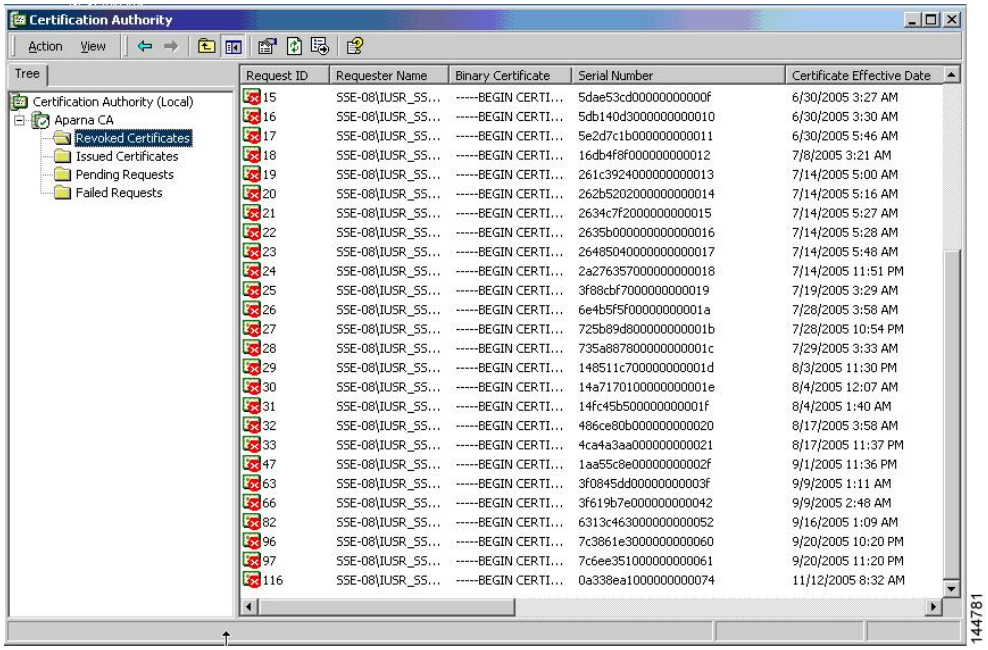
Step 2 Choose **All Tasks > Revoke Certificate**.



Step 3 From the Reason code drop-down list, choose a reason for the revocation and click **Yes**.



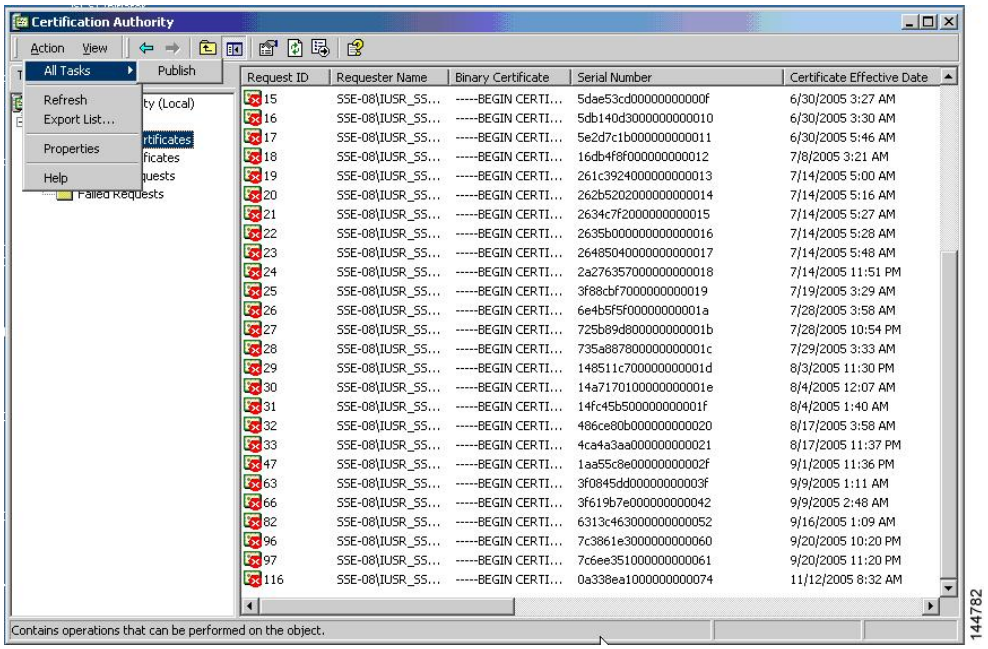
Step 4 Click the **Revoked Certificates** folder to list and verify the certificate revocation.



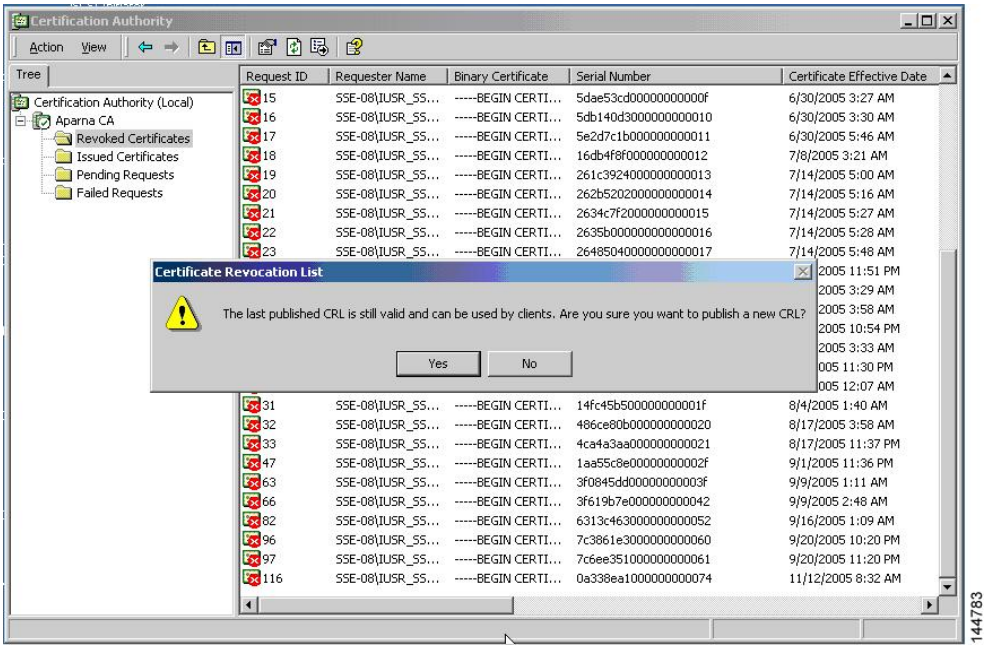
Generating and Publishing the CRL

To generate and publish the CRL using the Microsoft CA administrator program, follow these steps:

Step 1 From the Certification Authority screen, choose **Action** > **All Tasks** > **Publish**.



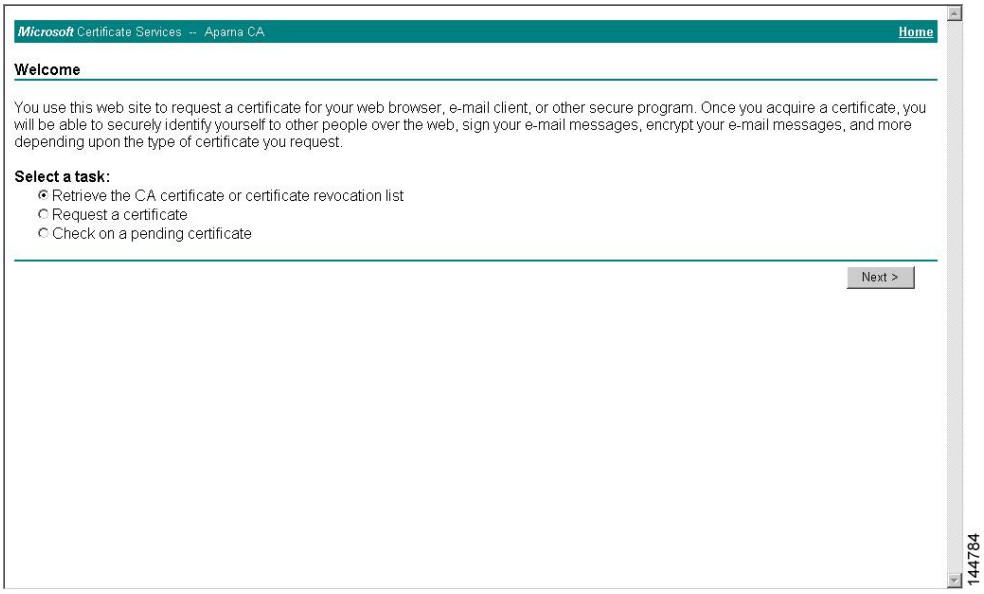
Step 2 In the Certificate Revocation List dialog box, click **Yes** to publish the latest CRL.



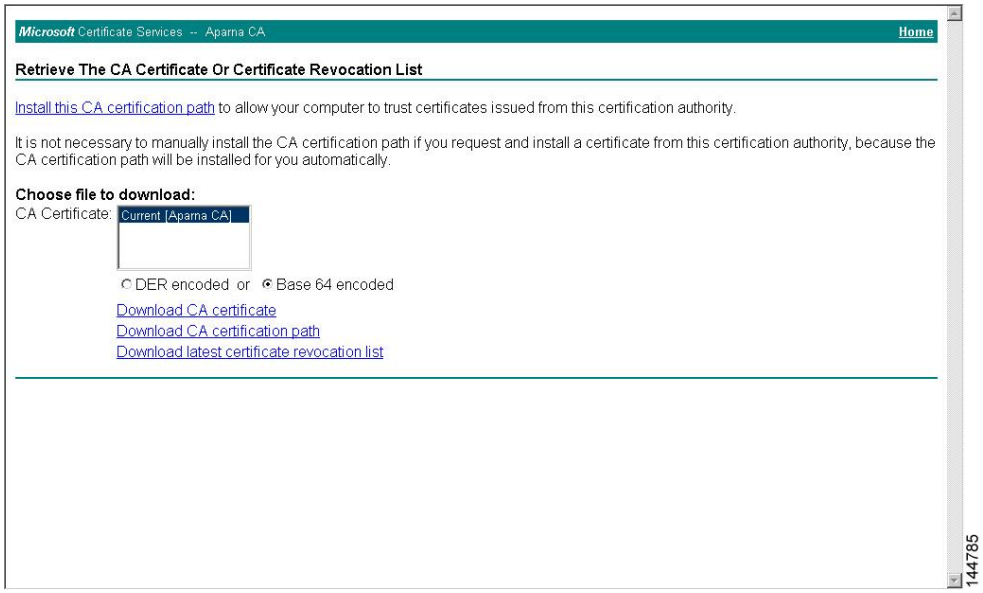
Downloading the CRL

To download the CRL from the Microsoft CA website, follow these steps:

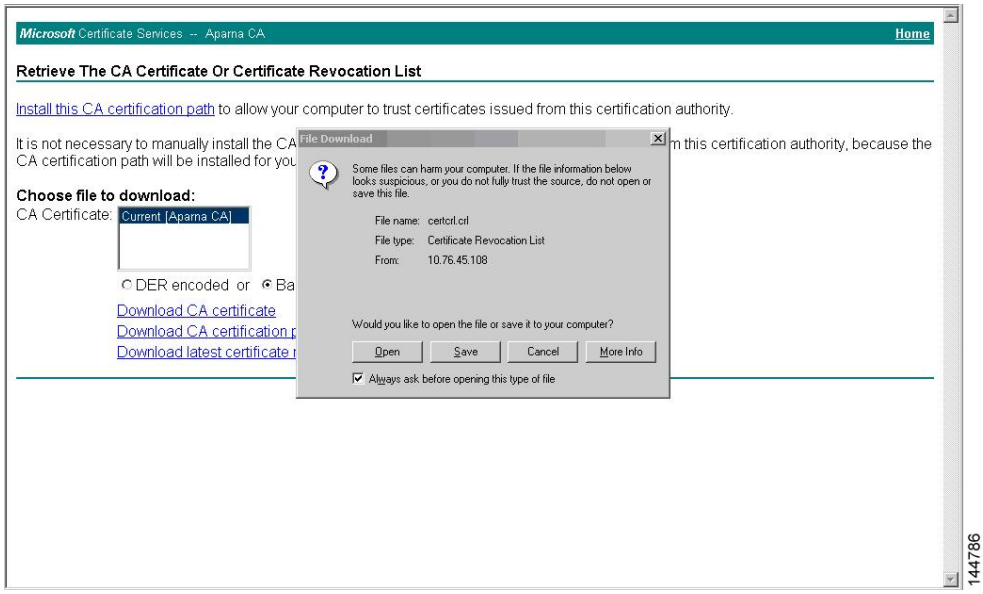
Step 1 From the Microsoft Certificate Services web interface, click **Retrieve the CA certificate or certificate revocation list** and click **Next**.



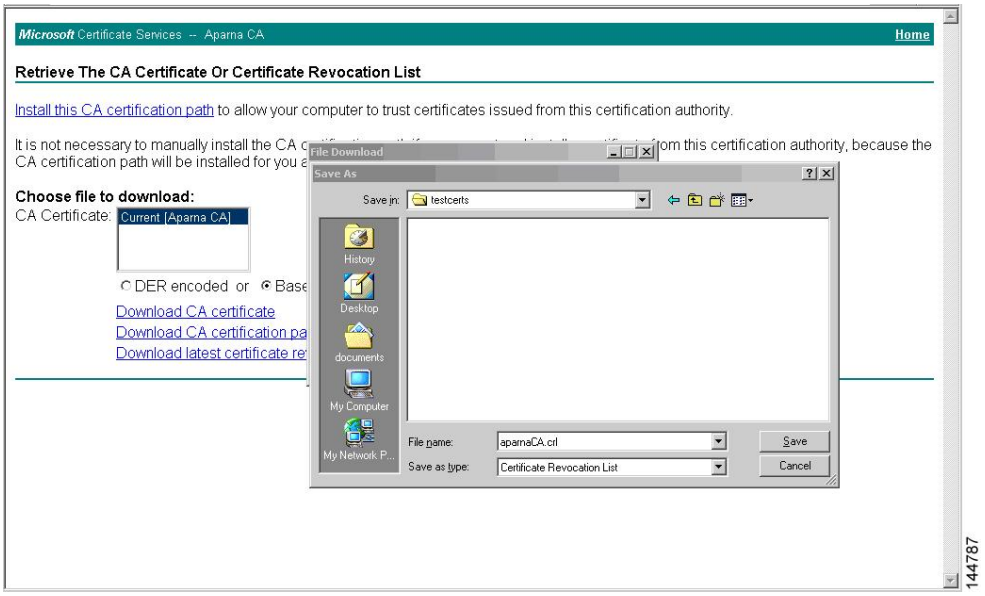
Step 2 Click **Download latest certificate revocation list**.



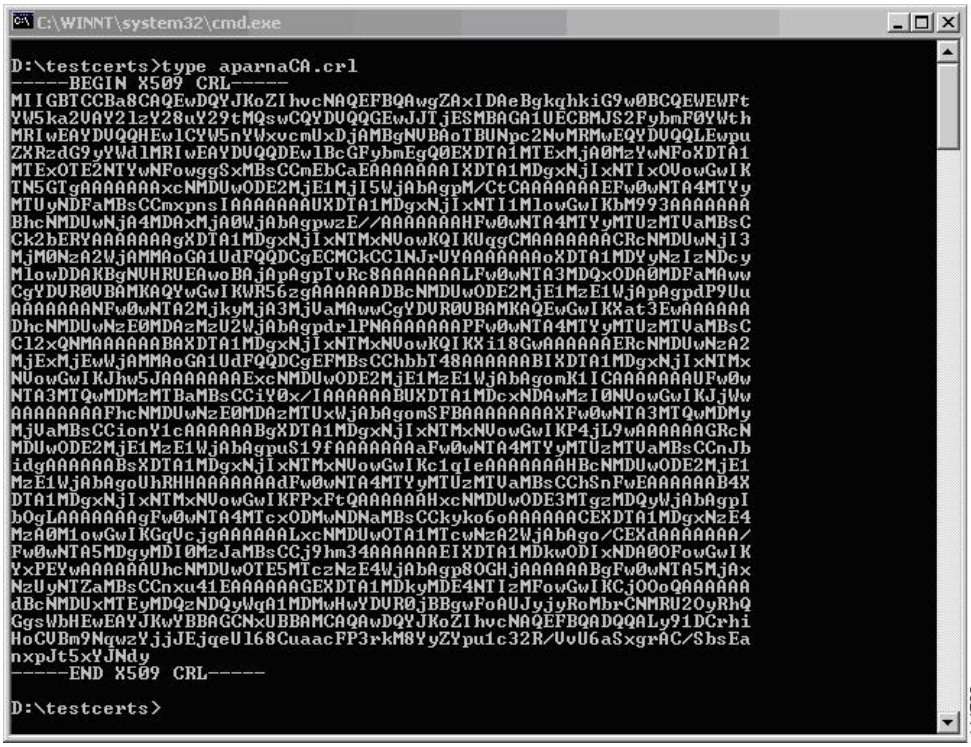
Step 3 In the File Download dialog box, click **Save**.



Step 4 In the Save As dialog box, enter the destination file name and click **Save**.



Step 5 Enter the Microsoft Windows **type** command to display the CRL.



