Security Configuration Guide, Cisco IOS XE Everest 16.6.x (Catalyst 9400 Switches)

First Published: 2017-07-31

Last Modified: 2017-11-03

CONTENTS

Preventing Unauthorized Access

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Preventing Unauthorized Access

You can prevent unauthorized users from reconfiguring your switch and viewing configuration information. Typically, you want network administrators to have access to your switch while you restrict access to users who dial from outside the network through an asynchronous port, connect from outside the network through a serial port, or connect through a terminal or workstation from within the local network.

To prevent unauthorized access into your switch, you should configure one or more of these security features:

- At a minimum, you should configure passwords and privileges at each switch port. These passwords are locally stored on the switch. When users attempt to access the switch through a port or line, they must enter the password specified for the port or line before they can access the switch.

- For an additional layer of security, you can also configure username and password pairs, which are locally stored on the switch. These pairs are assigned to lines or ports and authenticate each user before that user can access the switch. If you have defined privilege levels, you can also assign a specific privilege level (with associated rights and privileges) to each username and password pair.

- If you want to use username and password pairs, but you want to store them centrally on a server instead of locally, you can store them in a database on a security server. Multiple networking devices can then use the same database to obtain user authentication (and, if necessary, authorization) information.

- You can also enable the login enhancements feature, which logs both failed and unsuccessful login attempts. Login enhancements can also be configured to block future login attempts after a set number of unsuccessful attempts are made. For more information, see the Cisco IOS Login Enhancements documentation.

Feature Information for Preventing Unauthorized Access

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 1: Feature Information for Preventing Unauthorized Access

Feature Name	Releases	Feature Information
Preventing Unauthorized Access	Cisco IOS XE Everest 16.6.1	You can prevent unauthorized users from reconfiguring your switch and viewing configuration information.

Controlling Switch Access with Passwords and Privilege Levels

Restrictions for Controlling Switch Access with Passwords and Privileges

The following are the restrictions for controlling switch access with passwords and privileges:

- Disabling password recovery will not work if you have set the switch to boot up manually by using the boot manual global configuration command. This command produces the boot loader prompt (switch:) after the switch is power cycled.

- Password type 0 and type 7 are deprecated. So password type 0 and type 7, used for administrator login to Console, Telnet, SSH, webUI, and NETCONF, must be migrated to password type 8 or type 9.

- No action is required if username and password are type 0 and type 7 for local authentication such as CHAP, EAP and so on for ISG and Dot1x.

- Enable password type 0 and type 7 must be migrated to password type 8 or type 9.

- Password type 5 is deprecated. Password type 5 must be migrated to stronger password type 8 or type 9.

Information About Passwords and Privilege Levels

Default Password and Privilege Level Configuration

A simple way of providing terminal access control in your network is to use passwords and assign privilege levels. Password protection restricts access to a network or network device. Privilege levels define what commands users can enter after they have logged into a network device.

This table shows the default password and privilege level configuration.

Table 2: Default Password and Privilege Levels

Feature	Default Setting
Enable password and privilege level	No password is defined. The default is level 15 (privileged EXEC level). The password is not encrypted in the configuration file.
Enable secret password and privilege level	No password is defined. The default is level 15 (privileged EXEC level). The password is encrypted before it is written to the configuration file.
Line password	No password is defined.

Additional Password Security

To provide an additional layer of security, particularly for passwords that cross the network or that are stored on a Trivial File Transfer Protocol (TFTP) server, you can use either the enable password or enable secret global configuration commands. Both commands accomplish the same thing; that is, you can establish an encrypted password that users must enter to access privileged EXEC mode (the default) or any privilege level you specify.

We recommend that you use the enable secret command because it uses an improved encryption algorithm.

If you configure the enable secret command, it takes precedence over the enable password command; the two commands cannot be in effect simultaneously.

If you enable password encryption, it applies to all passwords including username passwords, authentication key passwords, the privileged command password, and console and virtual terminal line passwords.

Password Recovery

By default, any end user with physical access to the switch can recover from a lost password by interrupting the boot process while the switch is powering on and then by entering a new password.

The password-recovery disable feature protects access to the switch password by disabling part of this functionality. When this feature is enabled, the end user can interrupt the boot process only by agreeing to set the system back to the default configuration. With password recovery disabled, you can still interrupt the boot process and change the password, but the configuration file (config.text) and the VLAN database file (vlan.dat) are deleted.

If you disable password recovery, we recommend that you keep a backup copy of the configuration file on a secure server in case the end user interrupts the boot process and sets the system back to default values. Do not keep a backup copy of the configuration file on the switch. If the switch is operating in VTP transparent mode, we recommend that you also keep a backup copy of the VLAN database file on a secure server. When the switch is returned to the default system configuration, you can download the saved files to the switch by using the Xmodem protocol.

To re-enable password recovery, use the service password-recovery global configuration command.

Terminal Line Telnet Configuration

When you power-up your switch for the first time, an automatic setup program runs to assign IP information and to create a default configuration for continued use. The setup program also prompts you to configure your switch for Telnet access through a password. If you did not configure this password during the setup program, you can configure it when you set a Telnet password for a terminal line.

Username and Password Pairs

You can configure username and password pairs, which are locally stored on the switch. These pairs are assigned to lines or ports and authenticate each user before that user can access the switch. If you have defined privilege levels, you can also assign a specific privilege level (with associated rights and privileges) to each username and password pair.

Privilege Levels

Cisco devices use privilege levels to provide password security for different levels of switch operation. By default, the Cisco IOS software operates in two modes (privilege levels) of password security: user EXEC (Level 1) and privileged EXEC (Level 15). You can configure up to 16 hierarchical levels of commands for each mode. By configuring multiple passwords, you can allow different sets of users to have access to specified commands.

Privilege Levels on Lines

Users can override the privilege level you set using the privilege level line configuration command by logging in to the line and enabling a different privilege level. They can lower the privilege level by using the disable command. If users know the password to a higher privilege level, they can use that password to enable the higher privilege level. You might specify a high level or privilege level for your console line to restrict line usage.

For example, if you want many users to have access to the clear line command, you can assign it level 2 security and distribute the level 2 password fairly widely. But if you want more restricted access to the configure command, you can assign it level 3 security and distribute that password to a more restricted group of users.

Command Privilege Levels

When you set a command to a privilege level, all commands whose syntax is a subset of that command are also set to that level. For example, if you set the show ip traffic command to level 15, the show commands and show ip commands are automatically set to privilege level 15 unless you set them individually to different levels.

How to Control Switch Access with Passwords and Privilege Levels

Setting or Changing a Static Enable Password

The enable password controls access to the privileged EXEC mode. Follow these steps to set or change a static enable password:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	enable password password **Example:** Device(config)# **enable password secret321**	Defines a new password or changes an existing password for access to privileged EXEC mode. By default, no password is defined. For password, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, is case sensitive, and allows spaces but ignores leading spaces. It can contain the question mark (?) character if you precede the question mark with the key combination Crtl-v when you create the password; for example, to create the password abc?123, do this: a. Enter abc. b. Enter Crtl-v. c. Enter ?123. When the system prompts you to enter the enable password, you need not precede the question mark with the Ctrl-v; you can simply enter abc?123 at the password prompt.

	Command or Action	**Purpose**
Step 4	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 5	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Protecting Enable and Enable Secret Passwords with Encryption

Follow these steps to establish an encrypted password that users must enter to access privileged EXEC mode (the default) or any privilege level you specify:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	Use one of the following: • enable password [level level] password encryption-type encrypted-password • enable secret [level level] password encryption-type encrypted-password	• Defines a new password or changes an existing password for access to privileged EXEC mode. • Defines a secret password, which is saved using a nonreversible encryption method. • (Optional) For level, the range is from 0 to 15. Level 1 is normal user EXEC

		Command or Action	Purpose
		Example: Device(config)# **enable password example102** or Device(config)# **enable secret level 1 password secret123sample**	mode privileges. The default level is 15 (privileged EXEC mode privileges). • For password, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, is case sensitive, and allows spaces but ignores leading spaces. By default, no password is defined. • (Optional) For encryption-type, only type 5, a Cisco proprietary encryption algorithm, is available. If you specify an encryption type, you must provide an encrypted password—an encrypted password that you copy from another switch configuration. **Note** If you specify an encryption type and then enter a clear text password, you can not re-enter privileged EXEC mode. You cannot recover a lost encrypted password by any method.
Step 4		service password-encryption **Example:** Device(config)# **service password-encryption**	(Optional) Encrypts the password when the password is defined or when the configuration is written. Encryption prevents the password from being readable in the configuration file.
Step 5		end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6		show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 7		copy running-config startup-config **Example:**	(Optional) Saves your entries in the configuration file.

Command or Action	Purpose
Device# **copy running-config startup-config**	

Disabling Password Recovery

Follow these steps to disable password recovery to protect the security of your switch:

Before you begin

If you disable password recovery, we recommend that you keep a backup copy of the configuration file on a secure server in case the end user interrupts the boot process and sets the system back to default values. Do not keep a backup copy of the configuration file on the switch. If the switch is operating in VTP transparent mode, we recommend that you also keep a backup copy of the VLAN database file on a secure server. When the switch is returned to the default system configuration, you can download the saved files to the switch by using the Xmodem protocol.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	system disable password recovery switch {all | <1-9>} **Example:** Device(config)# **system disable password recovery switch all**	Disables password recovery. • all - Sets the configuration on switches in stack. • <1-9> - Sets the configuration on the Switch Number selected. This setting is saved in an area of the flash memory that is accessible by the boot loader and the Cisco IOS image, but it is not part of the file system and is not accessible by any user.
Step 4	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

What to do next

To remove disable password recovery, use the no system disable password recovery switch all global configuration command.

Setting a Telnet Password for a Terminal Line

Beginning in user EXEC mode, follow these steps to set a Telnet password for the connected terminal line:

Before you begin

- Attach a PC or workstation with emulation software to the switch console port, or attach a PC to the Ethernet management port.

- The default data characteristics of the console port are 9600, 8, 1, no parity. You might need to press the Return key several times to see the command-line prompt.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	**Note** If a password is required for access to privileged EXEC mode, you will be prompted for it. Enters privileged EXEC mode.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	line vty 0 15 **Example:** Device(config)# **line vty 0 15**	Configures the number of Telnet sessions (lines), and enters line configuration mode. There are 16 possible sessions on a command-capable Device. The 0 and 15 mean that you are configuring all 16 possible Telnet sessions.
Step 4	password password **Example:** Device(config-line)# **password abcxyz543**	Sets a Telnet password for the line or lines. For password, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, is case sensitive, and allows spaces but ignores leading spaces. By default, no password is defined.
Step 5	end **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config-line)# **end**	
Step 6	show running-config	Verifies your entries.
	Example:	
	Device# **show running-config**	
Step 7	copy running-config startup-config	(Optional) Saves your entries in the configuration file.
	Example:	
	Device# **copy running-config startup-config**	

Configuring Username and Password Pairs

Follow these steps to configure username and password pairs:

Procedure

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example:	• Enter your password if prompted.
	Device> **enable**	
Step 2	configure terminal	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	username name [privilege level] {password encryption-type password}	Sets the username, privilege level, and password for each user.
	Example:	• For name, specify the user ID as one word or the MAC address. Spaces and quotation marks are not allowed.
	Device(config)# **username adamsample privilege 1 password secret456**	• You can configure a maximum of 12000 clients each, for both username and MAC filter.
	Device(config)# **username 111111111111 mac attribute**	• (Optional) For level, specify the privilege level the user has after gaining access. The range is 0 to 15. Level 15 gives privileged

	Command or Action	Purpose
		EXEC mode access. Level 1 gives user EXEC mode access.
		• For encryption-type, enter 0 to specify that an unencrypted password will follow. Enter 7 to specify that a hidden password will follow.
		• For password, specify the password the user must enter to gain access to the Device. The password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the username command.
Step 4	Use one of the following: • line console 0 • line vty 0 15 **Example:** Device(config)# **line console 0** or Device(config)# **line vty 15**	Enters line configuration mode, and configures the console port (line 0) or the VTY lines (line 0 to 15).
Step 5	login local **Example:** Device(config-line)# **login local**	Enables local password checking at login time. Authentication is based on the username specified in Step 3.
Step 6	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 7	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 8	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Setting the Privilege Level for a Command

Follow these steps to set the privilege level for a command:

Procedure

	Command or Action	**Purpose**
Step 1	enable	Enables privileged EXEC mode.
	Example:	• Enter your password if prompted.
	Device> **enable**	
Step 2	configure terminal	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	privilege mode level level command	Sets the privilege level for a command.
	Example:	• For mode, enter configure for global configuration mode, exec for EXEC mode, interface for interface configuration mode, or line for line configuration mode.
	Device(config)# **privilege exec level 14 configure**	
		• For level, the range is from 0 to 15. Level 1 is for normal user EXEC mode privileges. Level 15 is the level of access permitted by the enable password.
		• For command, specify the command to which you want to restrict access.
Step 4	enable password level level password	Specifies the password to enable the privilege level.
	Example:	• For level, the range is from 0 to 15. Level 1 is for normal user EXEC mode privileges.
	Device(config)# **enable password level 14 SecretPswd14**	
		• For password, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, is case sensitive, and allows spaces but ignores leading spaces. By default, no password is defined.
Step 5	end	Returns to privileged EXEC mode.
	Example:	

	Command or Action	Purpose
	Device(config)# **end**	
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Changing the Default Privilege Level for Lines

Follow these steps to change the default privilege level for the specified line:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	line vty line **Example:** Device(config)# **line vty 10**	Selects the virtual terminal line on which to restrict access.
Step 4	privilege level level **Example:** Device(config)# **privilege level 15**	Changes the default privilege level for the line. For level, the range is from 0 to 15. Level 1 is for normal user EXEC mode privileges. Level 15 is the level of access permitted by the enable password.
Step 5	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

What to do next

Users can override the privilege level you set using the privilege level line configuration command by logging in to the line and enabling a different privilege level. They can lower the privilege level by using the disable command. If users know the password to a higher privilege level, they can use that password to enable the higher privilege level. You might specify a high level or privilege level for your console line to restrict line usage.

Logging into and Exiting a Privilege Level

Beginning in user EXEC mode, follow these steps to log into a specified privilege level and exit a specified privilege level.

Procedure

	Command or Action	Purpose
Step 1	enable level **Example:** Device> **enable 15**	Logs in to a specified privilege level. Following the example, Level 15 is privileged EXEC mode. For level, the range is 0 to 15.
Step 2	disable level **Example:** Device# **disable 1**	Exits to a specified privilege level. Following the example, Level 1 is user EXEC mode. For level, the range is 0 to 15.

Monitoring Switch Access

Table 3: Commands for Displaying DHCP Information

show privilege	Displays the privilege level configuration.

Configuration Examples for Setting Passwords and Privilege Levels

Example: Setting or Changing a Static Enable Password

This example shows how to change the enable password to l1u2c3k4y5. The password is not encrypted and provides access to level 15 (traditional privileged EXEC mode access):

```
Device(config)# enable password l1u2c3k4y5
```

Example: Protecting Enable and Enable Secret Passwords with Encryption

This example shows how to configure the encrypted password 1FaD0$Xyti5Rkls3LoyxzS8 for privilege level 2:

```
Device(config)# enable secret level 2 5 $1$FaD0$Xyti5Rkls3LoyxzS8
```

Example: Setting a Telnet Password for a Terminal Line

This example shows how to set the Telnet password to let45me67in89:

```
Device(config)# line vty 10
Device(config-line)# password let45me67in89
```

Example: Setting the Privilege Level for a Command

This example shows how to set the configure command to privilege level 14 and define SecretPswd14 as the password users must enter to use level 14 commands:

```
Device(config)# privilege exec level 14 configure
Device(config)# enable password level 14 SecretPswd14
```

Additional References

Error Message Decoder

Description	Link
To help you research and resolve system error messages in this release, use the Error Message Decoder tool.	https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History for Controlling Switch Access with Passwords and Privileges

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	Controlling Switch Access with Passwords and Privileges	Password protection restricts access to a network or network device. Privilege levels define what commands users can enter after they have logged into a network device.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

C H A P T E R 3

Configuring Authentication

Authentication provides a method to identify users, which includes the login and password dialog, challenge and response, messaging support, and encryption, depending on the selected security protocol. Authentication is the way a user is identified prior to being allowed access to the network and network services.

Prerequisites for Configuring Authentication

The implementation of authentication is divided into Authentication, Authorization, and Accounting (AAA) authentication and nonauthentication methods. Cisco recommends that, whenever possible, AAA security services be used to implement authentication.

Restrictions for Configuring Authentication

- The number of AAA method lists that can be configured is 250.

- If you configure one RADIUS server with the nonstandard option and another RADIUS server without the nonstandard option, the RADIUS-server host with the nonstandard option does not accept a predefined host. If you configure the same RADIUS server host IP address for a different UDP destination port for accounting requests by using the acct-port keyword and a UDP destination port for authentication requests by using the auth-port keyword with and without the nonstandard option, the RADIUS server does not accept the nonstandard option.

Information About Configuring Authentication

Named Method Lists for Authentication

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

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

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

Method Lists and Server Groups

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

Figure 1: Typical AAA Network Configuration
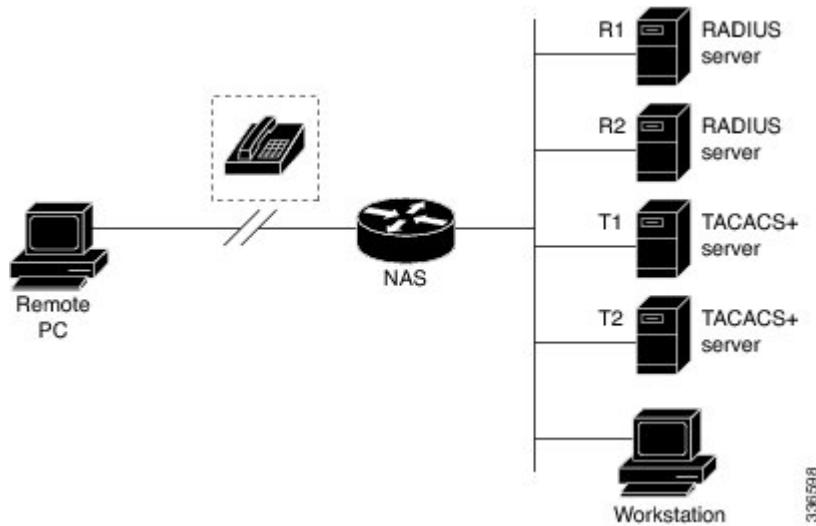
Using server groups, you can specify a subset of the configured server hosts and use them for a particular service. For example, server groups allow you to define R1 and R2 as a server group, and define T1 and T2 as a separate server group. For example, you can specify R1 and T1 in the method list for authentication login, while specifying R2 and T2 in the method list for PPP authentication.

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

For more information about configuring server groups and about configuring server groups based on Dialed Number Identification Service (DNIS) numbers, see the "Configuring RADIUS" or "Configuring TACACS+" chapters.

Example: Configuring Method Lists

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

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication ppp default group radius group tacacs+ local
Device(config)# exit
```

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

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

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

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

```
Device> enable
Device# configure terminal
```

```
Device(config)# Device(config)#
Device(config)# aaa authentication ppp server-group1 group radius group tacacs+ local none
Device(config)# interface gigabitethernet 1/0/3
Device(config-if)#  ppp authentication chap server-group1
Device(config-if)# end
```

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

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

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius rad2only
Device(config-sg-radius)# server 172.16.2.7
Device(config-sg-radius)# end
```

The TACACS+ server group "tac2only" is defined as follows using the aaa group server command:

```
Device> enable
Device# configure terminal
Device(config)# aaa group server tacacs+ tac2only
Device(config-sg-tacacs)# server 172.16.2.77
Device(config-sg-tacacs)# end
```

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

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication ppp default group rad2only group tac2only local
Device(config)# exit
```

If a method list is configured under VTY lines, the corresponding method list must be added to AAA. The following example shows how to configure a method list under a VTY line:

```
Device> enable
Device# configure terminal
Device(config)# line vty 0 4
Device(config-line)# authorization commands 15 auth1
Device(config-line)# exit
```

The following example shows how to configure a method list in AAA:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization commands 15 auth1 group tacacs+
Device(config)# exit
```

If no method list is configured under VTY lines, the default method list must be added to AAA. The following example shows a VTY configuration without a method list:

```
Device> enable
Device# configure terminal
Device(config)# line vty 0 4
Device(config-line)# end
```

The following example shows how to configure the default method list:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization commands 15 default group tacacs+
Device(config)# exit
```

About RADIUS Change of Authorization

A standard RADIUS interface is typically used in a pulled model, in which the request originates from a device attached to a network and the response is sent from the queried servers. The Cisco software supports the RADIUS CoA request defined in RFC 5176 that is used in a pushed model, in which the request originates from the external server to the device attached to the network, and enables the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

Use the following per-session CoA requests:

- Session reauthentication
- Session termination
- Session termination with port shutdown
- Session termination with port bounce
- Security and Password
- Accounting

CoA Requests

CoA requests, as described in RFC 5176, are used in a pushed model to allow for session identification, host reauthentication, and session termination. The model comprises one request (CoA-Request) and two possible response codes:

- CoA acknowledgment (ACK) [CoA-ACK]
- CoA non-acknowledgment (NAK) [CoA-NAK]

The request is initiated from a CoA client (typically a RADIUS or policy server) and directed to the device that acts as a listener.

RFC 5176 Compliance

The Disconnect Request message, which is also referred to as Packet of Disconnect (POD), is supported by the device for a session termination.

The following table shows the IETF attributes that are supported for the RADIUS Change of Authorization (CoA) feature.

Table 4: Supported IETF Attributes

Attribute Number	Attribute Name
24	State
31	Calling-Station-ID
44	Acct-Session-ID
80	Message-Authenticator
101	Error-Cause

The following table shows the possible values for the Error-Cause attribute.

Table 5: Error-Cause Values

Value	Explanation
201	Residual Session Context Removed
202	Invalid EAP Packet (Ignored)
401	Unsupported Attribute
402	Missing Attribute
403	NAS Identification Mismatch
404	Invalid Request
405	Unsupported Service
406	Unsupported Extension
407	Invalid Attribute Value
501	Administratively Prohibited
502	Request Not Routable (Proxy)
503	Session Context Not Found
504	Session Context Not Removable
505	Other Proxy Processing Error
506	Resources Unavailable
507	Request Initiated
508	Multiple Session Selection Unsupported

CoA Request Response Code

The CoA Request Response code can be used to issue a command to the device. The supported commands are listed in the "CoA Request Commands" section.

The packet format for a CoA Request Response code as defined in RFC 5176 consists of the following fields: Code, Identifier, Length, Authenticator, and Attributes in the Type:Length:Value (TLV) format.

The Attributes field is used to carry Cisco VSAs.

Session Identification

For disconnect and CoA requests targeted at a particular session, the device locates the session based on one or more of the following attributes:

- Acct-Session-Id (IETF attribute #44)

- Audit-Session-Id (Cisco vendor-specific attribute (VSA))

- Calling-Station-Id (IETF attribute #31, which contains the host MAC address)

Unless all session identification attributes included in the CoA message match the session, the device returns a Disconnect-NAK or CoA-NAK with the "Invalid Attribute Value" error-code attribute.

Note   A CoA NAK message is not sent for all CoA requests with a key mismatch. The message is sent only for the first three requests for a client. After that, all the packets from that client are dropped. When there is a key mismatch, the response authenticator sent with the CoA NAK message is calculated from a dummy key value.

CoA ACK Response Code

If an authorization state is changed successfully, a positive acknowledgment (ACK) is sent. The attributes returned within a CoA ACK can vary based on the CoA Request.

CoA NAK Response Code

A negative acknowledgment (NAK) indicates a failure to change the authorization state and can include attributes that indicate the reason for the failure.

CoA Request Commands

The commands supported on the device are shown in the table below. All CoA commands must include the session identifier between the device and the CoA client.

Table 6: CoA Request Commands Supported on the Device

Command	Cisco VSA
Bounce host port	Cisco:Avpair="subscriber:command=bounce-host-port"
Disable host port	Cisco:Avpair="subscriber:command=disable-host-port"
Reauthenticate host	Cisco:Avpair="subscriber:command=reauthenticate"
Terminate session	This is a standard disconnect request that does not require a VSA

Session Reauthentication

To initiate session reauthentication, the authentication, authorization, and accounting (AAA) server sends a standard CoA-Request message that contains a Cisco VSA and one or more session identification attributes. The Cisco VSA is in the form of Cisco:Avpair="subscriber:command=reauthenticate".

The current session state determines the device's response to the message in the following scenarios:

- If the session is currently authenticated by IEEE 802.1x, the device responds by sending an Extensible Authentication Protocol over LAN (EAPoL)-RequestId message to the server.

- If the session is currently authenticated by MAC authentication bypass (MAB), the device sends an access request to the server, passing the same identity attributes used for the initial successful authentication.

- If session authentication is in progress when the device receives the command, the device terminates the process and restarts the authentication sequence, starting with the method configured to be attempted first.

Session Termination

A CoA Disconnect-Request terminates the session without disabling the host port. CoA Disconnect-Request termination causes reinitialization of the authenticator state machine for the specified host, but does not restrict the host's access to the network. If the session cannot be located, the device returns a Disconnect-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the device terminates the session. After the session has been completely removed, the device returns a Disconnect-ACK message.

To restrict a host's access to the network, use a CoA Request with the Cisco:Avpair="subscriber:command=disable-host-port" VSA. This command is useful when a host is known to cause problems on the network and network access needs to be immediately blocked for the host. If you want to restore network access on the port, reenable it using a non-RADIUS mechanism.

CoA Request Disable Host Port

The RADIUS server CoA disable port command administratively shuts down the authentication port that is hosting a session, resulting in session termination. This command is useful when a host is known to cause problems on the network and network access needs to be immediately blocked for the host. If you want to restore network access on the port, reenable it using a non-RADIUS mechanism. This command is carried in a standard CoA-Request message that has the following VSA:

Cisco:Avpair="subscriber:command=disable-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification" section. If the device cannot locate the session, it returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the device locates the session, it disables the hosting port and returns a CoA-ACK message.

If the device fails before returning a CoA-ACK to the client, the process is repeated on the new active device when the request is re-sent from the client. If the device fails after returning a CoA-ACK message to the client but before the operation is complete, the operation is restarted on the new active device.

To ignore the RADIUS server CoA disable port command, see the "Configuring the Device to Ignore Bounce and Disable RADIUS CoA Requests" section.

CoA Request Bounce Port

A RADIUS server CoA bounce port sent from a RADIUS server can cause a link flap on an authentication port, which triggers DHCP renegotiation from one or more hosts connected to this port. This incident can occur when there is a VLAN change and the endpoint is a device (such as a printer) that does not have a mechanism to detect a change on this authentication port. The CoA bounce port is carried in a standard CoA-Request message that contains the following VSA:

Cisco:Avpair="subscriber:command=bounce-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the Session Identification. If the session cannot be located, the device returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the device disables the hosting port for a period of 10 seconds, reenables it (port-bounce), and returns a CoA-ACK.

To ignore the RADIUS server CoA bounce port, see the "Configuring the Device to Ignore Bounce and Disable RADIUS CoA Requests" section.

Domain Stripping

The AAA Broadcast Accounting feature allows accounting information to be sent to multiple AAA servers at the same time, that is, accounting information can be broadcast to one or more AAA servers simultaneously. This functionality allows you to send accounting information to private and public AAA servers. It also provides redundant billing information for voice applications.

The Domain Stripping feature allows domain stripping to be configured at the server group level.

Per-server group configuration overrides the global configuration. If domain stripping is not enabled globally, but it is enabled in a server group, then it is enabled only for that server group. Also, if virtual routing and forwarding (VRF)-specific domain stripping is configured globally and in a server group for a different VRF, domain stripping is enabled in both the VRFs. VRF configurations are taken from server-group configuration mode. If server-group configurations are disabled in global configuration mode but are available in server-group configuration mode, all configurations in server-group configuration mode are applicable.

After the domain stripping and broadcast accounting are configured, you can create separate accounting records as per the configurations.

If both domain-stripping and directed-request commands are enabled, domain stripping takes precedence and directed request functionality will not work.

How to Configure AAA Authentication Methods

Configuring Login Authentication Using AAA

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

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

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	aaa new-model **Example:** Device(config)# **aaa new-model**	Enables AAA.
Step 4	aaa authentication login{default | list-name} method1[method2...] **Example:** Device(config)# **aaa authentication login default local**	Creates a local authentication list.
Step 5	line [aux | console | tty | vty] line-number [ending-line-number] **Example:** Device(config)# **line vty 1**	Enters line configuration mode for the lines to which you want to apply the authentication list.
Step 6	login authentication {default | list-name} **Example:** Device(config-line)# **login authentication default**	Applies the authentication list to a line or set of lines.
Step 7	end **Example:** Device(config-line)# **end**	Exits line configuration mode and returns to privileged EXEC mode.

What to do next

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

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

```
Device(config)# aaa authentication login default group tacacs+ none
```

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

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

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

```
Device(config)# aaa authentication login default group radius
```

Login Authentication Using Enable Password

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

```
Device(config)# aaa authentication login default enable
```

Before you can use the enable password as the login authentication method, you need to define the enable password. For more information about defining enable passwords, see chapter "Controlling Switch Access with Passwords and Privilege Levels."

Login Authentication Using Kerberos

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

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

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

```
Device(config)# aaa authentication login default krb5
```

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

Login Authentication Using Line Password

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

```
Device(config)# aaa authentication login default line
```

Before you can use a line password as the login authentication method, you need to define a line password.

Login Authentication Using Local Password

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

```
Device(config)# aaa authentication login default local
```

Login Authentication Using Group RADIUS

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

```
Device(config)# aaa authentication login default group radius
```

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

RADIUS Attribute 8 in Access Requests

After you have used the aaa authentication login command to specify RADIUS and your login host has been configured to request its IP address from the NAS, you can send attribute 8 (Framed-IP-Address) in access-request packets by using the radius-server attribute 8 include-in-access-req command in global configuration mode. This command makes it possible for NAS to provide the RADIUS server a hint of the user IP address in advance for user authentication.

Login Authentication Using Group TACACS

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

```
Device(config)# aaa authentication login default group tacacs+
```

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

Login Authentication Using Group Name

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

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius loginrad
Device(config-sg-radius)# server 172.16.2.3
Device(config-sg-radius)# server 172.16.2 17
Device(config-sg-radius)# server 172.16.2.32
Device(config-sg-radius)# end
```

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

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

```
Device(config)# aaa authentication login default group loginrad
```

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

Configuring PPP Authentication Using AAA

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

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

Procedure

	Command or Action	**Purpose**
Step 1	enable	Enables privileged EXEC mode.
	Example:	• Enter your password if prompted.
	Device> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example:	

	Command or Action	Purpose
	Device# configure terminal	
Step 3	aaa new-model **Example:** Device(config)# **aaa new-model**	Enables AAA.
Step 4	aaa authentication ppp{default	list-name} method1[method2...] **Example:** Device(config)# **aaa authentication ppp-auth default local**	Creates a local authentication list.
Step 5	interface interface-type interface-number **Example:** Device(config)# **interface gigabitethernet 0/1/0**	Enters interface configuration mode for the interface to which you want to apply the authentication list.
Step 6	ppp authentication {protocol1 [protocol2...]} [if-needed] {default	list-name} [callin] [one-time][optional] **Example:** Device(config)# **ppp authentication ms-chap ppp-auth**	Applies the authentication list to a line or set of lines. In this command, protocol1 and protocol2 represent the following protocols: CHAP, MS-CHAP, and PAP. PPP authentication is attempted first using the first authentication method, specified by protocol1. If protocol1 is unable to establish authentication, the next configured protocol is used to negotiate authentication.
Step 7	end **Example:** Device(config)# **end**	Exits global configuration mode and returns to privileged EXEC mode.