Related Topics

Configuring Certificate Revocation Checking Methods, on page 145

Importing the CRL

To import the CRL to the trust point corresponding to the CA, follow these steps:

Step 1 Copy the CRL file to the Cisco NX-OS device bootflash.

```
Device-1# copy tftp:apranaCA.crl bootflash:aparnaCA.crl
```

Step 2 Configure the CRL.

```
Device-1# configure terminal
Device-1(config)# crypto ca crl request myCA bootflash:aparnaCA.crl
Device-1(config)#
```

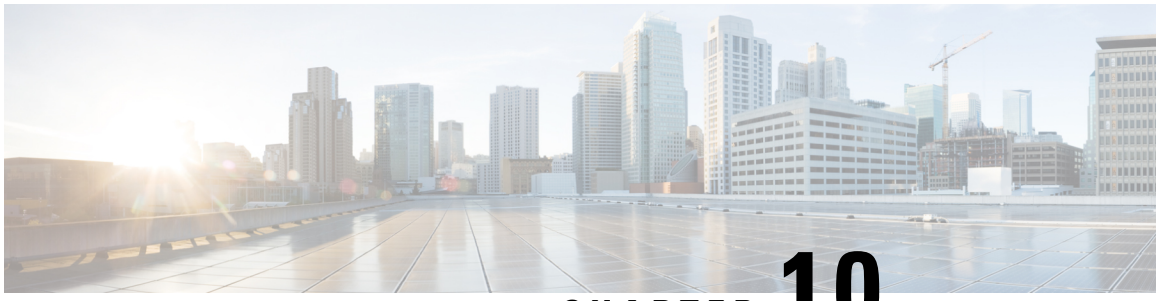
Step 3 Display the contents of the CRL.

```
Device-1(config)# show crypto ca crl myCA
Trustpoint: myCA
CRL:
Certificate Revocation List (CRL):
        Version 2 (0x1)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: /emailAddress=admin@yourcompany.com/C=IN/ST=Karnatak
Yourcompany/OU=netstorage/CN=Aparna CA
        Last Update: Nov 12 04:36:04 2005 GMT
        Next Update: Nov 19 16:56:04 2005 GMT
        CRL extensions:
            X509v3 Authority Key Identifier:
            keyid:27:28:F2:46:83:1B:AC:23:4C:45:4D:8E:C9:18:50:1
            1.3.6.1.4.1.311.21.1:
                ...
Revoked Certificates:
    Serial Number: 611B09A1000000000002
        Revocation Date: Aug 16 21:52:19 2005 GMT
Serial Number: 4CDE464E000000000003
        Revocation Date: Aug 16 21:52:29 2005 GMT
    Serial Number: 4CFC2B42000000000004
        Revocation Date: Aug 16 21:52:41 2005 GMT
    Serial Number: 6C699EC2000000000005
        Revocation Date: Aug 16 21:52:52 2005 GMT
    Serial Number: 6CCF7DDC000000000006
        Revocation Date: Jun  8 00:12:04 2005 GMT
    Serial Number: 70CC4FFF000000000007
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 4D9B1116000000000008
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 52A80230000000000009
        Revocation Date: Jun 27 23:47:06 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            CA Compromise
Serial Number: 5349AD4600000000000A
        Revocation Date: Jun 27 23:47:22 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            CA Compromise
Serial Number: 53BD173C00000000000B
        Revocation Date: Jul  4 18:04:01 2005 GMT
```

```
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Certificate Hold
Serial Number: 591E7ACE00000000000C
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5D3FD52E00000000000D
        Revocation Date: Jun 29 22:07:25 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Key Compromise
Serial Number: 5DAB771300000000000E
        Revocation Date: Jul 14 00:33:56 2005 GMT
    Serial Number: 5DAE53CD00000000000F
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5DB140D3000000000010
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 5E2D7C1B000000000011
        Revocation Date: Jul  6 21:12:10 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            Cessation Of Operation
Serial Number: 16DB4F8F000000000012
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 261C3924000000000013
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 262B5202000000000014
        Revocation Date: Jul 14 00:33:10 2005 GMT
    Serial Number: 2634C7F2000000000015
        Revocation Date: Jul 14 00:32:45 2005 GMT
    Serial Number: 2635B000000000000016
        Revocation Date: Jul 14 00:31:51 2005 GMT
    Serial Number: 26485040000000000017
        Revocation Date: Jul 14 00:32:25 2005 GMT
    Serial Number: 2A276357000000000018
Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 3F88CBF7000000000019
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 6E4B5F5F00000000001A
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 725B89D800000000001B
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 735A887800000000001C
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 148511C700000000001D
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 14A7170100000000001E
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 14FC45B500000000001F
        Revocation Date: Aug 17 18:30:42 2005 GMT
    Serial Number: 486CE80B000000000020
        Revocation Date: Aug 17 18:30:43 2005 GMT
    Serial Number: 4CA4A3AA000000000021
        Revocation Date: Aug 17 18:30:43 2005 GMT
    Serial Number: 1AA55C8E00000000002F
        Revocation Date: Sep  5 17:07:06 2005 GMT
    Serial Number: 3F0845DD00000000003F
        Revocation Date: Sep  8 20:24:32 2005 GMT
    Serial Number: 3F619B7E000000000042
        Revocation Date: Sep  8 21:40:48 2005 GMT
    Serial Number: 6313C463000000000052
        Revocation Date: Sep 19 17:37:18 2005 GMT
    Serial Number: 7C3861E3000000000060
        Revocation Date: Sep 20 17:52:56 2005 GMT
    Serial Number: 7C6EE351000000000061
```

```
    Revocation Date: Sep 20 18:52:30 2005 GMT
Serial Number: 0A338EA1000000000074   <-- Revoked identity certificate
    Revocation Date: Nov 12 04:34:42 2005 GMT
Signature Algorithm: sha1WithRSAEncryption
    0b:cb:dd:43:0a:b8:62:1e:80:95:06:6f:4d:ab:0c:d8:8e:32:
    44:8e:a7:94:97:af:02:b9:a6:9c:14:fd:eb:90:cf:18:c9:96:
    29:bb:57:37:d9:1f:d5:bd:4e:9a:4b:18:2b:00:2f:d2:6e:c1:
    1a:9f:1a:49:b7:9c:58:24:d7:72
```

Note The identity certificate for the device that was revoked (serial number 0A338EA1000000000074) is listed at the end.



CHAPTER 10

Configuring Access Control Lists

This chapter contains the following sections:

About ACLs

An access control list (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 switch determines that an ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether the packet is permitted or denied. If there is no match, the switch applies the applicable default rule. The switch continues processing packets that are permitted and drops packets that are denied.

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.

IP ACL Types and Applications

The Cisco Nexus device supports IPv4 for security traffic filtering. The switch allows you to use IP access control lists (ACLs) as port ACLs, VLAN ACLs, and Router ACLs as shown in the following table.

Table 12: Security ACL Applications

Application	Supported Interfaces	Types of ACLs Supported
Port ACL	An ACL is considered a port ACL when you apply it to one of the following: • Ethernet interface • Ethernet port-channel interface When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on the trunk port.	IPv4 ACLs
Router ACL	• VLAN interfaces **Note** You must enable VLAN interfaces globally before you can configure a VLAN interface. • Physical Layer 3 interfaces • Layer 3 Ethernet subinterfaces • Layer 3 Ethernet port-channel interfaces • Layer 3 Ethernet port-channel subinterfaces • Tunnels • Management interfaces	IPv4 ACLs
VLAN ACL (VACL)	An ACL is a VACL when you use an access map to associate the ACL with an action and then apply the map to a VLAN.	IPv4 ACLs
VTY ACL	VTYs	IPv4 ACLs

Application Order

When the device processes a packet, it determines the forwarding path of the packet. The path determines which ACLs that the device applies to the traffic. The device applies the ACLs in the following order:

1. Port ACL
2. Ingress VACL
3. Ingress Router ACL
4. Egress Router ACL
5. Egress VACL

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 supervisor module creates ACL entries from the rules in the running configuration and sends those ACL entries to the applicable I/O module. Depending upon how you configure the ACL, there may be more ACL entries than rules, especially if you implement policy-based ACLs by using object groups when you configure rules.

You can create rules in ACLs and tYou can create rules in access-list configuration mode by using the **permit** or **deny** command. The device 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.

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.

Protocols

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

You can specify any protocol by the integer that represents the Internet protocol number.

Implicit Rules

IP and MAC ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the switch applies them to traffic when no other rules in an ACL match.

All IPv4 ACLs include the following implicit rule:

```
deny ip any any
```

This implicit rule ensures that the switch denies unmatched IP traffic.

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

All MAC ACLs include the following implicit rule:

```
deny any any protocol
```

This implicit rule ensures that the device denies the unmatched traffic, regardless of the protocol specified in the Layer 2 header of the traffic.

Additional Filtering Options

You can identify traffic by using additional options. IPv4 ACLs support the following additional filtering options:

- Layer 4 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

Sequence Numbers

The Cisco Nexus device supports sequence numbers for rules. Every rule that you enter receives a sequence number, either assigned by you or assigned automatically by the device. 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:

  ```
  switch(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:

  ```
  switch(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.

If you enter a rule without a sequence number, the device 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 device assigns the sequence number 235 to the new rule.

In addition, the device 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 Nexus device stores operator-operand couples in registers called logical operation units (LOUs) to perform operations (greater than, less than, not equal to, and range) on the TCP and UDP ports specified in an IP ACL.

Note The range operator is inclusive of boundary values.

These LOUs minimize the number of ternary content addressable memory (TCAM) entries needed to perform these operations. A maximum of two LOUs are allowed for each feature on an interface. For example an ingress RACL can use two LOUs, and a QoS feature can use two LOUs. If an ACL feature requires more than two arithmetic operations, the first two operations use LOUs, and the remaining access control entries (ACEs) get expanded.

The following guidelines determine when the device stores 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.

- Whether the operator-operand couple is applied to a source port or a destination port in the rule affects LOU usage. Identical couples are stored separately when one of the identical couples is applied to a source port and the other couple is applied to a destination port.

  For example, if a rule applies the operator-operand couple "gt 10" to a source port and another rule applies a "gt 10" couple to a destination port, both couples would also be stored in half an LOU, resulting in the use of one whole LOU. Any additional rules using a "gt 10" couple would not result in further LOU usage.

ACL TCAM Regions

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

The IPv4 TCAMs are single wide.

TCAM region sizes have the following guidelines and limitations:

- To revert to the default ACL TCAM size, use the **no hardware profile tcam region** command. You no longer need to use the **write erase command** and reload the switch.
- Depending on the Cisco Nexus device, each TCAM region might have a different minimum/maximum/aggregate size restriction.
- The default size of the ARPACL TCAM is zero. Before you use the ARP ACLs in a Control Policing Plane (CoPP) policy, you must set the size of this TCAM to a non-zero size.
- You must set the VACL and egress VLAN ACL (E-VACL) size to the same value.
- The total TCAM depth is 4000 entries shared between ingress and egress, which can be carved in 16 entries blocks.
- TCAM supports 256 statistic entries per ACL feature.
- 64 ACL L4OPs are supported, 32 in each direction.
- 2 L4OPs are supported per label in each direction. Each label can be shared across multiple interfaces for same ACL.
- After TCAM carving, you must reload the switch.

- All existing TCAMs cannot be set to size 0.
- By default, all IPv6 TCAMs are disabled (the TCAM size is set to 0).

Table 13: TCAM Sizes by ACL Region

TCAM ACL Region	Default Size	Minimum Size	Incremental Size
SUP (ingress)	112	48	16
PACL (ingress)	400	0	16
VACL (ingress), VACL (egress)	640 (ingress), 640 (egress)	0 (ingress), 0 (egress)	16
RACL (ingress)	1536	0	16
QOS (ingress), QOS (egress)	192 (ingress), 64 (egress)	16 (ingress), 64 (egress)	16
E-VACL (egress)	640	0	16
E-RACL (egress)	256	0	16
NAT	256	0	16

Licensing Requirements for ACLs

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for ACLs

IP ACLs have the following 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.

VACLs have the following prerequisite:

- Ensure that the IP ACL that you want to use in the VACL exists and is configured to filter traffic in the manner that you need for this application.

Guidelines and Limitations for ACLs

IP ACLs have the following configuration guidelines and limitations:

- We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources that are required by the configuration are available before committing them to the running configuration. This is especially useful for ACLs that include more than about 1000 rules.

- 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).
- To apply an IP ACL to a VLAN interface, you must have enabled VLAN interfaces globally.
- One VLAN access map can match only one IP ACL.
- An IP ACL can have multiple permit/deny ACEs.
- One VLAN can have only one access map applied.
- Egress RACLs and VACLs should not be applied in warp mode, and it is not supported.
- Egress ACLs cannot be applied to multicast traffic.
- Egress ACL Logging not supported on Cisco Nexus 3548 platform.
- Although Ingress RACLs on SVI are supported for multicast traffic, if the ACL defining the multicast group where the traffic must be sent to or sourced from includes the **log** keyword, the ingress RACL application on SVI is not supported.
- To match the Ingress RACL ACEs for multicast traffic on SVI, the ACE should include a match on the multicast DIP. Also, before installing these ACEs, you must enable the **RACL-bridging using the hardware profile tcam mcast racl-bridge** command.
- PACL cannot be applied in warp mode.
- The same ingress RACL on an SVI and on a Layer 3 interface cannot share TCAM resources, and they individually use up TCAM resources. However, they share ACL statistics resources. If the RACL TCAM is almost depleted before upgrade, RACL application may fail after upgrade. If this happens, you can carve the RACL TCAM.
- ARP ACLs are not supported on the Nexus 3500 platform.
- Ingress RACL applied to physical or logical Layer 3 interface is supported. For the ingress RACL to be applied to the Layer 3 SVI, you can use the *hardware profile tcam mcast racl-bridge* configuration as a workaround to match multicast traffic.
- Upgrade from Cisco NX-OS Release 7.0(3)I7(6) or below, Cisco NX-OS Release 9.3(1) to 9.3(2) or above with default lou threshold config will set lou threshold as 1.
- In the Cisco Nexus 3548 Series switches, RACL with ACL log option will not take into effect as the sup-redirect ACLs will have higher priority for the traffic destined to SUP.

Below are the guidelines and limitations for wide IFACL:

- Same egress ports on two different flows with different **SET_VLAN id** is not possible if the ingress match VLAN is same for both the flows.
- Wide flow IFACL Redirect Action is Supported only on Trunk Ports.
- No other ACL feature will be supported on flow-redirect ports except PACL. PACL entries (Wide flow or not) will be installed in FIBACL TCAM & not in ACL TCAM like normal PACL when PACL_WIDE TCAM region is carved.
- During port flap, entries will not be removed from TCAM. They will stay as is like other security ACLs.

- Port range match give in CLI will be expanded in value & mask for L4 ports before writing to TCAM and LOU hardware resources will not be used. No impact to user and no impact in terms of existing scale of flows.
- Only Redirect/Set-vlan/Strip-vlan & Drop Actions are Supported. No Support for PUNT Action.
- Log keyword is not supported for wide IFACL ACLs.
- Max 4000 redirect ACLs are supported irrespective of TCAM size.
- Max 4k ACEs with stats can be supported.
- VLAN Range allowed for match & set/strip: 1 – 4094.
- ACE Match on TCP flags are not supported.
- Before changing the TCAM configuration from **ifacl-wide** to **ifacl**, ensure that all wideflow ACL's configurations are removed from interfaces.
- If ingress packets have with same VLAN matching wideflow ACEs with and without VLAN match conditions along with strip_vlan, VLAN header gets stripped even for packets matching no strip-vlan ACE.

Default ACL Settings

The following table lists the default settings for IP ACLs parameters.

Table 14: Default IP ACLs Parameters

Parameters	Default
IP ACLs	No IP ACLs exist by default.
ACL rules	Implicit rules apply to all ACLs .
Object groups	No object groups exist by default.

The following table lists the default settings for VACL parameters.

Table 15: Default VACL Parameters

Parameters	Default
VACLs	No IP ACLs exist by default.
ACL rules	Implicit rules apply to all ACLs.

Configuring IP ACLs

Creating an IP ACL

You can create an IPv4 ACL on the switch and add rules to it.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip access-list** *name*
3. switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*
4. (Optional) switch(config-acl)# **statistics**
5. (Optional) switch# **show ip access-lists** *name*
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **ip access-list** *name*	Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.
Step 3	switch(config-acl)# [*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. For more information, see the *Command Reference* for the specific Cisco Nexus device.
Step 4	(Optional) switch(config-acl)# **statistics**	Specifies that the switch maintains global statistics for packets that matches the rules in the ACL.
Step 5	(Optional) switch# **show ip access-lists** *name*	Displays the IP ACL configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to create an IPv4 ACL:

```
switch# configure terminal
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# statistics
```

Changing an IP ACL

You can add and remove rules in an existing IPv4 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.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip access-list** *name*
3. switch(config)# **ip access-list** *name*
4. switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*
5. (Optional) switch(config-acl)# **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*}
6. (Optional) switch(config-acl)# [**no**] **statistics**
7. (Optional) switch#**show ip access-lists** *name*
8. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip access-list** *name*	Enters IP ACL configuration mode for the ACL that you specify by name.
Step 3	switch(config)# **ip access-list** *name*	Enters IP ACL configuration mode for the ACL that you specify by name.
Step 4	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*	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. For more information, see the *Command Reference* for your Cisco Nexus device.
Step 5	(Optional) switch(config-acl)# **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*}	Removes the rule that you specified from the IP ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Command Reference* for your Cisco Nexus device.
Step 6	(Optional) switch(config-acl)# [**no**] **statistics**	Specifies that the switch maintains global statistics for packets that match the rules in the ACL. The **no** option stops the switch from maintaining global statistics for the ACL.
Step 7	(Optional) switch#**show ip access-lists** *name*	Displays the IP ACL configuration.

	Command or Action	Purpose
Step 8	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Related Topics

Changing Sequence Numbers in an IP ACL, on page 187

Removing an IP ACL

You can remove an IP ACL from the switch.

Before you remove an IP ACL from the switch, be sure that you know whether the ACL is applied to an interface. The switch 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 switch considers the removed ACL to be empty.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no ip access-list** *name*
3. switch(config)# no **ip access-list** *name*
4. (Optional) switch# **show running-config**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no ip access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 3	switch(config)# no **ip access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 4	(Optional) switch# **show running-config**	Displays the ACL configuration. The removed IP ACL should not appear.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Changing Sequence Numbers in an IP ACL

You can change all the sequence numbers assigned to the rules in an IP ACL.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **resequence ip access-list** *name starting-sequence-number increment*

3. (Optional) switch#**show ip access-lists** *name*
4. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **resequence ip 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	(Optional) switch#**show ip access-lists** *name*	Displays the IP ACL configuration.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Applying an IP ACL to mgmt0

You can apply an IPv4 ACL to the management interface (mgmt0).

Before you begin

Ensure that the ACL that you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

SUMMARY STEPS

1. **configure terminal**
2. **interface mgmt** *port*
3. **ip access-group** *access-list* {**in** | **out**}
4. (Optional) **show running-config aclmgr**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface mgmt** *port* **Example:**	Enters configuration mode for the management interface.

	Command or Action	Purpose
	`switch(config)# interface mgmt0` `switch(config-if)#`	
Step 3	**ip access-group** *access-list* {**in** | **out**} **Example:** `switch(config-if)#ip access-group acl-120 out`	Applies an IPv4 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 4	(Optional) **show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	Displays the ACL configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

- Creating an IP ACL

Applying an IP ACL as a Port ACL

You can apply an IPv4 ACL to a physical Ethernet interface or a PortChannel. ACLs applied to these interface types are considered port ACLs.