What to do next

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

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

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

```
Device(config)# aaa authentication ppp default local
```

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

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

```
Device(config)# aaa authentication ppp default group tacacs+ none
```

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

PPP Authentication Using Kerberos

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

```
Device(config)# aaa authentication ppp default krb5
```

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

Note Kerberos login authentication works only with PPP PAP authentication.

PPP Authentication Using Local Password

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

```
Device(config)# aaa authentication ppp default local
```

PPP Authentication Using Group RADIUS

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

```
Device(config)# aaa authentication ppp default group radius
```

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

RADIUS Attribute 44 in Access Requests

After you have used the aaa authentication ppp default group radius command to specify RADIUS as the login authentication method, you can configure your device to send attribute 44 (Acct-Session-ID) in access-request packets by using the radius-server attribute 44 include-in-access-req command in global configuration mode. This command allows the RADIUS daemon to track a call from the beginning to the end.

PPP Authentication Using Group TACACS

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

```
Device(config)# aaa authentication ppp default group tacacs+
```

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

PPP Authentication Using Group Name

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

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius ppprad
Device(config-sg-radius)# server 172.16.2.3
Device(config-sg-radius)# server 172.16.2 17
Device(config-sg-radius)# server 172.16.2.32
Device(config-sg-radius)# end
```

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

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

```
Device(config)# aaa authentication ppp default group ppprad
```

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

AAA Scalability for PPP Requests

You can configure and monitor the number of background processes allocated by the PPP manager in the network access server (NAS) to deal with AAA authentication and authorization requests. The AAA Scalability feature enables you to configure the number of processes used to handle AAA requests for PPP, thus increasing the number of users that can be simultaneously authenticated or authorized.

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

```
Device(config)# aaa processes 5000
```

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

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

Configuring ARAP Authentication Using AAA

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

Use the following commands starting in global configuration mode:

Procedure

	Command or Action	**Purpose**
Step 1	enable	Enables privileged EXEC mode.
	Example:	• Enter your password if prompted.
	Device> **enable**	
Step 2	configure terminal	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	aaa new-model	Enables AAA.
	Example:	
	Device(config)# **aaa new-model**	
Step 4	aaa authentication arap	
	Example:	
	Device(config)# **aaa authentication arap**	
	Example:	
	Enables authentication for ARAP users.	

	Command or Action	Purpose
Step 5	line number **Example:** Device(config)# **line 1**	(Optional) Changes to line configuration mode.
Step 6	Device(config-line)# autoselect arap **Example:** Device(config-line)# **auto-select arap**	(Optional) Enables autoselection of ARAP.
Step 7	autoselect during-login **Example:** Device(config-line)# **autoselect during-login**	(Optional) Starts the ARAP session automatically at user login.
Step 8	arap authentication list-name **Example:** Device(config-line)# **arap authentication arap-authen**	(Optional—not needed if default is used in the aaa authentication arap command) Enables TACACS+ authentication for ARAP on a line.
Step 9	end **Example:** Device(config-line)# **end**	Exits line configuration mode and returns to the privileged EXEC mode.

What to do next

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

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

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

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

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

```
Device(config)# aaa authentication arap default if-needed none
```

To create the same authentication method list for ARAP and name the list MIS-access, use the following command:

```
Device(config)# aaa authentication arap MIS-access if-needed none
```

ARAP Authentication Allowing Authorized Guest Logins

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

```
Device(config)# aaa authentication arap default auth-guest group radius
```

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

ARAP Authentication Allowing Guest Logins

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

```
Device(config)# aaa authentication arap default guest group radius
```

ARAP Authentication Using Line Password

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

```
Device(config)# aaa authentication arap default line
```

Before you can use a line password as the ARAP authentication method, you need to define a line password.

ARAP Authentication Using Local Password

Use the aaa authentication arap {default | authentication-list}command with the local keyword to specify that the Cisco device will use the local username database for authentication. For example, to specify the local username database as the method of ARAP user authentication when no other method list has been defined, enter the following command:

```
Device(config)# aaa authentication arap default local
```

ARAP Authentication Using Group RADIUS

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

```
Device(config)# aaa authentication arap default group radius
```

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

ARAP Authentication Using Group TACACS

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

```
Device(config)# aaa authentication arap default group tacacs+
```

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

ARAP Authentication Using a Group Name

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

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius araprad
Device(config-sg-radius)# server 172.16.2.3
Device(config-sg-radius)# server 172.16.2 17
Device(config-sg-radius)# server 172.16.2.32
Device(config-sg-radius)# end
```

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

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

```
Device(config)# aaa authentication arap default group araprad
```

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

Configuring NASI Authentication Using AAA

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

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

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	aaa new-model **Example:** Device(config)# **aaa new-model**	Enables AAA globally.
Step 4	aaa authentication nasi **Example:** Device(config)# **aaa authentication nasi**	Enables authentication for NASI users.
Step 5	line number **Example:** Device(config)# **line 4**	(Optional--not needed if default is used in the aaa authentication nasi command.) Enters line configuration mode.
Step 6	nasi authentication list-name **Example:** Device(config-line)# **nasi authentication nasi-authen**	(Optional--not needed if default is used in the aaa authentication nasi command.) Enables authentication for NASI on a line.
Step 7	end **Example:**	Exits line configuration mode and returns to the privileged EXEC mode.

Command or Action	Purpose
Device(config-line)# **end**	

What to do next

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

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

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

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

NASI Authentication Using Enable Password

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

```
Devie(config)# aaa authentication nasi default enable
```

Before you can use the enable password as the authentication method, you need to define the enable password.

NASI Authentication Using Line Password

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

```
Device(config)# aaa authentication nasi default line
```

Before you can use a line password as the NASI authentication method, you need to define a line password.

NASI Authentication Using Local Password

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

```
Device(config)# aaa authentication nasi default local
```

NASI Authentication Using Group RADIUS

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

```
Device(config)# aaa authentication nasi default group radius
```

Before you can use RADIUS as the NASI authentication method, you need to enable communication with the RADIUS security server.

NASI Authentication Using Group TACACS

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

```
Device(config)# aaa authentication nasi default group tacacs+
```

Before you can use TACACS+ as the authentication method, you need to enable communication with the TACACS+ security server.

NASI Authentication Using Group Name

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

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius nasirad
Device(config-sg-radius)# server 172.16.2.3
Device(config-sg-radius)# server 172.16.2 17
Device(config-sg-radius)# server 172.16.2.32
Device(config-sg-radius)# end
```

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

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

```
Device(config)# aaa authentication nasi default group nasirad
```

Before you can use a group name as the NASI authentication method, you need to enable communication with the RADIUS or TACACS+ security server.

Specifying the Amount of Time for Login Input

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

```
Device(config-line)# timeout login response 30
```

Password Protection at the Privileged Level

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

Use the following command in global configuration mode:

```
Device(config)# authentication enable default radius
```

or

```
Device(config)# authentication enable default tacacs
```

Changing the Text Displayed at the Password Prompt

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

```
Password:
```

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

The aaa authentication password-prompt command works when RADIUS is used as the login method. You will be able to see the password prompt defined in the command shown even when the RADIUS server is unreachable. The aaa authentication password-prompt command does not work with TACACS+. TACACS+ supplies the NAS with the password prompt to display to the users. If the TACACS+ server is reachable, the NAS gets the password prompt from the server and uses that prompt instead of the one defined in the aaa authentication password-prompt command. If the TACACS+ server is not reachable, the password prompt defined in the aaa authentication password-prompt command may be used.

Use the following command in global configuration mode:

```
Device(config)# aaa authentication password-prompt "Enter your password now:"
```

Preventing an Access Request with a Blank Username from Being Sent to the RADIUS Server

The following configuration steps provide the ability to prevent an Access Request with a blank username from being sent to the RADIUS server. This functionality ensures that unnecessary RADIUS server interaction is avoided, and RADIUS logs are kept short.

Procedure

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	aaa new-model **Example:** `Device(config)# configure terminal`	Enables AAA globally.
Step 4	aaa authentication suppress null-username **Example:** `Device(config)# aaa authentication suppress null-username`	Prevents an Access Request with a blank username from being sent to the RADIUS server.
Step 5	end **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Message Banners for AAA Authentication

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

Configuring a Login Banner

To configure a banner that is displayed when a user logs in (replacing the default message for login), perform the following task:

Before you begin

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

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	aaa new-model **Example:** Device(config)# **aaa new-model**	Enables AAA.
Step 4	aaa authentication banner delimiter string delimiter **Example:** Device(config)# **aaa authentication banner *Unauthorized use is prohibited.***	Creates a personalized login banner.
Step 5	end **Example:** Device(config)# **end**	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Failed-Login Banner

To configure a message that is displayed when a user login fails (replacing the default message for failed login), perform the following task:

Before you begin

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

Procedure

	Command or Action	**Purpose**
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	**Purpose**
	Example:	• Enter your password if prompted.
	Device> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example:	
	Device# configure terminal	
Step 3	aaa new-model	Enables AAA.
	Example:	
	Device(config)# **aaa new-model**	
Step 4	aaa authentication fail-message delimiter string delimiter	Creates a message to be displayed when a user login fails.
	Example:	
	Device(config)# **aaa authentication fail-message *Failed login. Try again.***	
Step 5	end	Exits global configuration mode and returns to privileged EXEC mode.
	Example:	
	Device(config)# **end**	

Configuring AAA Packet of Disconnect

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

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

Procedure

	Command or Action	**Purpose**
Step 1	enable	Enables privileged EXEC mode.
	Example:	• Enter your password if prompted.
	Device> **enable**	

	Command or Action	**Purpose**
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	aaa accounting network default start-stop radius **Example:** Device(config)# **aaa accounting network default start-stop radius**	Enables AAA accounting records.
Step 4	aaa accounting delay-start **Example:** Device(config)# **aaa accounting delay-start**	(Optional) Delays generation of the start accounting record until the Framed-IP-Address is assigned, allowing its use in the POD packet.
Step 5	aaa pod server server-key string **Example:** Device(config)# **aaa pod server server-key xyz123**	Enables POD reception.
Step 6	radius server name non-standard **Example:** Device(config)# **radius server radser**	Configures a RADIUS server and enters RADIUS server configuration mode.
Step 7	address {ipv4 | ipv6} hostname **Example:** Device(config-radius-server)# **address ipv4 radius-host**	Configures a RADIUS host.
Step 8	end **Example:** Device(config-radius-server)# **end**	Exits RADIUS server configuration mode and returns to privileged EXEC mode.

Double Authentication of PPP Sessions

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

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

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

How Double Authentication Works

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

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

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

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

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

Caution Double authentication can cause certain undesirable events if multiple hosts share a PPP connection to a network access server, as shown in the figure below. First, if a user, Bob, initiates a PPP session and activates double authentication at the network access server (per the figure below), any other user will automatically have the same network privileges as Bob until Bob's PPP session expires. This happens because Bob's authorization profile is applied to the network access server's interface during the PPP session and any PPP traffic from other users will use the PPP session Bob established. Second, if Bob initiates a PPP session and activates double authentication, and then--before Bob's PPP session has expired--another user, Jane, executes the access-profile command (or, if Jane Telnets to the network access server and autocommand access-profileis executed), a reauthorization will occur and Jane's authorization profile will be applied to the interface--replacing Bob's profile. This can disrupt or halt Bob's PPP traffic, or grant Bob additional authorization privileges Bob should not have.

Figure 2: Possibly Risky Topology: Multiple Hosts Share a PPP Connection to a Network Access Server
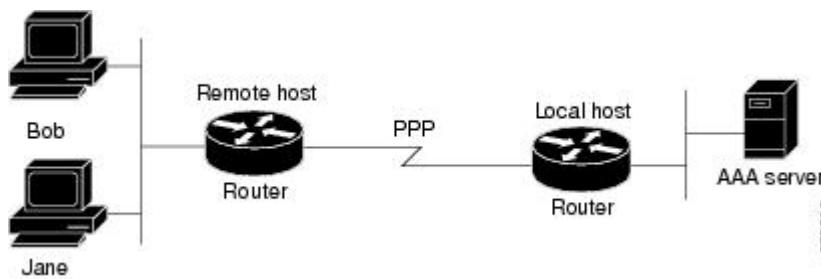
Configuring Double Authentication

To configure double authentication, perform the following steps:

1. Enable AAA by using the aaa-new modelglobal configuration command.

2. Use the aaa authentication command to configure your network access server to use login and PPP authentication method lists, then apply those method lists to the appropriate lines or interfaces.

3. Use the aaa authorization command to configure AAA network authorization at login.

4. Configure security protocol parameters (for example, RADIUS or TACACS+).

5. Use access control list AV pairs on the security server that the user can connect to the local host only by establishing a Telnet connection.

6. (Optional) Configure the access-profilecommand as an autocommand. If you configure the autocommand, remote users will not have to manually enter the access-profile command to access authorized rights associated with their personal user profile.

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

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

- Use valid AV pairs when configuring access control list AV pairs on the security server.

- For remote users to use the interface's existing authorization (that which existed prior to the second stage authentication/authorization), but you want them to have different access control lists (ACLs), you should specify only ACL AV pairs in the user-specific authorization definition. This might be desirable if you set up a default authorization profile to apply to the remote host, but want to apply specific ACLs to specific users.

- When these user-specific authorization statements are later applied to the interface, they can either be added to the existing interface configuration or they can replace the existing interface configuration--depending on which form of the access-profile command is used to authorize the user. You should understand how the access-profile command works before configuring the authorization statements.

- If you will be using ISDN or Multilink PPP, you must also configure virtual templates at the local host.

Accessing the User Profile After Double Authentication

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

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

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