Note Some configuration parameters when applied to an PortChannel are not reflected on the configuration of the member ports.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {**ethernet** [*chassis*/]*slot*/*port* | **port-channel** *channel-number*}
3. switch(config-if)# **ip port access-group** *access-list* **in**
4. (Optional) switch# **show running-config**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** {**ethernet** [*chassis*/]*slot*/*port* | **port-channel** *channel-number*}	Enters interface configuration mode for the specified interface.

	Command or Action	Purpose
Step 3	switch(config-if)# **ip port access-group** *access-list* **in**	Applies an IPv4 ACL to the interface or PortChannel. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.
Step 4	(Optional) switch# **show running-config**	Displays the ACL configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Applying an IP ACL as a Router ACL

You can apply an IPv4 ACL to any of the following types of interfaces:

- Physical Layer 3 interfaces and subinterfaces
- Layer 3 Ethernet port-channel interfaces and subinterfaces
- VLAN interfaces
- Tunnels
- Management interfaces

ACLs applied to these interface types are considered router ACLs.

Note Logical operation units (LOUs) are not available for router ACLs applied in the out direction. If an IPv4 ACL is applied as a router ACL in the out direction, access control entries (ACEs) that contain logical operators for TCP/UDP port numbers are expanded internally to multiple ACEs and might require more TCAM entries when compared to the same ACL applied in the in direction.

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.

SUMMARY STEPS

1. switch# **configure terminal**
2. Enter one of the following commands:
   - switch(config)# **interface ethernet** *slot*/*port*[**.** *number*]
   - switch(config)# **interface port-channel** *channel-number*[**.** *number*]
   - switch(config)# **interface tunnel** *tunnel-number*
   - switch(config)# **interface vlan** *vlan-ID*
   - switch(config)# **interface mgmt** *port*
3. switch(config-if)# **ip access-group** *access-list* {**in** | **out**}
4. (Optional) switch(config-if)# **show running-config aclmgr**
5. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	Enter one of the following commands: • switch(config)# **interface ethernet** *slot*/*port*[**.** *number*] • switch(config)# **interface port-channel** *channel-number*[**.** *number*] • switch(config)# **interface tunnel** *tunnel-number* • switch(config)# **interface vlan** *vlan-ID* • switch(config)# **interface mgmt** *port*	Enters configuration mode for the interface type that you specified.
Step 3	switch(config-if)# **ip access-group** *access-list* {**in** | **out**}	Applies an IPv4 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 4	(Optional) switch(config-if)# **show running-config aclmgr**	Displays the ACL configuration.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Verifying IP ACL Configurations

To display IP ACL configuration information, perform one of the following tasks:

Procedure

- switch# **show running-config**

  Displays ACL configuration, including IP ACL configuration and interfaces that IP ACLs are applied to.

- switch# **show running-config interface**

  Displays the configuration of an interface to which you have applied an ACL.

- switch# **show running-config aclmgr**

  Displays ACL configurations and the interfaces the ACLs are applied to.

Example

For detailed information about the fields in the output from these commands, refer to the *Command Reference* for your Cisco Nexus device.

Monitoring and Clearing IP ACL Statistics

Use the **show ip access-lists** command to display statistics about an IP ACL, including the number of packets that have matched each rule. For detailed information about the fields in the output from this command, see the *Command Reference* for your Cisco Nexus device.

Note The mac access-list is applicable to non-IPv4 traffic only.

Procedure

- switch# **show ip access-lists** *name*

  Displays IP ACL configuration. If the IP ACL includes the **statistics** command, then the **show ip access-lists** command output includes the number of packets that have matched each rule.

- switch#**show ip access-lists** *name*

  Displays IP ACL configuration. If the IP ACL includes the **statistics** command, then the **show ip access-lists** command output includes the number of packets that have matched each rule.

- switch# **clear access-list counters** [*access-list-name*]

  Clears statistics for all IP ACLs or for a specific IP ACL.

- switch# **clear ip access-list counters** [*access-list-name*]

  Clears statistics for all IP ACLs or for a specific IP ACL.

Information About VLAN ACLs

A VLAN ACL (VACL) is one application of an IP ACL. You can configure VACLs to apply to all packets that are bridged within a VLAN. VACLs are used strictly for security packet filtering. VACLs are not defined by direction (ingress or egress).

VACLs and Access Maps

VACLs use access maps to link an IP ACL to an action. The switch takes the configured action on packets that are permitted by the VACL.

VACLs and Actions

In access map configuration mode, you use the **action** command to specify one of the following actions:

- Forward—Sends the traffic to the destination determined by normal operation of the switch.
- Drop—Drops the traffic.

Statistics

The Cisco Nexus device can maintain global statistics for each rule in a VACL. If a VACL is applied to multiple VLANs, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that VACL is applied.

Note The Cisco Nexus device does not support interface-level VACL statistics.

For each VLAN access map that you configure, you can specify whether the switch maintains statistics for that VACL. This allows you to turn VACL statistics on or off as needed to monitor traffic filtered by a VACL or to help troubleshoot VLAN access-map configuration.

Configuring VACLs

Creating or Changing a VACL

You can create or change a VACL. Creating a VACL includes creating an access map that associates an IP ACL with an action to be applied to the matching traffic.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vlan access-map** *map-name*
3. switch(config-access-map)# **match ip address** *ip-access-list*
4. switch(config-access-map)# **action** {**drop** | **forward**}
5. (Optional) switch(config-access-map)# [**no**] **statistics**
6. (Optional) switch(config-access-map)# **show running-config**
7. (Optional) switch(config-access-map)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan access-map** *map-name*	Enters access map configuration mode for the access map specified.
Step 3	switch(config-access-map)# **match ip address** *ip-access-list*	Specifies an IPv4 ACL for the map.
Step 4	switch(config-access-map)# **action** {**drop** | **forward**}	Specifies the action that the switch applies to traffic that matches the ACL.
Step 5	(Optional) switch(config-access-map)# [**no**] **statistics**	Specifies that the switch maintains global statistics for packets matching the rules in the VACL. The **no** option stops the switch from maintaining global statistics for the VACL.
Step 6	(Optional) switch(config-access-map)# **show running-config**	Displays the ACL configuration.
Step 7	(Optional) switch(config-access-map)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Removing a VACL

You can remove a VACL, which means that you will delete the VLAN access map.

Be sure that you know whether the VACL is applied to a VLAN. The switch allows you to remove VACLs that are current applied. Removing a VACL does not affect the configuration of VLANs where you have applied the VACL. Instead, the switch considers the removed VACL to be empty.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no vlan access-map** *map-name*
3. (Optional) switch(config)# **show running-config**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no vlan access-map** *map-name*	Removes the VLAN access map configuration for the specified access map.
Step 3	(Optional) switch(config)# **show running-config**	Displays ACL configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Applying a VACL to a VLAN

You can apply a VACL to a VLAN.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **vlan filter** *map-name* **vlan-list** *list*
3. (Optional) switch(config)# **show running-config**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **vlan filter** *map-name* **vlan-list** *list*	Applies the VACL to the VLANs by the list that you specified. The **no** option unapplies the VACL. The **vlan-list** command can specify a list of up to 32 VLANs, but multiple **vlan-list** commands can be configured to cover more than 32 VLANs.

	Command or Action	Purpose
Step 3	(Optional) switch(config)# **show running-config**	Displays ACL configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Verifying VACL Configuration

To display VACL configuration information, perform one of the following tasks:

Procedure

- switch# **show running-config aclmgr**

  Displays ACL configuration, including VACL-related configuration.

- switch# **show vlan filter**

  Displays information about VACLs that are applied to a VLAN.

- switch# **show vlan access-map**

  Displays information about VLAN access maps.

Displaying and Clearing VACL Statistics

To display or clear VACL statistics, perform one of the following tasks:

Procedure

- switch# **show vlan access-list**

  Displays VACL configuration. If the VLAN access-map includes the **statistics** command, then the **show vlan access-list** command output includes the number of packets that have matched each rule.

- switch# **clear vlan access-list counters**

  Clears statistics for all VACLs or for a specific VACL.

Configuration Examples for VACL

The following example shows how to configure a VACL to forward traffic permitted by an IP ACL named acl-ip-01 and how to apply the VACL to VLANs 50 through 82:

```
switch# configure terminal
switch(config)# vlan access-map acl-ip-map
switch(config-access-map)# match ip address acl-ip-01
switch(config-access-map)# action forward
switch(config-access-map)# exit
 switch(config)# vlan filter acl-ip-map vlan-list 50-82
```

Configuring ACL TCAM Region Sizes

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

SUMMARY STEPS

1. **configure terminal**
2. **hardware profile tcam region** {**arpacl** | **e-racl**} | **ifacl** | **nat** | **qos**} |**qoslbl** | **racl**} | **vacl** } *tcam_size*
3. **copy running-config startup-config**
4. switch(config)# **show hardware profile tcam region**
5. switch(config)# **reload**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**hardware profile tcam region** {**arpacl** | **e-racl**} | **ifacl** | **nat** | **qos**} |**qoslbl** | **racl**} | **vacl** } *tcam_size*	Changes the ACL TCAM region size. • **arpacl**—Configures the size of the Address Resolution Protocol (ARP) ACL (ARPACL) TCAM region. • **e-racl**—Configures the size of the egress router ACL (ERACL) TCAM region. • **e-vacl**—Configures the size of the egress VLAN ACL (EVACL) TCAM region. • **ifacl**—Configures the size of the interface ACL (ifacl) TCAM region. The maximum number of entries is 1500. • **nat**—Configures the size of the NAT TCAM region. • **qos**—Configures the size of the quality of service (QoS) TCAM region. • **qoslbl**—Configures the size of the QoS Label (qoslbl) TCAM region. • **racl**—Configures the size of the router ACL (RACL) TCAM region. • **vacl**—Configures the size of the VLAN ACL (VACL) TCAM region. • *tcam_size*—TCAM size. The range is from 0 to 2,14,74, 83, 647 entries.

	Command or Action	Purpose
		Note **vacl** and **e-vacl** TCAM regions should be set to the same size.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 4	switch(config)# **show hardware profile tcam region** **Example:** `switch(config)# show hardware profile tcam region`	Displays the TCAM sizes that will be applicable on the next reload of the switch.
Step 5	switch(config)# **reload** **Example:** `switch(config)# reload`	Copies the running configuration to the startup configuration. **Note** The new size values are effective only upon the next reload after saving the **copy running-config to startup-config**.

Example

The following example shows how to change the size of the RACL TCAM region:

```
switch(config)# hardware profile tcam region racl 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'

switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

The following example shows how to configure the TCAM VLAN ACLs on a switch:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# hardware profile tcam region vacl 512
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'

switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

This example shows how to display the TCAM region sizes to verify your changes:

```
switch(config)# show hardware profile tcam region
         sup size =   16
        vacl size =  640
       ifacl size =  496
         qos size =  256
       rbacl size =    0
        span size =    0
        racl size = 1536
```

```
      e-racl size =  256
      e-vacl size =  640
      qoslbl size =    0
      arpacl size =    0
```

This example shows how to determine TCAM utilization for particular region. There are 5 RACL entries in this example:

```
switch(config)# show system internal aclqos platform mtc info tcam 0 region racl
         racl TCAM configuration for asic id 0:
[        sup tcam]: range     0 -   47
[       vacl tcam]: range   512 - 1087
[      ifacl tcam]: range   112 -  511
[        qos tcam]: range  3712 - 3903
[      rbacl tcam]: range     0 -    0
[       span tcam]: range     0 -    0
[       racl tcam]: range  1984 - 3455 *
[     e-racl tcam]: range  3456 - 3711
[     e-vacl tcam]: range  1088 - 1727
[     qoslbl tcam]: range     0 -    0
[       ipsg tcam]: range     0 -    0
[     arpacl tcam]: range     0 -    0
[  ipv6-racl tcam]: range     0 -    0
[ipv6-e-racl tcam]: range     0 -    0
[   ipv6-sup tcam]: range     0 -    0
[   ipv6-qos tcam]: range     0 -    0
[        nat tcam]: range  1728 - 1983
[      e-qos tcam]: range  3904 - 3967
[        pbr tcam]: range     0 -    0
[   ipv6-pbr tcam]: range     0 -    0
[       copp tcam]: range    48 -  111

    TCAM [racl tcam]: [v:1, size:1472, start:1984 end:3455]
    In use tcam entries: 5
        3451-3455
    Link Local Entries:
    nat size =  256
```

Reverting to the Default TCAM Region Sizes

SUMMARY STEPS

1. **configure terminal**
2. switch(config)# **no hardware profile tcam region** {**arpacl** | **e-racl**} | **ifacl** | **nat** | **qos**} |**qoslbl** | **racl**} | **vacl** } *tcam_size*
3. (Optional) **copy running-config startup-config**
4. switch(config)# **reload**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **no hardware profile tcam region** {**arpacl** | **e-racl**} | **ifacl** | **nat** | **qos**} |**qoslbl** | **racl**} | **vacl** } *tcam_size*	Reverts the configuration to the default ACL TCAM size.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 4	switch(config)# **reload**	Reloads the switch.

Example

The following example shows how to revert to the default RACL TCAM region sizes:

```
switch(config)# no hardware profile tcam region racl 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'

switch(config)# copy running-configur startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

Configuring ACLs on Virtual Terminal Lines

To restrict incoming and outgoing connections for IPv4 between a Virtual Terminal (VTY) line and the addresses in an access list, use the **access-class** command in line configuration mode. To remove access restrictions, use the **no** form of this command.

Follow these guidelines when configuring ACLs on VTY lines:

- Set identical restrictions on all VTY lines because a user can connect to any of them.
- Statistics per entry is not supported for ACLs on VTY lines.

Before you begin

Be sure that the ACL that you want to apply exists and is configured to filter traffic for this application.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **line vty**
3. switch(config-line)# **access-class access-list-number {in | out}**
4. (Optional) switch(config-line)# **no access-class access-list-number {in | out}**
5. switch(config-line)# **exit**
6. (Optional) switch# **show running-config aclmgr**
7. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# **line vty**
	Example:
	`switch(config)# line vty`
	`switch(config-line)#`
		Enters line configuration mode.

Step 3	switch(config-line)# **access-class access-list-number {in | out}**
	Example:
	`switch(config-line)# access-class ozi2 in`
	`switch(config-line)#access-class ozi3 out`
	`switch(config)#`
		Specifies inbound or outbound access restrictions.

Step 4	(Optional) switch(config-line)# **no access-class access-list-number {in | out}**
	Example:
	`switch(config-line)# no access-class ozi2 in`
	`switch(config-line)# no access-class ozi3 out`
	`switch(config)#`
		Removes inbound or outbound access restrictions.

Step 5	switch(config-line)# **exit**
	Example:
	`switch(config-line)# exit`
	`switch#`
		Exits line configuration mode.

Step 6	(Optional) switch# **show running-config aclmgr** **Example:** `switch# show running-config aclmgr`	Displays the running configuration of the ACLs on the switch.
Step 7	(Optional) switch# **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

The following example shows how to apply the access-class ozi2 command to the in-direction of the vty line.

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# line vty
switch(config-line)# access-class ozi2 in
switch(config-line)# exit
switch#
```

Verifying ACLs on VTY Lines

To display the ACL configurations on VTY lines, perform one of the following tasks:

Command	Purpose
show running-config aclmgr	Displays the running configuration of the ACLs configured on the switch.
show users	Displays the users that are connected.
show access-lists *access-list-name*	Display the statistics per entry.

Configuration Examples for ACLs on VTY Lines

The following example shows the connected users on the console line (ttyS0) and the VTY lines (pts/0 and pts/1).

```
switch# show users
NAME     LINE         TIME         IDLE          PID COMMENT
admin    ttyS0        Aug 27 20:45   .         14425 *
admin    pts/0        Aug 27 20:06 00:46       14176 (172.18.217.82) session=ssh
admin    pts/1        Aug 27 20:52   .         14584 (10.55.144.118)
```

The following example shows how to allow vty connections to all IPv4 hosts except 172.18.217.82 and how to deny vty connections to any IPv4 host except 10.55.144.118, 172.18.217.79, 172.18.217.82, 172.18.217.92:

```
switch# show running-config aclmgr
!Time: Fri Aug 27 22:01:09 2010
version 5.0(2)N1(1)
ip access-list ozi
  10 deny ip 172.18.217.82/32 any
  20 permit ip any any
ip access-list ozi2
  10 permit ip 10.55.144.118/32 any
  20 permit ip 172.18.217.79/32 any
  30 permit ip 172.18.217.82/32 any
  40 permit ip 172.18.217.92/32 any

line vty
  access-class ozi in
  access-class ozi2 out
```

The following example shows how to configure the IP access list by enabling per-entry statistics for the ACL:

```
switch# configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
switch(config)# ip access-list ozi2
switch(config-acl)# statistics per-entry
switch(config-acl)# deny tcp 172.18.217.83/32 any
switch(config-acl)# exit

switch(config)# ip access-list ozi
switch(config-acl)# statistics per-entry
switch(config-acl)# permit ip 172.18.217.20/24 any
switch(config-acl)# exit
switch#
```

The following example shows how to apply the ACLs on VTY in and out directions:

```
switch(config)# line vty
switch(config-line)# ip access-class ozi in
switch(config-line)# access-class ozi2 out
switch(config-line)# exit
switch#
```

The following example shows how to remove the access restrictions on the VTY line:

```
switch# configure terminal
Enter configuration commands, one per line. End
with CNTL/Z.
switch(config)# line vty
switch(config-line)# no access-class ozi2 in
switch(config-line)# no ip access-class ozi2 in
switch(config-line)# exit
switch#
```

Configuring Wideflow IFACL Redirect on IP Port ACLs

Until Cisco NX-OS Release 10.3(2)F the Tap Aggregation functionality in Cisco Nexus 3548 series switches are supported using Openflow. For more information, see Configuring the Cisco OpenFlow Agent.

Beginning with Cisco NX-OS Release 10.3(3)F Openflow is not supported on Cisco Nexus 3548 series switches. To cater all openflow or Tap aggregation functionalities, ACL redirect with wideflow feature is introduced with additional new match command options (srcmac, dstmac & vlan) and new actions (setvlan, strip-vlan).

Beginning with Cisco NX-OS Release 10.3(3)F, new CLI options are added in conjunction with keyword **wideflow** in the existing IP ACL CLI. Keyword **wideflow** protects the new CLI options and it is only enabled for Cisco Nexus 3548 switches.

Before you begin

To enable wideflow new command options, **IFACL-WIDE TCAM** needs to be configured. This requires a copy running configuration to the startup configuration and reload of the device. Hardware profile forwarding-mode will change from normal to flow-redirect post reload. For more information, see Achieving OpenFlow Functionality.

Note

- While changing from IFACL to IFACL-WIDE TCAM, make sure that all the existing IP access-lists are removed from interfaces and global configurations.
- After changing to IFACL-WIDE TCAM, legacy ACL's cannot be applied under interfaces.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip access-list** *name*
3. switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination* **redirect** *redirect-ports* **wideflow**
4. switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination* **redirect** *redirect-ports* **wideflow dstmac** *destination MAC address*

5. switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination* **redirect** *redirect-ports* **wideflow srcmac** *source MAC address*
6. switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination* **redirect** *redirect-ports* **wideflow vlan**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **ip access-list** *name*	Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.
Step 3	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination* **redirect** *redirect-ports* **wideflow**	Following are the options under wideflow. • **dstmac**—Configure the destination MAC Address. • **srcmac**—Configure the source MAC Address. • **vlan**—Configure the Vlan number.
Step 4	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination* **redirect** *redirect-ports* **wideflow dstmac** *destination MAC address*	Following are the options under **dstmac**. • **E.E.E**—Destination wildcard bits (Option 1). • **EE-EE-EE-EE-EE-EE**—Destination wildcard bits (Option 2). • **EE:EE:EE:EE:EE:EE**—Destination wildcard bits (Option 3). • **EEEE.EEEE.EEEE**—Destination wildcard bits (Option 4).
Step 5	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination* **redirect** *redirect-ports* **wideflow srcmac** *source MAC address*	Following are the options under **srcmac**. • **E.E.E**—Source MAC address (Option 1). • **EE-EE-EE-EE-EE-EE**—Source MAC address (Option 2). • **EE:EE:EE:EE:EE:EE**—Source MAC address (Option 3). • **EEEE.EEEE.EEEE**—Source MAC address (Option 4). • **any**—Any source address.
Step 6	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination* **redirect** *redirect-ports* **wideflow vlan**	Enter the Vlan number range from 0 to 4095.

Example

Following is the configuration example:

Step 1: If the switch is in openflow forwarding-mode, perform the following steps:

Note If the switch is in normal forwarding-mode, skip Step 1 and go to Step 2 directly.

- Remove all openflow configurations.
- Change hardware profile forwarding-mode to normal.
- Copy the running configuration to the startup configuration.
- Reload the switch.

```
switch#configure terminal
switch(config)# no openflow
switch(config)# no feature openflow
switch(config)# [optional] no hardware profile openflow forward-pdu
switch(config)# hardware profile forwarding-mode normal
switch(config)# copy r s
switch(config)# reload
```

Step 2: Upgrade to Cisco Nexus Release 10.3(3)F or later releases.

Step 3: After the switch is booted in Cisco Nexus Release 10.3(3)F or later releases, configure **TCAM for IFACL-WIDE** as follows:

```
switch# configure terminal
switch(config)# hardware profile tcam region ifacl 0
switch(config)# hardware profile tcam region ifacl-wide 4096
switch(config)# copy r s
switch(config)# reload
switch(config)# [optional] hardware profile flow-redirect forward-pdu
```

Following is the example for IP access-list configurations using redirect and wideflow commands:

```
switch# configure terminal
switch(config)# ip access-list ACL
switch(config-acl)# 10 permit ip host 1.1.1.1 host 1.1.1.2 dscp 52 redirect
Ethernet1/2,portchannel1 strip-vlan wideflow srcmac 00:16:3e:33:e1:84 0.0.0 dstmac
00:16:3e:4d:d6:dd 0.0.0 vlan 1000
switch(config-acl)# 20 permit icmp host 2.2.2.1 host 2.2.2.2 redirect
Ethernet1/34,portchannel2 wideflow
switch(config-acl)# 30 permit tcp host 3.3.3.1 host 3.3.3.2 dscp 28 redirect
Ethernet1/2,port-channel1 set-vlan 1002 wideflow srcmac 00:16:3e:12:e9:c4 0.0.0 dstmac
00:16:3e:0f:6a:48 0.0.0 vlan 1001
switch(config-acl)# 40 permit udp host 4.4.4.1 host 4.4.4.2 precedence 7 redirect
Ethernet1/2,port-channel1 wideflow srcmac 00:16:3e:07:aa:53 0.0.0 dstmac 00:16:3e:79:e4:a8
 0.0.0 vlan 1000
switch(config-acl)# 50 permit ethertype 0x0806 redirect Ethernet1/48 wideflow
```

Following is the example for applying IP ACL with redirect and wideflow commands under interface:

```
switch# configure terminal
switch(config)# interface Ethernet1/1
switch(config-if)# mode flow-redirect
switch(config-if)# ip port access-group ACL in
switch(config-if)# end
```

Configuring Redirect Action

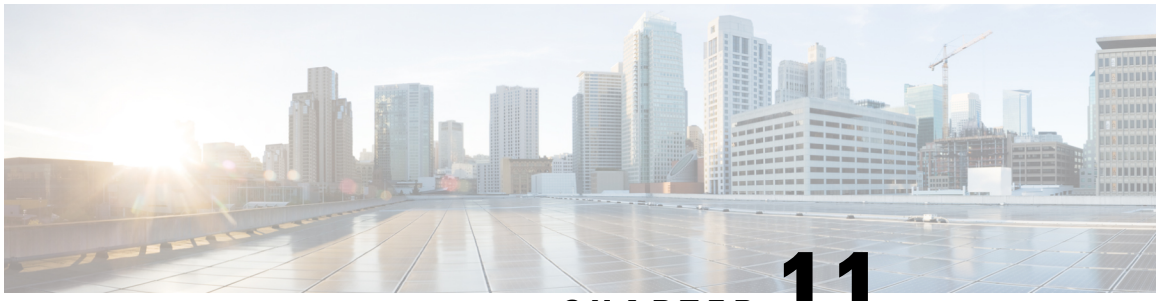
Redirect Action in CLI syntax must be present before **wideflow** keyword. Redirect Action Configuration is not accepted if **wideflow** keyword is missing. This check is performed at run time once user has entered the command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip access-list** *name*
3. switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination* **redirect** *redirect*
4. switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination* **redirect** *redirect*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **ip access-list** *name*	Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.
Step 3	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination* **redirect** *redirect*	Following are the options under **redirect**. • **redirect**—Redirect to interface(s). Syntax example: redirect Ethernet1/1,Ethernet1/2,port-channel1. • **wideflow**—Wide-flow options (mandatory).
Step 4	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination* **redirect** *redirect*	Following are the optional commands under redirect **redirect**. • **redirect**—Redirect to interface(s). Syntax example: redirect Ethernet1/1,Ethernet1/2,port-channel1. • **set-vlan**—Set vlan value for traffic egressing via redirect ports(s). • **strip-vlan**—Send vlan untagged packet from redirect port(s). • **wideflow**—Wide-flow options (mandatory).



CHAPTER 11

Configuring DHCP Snooping

This chapter includes the following sections:

About DHCP Snooping

DHCP snooping acts like a firewall between untrusted hosts and trusted DHCP servers. DHCP snooping performs the following activities:

- Validates DHCP messages received from untrusted sources and filters out invalid messages.
- Builds and maintains the DHCP snooping binding database, which contains information about untrusted hosts with leased IP addresses.
- Uses the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

DHCP snooping is enabled on a per-VLAN basis. By default, the feature is inactive on all VLANs. You can enable the feature on a single VLAN or a range of VLANs.

Feature Enabled and Globally Enabled

When you are configuring DHCP snooping, it is important that you understand the difference between enabling the DHCP snooping feature and globally enabling DHCP snooping.

Feature Enablement

The DHCP snooping feature is disabled by default. When the DHCP snooping feature is disabled, you cannot configure it or any of the features that depend on DHCP snooping. The commands to configure DHCP snooping and its dependent features are unavailable when DHCP snooping is disabled.

When you enable the DHCP snooping feature, the switch begins building and maintaining the DHCP snooping binding database. Features dependent on the DHCP snooping binding database can now make use of it and can therefore also be configured.

Enabling the DHCP snooping feature does not globally enable it. You must separately enable DHCP snooping globally.

Disabling the DHCP snooping feature removes all DHCP snooping configuration from the switch. If you want to disable DHCP snooping and preserve the configuration, globally disable DHCP snooping but do not disable the DHCP snooping feature.

Global Enablement

After DHCP snooping is enabled, DHCP snooping is globally disabled by default. Global enablement is a second level of enablement that allows you to have separate control of whether the switch is actively performing DHCP snooping that is independent from enabling the DHCP snooping binding database.

When you globally enable DHCP snooping, on each untrusted interface of VLANs that have DHCP snooping enabled, the switch begins validating DHCP messages that are received and used the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

When you globally disable DHCP snooping, the switch stops validating DHCP messages and validating subsequent requests from untrusted hosts. It also removes the DHCP snooping binding database. Globally disabling DHCP snooping does not remove any DHCP snooping configuration or the configuration of other features that are dependent upon the DHCP snooping feature.

Trusted and Untrusted Sources

You can configure whether DHCP snooping trusts traffic sources. An untrusted source might initiate traffic attacks or other hostile actions. To prevent such attacks, DHCP snooping filters messages from untrusted sources.

In an enterprise network, a trusted source is a switch that is under your administrative control. These switches include the switches, routers, and servers in the network. Any switch beyond the firewall or outside the network is an untrusted source. Generally, host ports are treated as untrusted sources.

In a service provider environment, any switch that is not in the service provider network is an untrusted source (such as a customer switch). Host ports are untrusted sources.

In a Cisco Nexus device, you indicate that a source is trusted by configuring the trust state of its connecting interface.

The default trust state of all interfaces is untrusted. You must configure DHCP server interfaces as trusted. You can also configure other interfaces as trusted if they connect to switches (such as switches or routers) inside your network. You usually do not configure host port interfaces as trusted.

Note For DHCP snooping to function properly, you must connect all DHCP servers to the switch through trusted interfaces.

DHCP Snooping Binding Database

Using information extracted from intercepted DHCP messages, DHCP snooping dynamically builds and maintains a database. The database contains an entry for each untrusted host with a leased IP address if the host is associated with a VLAN that has DHCP snooping enabled. The database does not contain entries for hosts that are connected through trusted interfaces.

Note The DHCP snooping binding database is also referred to as the DHCP snooping binding table.

DHCP snooping updates the database when the switch receives specific DHCP messages. For example, the feature adds an entry to the database when the switch receives a DHCPACK message from the server. The feature removes the entry in the database when the IP address lease expires or the switch receives a DHCPRELEASE message from the host.

Each entry in the DHCP snooping binding database includes the MAC address of the host, the leased IP address, the lease time, the binding type, and the VLAN number and interface information associated with the host.

You can remove entries from the binding database by using the **clear ip dhcp snooping binding** command.

Information About the DHCP Relay Agent

DHCP Relay Agent

You can configure the device to run a DHCP relay agent, which forwards DHCP packets between clients and servers. This feature is useful when clients and servers are not on the same physical subnet. Relay agents receive DHCP messages and then generate a new DHCP message to send out on another interface. The relay agent sets the gateway address (giaddr field of the DHCP packet) and, if configured, adds the relay agent information option (Option 82) in the packet and forwards it to the DHCP server. The reply from the server is forwarded back to the client after removing Option 82.

After you enable Option 82, the device uses the binary ifindex format by default. If needed, you can change the Option 82 setting to use an encoded string format instead.

Note When the device relays a DHCP request that already includes Option 82 information, the device forwards the request with the original Option 82 information without altering it.

VRF Support for the DHCP Relay Agent

You can configure the DHCP relay agent to forward DHCP broadcast messages from clients in a virtual routing and forwarding (VRF) instance to DHCP servers in a different VRF. By using a single DHCP server to provide DHCP support to clients in multiple VRFs, you can conserve IP addresses by using a single IP address pool rather than one for each VRF.

Enabling VRF support for the DHCP relay agent requires that you enable Option 82 for the DHCP relay agent.

If a DHCP request arrives on an interface that you have configured with a DHCP relay address and VRF information, and the address of the DHCP server belongs to a network on an interface that is a member of a different VRF, the device inserts Option 82 information in the request and forwards it to the DHCP server in the server VRF. The Option 82 information includes the following:

VPN identifier

Name of the VRF that the interface that receives the DHCP request is a member of.

Link selection

Subnet address of the interface that receives the DHCP request.

Server identifier override

IP address of the interface that receives the DHCP request.

Note The DHCP server must support the VPN identifier, link selection, and server identifier override options.

When the device receives the DHCP response message, it strips off the Option 82 information and forwards the response to the DHCP client in the client VRF.

DHCP Relay Binding Database

A relay binding is an entity that associates a DHCP or BOOTP client with a relay agent address and its subnet. Each relay binding stores the client MAC address, active relay agent address, active relay agent address mask, logical and physical interfaces to which the client is connected, giaddr retry count, and total retry count. The giaddr retry count is the number of request packets transmitted with that relay agent address, and the total retry count is the total number of request packets transmitted by the relay agent. One relay binding entry is maintained for each DHCP or BOOTP client.

Note When DHCP smart relay is enabled globally or at the interface level on any switch, the relay bindings on all switches should be synchronized with the vPC peer.

Prerequisites for DHCP Snooping

You should be familiar with DHCP before you configure DHCP snooping or the DHCP relay agent .

Guidelines and Limitations for DHCP Snooping

Consider the following guidelines and limitations when configuring DHCP snooping:

- The DHCP snooping database can store 2000 bindings.
- DHCP snooping is not active until you enable the feature, enable DHCP snooping globally, and enable DHCP snooping on at least one VLAN.

- Before globally enabling DHCP snooping on the switch, make sure that the switches that act as the DHCP server and the DHCP relay agent are configured and enabled.
- If a VLAN ACL (VACL) is configured on a VLAN that you are configuring with DHCP snooping, ensure that the VACL permits DHCP traffic between DHCP servers and DHCP hosts.
- DHCP snooping and DHCP relay feature are not supported on the same VLAN.

Default Settings for DHCP Snooping

This table lists the default settings for DHCP snooping parameters.

Table 16: Default DHCP Snooping Parameters

Parameters	Default
DHCP snooping feature	Disabled
DHCP snooping globally enabled	No
DHCP snooping VLAN	None
DHCP snooping Option 82 support	Disabled
DHCP snooping trust	Untrusted
VRF support for the DHCP relay agent	Disabled
DHCP relay agent	Disabled

Configuring DHCP Snooping

Minimum DHCP Snooping Configuration

Procedure

	Command or Action	Purpose
Step 1	Enable the DHCP snooping feature.	When the DHCP snooping feature is disabled, you cannot configure DHCP snooping. For details, see Enabling or Disabling the DHCP Snooping Feature, on page 212.
Step 2	Enable DHCP snooping globally.	For details, see Enabling or Disabling DHCP Snooping Globally, on page 213.
Step 3	Enable DHCP snooping on at least one VLAN.	By default, DHCP snooping is disabled on all VLANs.

	Command or Action	Purpose
		For details, see Enabling or Disabling DHCP Snooping on a VLAN, on page 213.
Step 4	Ensure that the DHCP server is connected to the switch using a trusted interface.	For details, see Configuring an Interface as Trusted or Untrusted, on page 217.

Enabling or Disabling the DHCP Snooping Feature

You can enable or disable the DHCP snooping feature on the switch. By default, DHCP snooping is disabled.

Before you begin

If you disable the DHCP snooping feature, all DHCP snooping configuration is lost. If you want to turn off DHCP snooping and preserve the DHCP snooping configuration, disable DHCP globally.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **feature dhcp**
3. (Optional) **show running-config dhcp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature dhcp** **Example:** `switch(config)# feature dhcp`	Enables the DHCP snooping feature. The **no** option disables the DHCP snooping feature and erases all DHCP snooping configuration.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling DHCP Snooping Globally

You can enable or disable the DHCP snooping globally on the switch. Globally disabling DHCP snooping stops the switch from performing any DHCP snooping or relaying DHCP messages but preserves DCHP snooping configuration.

Before you begin

Ensure that you have enabled the DHCP snooping feature. By default, DHCP snooping is globally disabled.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip dhcp snooping**
3. (Optional) **show running-config dhcp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp snooping** **Example:** `switch(config)# ip dhcp snooping`	Enables DHCP snooping globally. The **no** option disables DHCP snooping.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling DHCP Snooping on a VLAN

You can enable or disable DHCP snooping on one or more VLANs.

Before you begin

By default, DHCP snooping is disabled on all VLANs.

Ensure that DHCP snooping is enabled.

Note If a VACL is configured on a VLAN that you are configuring with DHCP snooping, ensure that the VACL permits DHCP traffic between DHCP servers and DHCP hosts.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip dhcp snooping vlan** *vlan-list*
3. (Optional) **show running-config dhcp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp snooping vlan** *vlan-list* **Example:** `switch(config)# ip dhcp snooping vlan 100,200,250-252`	Enables DHCP snooping on the VLANs specified by *vlan-list*. The **no** option disables DHCP snooping on the VLANs specified.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling Option 82 Data Insertion and Removal

You can enable or disable the insertion and removal of Option 82 information for DHCP packets forwarded without the use of the DHCP relay agent. By default, the device does not include Option 82 information in DHCP packets.

Note DHCP relay agent support for Option 82 is configured separately.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp snooping information option**
3. (Optional) [**no**] **ip dhcp snooping sub-option circuit-id format-type string format**
4. (Optional) **show running-config dhcp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp snooping information option** **Example:** `switch(config)# ip dhcp snooping information option`	Enables the insertion and removal of Option 82 information for DHCP packets. The **no** option disables the insertion and removal of Option 82 information.
Step 3	(Optional) [**no**] **ip dhcp snooping sub-option circuit-id format-type string format** **Example:** `switch(config)# ip dhcp snooping sub-option circuit-id format-type string format`	Configures Option 82 to use encoded string format of either ingress ifindex name or host name or a combination of both hostname and ifindex name. ["%h" for hostname, "%p" for ifindex and combination of %h and %p for both hostname and ifindex name.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling Option 82 User Defined Data Insertion and Removal

You can enable or disable the insertion and removal of Option 82 user-defined information for DHCP packets forwarded to server. The configuration is applicable per port and overrides option82 global configuration that uses encoded string format of ingress ifindex name. When DHCP Relay is configured on SVI, the user defined string from the ingress physical *ifindex* is appended to the DHCP Packet being relayed.

By default, the device does not include option 82 information in DHCP packets.

Note The user-defined option82 configuration applies to both DHCP Relay and DHCP Snooping.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp snooping information option**
3. **interface ethernet** *slot*/*port*
4. **ip dhcp option82 suboption circuit-id** *user-defined-circuit-id*
5. (Optional) **show ip dhcp option82 suboption info interface po5**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t**	Enters global configuration mode.
Step 2	[**no**] **ip dhcp snooping information option**	Enables the insertion and removal of Option 82 information for DHCP packets. The **no** option disables the insertion and removal of Option 82 information.
Step 3	**interface ethernet** *slot*/*port*	Enters interface configuration mode, where slot/port is the Layer 2 Ethernet ingress interface where you want to configure the option 82 string.
Step 4	**ip dhcp option82 suboption circuit-id** *user-defined-circuit-id* **Example:** switch(config-if)# ip dhcp option82 suboption circuit-id po5-option82-string	Enters user defined option82 string on port channel5. The string po5-option82-string is appended to the DHCP Packet being ingress on port channel 5. The same is configured on Ethernet interface.
Step 5	(Optional) **show ip dhcp option82 suboption info interface po5**	Displays the DHCP option 82 information and statistics.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Enabling or Disabling Strict DHCP Packet Validation

You can enable or disable the strict validation of DHCP packets by the DHCP snooping feature. By default, strict validation of DHCP packets is disabled.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip dhcp packet strict-validation**
3. (Optional) **show running-config dhcp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp packet strict-validation** **Example:** `switch(config)# ip dhcp packet strict-validation`	Enables the strict validation of DHCP packets by the DHCP snooping feature. The **no** option disables strict DHCP packet validation.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring an Interface as Trusted or Untrusted

You can configure whether an interface is a trusted or untrusted source of DHCP messages. You can configure DHCP trust on the following types of interfaces:

- Layer 2 Ethernet interfaces
- Layer 2 port-channel interfaces

Before you begin

By default, all interfaces are untrusted.

Ensure that DHCP snooping is enabled.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **interface ethernet** *port*/*slot*
   - **interface port-channel** *channel-number*
3. [**no**] **ip dhcp snooping trust**
4. (Optional) **show running-config dhcp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	• **interface ethernet** *port*/*slot*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		• Enters interface configuration mode, where *port* / *slot* is the Layer 2 Ethernet interface that you want to configure as trusted or untrusted for DHCP snooping.
		• Enters interface configuration mode, where *port* / *slot* is the Layer 2 port-channel interface that you want to configure as trusted or untrusted for DHCP snooping.

Step 3	[**no**] **ip dhcp snooping trust** **Example:** `switch(config-if)# ip dhcp snooping trust`	Configures the interface as a trusted interface for DHCP snooping. The **no** option configures the port as an untrusted interface.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Shows the DHCP snooping configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling the DHCP Relay Agent

You can enable or disable the DHCP relay agent. By default, the DHCP relay agent is enabled.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp relay**
3. (Optional) **show ip dhcp relay**
4. (Optional) **show running-config dhcp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp relay** **Example:** `switch(config)# ip dhcp relay`	Enables the DHCP relay agent. The **no** option disables the relay agent.
Step 3	(Optional) **show ip dhcp relay** **Example:** `switch(config)# show ip dhcp relay`	Displays the DHCP relay configuration.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling Option 82 for the DHCP Relay Agent

You can enable or disable the device to insert and remove Option 82 information on DHCP packets forwarded by the relay agent.

By default, the DHCP relay agent does not include Option 82 information in DHCP packets.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip dhcp relay**
3. [**no**] **ip dhcp relay information option**
4. (Optional) **show ip dhcp relay**
5. (Optional) **show running-config dhcp**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	[**no**] **ip dhcp relay** **Example:** `switch(config)# ip dhcp relay`	Enables the DHCP relay feature. The **no** option disables this behavior.
Step 3	[**no**] **ip dhcp relay information option** **Example:** `switch(config)# ip dhcp relay information option`	Enables the DHCP relay agent to insert and remove Option 82 information on the packets that it forwards. The Option 82 information is in binary ifindex format by default. The **no** option disables this behavior.
Step 4	(Optional) **show ip dhcp relay** **Example:** `switch(config)# show ip dhcp relay`	Displays the DHCP relay configuration.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Enabling or Disabling VRF Support for the DHCP Relay Agent

You can configure the device to support the relaying of DHCP requests that arrive on an interface in one VRF to a DHCP server in a different VRF instance.

Before you begin

You must enable Option 82 for the DHCP relay agent.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp relay information option vpn**
3. [**no**] **ip dhcp relay sub-option type cisco**
4. (Optional) **show ip dhcp relay**
5. (Optional) **show running-config dhcp**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# config t` `switch(config)#`	
Step 2	[**no**] **ip dhcp relay information option vpn** **Example:** `switch(config)# ip dhcp relay information option vpn`	Enables VRF support for the DHCP relay agent. The **no** option disables this behavior.
Step 3	[**no**] **ip dhcp relay sub-option type cisco** **Example:** `switch(config)# ip dhcp relay sub-option type cisco`	Enables DHCP to use Cisco proprietary numbers 150, 152, and 151 when filling the link selection, server ID override, and VRF name/VPN ID relay agent Option 82 suboptions. The **no** option causes DHCP to use RFC numbers 5, 11, and 151 for the link selection, server ID override, and VRF name/VPN ID suboptions.
Step 4	(Optional) **show ip dhcp relay** **Example:** `switch(config)# show ip dhcp relay`	Displays the DHCP relay configuration.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling Subnet Broadcast Support for the DHCP Relay Agent on a Layer 3 Interface

You can configure the device to support the relaying of DHCP packets from clients to a subnet broadcast IP address. When this feature is enabled, the VLAN ACLs (VACLs) accept IP broadcast packets and all subnet broadcast (primary subnet broadcast as well as secondary subnet broadcast) packets.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCP relay agent is enabled.

SUMMARY STEPS

1. **config t**
2. **interface** *interface slot/port*
3. [**no**] **ip dhcp relay subnet-broadcast**
4. **exit**
5. **exit**

6. (Optional) **show ip dhcp relay**
7. (Optional) **show running-config dhcp**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *interface slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Enters interface configuration mode, where *slot*/*port* is the interface for which you want to enable or disable subnet broadcast support for the DHCP relay agent.

Step 3	[**no**] **ip dhcp relay subnet-broadcast** **Example:** `switch(config-if)# ip dhcp relay subnet-broadcast`	Enables subnet broadcast support for the DHCP relay agent. The **no** option disables this behavior.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 6	(Optional) **show ip dhcp relay** **Example:** `switch# show ip dhcp relay`	Displays the DHCP relay configuration.
Step 7	(Optional) **show running-config dhcp** **Example:** `switch# show running-config dhcp`	Displays the DHCP configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring DHCP Server Addresses on an Interface

You can configure DHCP server IP addresses on an interface. When an inbound DHCP BOOTREQUEST packet arrives on the interface, the relay agent forwards the packet to all DHCP server IP addresses specified. The relay agent forwards replies from all DHCP servers to the host that sent the request.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCP server is correctly configured.

Determine the IP address for each DHCP server that you want to configure on the interface.

If the DHCP server is in a different VRF instance than the interface, ensure that you have enabled VRF support.

Note If an ingress router ACL is configured on an interface that you are configuring with a DHCP server address, ensure that the router ACL permits DHCP traffic between DHCP servers and DHCP hosts.

SUMMARY STEPS

1. **config t**
2. Do one of the following options:
   - **interface ethernet** *slot*/*port*[**.** *number*]
   - **interface vlan** *vlan-id*
   - **interface port-channel** *channel-id*[.*subchannel-id*]
3. **ip dhcp relay address** *IP-address* [**use-vrf** *vrf-name*]
4. (Optional) **show ip dhcp relay address**
5. (Optional) **show running-config dhcp**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Do one of the following options:
	• **interface ethernet** *slot*/*port*[**.** *number*]
	• **interface vlan** *vlan-id*
	• **interface port-channel** *channel-id*[.*subchannel-id*]
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		• Enters interface configuration mode, where *slot*/*port* is the physical Ethernet interface that you want to configure with a DHCP server IP address. If you want to configure a subinterface, include the *number* argument to specify the subinterface number.
		• Enters interface configuration mode, where *vlan-id* is the ID of the VLAN that you want to configure with a DHCP server IP address.

	Command or Action	Purpose
		• Enters interface configuration mode, where *channel-id* is the ID of the port channel that you want to configure with a DHCP server IP address. If you want to configure a subchannel, include the *subchannel-id* argument to specify the subchannel ID.
Step 3	**ip dhcp relay address** *IP-address* [**use-vrf** *vrf-name*] **Example:** `switch(config-if)# ip dhcp relay address 10.132.7.120 use-vrf red`	Configures an IP address for a DHCP server to which the relay agent forwards BOOTREQUEST packets received on this interface. To configure more than one IP address, use the **ip dhcp relay address** command once per address.
Step 4	(Optional) **show ip dhcp relay address** **Example:** `switch(config-if)# show ip dhcp relay address`	Displays all the configured DHCP server addresses.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating a DHCP Static Binding

You can create a static DHCP source binding to a Layer 2 interface.

Before you begin

Ensure that you have enabled the DHCP snooping feature.

SUMMARY STEPS

1. **configure terminal**
2. **ip source binding** *IP-address MAC-address* **vlan** *vlan-id* {**interface ethernet** *slot/port* | **port-channel** *channel-no*}
3. (Optional) **show ip dhcp snooping binding**
4. (Optional) **show ip dhcp snooping binding dynamic**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**ip source binding** *IP-address MAC-address* **vlan** *vlan-id* {**interface ethernet** *slot/port* | **port-channel** *channel-no*} **Example:** `switch(config)# ip source binding 10.5.22.7 001f.28bd.0013 vlan 100 interface ethernet 2/3`	Binds the static source address to the Layer 2 Ethernet interface.
Step 3	(Optional) **show ip dhcp snooping binding** **Example:** `switch(config)#  ip dhcp snooping binding`	Shows the DHCP snooping static and dynamic bindings.
Step 4	(Optional) **show ip dhcp snooping binding dynamic** **Example:** `switch(config)#  ip dhcp snooping binding dynamic`	Shows the DHCP snooping dynamic bindings.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

The following example shows how to create a static IP source entry associated with VLAN 100 on Ethernet interface 2/3:

```
switch# configure terminal
switch(config)# ip source binding 10.5.22.7 001f.28bd.0013 vlan 100 interface ethernet 2/3
switch(config)#
```

Verifying the DHCP Snooping Configuration

To display DHCP snooping configuration information, perform one of the following tasks. For detailed information about the fields in the output from these commands, see the System Management Configuration Guide for your Cisco Nexus device.

Command	Purpose
show running-config dhcp	Displays the DHCP snooping configuration.

Command	Purpose
show ip dhcp relay	Displays the DHCP relay configuration.
show ip dhcp snooping	Displays general information about DHCP snooping.

Displaying DHCP Bindings

Use the **show ip dhcp snooping binding** command to display the DHCP static and dynamic binding table. Use the **show ip dhcp snooping binding dynamic** to display the DHCP dynamic binding table.

For detailed information about the fields in the output from this command, see the *System Management Configuration Guide* for your Cisco Nexus device.

This example shows how to create a static DHCP binding and then verify the binding using the **show ip dhcp snooping binding** command.

```
switch# configuration terminal
switch(config)# ip source binding 10.20.30.40 0000.1111.2222 vlan 400 interface port-channel
 500

switch(config)# show ip dhcp snooping binding
MacAddress         IpAddress        LeaseSec  Type        VLAN  Interface
-----------------  ---------------  --------  ----------  ----  -------------
00:00:11:11:22:22  10.20.30.40      infinite  static      400   port-channel500
```

Clearing the DHCP Snooping Binding Database

You can remove entries from the DHCP snooping binding database, including a single entry, all entries associated with an interface, or all entries in the database.

Before you begin

Ensure that DHCP snooping is enabled.

SUMMARY STEPS

1. (Optional) **clear ip dhcp snooping binding**
2. (Optional) **clear ip dhcp snooping binding interface ethernet** *slot*/*port*[.*subinterface-number*]
3. (Optional) **clear ip dhcp snooping binding interface port-channel** *channel-number*[.*subchannel-number*]
4. (Optional) **clear ip dhcp snooping binding vlan** *vlan-id* **mac** *mac-address* **ip** *ip-address* **interface** {**ethernet** *slot*/*port*[.*subinterface-number* | **port-channel** *channel-number*[.*subchannel-number*] }
5. (Optional) **show ip dhcp snooping binding**

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **clear ip dhcp snooping binding** **Example:** `switch# clear ip dhcp snooping binding`	Clears all entries from the DHCP snooping binding database.
Step 2	(Optional) **clear ip dhcp snooping binding interface ethernet** *slot*/*port*[.*subinterface-number*] **Example:** `switch# clear ip dhcp snooping binding interface ethernet 1/4`	Clears entries associated with a specific Ethernet interface from the DHCP snooping binding database.
Step 3	(Optional) **clear ip dhcp snooping binding interface port-channel** *channel-number*[.*subchannel-number*] **Example:** `switch# clear ip dhcp snooping binding interface port-channel 72`	Clears entries associated with a specific port-channel interface from the DHCP snooping binding database.
Step 4	(Optional) **clear ip dhcp snooping binding vlan** *vlan-id* **mac** *mac-address* **ip** *ip-address* **interface** {**ethernet** *slot*/*port*[.*subinterface-number* | **port-channel** *channel-number*[.*subchannel-number*] } **Example:** `switch# clear ip dhcp snooping binding vlan 23 mac 0060.3aeb.54f0 ip 10.34.54.9 interface ethernet 2/11`	Clears a single, specific entry from the DHCP snooping binding database.
Step 5	(Optional) **show ip dhcp snooping binding** **Example:** `switch# show ip dhcp snooping binding`	Displays the DHCP snooping binding database.

Clearing DHCP Relay Statistics

Use the **clear ip dhcp relay statistics** command to clear the global DHCP relay statistics.

Use the **clear ip dhcp relay statistics interface** *interface* command to clear the DHCP relay statistics for a particular interface.

Use the **clear ip dhcp relay statistics interface** *interface* **serverip** *ip-address* [**use-vrf** *vrf-name*] command to clear the DHCP relay statistics at the server level for a particular interface.

Monitoring DHCP

Use the **show ip dhcp snooping statistics** command to monitor DHCP snooping.

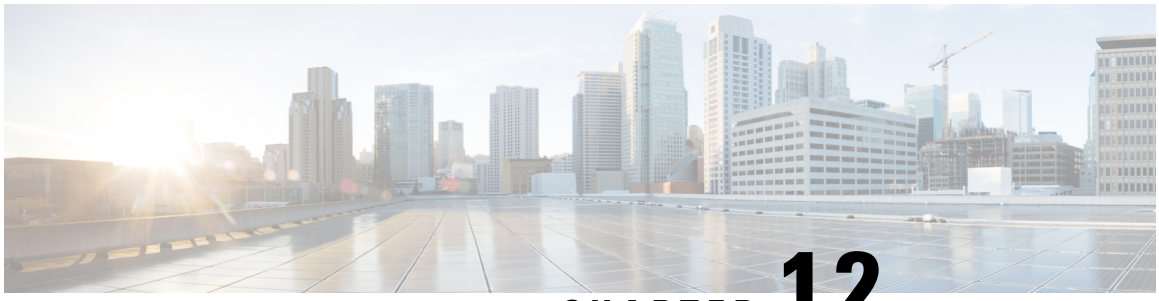
Use the **show ip dhcp relay statistics** [**interface** *interface* [**serverip** *ip-address* [**use-vrf** *vrf-name*]]] command to monitor DHCP relay statistics at the global, server, or interface level.

Use the (Optional) **show ip dhcp snooping statistics vlan** [*vlan-id*] **interface** [**ethernet**|*port-channel*][*id*] command to know the exact statistics about snooping statistics per interface under a vlan.

Configuration Examples for DHCP Snooping

The following example shows how to enable DHCP snooping on two VLANs, with Option 82 support enabled and Ethernet interface 2/5 trusted because the DHCP server is connected to that interface:

```
feature dhcp
ip dhcp snooping
ip dhcp snooping info option

interface Ethernet 2/5
  ip dhcp snooping trust
ip dhcp snooping vlan 1
ip dhcp snooping vlan 50
```



CHAPTER 12

Configuring MAC ACLs

This chapter describes how to configure MAC access lists (ACLs) on Cisco NX-OS devices.

Information About MAC ACLs

MAC ACLs are ACLs that use information in the Layer 2 header of packets to filter traffic. MAC ACLs share many fundamental concepts with IP ACLs, including support for virtualization.

MAC Packet Classification

MAC packet classification allows you to control whether a MAC ACL that is on a Layer 2 interface applies to all traffic entering the interface, including IP traffic, or to non-IP traffic only.

MAC packet classification does not work on the Layer 3 control plane protocols such as HSRP, VRRP, OSPF, and so on. If you enable MAC packet classification on the VLANs, the basic functionalities will break on these protocols.

MAC Packet Classification State	Effect on Interface
Enabled	• A MAC ACL that is on the interface applies to all traffic entering the interface, including IP traffic. • You can apply an IP port ACL on the interface, but it will not filter traffic.
Disabled	• A MAC ACL that is on the interface applies only to non-IP traffic entering the interface. • You can apply an IP port ACL on the interface and it will filter traffic.

Default Settings for MAC ACLs

This table lists the default settings for MAC ACL parameters.

Table 17: Default MAC ACLs Parameters

Parameters	Default
MAC ACLs	No MAC ACLs exist by default
ACL rules	Implicit rules apply to all ACLs

Guidelines and Limitations for MAC ACLs

MAC ACLs have the following configuration guidelines and limitations:

- MAC ACLs apply to ingress traffic only.
- Due to a hardware limitation, MAC ACL does not filter ARP packets on Cisco Nexus 3500 platform switches.

Configuring MAC ACLs

Creating a MAC ACL

You can create a MAC ACL and add rules to it.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **mac access-list** *name*
3. switch(config-mac-acl)# {**permit** | **deny**} *source destination protocol*
4. (Optional) switch(config-mac-acl)# **statistics per-entry**
5. (Optional) switch(config-mac-acl)# **show mac access-lists** *name*
6. (Optional) switch(config-mac-acl)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac access-list** *name*	Creates the MAC ACL and enters ACL configuration mode.
Step 3	switch(config-mac-acl)# {**permit** | **deny**} *source destination protocol*	Creates a rule in the MAC ACL.

	Command or Action	Purpose
		The **permit** and **deny** commands support many ways of identifying traffic.
Step 4	(Optional) switch(config-mac-acl)# **statistics per-entry**	Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	(Optional) switch(config-mac-acl)# **show mac access-lists** *name*	Displays the MAC ACL configuration.
Step 6	(Optional) switch(config-mac-acl)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create a MAC ACL:

```
switch# configure terminal
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# permit 00c0.4f00.0000 0000.00ff.ffff any
switch(config-mac-acl)# statistics per-entry
switch(config-mac-acl)# show mac access-lists acl-mac-01

MAC ACL acl-mac-01
        statistics per-entry
        10 permit 00c0.4f00.0000 0000.00ff.ffff any

switch(config-mac-acl)# copy running-config startup-config
```

Changing a MAC ACL

You can remove a MAC ACL from the device.

Before you begin

Use the **show mac access-lists** command with the **summary** keyword to find the interfaces that a MAC ACL is configured on.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **mac access-list** *name*
3. (Optional) switch(config-mac-acl)# [*sequence-number*] {**permit** | **deny**} *source destination protocol*
4. (Optional) switch(config-mac-acl)# **no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*}
5. (Optional) switch(config-mac-acl)# [**no**] **statistics per-entry**
6. (Optional) switch(config-mac-acl)# **show mac access-lists** *name*
7. (Optional) switch(config-mac-acl)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac access-list** *name*	Enters ACL configuration mode for the ACL that you specify by name.
Step 3	(Optional) switch(config-mac-acl)# [*sequence-number*] {**permit** | **deny**} *source destination protocol*	Creates a rule in the MAC 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 **permit** and **deny** commands support many ways of identifying traffic.
Step 4	(Optional) switch(config-mac-acl)# **no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*}	Removes the rule that you specify from the MAC ACL. The **permit** and **deny** commands support many ways of identifying traffic.
Step 5	(Optional) switch(config-mac-acl)# [**no**] **statistics per-entry**	Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.
Step 6	(Optional) switch(config-mac-acl)# **show mac access-lists** *name*	Displays the MAC ACL configuration.
Step 7	(Optional) switch(config-mac-acl)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to change a MAC ACL:

```
switch# configure terminal
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# 100 permit 00c0.4f00.00 0000.00ff.ffff any
switch(config-mac-acl)# 80 permit 00c0.4f00.00 0000.00ff.ffff any
switch(config-mac-acl)# no 80
switch(config-mac-acl)# statistics per-entry
switch(config-mac-acl)# show mac access-lists acl-mac-01

MAC ACL acl-mac-01
        statistics per-entry
        10 permit 00c0.4f00.0000 0000.00ff.ffff any
        100 permit 00c0.4f00.0000 0000.00ff.ffff any

switch(config-mac-acl)# copy running-config startup-config
```

Changing Sequence Numbers in a MAC ACL

You can change all the sequence numbers assigned to rules in a MAC ACL. Resequencing is useful when you need to insert rules into an ACL and there are not enough available sequence numbers.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **resequence mac access-list** *name starting-sequence-number increment*
3. (Optional) switch(config)# **show mac access-lists** *name*
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **resequence mac access-list** *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the number specified by 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 number that you specify.
Step 3	(Optional) switch(config)# **show mac access-lists** *name*	Displays the MAC ACL configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to change the sequence of a MAC ACL:

```
switch# configure terminal
switch(config)# resequence mac access-list acl-mac-01 100 15
switch(config)# show mac access-lists acl-mac-01

MAC ACL acl-mac-01
        statistics per-entry
        100 permit 00c0.4f00.0000 0000.00ff.ffff any
        115 permit 00c0.4f00.0000 0000.00ff.ffff any

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

Removing a MAC ACL

You can remove a MAC ACL from the device.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no mac access-list** *name*
3. (Optional) switch(config)# **show mac access-lists** *name* **summary**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no mac access-list** *name*	Removes the MAC ACL that you specify by name from the running configuration.
Step 3	(Optional) switch(config)# **show mac access-lists** *name* **summary**	Displays the MAC ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to remove a MAC ACL:

```
switch# configure terminal
switch(config)# show mac access-lists

MAC ACL acl-mac-01
        statistics per-entry
        100 permit 00c0.4f00.0000 0000.00ff.ffff any
        115 permit 00c0.4f00.0000 0000.00ff.ffff any
MAC ACL acl-mac-02
        statistics per-entry
        10 permit 00a0.3f00.0000 0000.00dd.ffff any
MAC ACL acl-mac-03
        statistics per-entry
        10 permit 00b0.5f00.0000 0000.00aa.fbbf any

switch(config)# no mac access-list acl-mac-02
switch(config)# show mac access-lists acl-mac-02 summary
switch(config)# show mac access-lists

MAC ACL acl-mac-01
        statistics per-entry
        100 permit 00c0.4f00.0000 0000.00ff.ffff any
        115 permit 00c0.4f00.0000 0000.00ff.ffff any
MAC ACL acl-mac-03
        statistics per-entry
        10 permit 00b0.5f00.0000 0000.00aa.fbbf any

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

Applying a MAC ACL as a Port ACL

You can apply a MAC ACL as a port ACL to any of the following interface types:

- Layer 2 or Layer 3 Ethernet interfaces
- Layer 2 or Layer 3 port-channel interfaces

Before you begin

Ensure that the ACL that you want to apply exists and is configured to filter traffic in the manner that you need for this application.

SUMMARY STEPS

1. switch# **configure terminal**
2. Enter one of the following commands:
   - switch(config)# **interface ethernet** *slot*/*port*
   - switch(config)# **interface port-channel** *channel-number*
3. switch(config-if)# **mac port access-group** *access-list*
4. (Optional) switch(config-if)#**show running-config aclmgr**
5. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	Enter one of the following commands: • switch(config)# **interface ethernet** *slot*/*port* • switch(config)# **interface port-channel** *channel-number*	• Enters interface configuration mode for a Layer 2 or Layer 3 interface. • Enters interface configuration mode for a Layer 2 or Layer 3 port-channel interface.
Step 3	switch(config-if)# **mac port access-group** *access-list*	Applies a MAC ACL to the interface.
Step 4	(Optional) switch(config-if)#**show running-config aclmgr**	Displays ACL configuration.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to apply a MAC ACL as a port ACL to an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# mac port access-group acl-mac-01
switch(config-if)# show running-config aclmgr

!Command: show running-config aclmgr
```

```
!Time: Sat Jul 19 23:36:04 2014

version 6.0(2)A4(1)
mac access-list acl-mac-01
  statistics per-entry
  100 permit 00C0.4F00.0000 0000.00FF.FFFF any
  115 permit 00C0.4F00.0000 0000.00FF.FFFF any
mac access-list acl-mac-03
  statistics per-entry
  10 permit 00B0.5F00.0000 0000.00AA.FBBF any
ip access-list copp-system-acl-bfd
  10 permit udp any any eq 3784
ip access-list copp-system-acl-eigrp
  10 permit eigrp any any
ip access-list copp-system-acl-ftp
  10 permit tcp any any eq ftp-data
  20 permit tcp any any eq ftp
  30 permit tcp any eq ftp-data any
  40 permit tcp any eq ftp any

...

interface Ethernet1/3
  mac port access-group acl-mac-01

switch(config-if)# copy running-config startup-config
```

This example shows how to apply a MAC ACL as a port ACL to a port-channel interface:

```
switch# configure terminal
switch(config)# interface port-channel 5
switch(config-if)# mac port access-group acl-mac-01
switch(config-if)# show running-config aclmgr

!Command: show running-config aclmgr
!Time: Sat Jul 19 23:37:04 2014

version 6.0(2)A4(1)
mac access-list acl-mac-01
  statistics per-entry
  100 permit 00C0.4F00.0000 0000.00FF.FFFF any
  115 permit 00C0.4F00.0000 0000.00FF.FFFF any
mac access-list acl-mac-03
  statistics per-entry
  10 permit 00B0.5F00.0000 0000.00AA.FBBF any
ip access-list copp-system-acl-bfd
  10 permit udp any any eq 3784
ip access-list copp-system-acl-eigrp
  10 permit eigrp any any
ip access-list copp-system-acl-ftp
  10 permit tcp any any eq ftp-data
  20 permit tcp any any eq ftp
  30 permit tcp any eq ftp-data any
  40 permit tcp any eq ftp any


...

interface port-channel5
  mac port access-group acl-mac-01
```

```
switch(config-if)# copy running-config startup-config
```

Enabling or Disabling MAC Packet Classification

You can enable or disable MAC packet classification on a per VLAN basis.

SUMMARY STEPS

1. **config t**
2. **vlan** *vlan-number*
3. [**no**] **mac packet-classify**
4. **exit**
5. (Optional) **show running-config vlan** *vlan-number*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**vlan** *vlan-number*
	Example:
	`switch(config)# vlan 10`
	`switch(config-vlan)#`
		Creates a VLAN interface. The number range is from 1 to 4094.

Step 3	[**no**] **mac packet-classify**
	Example:
	`switch(config-vlan)# mac packet-classify`
	`switch(config-vlan)#`
		Enables MAC packet classification on the vlan. The **no** option disables MAC packet classification on the vlan.

Step 4	**exit**
	Example:
	`switch(config-vlan)# exit`
	`switch(config)#`
		Exits the vlan configuration.

Step 5	(Optional) **show running-config vlan** *vlan-number*	Displays the running configuration.

Example

This example shows how to enable MAC packet classification on a per VLAN basis:

```
switch# configure terminal
switch(config)# vlan 50
switch(config-vlan)# mac packet-classify
switch(config-vlan)# exit
switch(config)# show running-config vlan 50
```

```
!Command: show running-config interface Vlan50
!Time: Wed Aug  6 20:39:03 2014

version 6.0(2)A4(1)

interface Vlan50
  mac packet-classify

switch(config-if)# copy running-config startup-config
```

Verifying the MAC ACL Configuration

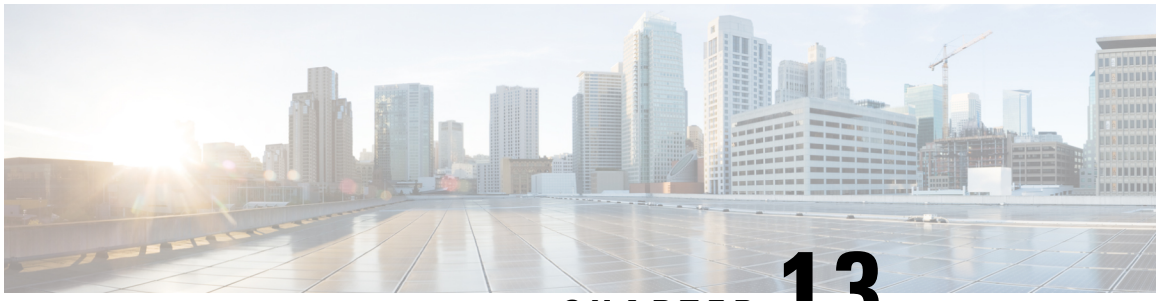
To display MAC ACL configuration information, perform one of the following tasks.

Command	Purpose
show mac access-lists	Displays the MAC ACL configuration.
show running-config aclmgr [**all**]	Displays the ACL configuration, including MAC ACLs and the interfaces to which MAC ACLs are applied. **Note** The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show startup-config aclmgr [**all**]	Displays the ACL startup configuration. **Note** The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

Clearing MAC ACL Statistics

You can clear MAC ACL statistics by using the **clear mac access-list counters** command

Command	Purpose
clear mac access-list counters	Clears statistics for all MAC ACLs or for a specific MAC ACL.



CHAPTER 13

Configuring Unicast RPF

This chapter describes how to configure rate limits for egress traffic on Cisco NX-OS devices and includes the following sections:.

Information About Unicast RPF

The Unicast RPF feature reduces problems that are caused by the introduction of malformed or forged (spoofed) IPv4 source addresses into a network by discarding IPv4 packets that lack a verifiable IP source address. For example, a number of common types of Denial-of-Service (DoS) attacks, including Smurf and Tribal Flood Network (TFN) attacks, can take advantage of forged or rapidly changing source IPv4 addresses to allow attackers to thwart efforts to locate or filter the attacks. Unicast RPF deflects attacks by forwarding only the packets that have source addresses that are valid and consistent with the IP routing table.

When you enable Unicast RPF on an interface, the switch examines all ingress packets received on that interface to ensure that the source address and source interface appear in the routing table and match the interface on which the packet was received. This examination of source addresses relies on the Forwarding Information Base (FIB).

Note Unicast RPF is an ingress function and is applied only on the ingress interface of a switch at the upstream end of a connection.

Unicast RPF verifies that any packet received at a switch interface arrives on the best return path (return route) to the source of the packet by doing a reverse lookup in the FIB. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. If there is no reverse path route on the same interface from which the packet was received, the source address might have been modified by the attacker. If Unicast RPF does not find a reverse path for the packet, the packet is dropped.

Note With Unicast RPF, all equal-cost “best” return paths are considered valid, which means that Unicast RPF works where multiple return paths exist, if each path is equal to the others in terms of the routing cost (number of hops, weights, and so on) and as long as the route is in the FIB. Unicast RPF also functions where Enhanced Interior Gateway Routing Protocol (EIGRP) variants are being used and unequal candidate paths back to the source IP address exist.

Unicast RPF

The Unicast Reverse Path Forwarding (RPF) feature reduces problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of Denial-of-Service (DoS) attacks, including Smurf and Tribal Flood Network (TFN) attacks, can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. Unicast RPF deflects attacks by forwarding only the packets that have source addresses that are valid and consistent with the IP routing table.

Global Statistics

Each time the Cisco NX-OS device drops a packet at an interface due to a failed unicast RPF check, that information is counted globally on the device on a per-forwarding engine (FE) basis. Global statistics on dropped packets provide information about potential attacks on the network, but they do not specify which interface is the source of the attack. Per-interface statistics on packets dropped due to a failed unicast RPF check are not available.

Guidelines and Limitations for Unicast RPF

Unicast RPF has the following configuration guidelines and limitations:

- In Warp mode that is unique to Cisco Nexus 3548 Series switches, when URPF is enabled, the number of multicast entries is halved from 8k to 4k. Similarly, the number of host entries is also halved from 8k to 4k. In Normal mode, the number of LPM entries supported is halved (from 24k to 12k) but this is similar to that in Cisco Nexus 3000 Series switches.
- You must apply Unicast RPF at the interface downstream from the larger portion of the network, preferably at the edges of your network.
- The further downstream that you apply Unicast RPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying Unicast RPF on an aggregation device helps to mitigate attacks from many downstream networks or clients and is simple to administer, but it does not help identify the source of the attack. Applying Unicast RPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying Unicast RPF across many sites does add to the administration cost of operating the network.
- The more entities that deploy Unicast RPF across Internet, intranet, and extranet resources, means that the better the chances are of mitigating large-scale network disruptions throughout the Internet community, and the better the chances are of tracing the source of an attack.

- Unicast RPF will not inspect IP packets that are encapsulated in tunnels, such as generic routing encapsulation (GRE) tunnels. You must configure Unicast RPF at a home gateway so that Unicast RPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.
- You can use Unicast RPF in any "single-homed" environment where there is only one access point out of the network or one upstream connection. Networks that have one access point provide symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet.
- Do not use Unicast RPF on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry, which means that multiple routes to the source of a packet exist. You should configure Unicast RPF only where there is natural or configured symmetry. Do not configure strict Unicast RPF.
- Unicast RPF allows packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that the Bootstrap Protocol (BOOTP) and the Dynamic Host Configuration Protocol (DHCP) can operate correctly.

Default Settings for Unicast RPF

This table lists the default settings for Unicast RPF parameters.

Table 18: Default Unicast RPF Parameter Settings

Parameters	Default
Unicast RPF	Disabled

Configuring Unicast RPF

You can configure one the following Unicast RPF modes on an ingress interface:

Strict Unicast RPF mode
A strict mode check is successful when Unicast RFP finds a match in the FIB for the packet source address and the ingress interface through which the packet is received matches one of the Unicast RPF interfaces in the FIB match. If this check fails, the packet is discarded. You can use this type of Unicast RPF check where packet flows are expected to be symmetrical.

Loose Unicast RPF mode
A loose mode check is successful when a lookup of a packet source address in the FIB returns a match and the FIB result indicates that the source is reachable through at least one real interface. The ingress interface through which the packet is received is not required to match any of the interfaces in the FIB result.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **ip verify unicast source reachable-via** {**any** [**allow-default**] | **rx**}
4. **exit**
5. (Optional) **show ip interface ethernet** *slot/port*

6. (Optional) **show running-config interface ethernet** *slot/port*
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		Specifies an Ethernet interface and enters interface configuration mode.

Step 3	**ip verify unicast source reachable-via** {**any** [**allow-default**] | **rx**} **Example:** `switch(config-if)# ip verify unicast source reachable-via any`	Configures Unicast RPF on the interface for IPv4. The **any** keyword specifies loose Unicast RPF. If you specify the **allow-default** keyword, the source address lookup can match the default route and use that for verification. The **rx** keyword specifies strict Unicast RPF.

Step 4	**exit**
	Example:
	`switch(config-cmap)# exit`
	`switch(config)#`
		Exits class map configuration mode.

Step 5	(Optional) **show ip interface ethernet** *slot/port* **Example:** `switch(config)# show ip interface ethernet 2/3`	Displays the IP information for an interface.
Step 6	(Optional) **show running-config interface ethernet** *slot/port* **Example:** `switch(config)# show running-config interface ethernet 2/3`	Displays the configuration for an interface in the running configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuration Examples for Unicast RPF

The following example shows how to configure loose Unicast RFP for IPv4 packets:

```
interface Ethernet2/3
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via any
```

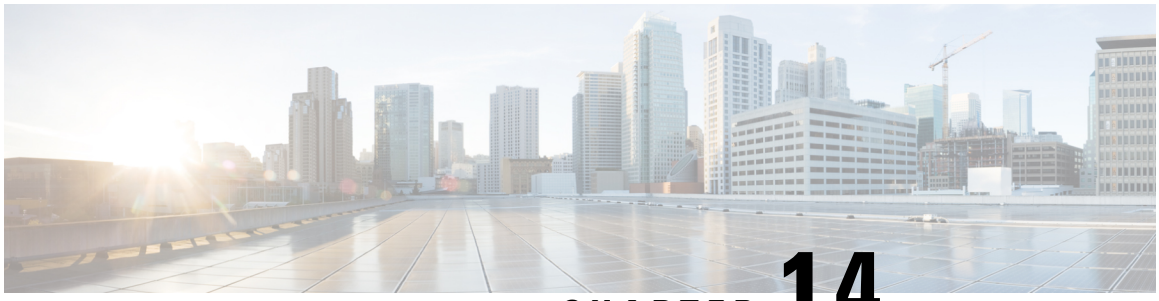
The following example shows how to configure strict Unicast RPF for IPv4 packets:

```
interface Ethernet2/2
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via rx
```

Verifying the Unicast RPF Configuration

To display Unicast RPF configuration information, perform one of the following tasks:

Command	Purpose
show running-config interface ethernet *slot/port*	Displays the interface configuration in the running configuration.
show running-config ip [**all**]	Displays the IPv4 configuration in the running configuration.
show startup-config interface ethernet *slot/port*	Displays the interface configuration in the startup configuration.
show startup-config ip	Displays the IP configuration in the startup configuration.