```
Device> access-profile merge ignore-sanity-checks
```

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

Enabling Automated Double Authentication

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

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

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

1. Enable AAA by using the aaa-new model global configuration command.

2. Use the aaa authentication command to configure your network access server to use login and PPP authentication method lists, then apply those method lists to the appropriate lines or interfaces.

3. Use the aaa authorization command to configure AAA network authorization at login.

4. Configure security protocol parameters (for example, RADIUS or TACACS+).

5. Use access control list AV pairs on the security server that the user can connect to the local host only by establishing a Telnet connection.

6. Configure the access-profilecommand as an autocommand. If you configure the autocommand, remote users will not have to manually enter the access-profile command to access authorized rights associated with their personal user profile.

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

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

- Use valid AV pairs when configuring access control list AV pairs on the security server.

- If you want remote users to use the interface's existing authorization (that which existed prior to the second stage authentication/authorization), but you want them to have different access control lists (ACLs), you should specify only ACL AV pairs in the user-specific authorization definition. This might be desirable if you set up a default authorization profile to apply to the remote host, but want to apply specific ACLs to specific users.

- When these user-specific authorization statements are later applied to the interface, they can either be added to the existing interface configuration, or replace the existing interface configuration--depending on which form of the access-profile command is used to authorize the user. You should understand how the access-profile command works before configuring the authorization statements.

- If you will be using ISDN or Multilink PPP, you must also configure virtual templates at the local host.

Configuring Automated Double Authentication

To configure automated double authentication, perform the following task:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	ip trigger-authentication [timeout seconds] [port number] **Example:** `Device(config)# ip trigger-authentication timeout 120`	Enables automation of double authentication.

	Command or Action	Purpose
Step 4	interface type number **Example:** Device(config)# **interface gigabitethernet 1/0/17**	Configures an interface and enter the interface configuration mode.
Step 5	ip trigger-authentication **Example:** Device(config-if)# **ip rigger-authentication**	Applies automated double authentication to the interface.
Step 6	end **Example:** Device(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.

Troubleshooting Automated Double Authentication

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

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	show ip trigger-authentication **Example:** Device# show ip trigger-authentication	Displays the list of remote hosts for which automated double authentication has been attempted (successfully or unsuccessfully).
Step 3	clear ip trigger-authentication **Example:** Device# clear ip trigger-authentication	Clears the list of remote hosts for which automated double authentication has been attempted. (This clears the table displayed by the show ip trigger-authentication command.)
Step 4	debug ip trigger-authentication **Example:** Device# debug ip trigger-authentication	Displays debug output related to automated double authentication.

Configuring the Dynamic Authorization Service for RADIUS CoA

Use the following procedure to enable the router as an authentication, authorization, and accounting (AAA) server for dynamic authorization service to support the CoA functionality that pushes the policy map in an input and output direction.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	aaa new-model **Example:** `Router(config)# aaa new-model`	Enables AAA.
Step 4	aaa server radius dynamic-author **Example:** `Router(config)# aaa server radius dynamic-author`	Sets up the local AAA server for dynamic authorization service, which must be enabled to support the CoA functionality to push the policy map in an input and output direction and enter dynamic authorization local server configuration mode. In this mode, the RADIUS application commands are configured.
Step 5	client {ip_addr | hostname} [server-key [0 | 7] string] **Example:** `Router(config-locsvr-da-radius)#client 192.168.0.5 server-key cisco1`	Configures the IP address or hostname of the AAA server client. Use the optional server-key keyword and string argument to configure the server key at the "client" level. **Note** Configuring the server key at the client level overrides the server key configured at the global level.
Step 6	domain {delimiter character| stripping [right-to-left]} **Example:** `Router(config-locsvr-da-radius)# domain stripping right-to-left` **Example:**	(Optional) Configures username domain options for the RADIUS application. • The delimiter keyword specifies the domain delimiter. One of the following options can be specified for the character argument: @, /, $, %, \, # or - • The stripping keyword compares the incoming username with the names

	Command or Action	Purpose
	`Router(config-locsvr-da-radius)# domain delimiter @`	oriented to the left of the @ domain delimiter.
		• The right-to-left keyword terminates the string at the first delimiter going from right to left.
Step 7	port {port-num} **Example:** `Router(config-locsvr-da-radius)# port 3799`	Configures UDP port 3799 for CoA requests.

Configuring a Device to Ignore Bounce and Disable RADIUS CoA Requests

When an authentication port is authenticated with multiple hosts and there is a Change of Authorization (CoA) request for one host to flap on this port or one host session to be terminated on this port, the other hosts on this port are also affected. Thus, an authenticated port with multiple hosts can trigger a DHCP renegotiation from one or more hosts in the case of a flap, or it can administratively shut down the authentication port that is hosting the session for one or more hosts.

Perform the following steps to configure the device to ignore RADIUS server Change of Authorization (CoA) requests in the form of a bounce port command or disable port command.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	aaa new-model **Example:** `Device(config)# aaa new-model`	Enables authentication, authorization, and accounting (AAA) globally.
Step 4	authentication command bounce-port ignore **Example:** `Device(config)# authentication command bounce-port ignore`	(Optional) Configures the device to ignore a RADIUS server bounce port command that causes a host to link flap on an authentication port, which causes DHCP renegotiation from one or more hosts connected to this port.

	Command or Action	Purpose
Step 5	authentication command disable-port ignore **Example:** `Device(config)# authentication command disable-port ignore`	(Optional) Configures the device to ignore a RADIUS server CoA disable port command that administratively shuts down the authentication port that hosts one or more host sessions. • The shutting down of the port causes session termination.
Step 6	end **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring Domain Stripping at the Server Group Level

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	aaa group server radius server-name **Example:** `Device(config)# aaa group server radius rad1`	Adds the RADIUS server and enters server group RADIUS configuration mode. • The server-name argument specifies the RADIUS server group name.
Step 4	domain-stripping [strip-suffix word] [right-to-left] [prefix-delimiter word] [delimiter word] **Example:** `Device(config-sg-radius)# domain-stripping delimiter username@example.com`	Configures domain stripping at the server group level.
Step 5	end **Example:** `Device(config-sg-radius)# end`	Exits server group RADIUS configuration mode and returns to the privileged EXEC mode.

Non-AAA Authentication Methods

Configuring Line Password Protection

This task is used to provide access control on a terminal line by entering the password and establishing password checking.

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

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	line [aux | console | tty | vty] line-number [ending-line-number] **Example:** Device(config)# line console 0	Enters line configuration mode.
Step 4	password password **Example:** Device(config-line)# secret word	Assigns a password to a terminal or other device on a line. The password checker is case sensitive and can include spaces; for example, the password "Secret" is different from the password "secret," and "two words" is an acceptable password.
Step 5	login **Example:** Device(config-line)# login	Enables password checking at login. You can disable line password verification by disabling password checking by using the no version of this command.

	Command or Action	Purpose
		Note The login command only changes username and privilege level but it does not execute a shell; therefore autocommands will not be executed. To execute autocommands under this circumstance, you need to establish a Telnet session back into the device (loop-back). Make sure that the device has been configured for secure Telnet sessions if you choose to implement autocommands this way.
Step 6	end **Example:** Device(config-line)# **end**	Exits line configuration mode and returns to privileged EXEC mode.

Establishing Username Authentication

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

- To provide a TACACS-like username and encrypted password-authentication system for networks that cannot support TACACS

- To provide special-case logins: for example, access list verification, no password verification, autocommand execution at login, and "no escape" situations

To establish username authentication, perform the following task:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	Do one of the following: - username name [nopassword | password password | password encryption-type encrypted password]	Establishes username authentication with encrypted passwords. or

	Command or Action	Purpose
	• username name [access-class number] **Example:** Device(config)# **username superuser password superpassword password 7 encrypted-password** Device(config)# **username user1 access-class access-user**	(Optional) Establishes username authentication by access list.
Step 4	username name [privilege level] **Example:** Device(config)# **username user1 privilege 5**	(Optional) Sets the privilege level for the user.
Step 5	username name [autocommand command] **Example:** Device(config)# **username user1 autocommand show users**	(Optional) Specifies a command to be executed automatically.
Step 6	username name [noescape] [nohangup] **Example:** Device(config)# **username user1 noescapee**	(Optional) Sets a "no escape" login environment.
Step 7	end **Example:** Device(config)# **end**	Exits global configuration mode and returns to privileged EXEC mode.

What to do next

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

⚠️ **Caution** Passwords will be displayed in clear text in your configuration unless you enable the service password-encryption command.

CHAP or PAP Authentication

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- Enable PPP encapsulation.

- Enable CHAP or PAP on the interface.

- For CHAP, configure host name authentication and the secret or password for each remote system with which authentication is required.

Enabling PPP Encapsulation

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

```
Device(config-if)# encapsulation ppp
```

This command enables PPP on an interface.

Enabling PAP or CHAP

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

```
Device(config-if)# ppp authentication chap pap
```

Defines the authentication protocols supported and the order in which they are used. In this command, protocol1, protocol2 represent the following protocols: CHAP, MS-CHAP, and PAP. PPP authentication is attempted first using the first authentication method, which is protocol1. If protocol1 is unable to establish authentication, the next configured protocol is used to negotiate authentication.

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

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

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

⚠️

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

Inbound and Outbound Authentication

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

Enabling Outbound PAP Authentication

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

```
Device(config-if)# ppp pap sent-username username1 password password1
```

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

Refusing PAP Authentication Requests

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

```
Device(config-if)# ppp pap refuse
```

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

Creating a Common CHAP Password

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

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

```
Device(config-if)# ppp chap password secret
```

Refusing CHAP Authentication Requests

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

```
Device(config-if)# ppp chap refuse calling
```

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

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

Delaying CHAP Authentication Until Peer Authenticates

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

```
Device(config-if)# ppp chap wait secret
```

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

Using MS-CHAP

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

MS-CHAP differs from the standard CHAP as follows:

- MS-CHAP is enabled by negotiating CHAP Algorithm 0x80 in LCP option 3, Authentication Protocol.

- The MS-CHAP Response packet is in a format designed to be compatible with Microsoft Windows NT 3.5 and 3.51, Microsoft Windows 95, and Microsoft LAN Manager 2.x. This format does not require the authenticator to store a clear or reversibly encrypted password.

- MS-CHAP provides an authenticator-controlled authentication retry mechanism.

- MS-CHAP provides an authenticator-controlled change password mechanism.

- MS-CHAP defines a set of "reason-for-failure" codes returned in the Failure packet message field.

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

Table 7: Vendor-Specific RADIUS Attributes for MS-CHAP

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

Defining PPP Authentication Using MS-CHAP

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

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	encapsulation ppp **Example:** `Device(config)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	interface type number **Example:** `Device(config)# interface gigabitethernet 1/0/2`	Configures an interface and enters interface configuration mode.
Step 5	ppp authentication ms-chap [if-needed] [list-name | default] [callin] [one-time] **Example:** `Device(config-if)# ppp authentication ms-chap default callin`	Defines PPP authentication using MS-CHAP.
Step 6	end **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

What to do next

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

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

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

Note   If PPP authentication using MS-CHAP is used with username authentication, you must include the MS-CHAP secret in the local username/password database.

Authentication Examples

Example: RADIUS Authentication

This section provides two sample configurations using RADIUS.

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

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication login radius-login group radius local
Device(config)# aaa authentication ppp radius-ppp if-needed group radius
Device(config)# aaa authorization exec default group radius if-authenticated
Device(config)# aaa authorization network default group radius
Device(config)# line 3
Device(config-line)# login authentication radius-login
Device(config-line)# exit
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# ppp authentication radius-ppp
Device(config-if)# end
```

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

   • The aaa authentication login radius-login group radius local command configures the router to use RADIUS for authentication at the login prompt. If RADIUS returns an error, the user is authenticated using the local database.

   • The aaa authentication ppp radius-ppp if-needed group radius command configures the Cisco IOS XE software to use PPP authentication using CHAP or PAP if the user has not already logged in. If the EXEC facility has authenticated the user, PPP authentication is not performed.

   • The aaa authorization exec default group radius if-authenticated command queries the RADIUS database for information that is used during EXEC authorization, such as autocommands and privilege levels, but only provides authorization if the user has successfully authenticated.

   • The aaa authorization network default group radius command queries RADIUS for network authorization, address assignment, and other access lists.

   • The login authentication radius-login command enables the radius-login method list for line 3.

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

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

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

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication login default group radius local
Device(config)# aaa authorization exec default group radius local
Device(config)# aaa authorization command 2 default group tacacs+ if-authenticated
Device(config)# radius server radserver
Device(config-sg-radius)# address ipv4 10.2.3.1
Device(config-sg-radius)# exit
Device(config)# radius-server attribute 44 include-in-access-req
Device(config)# radius-server attribute 8 include-in-access-req
Device(config)# end
```

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

- The aaa authentication login default group radius local command specifies that the username and password are verified by RADIUS or, if RADIUS is not responding, by the router's local user database.

- The aaa authorization exec default group radius local command specifies that RADIUS authentication information be used to set the user's EXEC level if the user authenticates with RADIUS. If no RADIUS information is used, this command specifies that the local user database be used for EXEC authorization.

- The aaa authorization command 2 default group tacacs+ if-authenticated command specifies TACACS+ authorization for commands set at privilege level 2, if the user has already successfully authenticated.

- The radius-server host 172.16.71.146 auth-port 1645 acct-port 1646 command specifies the IP address of the RADIUS server host, the UDP destination port for authentication requests, and the UDP destination port for accounting requests.

- The radius-server attribute 44 include-in-access-req command sends RADIUS attribute 44 (Acct-Session-ID) in access-request packets.

- The radius-server attribute 8 include-in-access-req command sends RADIUS attribute 8 (Framed-IP-Address) in access-request packets.

Example: TACACS Authentication

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

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication ppp test group tacacs+ local
Device(config)# interface gigabitethernet 1/1/2
Device(config-if)# ppp authentication chap pap test
Device(config-if)# exit
Device(config)# tacacs-server host 192.0.2.3
Device(config)# tacacs-server key key1
```

```
Device(config)# end
```

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

- The aaa new-model command enables the AAA security services.

- The aaa authentication command defines a method list, "test," to be used on serial interfaces running PPP. The keywords group tacacs+ means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword local indicates that authentication will be attempted using the local database on the network access server.

- The interface command selects the line.

- The ppp authentication command applies the test method list to this line.

- The tacacs-server host command identifies the TACACS+ daemon as having an IP address of 192.0.2.3.

- The tacacs-server key command defines the shared encryption key to be "key1."

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

```
Device(config)# aaa authentication ppp default if-needed group tacacs+ local
```

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

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

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication ppp MIS-access if-needed group tacacs+ local
Device(config)# interface gigabitethernet 1/1/2
Device(config)# ppp authentication pap MIS-access
Device(config)# end
```

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

Example: Kerberos Authentication

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

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication login default krb5
Device(config)# end
```

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

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication ppp default krb5
```

```
Device(config)# end
```

Example: AAA Scalability

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

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# radius server radserver
Device(config-sg-radius)# address ipv4 radius-host
Device(config-sg-radius)# key myRaDiUSpassWoRd
Device(config-sg-radius)# exit
Device(config)# radius-server configure-nas
Device(config)# username root password ALongPassword
Device(config)# aaa authentication ppp dialins group radius local
Device(config)# aaa authentication login admins local
Device(config)# aaa authorization network default group radius local
Device(config)# aaa accounting network default start-stop group radius
Device(config)# aaa processes 16
Device(config)# line 1 16
Device(config-line)# autoselect ppp
Device(config-line)# autoselect during-login
Device(config-line)# login authentication admins
Device(config-line)# modem dialin
Device(config-line)# exit
Device(config)# interface gigabitethernet 1/2/0
Device(config-if)# group-range 1 16
Device(config-if)# encapsulation ppp
Device(config-if)# ppp authentication pap dialins
Device(config-if)# end
```

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

- The aaa new-model command enables AAA network security services.

- The address ipv4 {hostname | host-address} command defines the name of the RADIUS server host.

- The key command defines the shared secret text string between the network access server and the RADIUS server host.

- The radius-server configure-nas command defines that the Cisco router or access server will query the RADIUS server for static routes and IP pool definitions when the device first starts up.

- The username command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.

- The aaa authentication ppp dialins group radius local command defines the authentication method list "dialins," which specifies that RADIUS authentication, then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.

- The aaa authentication login admins local command defines another method list, "admins," for login authentication.

- The aaa authorization network default group radius local command is used to assign an address and other network parameters to the RADIUS user.