CHAPTER 14

Configuring Control Plane Policing

This chapter contains the following sections:

Information About CoPP

Control Plane Policing (CoPP) protects the control plane and separates it from the data plane, which ensures network stability, reachability, and packet delivery.

This feature allows a policy map to be applied to the control plane. This policy map looks like a normal QoS policy and is applied to all traffic destined to any of the IP addresses of the router or Layer 3 switch.. A common attack vector for network devices is the denial-of-service (DoS) attack, where excessive traffic is directed at the device interfaces.

The Cisco NX-OS device provides CoPP to prevent DoS attacks from impacting performance. Such attacks, which can be perpetrated either inadvertently or maliciously, typically involve high rates of traffic destined to the supervisor module or CPU itself.

The supervisor module divides the traffic that it manages into three functional components or planes:

Data plane
Handles all the data traffic. The basic functionality of a Cisco NX-OS device is to forward packets from one interface to another. The packets that are not meant for the switch itself are called the transit packets. These packets are handled by the data plane.

Control plane
Handles all routing protocol control traffic. These protocols, such as the Border Gateway Protocol (BGP) and the Open Shortest Path First (OSPF) Protocol, send control packets between devices. These packets are destined to router addresses and are called control plane packets.

Management plane
Runs the components meant for Cisco NX-OS device management purposes such as the command-line interface (CLI) and Simple Network Management Protocol (SNMP).

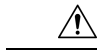
The supervisor module has both the management plane and control plane and is critical to the operation of the network. Any disruption or attacks to the supervisor module will result in serious network outages. For example, excessive traffic to the supervisor module could overload and slow down the performance of the entire Cisco NX-OS device. Another example is a DoS attack on the supervisor module that could generate IP traffic streams to the control plane at a very high rate, forcing the control plane to spend a large amount of time in handling these packets and preventing the control plane from processing genuine traffic.

Examples of DoS attacks are as follows:

- Internet Control Message Protocol (ICMP) echo requests
- IP fragments
- TCP SYN flooding

These attacks can impact the device performance and have the following negative effects:

- Reduced service quality (such as poor voice, video, or critical applications traffic)
- High route processor or switch processor CPU utilization
- Route flaps due to loss of routing protocol updates or keepalives
- Unstable Layer 2 topology
- Slow or unresponsive interactive sessions with the CLI
- Processor resource exhaustion, such as the memory and buffers
- Indiscriminate drops of incoming packets



Caution It is important to ensure that you protect the supervisor module from accidental or malicious attacks by configuring control plane protection.

Control Plane Protection

To protect the control plane, the Cisco NX-OS device segregates different packets destined for the control plane into different classes. Once these classes are identified, the Cisco NX-OS device polices the packets, which ensures that the supervisor module is not overwhelmed.

Control Plane Packet Types

Different types of packets can reach the control plane:

Receive packets
Packets that have the destination address of a router. The destination address can be a Layer 2 address (such as a router MAC address) or a Layer 3 address (such as the IP address of a router interface). These packets include router updates and keepalive messages. Multicast packets can also be in this category where packets are sent to multicast addresses that are used by a router.

Exception packets
Packets that need special handling by the supervisor module. For example, if a destination address is not present in the Forwarding Information Base (FIB) and results in a miss, the supervisor module sends an ICMP unreachable packet back to the sender. Another example is a packet with IP options set.

Redirected packets
Packets that are redirected to the supervisor module. Features such as Dynamic Host Configuration Protocol (DHCP) snooping or dynamic Address Resolution Protocol (ARP) inspection redirect some packets to the supervisor module.

Glean packets
If a Layer 2 MAC address for a destination IP address is not present in the FIB, the supervisor module receives the packet and sends an ARP request to the host.

All of these different packets could be maliciously used to attack the control plane and overwhelm the Cisco NX-OS device. CoPP classifies these packets to different classes and provides a mechanism to individually control the rate at which the supervisor module receives these packets.

Classification for CoPP

For effective protection, the Cisco NX-OS device classifies the packets that reach the supervisor modules to allow you to apply different rate controlling policies based on the type of the packet. For example, you might want to be less strict with a protocol packet such as Hello messages but more strict with a packet that is sent to the supervisor module because the IP option is set. ou configure packet classifications and rate controlling policies using class-maps and policy-maps.

The following parameters can be used to classify a packet:

- Source IP address
- Destination IP address
- Source port
- Destination port
- Layer 4 protocol

Rate Controlling Mechanisms

Once the packets are classified, the Cisco NX-OS device has different mechanisms to control the rate at which packets arrive at the supervisor module.

The policing rate is specified in terms of packets per second (PPS). Each classified flow can be policed individually by specifying a policing rate limit in PPS.

CoPP Policy Templates

When you bring up your Cisco NX-OS device for the first time, the Cisco NX-OS software installs the default copp-system-policy to protect the supervisor module from DoS attacks. You can choose the CoPP policy template for your deployment scenario by specifying CoPP policy options from the initial setup utility:

- Default—Layer 2 and Layer 3 policy which provides a good balance of policing between switched and routed traffic bound to CPU.
- Layer 2—Layer 2 policy which gives more preference to the Layer 2 traffic (eg BPDU) bound to the CPU
- Layer 3—Layer 3 policy which gives more preference to the Layer 3 traffic (eg BGP, RIP, OSPF etc) bound to the CPU

If you do not select an option or choose not to execute the setup utility, the Cisco NX-OS software applies the Default policing. Cisco recommends starting with the default policy and later modifying the CoPP policies as required.

The default copp-system-policy policy has optimized values suitable for basic device operations. You must add specific class and access-control list (ACL) rules that meet your DoS protection requirements.

You can switch across default, Layer 2 and Layer 3 templates by entering the setup utility again using the setup command.

Default CoPP Policy

This policy is applied to the switch by default. It has the classes with police rates that should suit most network installations. You cannot modify this policy template, but you can modify the CoPP configuration on the device. After you run the setup utility and set up the default CoPP policy profile, all modifications that were made to the CoPP policy will be removed.

This policy has the following configuration:

```
policy-map type control-plane copp-system-policy
  class copp-s-default
    police pps 400
  class copp-s-ping
    police pps 100
  class copp-s-l3destmiss
    police pps 100
  class copp-s-glean
    police pps 500
  class copp-s-l3mtufail
    police pps 100
  class copp-s-ttl1
    police pps 100
  class copp-s-ip-options
    police pps 100
  class copp-s-ip-nat
    police pps 100
  class copp-s-ipmcmiss
    police pps 400
  class copp-s-ipmc-g-hit
    police pps 400
  class copp-s-ipmc-rpf-fail-g
```

```
    police pps 400
  class copp-s-ipmc-rpf-fail-sg
    police pps 400
  class copp-s-dhcpreq
    police pps 300
  class copp-s-dhcpresp
    police pps 300
  class copp-s-igmp
    police pps 400
  class copp-s-routingProto2
    police pps 1300
  class copp-s-eigrp
    police pps 200
  class copp-s-pimreg
    police pps 200
  class copp-s-pimautorp
    police pps 200
  class copp-s-routingProto1
    police pps 1000
  class copp-s-arp
    police pps 200
  class copp-s-ptp
    police pps 1000
  class copp-s-bpdu
    police pps 12000
  class copp-s-cdp
    police pps 400
  class copp-s-lacp
    police pps 400
  class copp-s-lldp
    police pps 200
  class copp-icmp
    police pps 200
  class copp-telnet
    police pps 500
  class copp-ssh
    police pps 500
  class copp-snmp
    police pps 500
  class copp-ntp
    police pps 100
  class copp-tacacsradius
    police pps 400
  class copp-stftp
    police pps 400
  class copp-ftp
    police pps 100
  class copp-http
    police pps 100
```

Layer 2 CoPP Policy

You cannot modify this policy template, but you can modify the CoPP configuration on the device. After you run the setup utility and set up the Layer 2 CoPP policy profile, all modifications that were made to the CoPP policy will be removed.

This policy has the following configuration:

```
policy-map type control-plane copp-system-policy
  class copp-s-default
    police pps 400
  class copp-s-ping
    police pps 100
```

```
class copp-s-l3destmiss
  police pps 100
class copp-s-glean
  police pps 500
class copp-s-l3mtufail
  police pps 100
class copp-s-ttl1
  police pps 100
class copp-s-ip-options
  police pps 100
class copp-s-ip-nat
  police pps 100
class copp-s-ipmcmiss
  police pps 400
class copp-s-ipmc-g-hit
  police pps 400
class copp-s-ipmc-rpf-fail-g
  police pps 400
class copp-s-ipmc-rpf-fail-sg
  police pps 400
class copp-s-dhcpreq
  police pps 300
class copp-s-dhcpresp
  police pps 300
class copp-s-igmp
  police pps 400
class copp-s-routingProto2
  police pps 1200
class copp-s-eigrp
  police pps 200
class copp-s-pimreg
  police pps 200
class copp-s-pimautorp
  police pps 200
class copp-s-routingProto1
  police pps 900
class copp-s-arp
  police pps 200
class copp-s-ptp
  police pps 1000
class copp-s-bpdu
  police pps 12300
class copp-s-cdp
  police pps 400
class copp-s-lacp
  police pps 400
class copp-s-lldp
  police pps 200
class copp-icmp
  police pps 200
class copp-telnet
  police pps 500
class copp-ssh
  police pps 500
class copp-snmp
  police pps 500
class copp-ntp
  police pps 100
class copp-tacacsradius
  police pps 400
class copp-stftp
  police pps 400
class copp-ftp
  police pps 100
```

```
  class copp-http
    police pps 100
```

Layer 3 CoPP Policy

You cannot modify this policy template, but you can modify the CoPP configuration on the device. After you run the setup utility and set up the Layer 3 CoPP policy profile, all modifications that were made to the CoPP policy will be removed.

This policy has the following configuration:

```
policy-map type control-plane copp-system-policy
  class copp-s-default
    police pps 400
  class copp-s-ping
    police pps 100
  class copp-s-l3destmiss
    police pps 100
  class copp-s-glean
    police pps 500
  class copp-s-l3mtufail
    police pps 100
  class copp-s-ttl1
    police pps 100
  class copp-s-ip-options
    police pps 100
  class copp-s-ip-nat
    police pps 100
  class copp-s-ipmcmiss
    police pps 400
  class copp-s-ipmc-g-hit
    police pps 400
  class copp-s-ipmc-rpf-fail-g
    police pps 400
  class copp-s-ipmc-rpf-fail-sg
    police pps 400
  class copp-s-dhcpreq
    police pps 300
  class copp-s-dhcpresp
    police pps 300
  class copp-s-igmp
    police pps 400
  class copp-s-routingProto2
    police pps 4000
  class copp-s-eigrp
    police pps 200
  class copp-s-pimreg
    police pps 200
  class copp-s-pimautorp
    police pps 200
  class copp-s-routingProto1
    police pps 4000
  class copp-s-arp
    police pps 200
  class copp-s-ptp
    police pps 1000
  class copp-s-bpdu
    police pps 6000
  class copp-s-cdp
    police pps 200
  class copp-s-lacp
    police pps 200
```

```
class copp-s-lldp
  police pps 200
class copp-icmp
  police pps 200
class copp-telnet
  police pps 500
class copp-ssh
  police pps 500
class copp-snmp
  police pps 500
class copp-ntp
  police pps 100
class copp-tacacsradius
  police pps 400
class copp-stftp
  police pps 400
class copp-ftp
  police pps 100
class copp-http
  police pps 100
```

CoPP Class Maps

Classes within a policy are of two types:

- Static—These classes are part of every policy template and cannot be removed from the policy or CoPP configuration. Static classes would typically contain the traffic which is deemed critical to device operation and is required in the policy.
- Dynamic—These classes can be created, added or removed from a policy. Using dynamic classes, you can create classes/policing for CPU bound traffic (unicast) specific to their requirements.

Note Classes with names copp-s-x are static classes. ACLs can be associated with both static and dynamic classes.

A new CoPP class "copp-s-pim-datareg" is added to match Protocol-Independent Multicast (PIM) data register packets destined to the switch. This CoPP class help classify PIM data register packets to a separate queue, with a policer rate of 600 Packets-Per-Second (pps). The following are the three CoPP classes for the PIM protocol:

- **copp-s-pimreg -** Matches PIM protocol packets which are multicast packets such as PIM hello, join-prune etc.
- **copp-s-pimautorp** – Matches PIM RP election protocol packets.
- **copp-s-pim-datareg** - Matches PIM data register packets.

Packets Per Second Credit Limit

The aggregate packets per second (PPS) for a given policy (sum of PPS of each class part of the policy) is capped by an upper PPS Credit Limit (PCL). If an increase in PPS of a given class causes a PCL exceed, the configuration is rejected. To increase the desired PPS, the additional PPS beyond PCL should be decreased from other class(es).

CoPP and the Management Interface

The Cisco NX-OS device supports only hardware-based CoPP which does not support the management interface (mgmt0). The out-of-band mgmt0 interface connects directly to the CPU and does not pass through the in-band traffic hardware where CoPP is implemented.

On the mgmt0 interface, ACLs can be configured to give or deny access to a particular type of traffic.

Guidelines and Limitations for CoPP

CoPP has the following configuration guidelines and limitations:

- Cisco recommends that you choose the default, L2, or L3 policy, depending upon your deployment scenario and later modify the CoPP policies based on observed behavior.
- If you observe +/- 2-5% irregularity in the traffic around 30-40s after the traffic has fully converged after fast-reload, use a higher COPP value for the ARP packets.
- Customizing CoPP is an ongoing process. CoPP must be configured according to the protocols and features used in your specific environment as well as the supervisor features that are required by the server environment. As these protocols and features change, CoPP must be modified.
- The default police packets per second (PPS) value is changed to 900 for **copp-s-bfd** command with **write erase** command and reload.
- Cisco recommends that you continuously monitor CoPP. If drops occur, determine if CoPP dropped traffic unintentionally or in response to a malfunction or attack. In either event, analyze the situation and evaluate the need to use a different CoPP policy or modify the customized CoPP policy.
- The Cisco NX-OS software does not support egress CoPP or silent mode. CoPP is supported only on ingress (**service-policy output copp** cannot be applied to the control plane interface).
- The creation of new CoPP policies is not supported.
- When upgrading, check whether the default LLDP CoPP value is less than 500 pps. If it is less than 500 pps, manually change it to 500 pps by using the following commands:

```
switch(config)# policy-map type control-plane policy-map-name
switch(config-pmap)# class copp-s-lldp
switch(config-pmap-c)# police pps 500
```

- There are no hardware counters for glean, class-default class-map in cache mode.
- There are no counters for the MTU fail class-map.
- There are no hardware counters for NAT.
- There are no hardware counter for IPMCMISS.
- You cannot add match ACL statements to a static class-map.
- Cisco Nexus 3500 Series switches drop all the packets when the tunnel is not configured. The packets are also dropped when the tunnel is configured but the tunnel interface is not configured or the tunnel interface is in shut down state.

Point to Point tunnel (Source and Destination) – Cisco Nexus 3500 Series switches decapsulate all IP-in-IP packets destined to it when the command **feature tunnel** is configured and there is an operational tunnel interface configured with the tunnel source and the destination address that matches the incoming packets' outer source and destination addresses. If there is not a source and destination packet match or if the interface is in shutdown state, the packet is dropped.

Decapsulate Tunnel (Source only) - Cisco Nexus 3500 Series switches decapsulate all IP-in-IP packets destined to it when the command **feature tunnel** is configured and there is an operational tunnel interface configured with the tunnel source address that matches the incoming packets' outer destination addresses. If there is not a source packet match or if the interface is in shutdown state, the packet is dropped.

- If you use NXAPI over the front panel port, you must increase the CoPP policy (for http) to allow 3000 PPS traffic so that there is no packet drop and the CLIs with a larger output return within the expected time.
- When you execute the setup script you will be prompted with *Enter to basic configuration (yes/no)?*.
  - If you answer *no*, then the default CoPP policy template will not be applied to the system.
  - If you answer *yes*, then the default CoPP policy template of the running version will be applied to the system. This action will overwrite the non-default policer rates configured on system CoPP classes.

Note If you press CTRL+C during the setup script execution of the script, default CoPP policy template will not be applied into the system and there will be no changes in the existing CoPP policy

- If you press CTRL+C after executing the setup script and entering into basic configuration, it skips all the remaining steps and you will be prompted to *Apply and save the config before exiting (yes/no)?*.
  - If you answer *no*, then the default CoPP policy template will not be applied to the system.
  - If you answer *yes*, then the default CoPP policy template of the running version is applied. This action will overwrite the non-default policer rates configured on system CoPP classes.
- The setup script will not alter any user defined CoPP classes.
- When a default CoPP policy template is applied as part of successful setup script execution, the control packets may be dropped for a brief period of time. During this window, control plane protocols may flap.
- The setup script may fail to configure the default CoPP policy template when PPS credits are exhausted. This may result in one or more system CoPP classes with zero PPS. This may happen, when there are user defined classes with high PPS values. To apply the default CoPP policy, you must reconfigure the PPS values of user defined CoPP classes and run the setup script once again.
- Hardware and software match packet counters for CDP (copp-s-cdp), LLDP (copp-s-lldp), LACP (copp-s-lacp), BPDU (copp-s-bpdu) classes are aggregated on Cisco Nexus 3548 platform switches. Likewise, hardware and software match packet counters for copp-s-dhcpreq and copp-s-dhcpresp classes are aggregated.
- When a change is applied to the COPP policy, the configuration is implemented in the hardware as a non-atomic operation. This process may cause disruption to the control traffic.

Upgrade Guidelines for CoPP

CoPP has the following upgrade guidelines:

- If you upgrade from a Cisco NX-OS release that does not support the CoPP feature to a release that supports the CoPP feature, CoPP is automatically enabled with the default policy when the switch boots up. You must run the setup script after the upgrade to enable a different policy (default, l3, ,l2). Not configuring CoPP protection can leave your NX-OS device vulnerable to DoS attacks.
- If you upgrade from a Cisco NX-OS release that supports the CoPP feature to a Cisco NX-OS release that supports the CoPP feature with additional classes for new protocols, you must run the setup utility for the new CoPP classes to be available.
- We recommend that you run the setup script during a scheduled maintenance period and not during a time when there is traffic on the device, because the setup script modifies the policing rates corresponding to different flows coming into the CPU.
- When upgrading to Cisco NX-OS Release 6.0(2)A3(1), check whether the default LLDP CoPP value is less than 500 pps. If it is less than 500, manually change it to 500 by using the following commands:

```
switch(config)# policy-map type control-plane copp-system-policy
switch(config-pmap)# class copp-s-lldp
switch(config-pmap-c)# police pps 500
```

Configuring CoPP

Configuring a Control Plane Class Map

You must configure control plane class maps for control plane policies.

You can classify traffic by matching packets based on existing ACLs. The permit and deny ACL keywords are ignored in the matching.

Before you begin

Ensure that you have configured the IP ACLs if you want to use ACE hit counters in the class maps.

SUMMARY STEPS

1. **configure terminal**
2. **class-map type control-plane match-any** *class-map-name*
3. (Optional) **match access-group name** *access-list-name*
4. **exit**
5. (Optional) **show class-map type control-plane** [*class-map-name*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**class-map type control-plane match-any** *class-map-name*
	Example:
	`switch(config)# class-map type control-plane ClassMapA`
	`switch(config-cmap)#`
		Specifies a control plane class map and enters class map configuration mode. The default class matching is match-any. The name can be a maximum of 64 characters long and is case sensitive.
		Note You cannot use class-default, match-all, or match-any as class map names.

Step 3	(Optional) **match access-group name** *access-list-name* **Example:** `switch(config-cmap)# match access-group name MyAccessList`	Specifies matching for an IP ACL. You can repeat this step to match more than one IP ACL. **Note** The **permit** and **deny** ACL keywords are ignored in the CoPP matching.

Step 4	**exit**
	Example:
	`switch(config-cmap)# exit`
	`switch(config)#`
		Exits class map configuration mode.

Step 5	(Optional) **show class-map type control-plane** [*class-map-name*] **Example:** `switch(config)# show class-map type control-plane`	Displays the control plane class map configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a Control Plane Policy Map

You must configure a policy map for CoPP, which includes policing parameters. If you do not configure a policer for a class, the default PPS for that class is 0.

You can configure policies for IPv4 packets.

Before you begin

Ensure that you have configured a control plane class map.

SUMMARY STEPS

1. **configure terminal**

2. **policy-map type control-plane** *policy-map-name*
3. **class** {*class-map-name* | **class**}
4. **police** [**pps**] {*pps-value*} [**bc**] *burst-size* [**bytes** | **kbytes** | **mbytes** | **ms** | **packets** | **us**]
5. **exit**
6. **exit**
7. (Optional) **show policy-map type control-plane** [**expand**] [**name** *class-map-name*]
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**policy-map type control-plane** *policy-map-name*
	Example:
	`switch(config)# policy-map type control-plane copp-system-policy`
	`switch(config-pmap)#`
		Specifies a control plane policy map and enters the policy map configuration mode. The policy map name is case sensitive.
		Note The name of the policy-map cannot be changed. You can only use the **copp-system-policy** name for the policy-map. The system allows only a single **type control-plane** policy-map to be configured.

Step 3	**class** {*class-map-name*	**class**}
		Example:
		`switch(config-pmap)# class ClassMapA`
		`switch(config-pmap-c)#`
			Specifies a control plane class map name or the class default and enters control plane class configuration mode.

Step 4	**police** [**pps**] {*pps-value*} [**bc**] *burst-size* [**bytes**	**kbytes**	**mbytes**	**ms**	**packets**	**us**] **Example:** `switch(config-pmap-c)# police pps 100 bc 10`	Specifies the rate limit in terms of packets per second (PPS) and the committed burst (BC). The PPS range is 0 - 20,000. The default PPS is 0. The BC range is from 0 to 512000000. The default BC size unit is bytes.

Step 5	**exit**
	Example:
	`switch(config-pmap-c)# exit`
	`switch(config-pmap)#`
		Exits policy map class configuration mode.

Step 6	**exit**
	Example:
	`switch(config-pmap)# exit`
	`switch(config)#`
		Exits policy map configuration mode.

Step 7	(Optional) **show policy-map type control-plane** [**expand**] [**name** *class-map-name*] **Example:**	Displays the control plane policy map configuration.

	Command or Action	Purpose
	`switch(config)# show policy-map type control-plane`	
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Control Plane Service Policy

Before you begin

Configure a control plane policy map.

SUMMARY STEPS

1. **configure terminal**
2. **control-plane**
3. **exit**
4. (Optional) **show running-config copp** [**all**]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**control-plane**
	Example:
	`switch(config) # control-plane`
	`switch(config-cp)#`
		Enters control plane configuration mode.

Step 3	**exit**
	Example:
	`switch(config-cp)# exit`
	`switch(config)#`
		Exits control plane configuration mode.

Step 4	(Optional) **show running-config copp** [**all**] **Example:** `switch(config)# show running-config copp`	Displays the CoPP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

CoPP Show Commands

To display CoPP configuration information, enter one of the following show commands:

Command	Purpose
show ip access-lists [*acl-name*]	Displays all IPv4 ACLs configured in the system, including the CoPP ACLs.
show class-map type control-plane [*class-map-name*]	Displays the control plane class map configuration, including the ACLs that are bound to this class map.
show policy-map type control-plane [**expand**] [**name** *policy-map-name*]	Displays the control plane policy map with associated class maps and PPS values.
show running-config copp [**all**]	Displays the CoPP configuration in the running configuration.
show running-config aclmgr [**all**]	Displays the user-configured access control lists (ACLs) in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show startup-config copp [**all**]	Displays the CoPP configuration in the startup configuration.
show startup-config aclmgr [**all**]	Displays the user-configured access control lists (ACLs) in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

Displaying the CoPP Configuration Status

SUMMARY STEPS

1. switch# **show copp status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show copp status**	Displays the configuration status for the CoPP feature.

Example

This example shows how to display the CoPP configuration status:

```
switch# show copp status
```

Monitoring CoPP

SUMMARY STEPS

1. switch# **show policy-map interface control-plane**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show policy-map interface control-plane**	Displays packet-level statistics for all classes that are part of the applied CoPP policy.

Example

This example shows how to monitor CoPP:

```
switch# show policy-map interface control-plane
Control Plane

service-policy  input: copp-system-policy

  class-map copp-s-default (match-any)
    police pps 400 , bc 0 packets
      HW Matched Packets    0
      SW Matched Packets   0
  class-map copp-s-ping (match-any)
    match access-group name copp-system-acl-ping
    police pps 100 , bc 0 packets
      HW Matched Packets    0
      SW Matched Packets   0
....
```

Clearing the CoPP Statistics

SUMMARY STEPS

1. (Optional) switch# **show policy-map interface control-plane**
2. switch# **clear copp statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) switch# **show policy-map interface control-plane**	Displays the currently applied CoPP policy and per-class statistics.
Step 2	switch# **clear copp statistics**	Clears the CoPP statistics.

Example

This example shows how to clear the CoPP statistics for your installation:

```
switch# show policy-map interface control-plane
switch# clear copp statistics
```

CoPP Configuration Examples

Creating an IP ACL

```
ip access-list  copp-sample-acl
permit udp any any eq 3333
permit udp any any eq 4444
```

The following example shows how to modify the CoPP Policy to drop all IP-in-IP (Protocol 4) packets immediately if there is not an operational tunnel that matches the incoming packet. Create copp-s-ipinip before the default copp-s-selfip policy as displayed in the following example.

```
ip access-list copp-s-ipinip
10 permit 4 any any
class-map type control-plane match-any copp-s-ipinip
match access-group name copp-s-ipinip
policy-map type control-plane copp-system-policy
class copp-s-ipinip
police pps 0
class copp-s-selfIp
police pps 500
class copp-s-default
police pps 400
```

Creating a Sample CoPP Class with an Associated IP ACL

The following example shows how to create a new CoPP class and associated ACL:

```
class-map type control-plane copp-sample-class
match access-group name copp-sample-acl
```

The following example shows how to add a class to a CoPP policy:

```
policy-map type control-plane copp-system-policy
Class copp-sample-class
 Police pps 100
```

The following example shows how to modify the PPS for an existing class (copp-s-bpdu):

```
policy-map type control-plane copp-system-policy
 Class copp-s-bpdu
 Police pps <new_pps_value>
```

Associating an ACL with an Existing or New CoPP Class

The following example shows how to associate an ACL with an existing or new CoPP class:

```
class-map type control-plane copp-s-eigrp
match access-grp name copp-system-acl-eigrp6
```

Adding a Class to a CoPP Policy

The following example shows how to add a class to a CoPP policy, if the class has not already been added:

```
policy-map type control-plane copp-system-policy
class copp-s-eigrp
police pps 100
```

Creating an ARP ACL-Based Dynamic Class

ARP ACLs use ARP TCAM. The default size of this TCAM is 0. Before ARP ACLs can be used with CoPP, this TCAM needs to be carved for a non-zero size.

```
hardware profile tcam region arpacl 128
copy running-config startup-config
reload
```

Creating an ARP ACL

```
arp access-list copp-arp-acl
permit ip 20.1.1.1 255.255.255.0 mac any
```

The procedure to associate an ARP ACLs with a class, and adding that class to the CoPP policy, is the same as the procedure for IP ACLs.

Creating a CoPP Class and Associating an ARP ACL

```
class-map type control-plane copp-sample-class
match access-group name copp-arp-acl
```

Removing a Class from a CoPP Policy

```
policy-map type control-plane copp-system-policy
   no class-abc
```

Removing a Class from the System

```
no class-map type control-plane copp-abc
```

Displaying the control plane class map configuration

```
show class-map type control-plane copp-s-pim-datareg
class-map type control-plane match-any copp-s-pim-datareg
```

The following example shows the interface control plane information of the copp-s-pim-datareg class:

```
switch# sh policy-map interface control-plane class copp-s-pim-datareg

       Control Plane

  service-policy  input: copp-system-policy

    class-map copp-s-pim-datareg (match-any)
      police pps 600 , bc 0 packets
        HW Matched Packets    55753
        SW Matched Packets   33931

switch#
```

Using the insert-before option to see if a packet matches multiple classes and the priority needs to be assigned to one of them

```
policy-map type control-plan copp-system-policy
class copp-ping insert-before copp-icmp
```

Sample CoPP Configuration

The following example shows a sample CoPP configuration with ACLs, classes, policies, and individual class policing:

```
IP access list copp-system-acl-eigrp
        10 permit eigrp any 224.0.0.10/32
IP access list copp-system-acl-icmp
        10 permit icmp any any
IP access list copp-system-acl-igmp
        10 permit igmp any any
IP access list copp-system-acl-ntp
        10 permit udp any any eq ntp
        20 permit udp any eq ntp any
IP access list copp-system-acl-pimreg
        10 permit pim any any
IP access list copp-system-acl-ping
        10 permit icmp any any echo
        20 permit icmp any any echo-reply
IP access list copp-system-acl-routingproto1
        10 permit tcp any gt 1024 any eq bgp
        20 permit tcp any eq bgp any gt 1024
        30 permit udp any 224.0.0.0/24 eq rip
        40 permit tcp any gt 1024 any eq 639
        50 permit tcp any eq 639 any gt 1024
        70 permit ospf any any
        80 permit ospf any 224.0.0.5/32
        90 permit ospf any 224.0.0.6/32
IP access list copp-system-acl-routingproto2
        10 permit udp any 224.0.0.0/24 eq 1985
        20 permit 112 any 224.0.0.0/24
IP access list copp-system-acl-snmp
        10 permit udp any any eq snmp
        20 permit udp any any eq snmptrap
IP access list copp-system-acl-ssh
```

```
        10 permit tcp any any eq 22
        20 permit tcp any eq 22 any
IP access list copp-system-acl-stftp
        10 permit udp any any eq tftp
        20 permit udp any any eq 1758
        30 permit udp any eq tftp any
        40 permit udp any eq 1758 any
        50 permit tcp any any eq 115
        60 permit tcp any eq 115 any
IP access list copp-system-acl-tacacsradius
        10 permit tcp any any eq tacacs
        20 permit tcp any eq tacacs any
        30 permit udp any any eq 1812
        40 permit udp any any eq 1813
        50 permit udp any any eq 1645
        60 permit udp any any eq 1646
        70 permit udp any eq 1812 any
        80 permit udp any eq 1813 any
        90 permit udp any eq 1645 any
        100 permit udp any eq 1646 any
IP access list copp-system-acl-telnet
        10 permit tcp any any eq telnet
        20 permit tcp any any eq 107
        30 permit tcp any eq telnet any
        40 permit tcp any eq 107 any
IP access list copp-system-dhcp-relay
        10 permit udp any eq bootps any eq bootps
IP access list test
        statistics per-entry
        10 permit ip 1.2.3.4/32 5.6.7.8/32 [match=0]
        20 permit udp 11.22.33.44/32 any [match=0]
        30 deny udp 1.1.1.1/32 any [match=0]

class-map type control-plane match-any copp-icmp
  match access-group name copp-system-acl-icmp
class-map type control-plane match-any copp-ntp
  match access-group name copp-system-acl-ntp
class-map type control-plane match-any copp-s-arp
class-map type control-plane match-any copp-s-bfd
class-map type control-plane match-any copp-s-bpdu
class-map type control-plane match-any copp-s-dai
class-map type control-plane match-any copp-s-default
class-map type control-plane match-any copp-s-dhcpreq
  match access-group name copp-system-acl-dhcps6
class-map type control-plane match-any copp-s-dhcpresp
  match access-group name copp-system-acl-dhcpc6
  match access-group name copp-system-dhcp-relay
class-map type control-plane match-any copp-s-eigrp
  match access-group name copp-system-acl-eigrp
  match access-group name copp-system-acl-eigrp6
class-map type control-plane match-any copp-s-glean
class-map type control-plane match-any copp-s-igmp
  match access-group name copp-system-acl-igmp
class-map type control-plane match-any copp-s-ipmcmiss
class-map type control-plane match-any copp-s-l2switched
class-map type control-plane match-any copp-s-l3destmiss
class-map type control-plane match-any copp-s-l3mtufail
class-map type control-plane match-any copp-s-l3slowpath
class-map type control-plane match-any copp-s-pimautorp
class-map type control-plane match-any copp-s-pimreg
  match access-group name copp-system-acl-pimreg
class-map type control-plane match-any copp-s-ping
  match access-group name copp-system-acl-ping
class-map type control-plane match-any copp-s-ptp
```

```
class-map type control-plane match-any copp-s-routingProto1
  match access-group name copp-system-acl-routingproto1
  match access-group name copp-system-acl-v6routingproto1
class-map type control-plane match-any copp-s-routingProto2
  match access-group name copp-system-acl-routingproto2
class-map type control-plane match-any copp-s-selfIp
class-map type control-plane match-any copp-s-ttl1
class-map type control-plane match-any copp-s-v6routingProto2
  match access-group name copp-system-acl-v6routingProto2
class-map type control-plane match-any copp-snmp
  match access-group name copp-system-acl-snmp
class-map type control-plane match-any copp-ssh
  match access-group name copp-system-acl-ssh
class-map type control-plane match-any copp-stftp
  match access-group name copp-system-acl-stftp
class-map type control-plane match-any copp-tacacsradius
  match access-group name copp-system-acl-tacacsradius
class-map type control-plane match-any copp-telnet
  match access-group name copp-system-acl-telnet
policy-map type control-plane copp-system-policy
  class copp-s-selfIp
    police pps 500
  class copp-s-default
    police pps 400
  class copp-s-l2switched
    police pps 200
  class copp-s-ping
    police pps 100
  class copp-s-l3destmiss
    police pps 100
  class copp-s-glean
    police pps 500
  class copp-s-l3mtufail
    police pps 100
  class copp-s-ttl1
    police pps 100
  class copp-s-ipmcmiss
    police pps 400
  class copp-s-l3slowpath
    police pps 100
  class copp-s-dhcpreq
    police pps 300
  class copp-s-dhcpresp
    police pps 300
  class copp-s-dai
    police pps 300
  class copp-s-igmp
    police pps 400
  class copp-s-routingProto2
    police pps 1300
  class copp-s-v6routingProto2
    police pps 1300
  class copp-s-eigrp
    police pps 200
  class copp-s-pimreg
    police pps 200
  class copp-s-pimautorp
    police pps 200
  class copp-s-routingProto1
    police pps 1000
  class copp-s-arp
    police pps 200
  class copp-s-ptp
    police pps 1000
```

```
  class copp-s-bfd
    police pps 350
  class copp-s-bpdu
    police pps 12000
  class copp-icmp
    police pps 200
  class copp-telnet
    police pps 500
  class copp-ssh
    police pps 500
  class copp-snmp
    police pps 500
  class copp-ntp
    police pps 100
  class copp-tacacsradius
    police pps 400
  class copp-stftp
    police pps 400
control-plane
  service-policy input copp-system-policy
```

Example: Changing or Reapplying the Default CoPP Policy Using the Setup Utility

The following example shows how to change or reapply the default CoPP policy using the setup utility:

```
switch# setup

         ---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.

*Note: setup is mainly used for configuring the system initially,
when no configuration is present. So setup always assumes system
defaults and not the current system configuration values.

Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.

Would you like to enter the basic configuration dialog (yes/no): yes

  Create another login account (yes/no) [n]: n

  Configure read-only SNMP community string (yes/no) [n]: n

  Configure read-write SNMP community string (yes/no) [n]: n

  Enter the switch name : switch

  Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: n

  Configure the default gateway for mgmt? (yes/no) [y]: n

  Enable the telnet service? (yes/no) [n]: y

  Enable the ssh service? (yes/no) [y]: n

  Configure the ntp server? (yes/no) [n]: n
```

```
  Configure CoPP System Policy Profile ( default / l2 / l3 ) [default]: l2

The following configuration will be applied:
  switchname switch
  telnet server enable
  no ssh server enable
  policy-map type control-plane copp-system-policy ( l2 )

Would you like to edit the configuration? (yes/no) [n]: n

Use this configuration and save it? (yes/no) [y]: y

[########################################] 100%
```

INDEX

A

C

D

E

F

G

H

I

L

M

P

R

S

T

U

V