- The aaa accounting network default start-stop group radius command tracks PPP usage.

- The aaa processes command allocates 16 background processes to handle AAA requests for PPP.

- The line command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.

- The autoselect ppp command allows a PPP session to start up automatically on these selected lines.

- The autoselect during-login command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.

- The login authentication admins command applies the "admins" method list for login authentication.

- The modem dialin command configures modems attached to the selected lines to only accept incoming calls.

- The interface group-async command selects and defines an asynchronous interface group.

- The group-range command defines the member asynchronous interfaces in the interface group.

- The encapsulation ppp command sets PPP as the encapsulation method used on the specified interfaces.

- The ppp authentication pap dialinscommand applies the "dialins" method list to the specified interfaces.

Example: Configuring Login and Failed-Login Banners for AAA Authentication

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

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication banner *Unauthorized Access Prohibited*
Device(config)# aaa authentication login default group radius
Device(config)# end
```

This configuration displays the following login banner:

```
Unauthorized Access Prohibited
Username:
```

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

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication banner *Unauthorized Access Prohibited*
Device(config)# aaa authentication fail-message *Failed login. Try again.*
Device(config)# aaa authentication login default group radius
Device(config)# end
```

This configuration displays the following login and failed-login banner:

```
Unauthorized Access Prohibited
Username:
```

```
Password:
Failed login. Try again.
```

Example: AAA Packet of Disconnect Server Key

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

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication ppp default radius
Device(config)# aaa accounting network default start-stop radius
Device(config)# aaa accounting delay-start
Device(config)# aaa pod server server-key xyz123
Device(config)# radius server non-standard
Device(config-sg-radius)# address ipv4 10.2.1.1
Device(config-sg-radius)# key rad123
Device(config-sg-radius)# end
```

Example: Double Authentication

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

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

Example: Configuration of the Local Host for AAA with Double Authentication

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

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

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

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# radius server radserver
Device(config-sg-radisu)# address ipv4 secureserver
Device(config-sg-radius)# key myradiuskey
Device(config-sg-radius)# exit
Device(config)# aaa authentication ppp default group radius
Device(config)# aaa authentication login default group radius
Device(config)# aaa authorization network default group radius
Device(config)# aaa authorization exec default group radius
Device(config)# end
```

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

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# tacacs-server host security
Device(config)# tacacs-server key mytacacskey
Device(config)# aaa authentication ppp default group tacacs+
Device(config)# aaa authentication login default group tacacs+
Device(config)# aaa authorization network default group tacacs+
Device(config)# aaa authorization exec default group tacacs+
Device(config)# end
```

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

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

TACACS+ servers can be configured similarly. (See the Complete Configuration with TACACS Example.)

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

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

```
hostx    Password = "welcome"
         User-Service-Type = Framed-User,
         Framed-Protocol = PPP,
         cisco-avpair = "lcp:interface-config=ip unnumbered fastethernet 0",
         cisco-avpair = "ip:inacl#3=permit tcp any 172.21.114.0 0.0.0.255 eq telnet",
         cisco-avpair = "ip:inacl#4=deny icmp any any",
         cisco-avpair = "ip:route#5=10.0.0.0 255.0.0.0",
         cisco-avpair = "ip:route#6=10.10.0.0 255.0.0.0",
         cisco-avpair = "ipx:inacl#3=deny any"
```

Example: Configuration of the AAA Server for Second-Stage Per-User Authentication and Authorization

This section contains partial sample AAA configurations on a RADIUS server. These configurations define authentication and authorization for a user with the username "user1," who will be user-authenticated in the second stage of double authentication.

TACACS+ servers can be configured similarly.

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

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

```
user1    Password = "welcome"
         User-Service-Type = Shell-User,
         cisco-avpair = "shell:autocmd=access-profile"
         User-Service-Type = Framed-User,
         Framed-Protocol = PPP,
         cisco-avpair = "ip:inacl#3=permit tcp any host 10.0.0.2 eq telnet",
         cisco-avpair = "ip:inacl#4=deny icmp any any"
```

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

```
user1     Password = "welcome"
          User-Service-Type = Shell-User,
          cisco-avpair = "shell:autocmd=access-profile merge"
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = "ip:inacl#3=permit tcp any any"
          cisco-avpair = "ip:route=10.0.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.1.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.2.0.0 255.255.0.0"
```

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

```
user1     Password = "welcome"
          User-Service-Type = Shell-User,
          cisco-avpair = "shell:autocmd=access-profile replace"
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = "ip:inacl#3=permit tcp any any",
          cisco-avpair = "ip:inacl#4=permit icmp any any",
          cisco-avpair = "ip:route=10.10.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.11.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.12.0.0 255.255.0.0"
```

Example: Complete Configuration with TACACS

This example shows TACACS+ authorization profile configurations both for the remote host (used in the first stage of double authentication) and for specific users (used in the second stage of double authentication).

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

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

Figure 3: Example Topology for Double Authentication
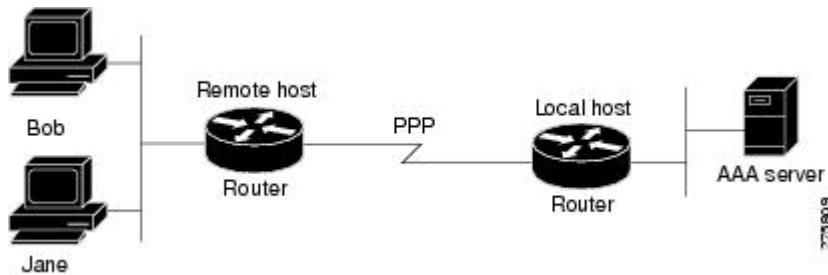
This sample configuration shows authentication/authorization profiles on the TACACS+ server for the remote host "hostx" and for three users, with the usernames "user_default," "user_merge," and "user_replace."

```
key = "mytacacskey"
default authorization = permit
#--------------------------Remote Host (BRI)------------------------
#
# This allows the remote host to be authenticated by the local host
# during fist-stage authentication, and provides the remote host
# authorization profile.
#
#------------------------------------------------------------------------
user = hostx
{
    login = cleartext "welcome"
    chap = cleartext "welcome"
    service = ppp protocol = lcp {
                interface-config="ip unnumbered fastethernet 0"
    }
    service = ppp protocol = ip {
            # It is important to have the hash sign and some string after
            # it. This indicates to the NAS that you have a per-user
            # config.
            inacl#3="permit tcp any 172.21.114.0 0.0.0.255 eq telnet"
            inacl#4="deny icmp any any"
            route#5="10.0.0.0 255.0.0.0"
            route#6="10.10.0.0 255.0.0.0"
    }
    service = ppp protocol = ipx {
            # see previous comment about the hash sign and string, in protocol = ip
            inacl#3="deny any"
    }
}
#------------------ "access-profile" default user "only acls" -----------------
#
# Without arguments, access-profile removes any access-lists it can find
# in the old configuration (both per-user and per-interface), and makes sure
# that the new profile contains ONLY access-list definitions.
#
#------------------------------------------------------------------------------
user = user_default
{
        login = cleartext "welcome"
        chap = cleartext "welcome"
        service = exec
        {
                # This is the autocommand that executes when user_default logs in.
                autocmd = "access-profile"
        }
        service = ppp protocol = ip {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!
                inacl#3="permit tcp any host 10.0.0.2 eq telnet"
                inacl#4="deny icmp any any"
        }
        service = ppp protocol = ipx {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }
}
```

```
#-------------------- "access-profile merge" user --------------------------
#
# With the 'merge' option, first all old access-lists are removed (as before),
# but then (almost) all AV pairs are uploaded and installed. This will allow
# for uploading any custom static routes, sap-filters, and so on, that the user
# may need in his or her profile. This needs to be used with care, as it leaves
# open the possibility of conflicting configurations.
#
#-----------------------------------------------------------------------------
user = user_merge
{
        login = cleartext "welcome"
        chap = cleartext "welcome"
        service = exec
        {
                # This is the autocommand that executes when user_merge logs in.
                autocmd = "access-profile merge"
        }
        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!
                inacl#3="permit tcp any any"
                route#2="10.0.0.0 255.255.0.0"
                route#3="10.1.0.0 255.255.0.0"
                route#4="10.2.0.0 255.255.0.0"
        }
        service = ppp protocol = ipx
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }
}
#-------------------- "access-profile replace" user --------------------------
#
# With the 'replace' option, ALL old configuration is removed and ALL new
# configuration is installed.
#
# One caveat: access-profile checks the new configuration for address-pool and
# address AV pairs. As addresses cannot be renegotiated at this point, the
# command will fail (and complain) when it encounters such an AV pair.
# Such AV pairs are considered to be "invalid" for this context.
#-----------------------------------------------------------------------------
user = user_replace
{
        login = cleartex
t
"
welcome
"
        chap = cleartext "welcome"
        service = exec
        {
                # This is the autocommand that executes when user_replace logs in.
                autocmd = "access-profile replace"
        }
        service = ppp protocol = ip
        {
```

```
                         # Put whatever access-lists, static routes, whatever
                         # here.
                         # If you leave this blank, the user will have NO IP
                         # access-lists (not even the ones installed prior to
                         # this)!
                         inacl#3="permit tcp any any"
                         inacl#4="permit icmp any any"
                         route#2="10.10.0.0 255.255.0.0"
                         route#3="10.11.0.0 255.255.0.0"
                         route#4="10.12.0.0 255.255.0.0"
          }
          service = ppp protocol = ipx
          {
                         # put whatever access-lists, static routes, whatever
                         # here.
                         # If you leave this blank, the user will have NO IPX
                         # access-lists (not even the ones installed prior to
                         # this)!
          }
}
```

Example: Automated Double Authentication

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

```
Current configuration:
!
version 16.10
no service password-encryption
!
hostname myrouter
!
!
! **The following AAA commands are used to configure double authentication:
!
! **The following command enables AAA:
aaa new-model
! **The following command enables user authentication via the RADIUS AAA server:
!
aaa authentication login default none
aaa authentication ppp default group radius
! **The following command causes the remote user's authorization profile to be
! downloaded from the AAA server to the router when required:
!
aaa authorization network default group radius
!
enable password mypassword
!
ip host blue 172.21.127.226
ip host green 172.21.127.218
ip host red 172.21.127.114
ip domain-name example.com
ip name-server 172.16.2.75
!
!
interface GigabitEthernet0/0/0
 ip address 172.21.127.186 255.255.255.248
 no ip route-cache
 no ip mroute-cache
```

```
 no keepalive
 ntp disable
 no cdp enable
!
interface Virtual-Template1
 ip unnumbered loopback0
 no ip route-cache
 no ip mroute-cache
!
! **The following command specifies that device authentication occurs via PPP CHAP:
 ppp authentication chap
!
router eigrp 109
 network 172.21.0.0
 no auto-summary
!
ip default-gateway 172.21.127.185
no ip classless
ip route 172.21.127.114 255.255.255.255 172.21.127.113
! **Virtual profiles are required for double authentication to work:
virtual-profile virtual-template 1
dialer-list 1 protocol ip permit
no cdp run
! **The following command defines where the TACACS+ AAA server is:
tacacs-server host 172.16.57.35 port 1049
tacacs-server timeout 90
! **The following command defines the key to use with TACACS+ traffic (required):
tacacs-server key mytacacskey
snmp-server community public RO
!
line con 0
 exec-timeout 0 0
 login authentication console
line aux 0
 transport input all
line vty 0 4
 exec-timeout 0 0
 password lab
!
end
```

Feature History for Configuring Authentication

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	AAA Authentication	Authentication provides a method to identify users, which includes the login and password dialog, challenge and response, messaging support, and encryption, depending on the selected security protocol. Authentication is the way a user is identified prior to being allowed access to the network and network services.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

Configuring Authorization

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

Prerequisites for Configuring Authorization

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

- Enable authentication, authorization, and accounting (AAA) on your network access server.

- Configure AAA authentication. Authorization generally takes place after authentication and relies on authentication to work properly. For more information about AAA authentication, refer to the "Configuring Authentication" module.

- Define the characteristics of your RADIUS or TACACS+ security server if you are issuing RADIUS or TACACS+ authorization. For more information about configuring your Cisco network access server to communicate with your RADIUS security server, refer to the chapter "Configuring RADIUS". For more information about configuring your Cisco network access server to communicate with your TACACS+ security server, refer to the "Configuring TACACS+" module.

- Define the rights associated with specific users by using the username command if you are issuing local authorization.

Information About Configuring Authorization

Named Method Lists for Authorization

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

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

Method lists are specific to the authorization type requested:

- Commands: Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.

- EXEC: Applies to the attributes associated with a user EXEC terminal session.

- Network: Applies to network connections. This can include a PPP, SLIP, or ARAP connection.

- Reverse Access: Applies to reverse Telnet sessions.

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

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

AAA Authorization Methods

AAA supports five different methods of authorization:

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

- If-Authenticated: The user is allowed to access the requested function provided the user has been authenticated successfully.

- None: The network access server does not request authorization information; authorization is not performed over this line/interface.

- Local: The router or access server consults its local database, as defined by the username command, for example, to authorize specific rights for users. Only a limited set of functions can be controlled via the local database.

- RADIUS: The network access server requests authorization information from the RADIUS security server. RADIUS authorization defines specific rights for users by associating attributes, which are stored in a database on the RADIUS server, with the appropriate user.

Note   With CSCuc32663, passwords and authorization logs are masked before being sent to the TACACS+, LDAP, or RADIUS security servers. Use the aaa authorization commands visible-keys command to send unmasked information to the TACACS+, LDAP, or RADIUS security servers.

Authorization Methods

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

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

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

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

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

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

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

Method Lists and Server Groups

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

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

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

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

AAA Authorization Types

Cisco IOS XE software supports five different types of authorization:

- Commands: Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.

- EXEC: Applies to the attributes associated with a user EXEC terminal session.

- Network: Applies to network connections. This can include a PPP, SLIP, or ARAP connection.

- Reverse Access: Applies to reverse Telnet sessions.

- Configuration: Applies to downloading configurations from the AAA server.

- IP Mobile: Applies to authorization for IP mobile services.

Authorization Types

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

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

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

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

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

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

For information about the types of authorization supported by the Cisco IOS XE software, refer to the AAA Authorization Types.

Authorization Attribute-Value Pairs

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

For a list of supported RADIUS attributes, refer to the "RADIUS Attributes Overview and RADIUS IETF Attributes" chapter. For a list of supported TACACS+ AV pairs, refer to the "Configuring TACACS+" chapter.

How to Configure Authorization

Configuring AAA Authorization Using Named Method Lists

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

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	**Purpose**
Step 3	aaa authorization {auth-proxy | network | exec | commands level | reverse-access | configuration | ipmobile} {default | list-name} [method1 [method2...]] **Example:** Device(config)# aaa authorization auth-proxy default	Creates an authorization method list for a particular authorization type and enable authorization.

Step 4	Do one of the following:
	
	    • line [aux | console | tty | vty] line-number [ending-line-number]
	    • interface interface-type interface-number
	
	Example:
	
	Device(config)# line 1
	
	Device(config)# interface gigabitethernet 0/1/1
		Enters the line configuration mode for the lines to which you want to apply the authorization method list.
		
		Alternately, enters the interface configuration mode for the interfaces to which you want to apply the authorization method list.

Step 5	Do one of the following:
	
	    • authorization{arap | commands level | exec | reverse-access} {default | list-name}
	    • ppp authorization{default | list-name}
	
	Example:
	Device(config-line)# authorization commands default
	
	Device(config-if)# ppp authorization default
		Applies the authorization list to a line or set of lines.
		
		Alternately, applies the authorization list to an interface or set of interfaces.

Step 6	end **Example:** Device(config-line)# end Device(config-if)# end	Exits line configuration mode and returns to privileged EXEC mode. Exits interface configuration mode and returns to privileged EXEC mode.

Disabling Authorization for Global Configuration Commands

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

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

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

To disable AAA authorization on the console, use the following command in global configuration mode:

Note   AAA authorization is disabled on the console by default. If AAA authorization is enabled on the console, disable it by configuring the no aaa authorization console command during the AAA configuration stage. AAA should be disabled on the console for user authentication.

Command	Purpose
Device(config)# **no aaa authorization console**	Disables authorization on the console.

Configuring Authorization for Reverse Telnet

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

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

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

Reverse Telnet authorization offers the following benefits:

- An additional level of protection by ensuring that users engaged in reverse Telnet activities are indeed authorized to access a specific asynchronous port using reverse Telnet.

- An alternative method (other than access lists) to manage reverse Telnet authorization.

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

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

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

Authorization Configuration Examples

Example: TACACS Authorization

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

```
Device(config)# aaa authorization network default group tacacs+ none
```

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

```
Device(config)# aaa authorization network default group tacacs+
```

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

```
Device> enable
Device# configure terminal
Device(config)# aaa authorization network default group tacacs+
Device(config)# interface gigabitethernet 01/1/
Device(config-if)# ip address-pool local
Device(config-if)# exit
Device(config)# ip local-pool mci 172.16.0.1 172.16.0.255
Device(config)# ip local-pool att 172.17.0.1 172.17.0.255
Device(config-if)# end
```

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

```
user = mci_customer1 {
    login = cleartext "some password"
    service = ppp protocol = ip {
        addr-pool=mci
    }
}
user = att_customer1 {
    login = cleartext "some other password"
    service = ppp protocol = ip {
        addr-pool=att
     }
```

Example: RADIUS Authorization

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

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization exec default group radius if-authenticated
Device(config)# aaa authorization network default group radius
Device(config)# radius server ip
Device(config-radius-server)# key sharedkey
Device(config-radius-server)# end
```

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

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

The RADIUS information returned may be used to specify an autocommand or a connection access list be applied to this connection.

- The aaa authorization network default group radius command configures network authorization via RADIUS. This can be used to govern address assignment, the application of access lists, and various other per-user quantities.

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

Example: Reverse Telnet Authorization

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

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login default group tacacs+
Device(config)# aaa authorization reverse-access default group tacacs+
Device(config)# tacacs-server host 172.31.255.0
Device(config)# tacacs-server timeout 90
Device(config)# tacacs-server key sharedkey
Device(config)# end
```

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

- The aaa new-model command enables AAA.

- The aaa authentication login default group tacacs+ command specifies TACACS+ as the default method for user authentication during login.

- The aaa authorization reverse-access default group tacacs+ command specifies TACACS+ as the method for user authorization when trying to establish a reverse Telnet session.

- The tacacs-server host command identifies the TACACS+ server.

- The tacacs-server timeoutcommand sets the interval of time that the network access server waits for the TACACS+ server to reply.

- The tacacs-server key command defines the encryption key used for all TACACS+ communications between the network access server and the TACACS+ daemon.

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

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

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

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

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

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

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

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

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

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
```

```
Device(config)# aaa authentication login default group radius
Device(config)# aaa authorization reverse-access default group radius
Device(config)# radius server ip
Device(config-radius-server)# key sharedkey
Device(config-radius-server)# address ipv4 172.31.255.0 auth-port 1645 acct-port 1646
Device(config-radius-server)# end
```

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

- The aaa new-model command enables AAA.

- The aaa authentication login default group radius command specifies RADIUS as the default method for user authentication during login.

- The aaa authorization reverse-access default group radius command specifies RADIUS as the method for user authorization when trying to establish a reverse Telnet session.

- The radius command identifies the RADIUS server.

- The key command defines the encryption key used for all RADIUS communications between the network access server and the RADIUS daemon.

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

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

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

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

Additional References for Configuring Authorization

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature History for Configuring Authorization

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	AAA Authorization	AAA authorization enables you to limit the services available to a user. When AAA authorization is enabled, the network access server uses information retrieved from the user's profile, which is located either in the local user database or on the security server, to configure the user's session. Once this is done, the user will be granted access to a requested service only if the information in the user profile allows it.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

C H A P T E R 5

Configuring Accounting

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

Prerequisites for Configuring Accounting

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

- Enable AAA on the network access server by using the aaa new-modelcommand in global configuration mode.

- Define the characteristics of the RADIUS or TACACS+ security server if RADIUS or TACACS+ authorization is issued. For more information about configuring the Cisco network access server to communicate with the RADIUS security server, see the Configuring RADIUS module. For more information about configuring the Cisco network access server to communicate with the TACACS+ security server, see the Configuring TACACS+ module.

Restrictions for Configuring Accounting

- Accounting information can be sent simultaneously to a maximum of only four AAA servers.

Topic 2.1

Information About Configuring Accounting

Named Method Lists for Accounting

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

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

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

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

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

- Network : Provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.

- EXEC : Provides information about user EXEC terminal sessions of the network access server.

- Commands : Provides information about the EXEC mode commands that a user issues. Command accounting generates accounting records for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.

- Connection : Provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembler/disassembler (PAD), and rlogin.

- System : Provides information about system-level events.

- Resource : Provides "start" and "stop" records for calls that have passed user authentication, and provides "stop" records for calls that fail to authenticate.

- VRRS : Provides information about Virtual Router Redundancy Service (VRRS).

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

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

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

This section includes the following subsections:

Method Lists and Server Groups

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

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

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

For more information about configuring server groups and about configuring server groups based on Dialed Number Identification Service (DNIS) numbers, see the "Configuring RADIUS" or "Configuring TACACS+" modules.

AAA Accounting Methods

The following two methods of accounting are supported:

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

• RADIUS: The network access server reports user activity to the RADIUS security server in the form of accounting records. Each accounting record contains accounting AV pairs and is stored on the security server.

✎

Note Passwords and accounting logs are masked before being sent to the TACACS+ or RADIUS security servers. Use the aaa accounting commands visible-keys command to send unmasked information to the TACACS+ or RADIUS security servers.

Accounting Record Types

For minimal accounting, use the stop-only keyword, which instructs the specified method (RADIUS or TACACS+) to send a stop record accounting notice at the end of the requested user process. For more accounting information, use the start-stop keyword to send a start accounting notice at the beginning of the requested event and a stop accounting notice at the end of the event. To stop all accounting activities on this line or interface, use the none keyword.

Accounting Methods

The table below lists the supported accounting methods.

Table 8: AAA Accounting Methods

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

The method argument refers to the actual method the authentication algorithm tries. Additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all other methods return an error, specify additional methods in the command. For example, to create a method list named acct_tac1 that specifies RADIUS as the backup method of authentication in the event that TACACS+ authentication returns an error, enter the following command:

```
aaa accounting network acct_tac1 stop-only group tacacs+ group radius
```

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

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

```
aaa accounting network default stop-only group radius
```

AAA Accounting supports the following methods:

• group tacacs : To have the network access server send accounting information to a TACACS+ security server, use the group tacacs+ method keyword.

- group radius : To have the network access server send accounting information to a RADIUS security server, use the group radius method keyword.

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

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

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

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

To specify group loginrad as the method of network accounting when no other method list has been defined, enter the following command:

```
aaa accounting network default start-stop group loginrad
```

Before a group name can be used as the accounting method, communication with the RADIUS or TACACS+ security server must be enabled.

AAA Accounting Types

Network Accounting

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

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

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

Wed Jun 27 04:45:00 2001
```

```
                     NAS-IP-Address = "172.16.25.15"
                     NAS-Port = 5
                     User-Name = "username1"
                     Client-Port-DNIS = "4327528"
                     Caller-ID = "562"
                     Acct-Status-Type = Start
                     Acct-Authentic = RADIUS
                     Service-Type = Framed
                     Acct-Session-Id = "0000000E"
                     Framed-IP-Address = "10.1.1.2"
                     Framed-Protocol = PPP
                     Acct-Delay-Time = 0
                     User-Id = "username1"
                     NAS-Identifier = "172.16.25.15"
Wed Jun 27 04:47:46 2001
                     NAS-IP-Address = "172.16.25.15"
                     NAS-Port = 5
                     User-Name = "username1"
                     Client-Port-DNIS = "4327528"
                     Caller-ID = "562"
                     Acct-Status-Type = Stop
                     Acct-Authentic = RADIUS
                     Service-Type = Framed
                     Acct-Session-Id = "0000000E"
                     Framed-IP-Address = "10.1.1.2"
                     Framed-Protocol = PPP
                     Acct-Input-Octets = 3075
                     Acct-Output-Octets = 167
                     Acct-Input-Packets = 39
                     Acct-Output-Packets = 9
                     Acct-Session-Time = 171
                     Acct-Delay-Time = 0
                     User-Id = "username1"
                     NAS-Identifier = "172.16.25.15"
Wed Jun 27 04:48:45 2001
                     NAS-IP-Address = "172.16.25.15"
                     NAS-Port = 5
                     User-Name = "username1"
                     Client-Port-DNIS = "4327528"
                     Caller-ID = "408"
                     Acct-Status-Type = Stop
                     Acct-Authentic = RADIUS
                     Service-Type = Exec-User
                     Acct-Session-Id = "0000000D"
                     Acct-Delay-Time = 0
                     User-Id = "username1"
                     NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ network accounting record for a PPP user who first started an EXEC session:

```
Wed Jun 27 04:00:35 2001 172.16.25.15   username1   tty4    562/4327528      starttask_id=28
        service=shell
Wed Jun 27 04:00:46 2001 172.16.25.15   username1   tty4    562/4327528      starttask_id=30
        addr=10.1.1.1    service=ppp
Wed Jun 27 04:00:49 2001 172.16.25.15   username1   tty4    408/4327528      updattask_id=30
        addr=10.1.1.1    service=ppp    protocol=ip    addr=10.1.1.1
Wed Jun 27 04:01:31 2001 172.16.25.15   username1   tty4    562/4327528      stoptask_id=30
        addr=10.1.1.1    service=ppp    protocol=ip    addr=10.1.1.1
                                        bytes_in=2844      bytes_out=1682  paks_in=36
    paks_out=24     elapsed_time=51
Wed Jun 27 04:01:32 2001 172.16.25.15   username1   tty4    562/4327528      stoptask_id=28
        service=shell   elapsed_time=57
```

> **Note**   The precise format of accounting packets records may vary depending on the security server daemon.

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

```
Wed Jun 27 04:30:52 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 3
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000B"
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:36:49 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 3
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000B"
        Framed-Protocol = PPP
        Framed-IP-Address = "10.1.1.1"
        Acct-Input-Octets = 8630
        Acct-Output-Octets = 5722
        Acct-Input-Packets = 94
        Acct-Output-Packets = 64
        Acct-Session-Time = 357
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ network accounting record for a PPP user who comes in through autoselect:

```
Wed Jun 27 04:02:19 2001 172.16.25.15    username1   Async5  562/4327528       starttask_id=35
        service=ppp
Wed Jun 27 04:02:25 2001 172.16.25.15    username1   Async5  562/4327528       updatetask_id=35
        service=ppp      protocol=ip      addr=10.1.1.2
Wed Jun 27 04:05:03 2001 172.16.25.15    username1   Async5  562/4327528       stoptask_id=35
        service=ppp      protocol=ip      addr=10.1.1.2
                                            bytes_in=3366    bytes_out=2149       paks_in=42
      paks_out=28       elapsed_time=164
```

EXEC Accounting

EXEC accounting provides information about user EXEC terminal sessions (user shells) on the network access server, including username, date, start and stop times, the access server IP address, and (for dial-in users) the telephone number the call originated from.

The following example shows the information contained in a RADIUS EXEC accounting record for a dial-in user:

```
Wed Jun 27 04:26:23 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 1
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329483"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000006"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
Wed Jun 27 04:27:25 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 1
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329483"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000006"
        Acct-Session-Time = 62
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ EXEC accounting record for a dial-in user:

```
Wed Jun 27 03:46:21 2001        172.16.25.15    username1    tty3    5622329430/4327528
start
 task_id=2       service=shell
Wed Jun 27 04:08:55 2001        172.16.25.15    username1    tty3    5622329430/4327528
stop
 task_id=2       service=shell   elapsed_time=1354
```

The following example shows the information contained in a RADIUS EXEC accounting record for a Telnet user:

```
Wed Jun 27 04:48:32 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 26
        User-Name = "username1"
        Caller-ID = "10.68.202.158"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000010"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:48:46 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 26
        User-Name = "username1"
        Caller-ID = "10.68.202.158"
```

```
              Acct-Status-Type = Stop
              Acct-Authentic = RADIUS
              Service-Type = Exec-User
              Acct-Session-Id = "00000010"
              Acct-Session-Time = 14
              Acct-Delay-Time = 0
              User-Id = "username1"
              NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ EXEC accounting record for a Telnet user:

```
Wed Jun 27 04:06:53 2001        172.16.25.15    username1   tty26   10.68.202.158
starttask_id=41      service=shell
Wed Jun 27 04:07:02 2001        172.16.25.15    username1   tty26   10.68.202.158
stoptask_id=41        service=shell   elapsed_time=9
```

Command Accounting

Command accounting provides information about the EXEC shell commands for a specified privilege level that are being executed on a network access server. Each command accounting record includes a list of the commands executed for that privilege level, as well as the date and time each command was executed, and the user who executed it.

The following example shows the information contained in a TACACS+ command accounting record for privilege level 1:

```
Wed Jun 27 03:46:47 2001         172.16.25.15    username1   tty3   5622329430/4327528
stop     task_id=3       service=shell   priv-lvl=1      cmd=show version <cr>
Wed Jun 27 03:46:58 2001         172.16.25.15    username1   tty3   5622329430/4327528
stop     task_id=4       service=shell   priv-lvl=1      cmd=show interfaces Ethernet 0
<cr>
Wed Jun 27 03:47:03 2001         172.16.25.15    username1   tty3   5622329430/4327528
stop     task_id=5       service=shell   priv-lvl=1      cmd=show ip route <cr>
```

The following example shows the information contained in a TACACS+ command accounting record for privilege level 15:

```
Wed Jun 27 03:47:17 2001         172.16.25.15    username1   tty3   5622329430/4327528
stop     task_id=6       service=shell   priv-lvl=15     cmd=configure terminal <cr>
Wed Jun 27 03:47:21 2001         172.16.25.15    username1   tty3   5622329430/4327528
stop     task_id=7       service=shell   priv-lvl=15     cmd=interface Serial 0 <cr>
Wed Jun 27 03:47:29 2001         172.16.25.15    username1   tty3   5622329430/4327528
stop     task_id=8       service=shell   priv-lvl=15     cmd=ip address 10.1.1.1 255.255.255.0
 <cr>
```

Note   The Cisco implementation of RADIUS does not support command accounting.

Connection Accounting

Connection accounting provides information about all outbound connections made from the network access server such as Telnet, LAT, TN3270, PAD, and rlogin.

The following example shows the information contained in a RADIUS connection accounting record for an outbound Telnet connection:

```
Wed Jun 27 04:28:00 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "00000008"
        Login-Service = Telnet
        Login-IP-Host = "10.68.202.158"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:28:39 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "00000008"
        Login-Service = Telnet
        Login-IP-Host = "10.68.202.158"
        Acct-Input-Octets = 10774
        Acct-Output-Octets = 112
        Acct-Input-Packets = 91
        Acct-Output-Packets = 99
        Acct-Session-Time = 39
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound Telnet connection:

```
Wed Jun 27 03:47:43 2001          172.16.25.15    username1   tty3    5622329430/4327528
start    task_id=10     service=connection     protocol=telnet addr=10.68.202.158 cmd=telnet
 username1-sun
Wed Jun 27 03:48:38 2001          172.16.25.15    username1   tty3    5622329430/4327528
stop     task_id=10     service=connection     protocol=telnet addr=10.68.202.158 cmd=telnet
 username1-sun    bytes_in=4467   bytes_out=96    paks_in=61     paks_out=72 elapsed_time=55
```

The following example shows the information contained in a RADIUS connection accounting record for an outbound rlogin connection:

```
Wed Jun 27 04:29:48 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "0000000A"
        Login-Service = Rlogin
```

```
            Login-IP-Host = "10.68.202.158"
            Acct-Delay-Time = 0
            User-Id = "username1"
            NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:30:09 2001
            NAS-IP-Address = "172.16.25.15"
            NAS-Port = 2
            User-Name = "username1"
            Client-Port-DNIS = "4327528"
            Caller-ID = "5622329477"
            Acct-Status-Type = Stop
            Acct-Authentic = RADIUS
            Service-Type = Login
            Acct-Session-Id = "0000000A"
            Login-Service = Rlogin
            Login-IP-Host = "10.68.202.158"
            Acct-Input-Octets = 18686
            Acct-Output-Octets = 86
            Acct-Input-Packets = 90
            Acct-Output-Packets = 68
            Acct-Session-Time = 22
            Acct-Delay-Time = 0
            User-Id = "username1"
            NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound rlogin connection:

```
Wed Jun 27 03:48:46 2001        172.16.25.15    username1   tty3   5622329430/4327528
start    task_id=12     service=connection    protocol=rlogin addr=10.68.202.158 cmd=rlogin
 username1-sun /user username1
Wed Jun 27 03:51:37 2001        172.16.25.15    username1   tty3   5622329430/4327528
stop     task_id=12     service=connection    protocol=rlogin addr=10.68.202.158 cmd=rlogin
 username1-sun /user username1 bytes_in=659926 bytes_out=138   paks_in=2378     paks_
out=1251        elapsed_time=171
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound LAT connection:

```
Wed Jun 27 03:53:06 2001        172.16.25.15    username1   tty3   5622329430/4327528
start    task_id=18      service=connection     protocol=lat    addr=VAX        cmd=lat
VAX
Wed Jun 27 03:54:15 2001        172.16.25.15    username1   tty3   5622329430/4327528
stop     task_id=18      service=connection     protocol=lat    addr=VAX        cmd=lat
VAX  bytes_in=0      bytes_out=0     paks_in=0     paks_out=0      elapsed_time=6
```

System Accounting

System accounting provides information about all system-level events (for example, when the system reboots or when accounting is turned on or off).

The following accounting record shows a typical TACACS+ system accounting record server indicating that AAA Accounting has been turned off:

```
Wed Jun 27 03:55:32 2001        172.16.25.15    unknown unknown unknown start   task_id=25
    service=system
event=sys_acct  reason=reconfigure
```

Note The precise format of accounting packets records may vary depending on the TACACS+ daemon.

The following accounting record shows a TACACS+ system accounting record indicating that AAA Accounting has been turned on:

```
Wed Jun 27 03:55:22 2001        172.16.25.15     unknown unknown unknown stop    task_id=23
    service=system
event=sys_acct  reason=reconfigure
```

Resource Accounting

The Cisco implementation of AAA accounting provides "start" and "stop" record support for calls that have passed user authentication. The additional feature of generating "stop" records for calls that fail to authenticate as part of user authentication is also supported. Such records are necessary for users employing accounting records to manage and monitor their networks.

This section includes the following subsections:

AAA Resource Failure Stop Accounting

Before AAA resource failure stop accounting, there was no method of providing accounting records for calls that failed to reach the user authentication stage of a call setup sequence. Such records are necessary for users employing accounting records to manage and monitor their networks and their wholesale customers.

This functionality generates a "stop" accounting record for any calls that do not reach user authentication; "stop" records are generated from the moment of call setup. All calls that pass user authentication behave as they did before; that is, no additional accounting records are seen.

The figure below illustrates a call setup sequence with normal call flow (no disconnect) and without AAA resource failure stop accounting enabled.

Figure 4: Modem Dial-In Call Setup Sequence With Normal Flow and Without Resource Failure Stop Accounting Enabled
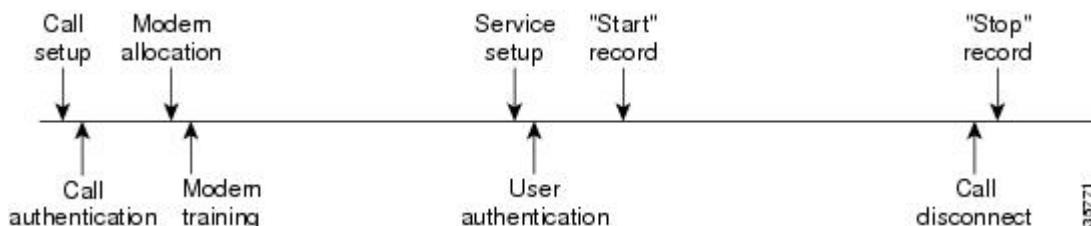
The figure below illustrates a call setup sequence with normal call flow (no disconnect) and with AAA resource failure stop accounting enabled.

Figure 5: Modem Dial-In Call Setup Sequence With Normal Flow and WIth Resource Failure Stop Accounting Enabled
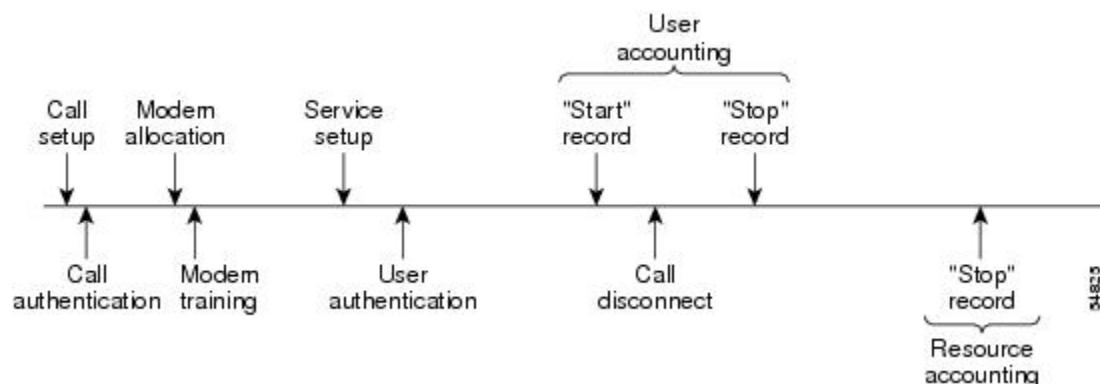
The figure below illustrates a call setup sequence with call disconnect occurring before user authentication and with AAA resource failure stop accounting enabled.

Figure 6: Modem Dial-In Call Setup Sequence With Call Disconnect Occurring Before User Authentication and With Resource Failure Stop Accounting Enabled

The figure below illustrates a call setup sequence with call disconnect occurring before user authentication and without AAA resource failure stop accounting enabled.

Figure 7: Modem Dial-In Call Setup Sequence With Call Disconnect Occurring Before User Authentication and Without Resource Failure Stop Accounting Enabled
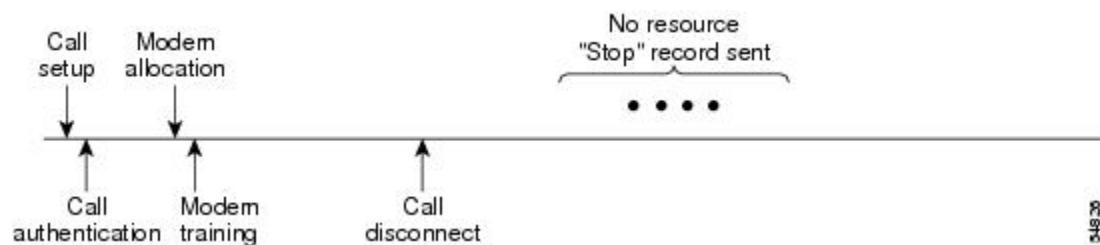
AAA Resource Accounting for Start-Stop Records

AAA resource accounting for start-stop records supports the ability to send a "start" record at each call setup, followed by a corresponding "stop" record at the call disconnect. This functionality can be used to manage and monitor wholesale customers from one source of data reporting, such as accounting records.

With this feature, a call setup and call disconnect "start-stop" accounting record tracks the progress of the resource connection to the device. A separate user authentication "start-stop" accounting record tracks the user management progress. These two sets of accounting records are interlinked by using a unique session ID for the call.

The figure below illustrates a call setup sequence with AAA resource start-stop accounting enabled.

Figure 8: Modem Dial-In Call Setup Sequence With Resource Start-Stop Accounting Enabled
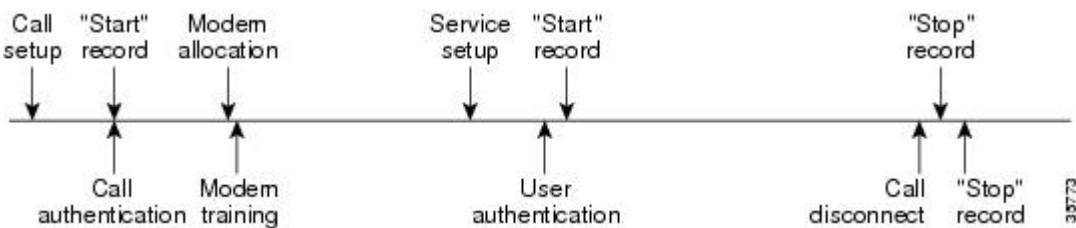
AAA Accounting Enhancements

AAA Broadcast Accounting

AAA broadcast accounting allows accounting information to be sent to multiple AAA servers at the same time; that is, accounting information can be broadcast to one or more AAA servers simultaneously. This functionality allows service providers to send accounting information to their own private AAA servers and to the AAA servers of their end customers. It also provides redundant billing information for voice applications.

Broadcasting is allowed among groups of RADIUS or TACACS+ servers, and each server group can define its backup servers for failover independently of other groups.

Thus, service providers and their end customers can use different protocols (RADIUS or TACACS+) for the accounting server. Service providers and their end customers can also specify their backup servers independently. As for voice applications, redundant accounting information can be managed independently through a separate group with its own failover sequence.

AAA Session MIB

The AAA session MIB feature allows customers to monitor and terminate their authenticated client connections using Simple Network Management Protocol (SNMP). The data of the client is presented so that it correlates directly to the AAA Accounting information reported by either the RADIUS or the TACACS+ server. AAA session MIB provides the following information:

  • Statistics for each AAA function (when used in conjunction with the show radius statistics command)

  • Status of servers providing AAA functions

  • Identities of external AAA servers

  • Real-time information (such as idle times), providing additional criteria for use by SNMP networks for assessing whether or not to terminate an active call

The table below shows the SNMP user-end data objects that can be used to monitor and terminate authenticated client connections with the AAA session MIB feature.

Table 9: SNMP End-User Data Objects

SessionId	The session identification used by the AAA Accounting protocol (same value as reported by RADIUS attribute 44 (Acct-Session-ID)).
UserId	The user login ID or zero-length string if a login is unavailable.
IpAddr	The IP address of the session or 0.0.0.0 if an IP address is not applicable or unavailable.

IdleTime	The elapsed time in seconds that the session has been idle.
Disconnect	The session termination object used to disconnect the given client.
CallId	The entry index corresponding to this accounting session that the Call Tracker record stored.

The table below describes the AAA summary information provided by the AAA session MIB feature using SNMP on a per-system basis.

Table 10: SNMP AAA Session Summary

ActiveTableEntries	Number of sessions currently active.
ActiveTableHighWaterMark	Maximum number of sessions present at once since last system reinstallation.
TotalSessions	Total number of sessions since last system reinstallation.
DisconnectedSessions	Total number of sessions that have been disconnected using since last system reinstallation.

Accounting Attribute-Value Pairs

The network access server monitors the accounting functions defined in either TACACS+ AV pairs or RADIUS attributes, depending on which security method is implemented.

How to Configure AAA Accounting

Configuring AAA Accounting Using Named Method Lists

To configure AAA Accounting using named method lists, perform the following steps:

Note System accounting does not use named method lists. For system accounting, define only the default method list.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:**	Enters global configuration mode.

	Command or Action	**Purpose**
	`Device# configure terminal`	
Step 3	aaa accounting {system | network | exec | connection | commands level} {default | list-name} {start-stop | stop-only | none} [method1 [method2...]] **Example:** `Device(config)# aaa accounting system default start-stop`	Creates an accounting method list and enables accounting. The argument list-name is a character string used to name the created list.
Step 4	Do one of the following: • line [aux | console | tty | vty] line-number [ending-line-number] • interface interface-type interface-number **Example:** `Device(config)# line aux line1`	Enters the line configuration mode for the lines to which the accounting method list is applied. or Enters the interface configuration mode for the interfaces to which the accounting method list is applied.
Step 5	Do one of the following: • accounting {arap | commands level | connection | exec} {default | list-name} • ppp accounting{default | list-name} **Example:** `Device(config-line)# accounting arap default`	Applies the accounting method list to a line or set of lines. or Applies the accounting method list to an interface or set of interfaces.
Step 6	end **Example:** `Device(config-line)# end`	(Optional) Exits line configuration mode and returns to privileged EXEC mode.

Suppressing Generation of Accounting Records for Null Username Sessions

When AAA Accounting is activated, the Cisco IOS software issues accounting records for all users on the system, including users whose username string, because of protocol translation, is NULL. An example of this is users who come in on lines where the aaa authentication login method-list none command is applied. To prevent accounting records from being generated for sessions that do not have usernames associated with them, use the following command in global configuration mode:

Command	Purpose
Device(config)# **aaa accounting suppress null-username**	Prevents accounting records from being generated for users whose username string is NULL.

Generating Interim Accounting Records

To enable periodic interim accounting records to be sent to the accounting server, use the following command in global configuration mode:

Command	Purpose
Device(config)# **aaa accounting update** [**newinfo**] [**periodic**] *number*	Enables periodic interim accounting records to be sent to the accounting server.

When the aaa accounting updatecommandis activated, the Cisco IOS software issues interim accounting records for all users on the system. If the keyword newinfo is used, interim accounting records are sent to the accounting server every time there is new accounting information to report. An example of this would be when IPCP completes IP address negotiation with the remote peer. The interim accounting record includes the negotiated IP address used by the remote peer.

When used with the keyword periodic, interim accounting records are sent periodically as defined by the number argument. The interim accounting record contains all of the accounting information recorded for that user up to the time the interim accounting record is sent.

⚠️

Caution   Using the aaa accounting update periodic command can cause heavy congestion when many users are logged in to the network.

Configuring an Alternate Method to Enable Periodic Accounting Records

You can use the following alternative method to enable periodic interim accounting records to be sent to the accounting server.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	aaa accounting network default **Example:** Device(config)# aaa accounting network default	Configures the default accounting for all network-related service requests and enters accounting method list configuration mode.
Step 4	action-type {none | start-stop [periodic {disable | interval minutes}] | stop-only} **Example:** Device(cfg-acct-mlist)# action-type start-stop **Example:** periodic interval 5	Specifies the type of action to be performed on accounting records. • (Optional) The periodic keyword specifies periodic accounting action. • The interval keyword specifies the periodic accounting interval. • The value argument specifies the intervals for accounting update records (in minutes). • The disable keyword disables periodic accounting.
Step 5	end **Example:** Device(cfg-acct-mlist)# end	Exits accounting method list configuration mode and returns to privileged EXEC mode.

Generating Interim Service Accounting Records

Perform this task to enable the generation of interim service accounting records at periodic intervals for subscribers.

Before you begin

RADIUS Attribute 85 in the user service profile always takes precedence over the configured interim-interval value. RADIUS Attribute 85 must be in the user service profile. See the RADIUS Attributes Overview and RADIUS IETF Attributes feature document for more information.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	subscriber service accounting interim-interval minutes **Example:** Device(config)# subscriber service accounting interim-interval 10	Enables the generation of interim service accounting records at periodic intervals for subscribers. The minutes argument indicates the number of periodic intervals to send accounting update records from 1 to 71582 minutes.
Step 4	end **Example:** Device(config)# end	Exits global configuration mode and returns to privileged EXEC mode.

Generating Accounting Records for a Failed Login or Session

When AAA accounting is activated, the Cisco IOS XE software does not generate accounting records for system users who fail login authentication, or who succeed in login authentication but fail PPP negotiation for some reason.

To specify that accounting stop records be generated for users who fail to authenticate at login or during session negotiation, use the following command in global configuration mode:

Command or Action	**Purpose**
aaa accounting send stop-record authentication failure	Generates "stop" records for users who fail to authenticate at login or during session negotiation using PPP.

Specifying Accounting NETWORK-Stop Records Before EXEC-Stop Records

For PPP users who start EXEC terminal sessions, it can be specified that NETWORK records be generated before EXEC-stop records. In some cases, such as billing customers for specific services, is can be desirable

to keep network start and stop records together, essentially "nesting" them within the framework of the EXEC start and stop messages. For example, a user dialing in using PPP can create the following records: EXEC-start, NETWORK-start, EXEC-stop, NETWORK-stop. By nesting the network accounting records, NETWORK-stop records follow NETWORK-start messages: EXEC-start, NETWORK-start, NETWORK-stop, EXEC-stop.

To nest accounting records for user sessions, use the following command in global configuration mode:

Command or Action	Purpose
aaa accounting nested	Nests network accounting records.

Suppressing System Accounting Records over Switchover

To suppress the system accounting-on and accounting-off messages during switchover, use the following command in global configuration mode:

Command or Action	Purpose
aaa accounting redundancy suppress system-records	Suppresses the system accounting messages during switchover.

Configuring AAA Resource Failure Stop Accounting

To enable resource failure stop accounting, use the following command in global configuration mode:

Command	Purpose
Device(config)# **aaa accounting resource** *method-list* **stop-failure group** *server-group*	Generates a "stop" record for any calls that do not reach user authentication. **Note** Before configuring this feature, the tasks described in the Prerequisites for Configuring Accounting, on page 87 section must be performed, and SNMP must be enabled on the network access server.

Configuring AAA Resource Accounting for Start-Stop Records

To enable full resource accounting for start-stop records, use the following command in global configuration mode:

Command	Purpose
Device(config)# **aaa accounting resource** *method-list* **start-stop group** *server-group*	Supports the ability to send a "start" record at each call setup. followed with a corresponding "stop" record at the call disconnect. **Note** Before configuring this feature, the tasks described in the Prerequisites for Configuring Accounting, on page 87 section must be performed, and SNMP must be enabled on the network access server.

AAA Broadcast Accounting

AAA broadcast accounting allows accounting information to be sent to multiple AAA servers at the same time; that is, accounting information can be broadcast to one or more AAA servers simultaneously. This functionality allows service providers to send accounting information to their own private AAA servers and to the AAA servers of their end customers. It also provides redundant billing information for voice applications.

Broadcasting is allowed among groups of RADIUS or TACACS+ servers, and each server group can define its backup servers for failover independently of other groups.

Thus, service providers and their end customers can use different protocols (RADIUS or TACACS+) for the accounting server. Service providers and their end customers can also specify their backup servers independently. As for voice applications, redundant accounting information can be managed independently through a separate group with its own failover sequence.

Configuring Per-DNIS AAA Broadcast Accounting

To configure AAA broadcast accounting per DNIS, use the aaa dnis map accounting networkcommand in global configuration mode:

Command	Purpose
Device(config)# **aaa dnis map** *dnis-number* **accounting network** [**start-stop** | **stop-only** | **none**] [**broadcast**] *method1* [*method2...*]	Allows per-DNIS accounting configuration. This command has precedence over the global aaa accounting command. Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group.

AAA Session MIB

The AAA session MIB feature allows customers to monitor and terminate their authenticated client connections using Simple Network Management Protocol (SNMP). The data of the client is presented so that it correlates directly to the AAA Accounting information reported by either the RADIUS or the TACACS+ server. AAA session MIB provides the following information:

- Statistics for each AAA function (when used in conjunction with the show radius statistics command)

- Status of servers providing AAA functions

- Identities of external AAA servers

- Real-time information (such as idle times), providing additional criteria for use by SNMP networks for assessing whether or not to terminate an active call

The table below shows the SNMP user-end data objects that can be used to monitor and terminate authenticated client connections with the AAA session MIB feature.

Table 11: SNMP End-User Data Objects

SessionId	The session identification used by the AAA Accounting protocol (same value as reported by RADIUS attribute 44 (Acct-Session-ID)).
UserId	The user login ID or zero-length string if a login is unavailable.
IpAddr	The IP address of the session or 0.0.0.0 if an IP address is not applicable or unavailable.
IdleTime	The elapsed time in seconds that the session has been idle.
Disconnect	The session termination object used to disconnect the given client.
CallId	The entry index corresponding to this accounting session that the Call Tracker record stored.

The table below describes the AAA summary information provided by the AAA session MIB feature using SNMP on a per-system basis.

Table 12: SNMP AAA Session Summary

ActiveTableEntries	Number of sessions currently active.
ActiveTableHighWaterMark	Maximum number of sessions present at once since last system reinstallation.
TotalSessions	Total number of sessions since last system reinstallation.
DisconnectedSessions	Total number of sessions that have been disconnected using since last system reinstallation.

Establishing a Session with a Device if the AAA Server Is Unreachable

To establish a console session with a device if the AAA server is unreachable, use the following command in global configuration mode:

Command or Action	Purpose
no aaa accounting system guarantee-first	The aaa accounting system guarantee-first command guarantees system accounting as the first record, which is the default condition.
	In some situations, users may be prevented from starting a session on the console or terminal connection until after the system reloads, which can take more than three minutes. To resolve this problem, use the no aaa accounting system guarantee-first command.

Monitoring Accounting

No specific show command exists for either RADIUS or TACACS+ accounting. To obtain accounting records displaying information about users logged in, use the following command in privileged EXEC mode:

Command or Action	Purpose
show accounting	Allows display of the active accountable events on the network and helps collect information in the event of a data loss on the accounting server.

Troubleshooting Accounting

To troubleshoot accounting information, use the following command in privileged EXEC mode:

Command or Action	Purpose
debug aaa accounting	Displays information on accountable events as they occur.

Configuration Examples for AAA Accounting

Example: Configuring a Named Method List

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

```
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login admins local
Device(config)# aaa authentication ppp dialins group radius local
Device(config)# aaa authorization network network1 group radius local
Device(config)# aaa accounting network network2 start-stop group radius group tacacs+
Device(config)# username root password ALongPassword
Device(config)# tacacs-server host 172.31.255.0
Device(config)# tacacs-server key goaway
Device(config)# radius server isp
Device(config-sg-radius)# key myRaDiUSpassWoRd
Device(config-sg-radius)# exit
Device(config)# interface group-async 1
Device(config-if)# group-range 1 16
Device(config-if)# encapsulation ppp
Device(config-if)# ppp authentication chap dialins
Device(config-if)# ppp authorization network1
Device(config-if)# ppp accounting network2
Device(config-if)# exit
Device(config)# line 1 16
Device(config-line)# autoselect ppp
Device(config-line)# autoselect during-login
Device(config-line)# login authentication admins
Device(config-line)# modem dialin

Device(config-line)# end
```

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

• The aaa new-model command enables AAA network security services.

- The aaa authentication login admins local command defines a method list, "admins", for login authentication.

- The aaa authentication ppp dialins group radius local command defines the authentication method list "dialins", which specifies that first RADIUS authentication and then (if the RADIUS server does not respond) local authentication is used on serial lines using PPP.

- The aaa authorization network network1 group radius local command defines the network authorization method list named "network1", which specifies that RADIUS authorization is used on serial lines using PPP. If the RADIUS server fails to respond, then local network authorization is performed.

- The aaa accounting network network2 start-stop group radius group tacacs+ command defines the network accounting method list named "network2", which specifies that RADIUS accounting services (in this case, start and stop records for specific events) are used on serial lines using PPP. If the RADIUS server fails to respond, accounting services are handled by a TACACS+ server.

- The username command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.

- The tacacs-server host command defines the name of the TACACS+ server host.

- The tacacs-server key command defines the shared secret text string between the network access server and the TACACS+ server host.

- The radius server command defines the name of the RADIUS server host.

- The key command defines the shared secret text string between the network access server and the RADIUS server host.

- The interface group-async command selects and defines an asynchronous interface group.

- The group-range command defines the member asynchronous interfaces in the interface group.

- Theencapsulation ppp command sets PPP as the encapsulation method used on the specified interfaces.

- Theppp authentication chap dialins command selects Challenge Handshake Authentication Protocol (CHAP) as the method of PPP authentication and applies the "dialins" method list to the specified interfaces.

- Theppp authorization network1 command applies the blue1 network authorization method list to the specified interfaces.

- Theppp accounting network2 command applies the red1 network accounting method list to the specified interfaces.

- The line command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.

- The autoselect ppp command configures the Cisco IOS XE software to allow a PPP session to start up automatically on these selected lines.

- The autoselect during-login command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.

- The login authentication admins command applies the admins method list for login authentication.

- The modem dialin command configures modems attached to the selected lines to accept only incoming calls.

The show accounting command yields the following output for the preceding configuration:

```
Active Accounted actions on tty1, User username2 Priv 1
 Task ID 5, Network Accounting record, 00:00:52 Elapsed
 task_id=5 service=ppp protocol=ip address=10.0.0.98
```

The table below describes the fields contained in the preceding output.

Table 13: show accounting Field Descriptions

Field	Description
Active Accounted actions on	Terminal line or interface name user with which the user logged in.
User	User's ID.
Priv	User's privilege level.
Task ID	Unique identifier for each accounting session.
Accounting Record	Type of accounting session.
Elapsed	Length of time (hh:mm:ss) for this session type.
attribute=value	AV pairs associated with this accounting session.

Example: Configuring AAA Resource Accounting

The following example shows how to configure the resource failure stop accounting and resource accounting for start-stop records functions:

```
!Enable AAA on your network access server.
Device(config)# aaa new-model
!Enable authentication at login and list the AOL string name to use for login authentication.
Device(config)# aaa authentication login AOL group radius local
!Enable authentication for ppp and list the default method to use for PPP authentication.
Device(config)# aaa authentication ppp default group radius local
!Enable authorization for all exec sessions and list the AOL string name to use for
authorization.
Device(config)# aaa authorization exec AOL group radius if-authenticated
!Enable authorization for all network-related service requests and list the default method

 to use for all network-related authorizations.
Device(config)# aaa authorization network default group radius if-authenticated
!Enable accounting for all exec sessions and list the default method to use for all start-stop

  accounting services.
Device(config)# aaa accounting exec default start-stop group radius
!Enable accounting for all network-related service requests and list the default method to
 use
 for all start-stop accounting services.
Device(config)# aaa accounting network default start-stop group radius
!Enable failure stop accounting.
Device(config)# aaa accounting resource default stop-failure group radius
!Enable resource accounting for start-stop records.
Device(config)# aaa accounting resource default start-stop group radius
```

Example: Configuring AAA Broadcast Accounting

The following example shows how to turn on broadcast accounting using the global aaa accounting command:

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius isp
Device(config-sg-radius)# server 10.0.0.1
Device(config-sg-radius)# server 10.0.0.2
Device(config-sg-radius)# exit
Device(config)# aaa group server tacacs+ isp_customer
Device config-sg-tacacs+)# server 172.0.0.1
Device config-sg-tacacs+)# exit
Device(config)# aaa accounting network default start-stop broadcast group isp group
isp_customer
Device(config)# tacacs-server host 172.0.0.1 key key2
Device(config)# end
```

The broadcast keyword causes "start" and "stop" accounting records for network connections to be sent simultaneously to server 10.0.0.1 in the group isp and to server 172.0.0.1 in the group isp_customer. If server 10.0.0.1 is unavailable, failover to server 10.0.0.2 occurs. If server 172.0.0.1 is unavailable, no failover occurs because backup servers are not configured for the group isp_customer.

Example: Configuring per-DNIS AAA Broadcast Accounting

The following example shows how to turn on per-DNIS broadcast accounting using the global aaa dnis map accounting networkcommand:

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius isp
Device(config-sg-radius)# server 10.0.0.1
Device(config-sg-radius)# server 10.0.0.2
Device(config-sg-radius)# exit
Device(config)# aaa group server tacacs+ isp_customer
Device config-sg-tacacs+)# server 172.0.0.1
Device config-sg-tacacs+)# exit
Device(config)# aaa dnis map enable
Device(config)# aaa dnis map 7777 accounting network start-stop broadcast group isp group
isp_customer
Device(config)# tacacs-server host 172.0.0.1 key key_2
Device(config)# end
```

The broadcast keyword causes "start" and "stop" accounting records for network connection calls having DNIS number 7777 to be sent simultaneously to server 10.0.0.1 in the group isp and to server 172.0.0.1 in the group isp_customer. If server 10.0.0.1 is unavailable, failover to server 10.0.0.2 occurs. If server 172.0.0.1 is unavailable, no failover occurs because backup servers are not configured for the group isp_customer.

Example: AAA Session MIB

The following example shows how to set up the AAA session MIB feature to disconnect authenticated client connections for PPP users:

```
Device> enable
```

```
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication ppp default group radius
Device(config)# aaa authorization network default group radius
Device(config)# aaa accounting network default start-stop group radius
Device(config)# aaa session-mib disconnect
Device(config)# end
```

Additional References for Configuring Accounting

The following sections provide references related to the Configuring Accounting feature.

Related Documents

Related Topic	Document Title
IOS commands	Cisco IOS Master Command List, All Releases

MIBs

MIB	MIBs Link
• CISCO-AAA-SESSION-MIB	To locate and download MIBs for selected platforms, Cisco IOS XE software releases , and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature History for Configuring Accounting

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	AAA Broadcast Accounting	AAA broadcast accounting allows accounting information to be sent to multiple AAA servers at the same time; that is, accounting information can be broadcast to one or more AAA servers simultaneously.
Cisco IOS XE Everest 16.6.1	AAA Session MIB	The AAA session MIB feature allows customers to monitor and terminate their authenticated client connections using SNMP.
Cisco IOS XE Everest 16.6.1	Connection Accounting	Connection accounting provides information about all outbound connections made from the network access server, such as Telnet, local-area transport, TN3270, packet assembler/disassembler (PAD), and rlogin.
Cisco IOS XE Everest 16.6.1	AAA Interim Accounting	AAA interim accounting allows accounting records to be sent to the accounting server every time there is new accounting information to report, or on a periodic basis.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

Configuring Local Authentication and Authorization

How to Configure Local Authentication and Authorization

Configuring the Switch for Local Authentication and Authorization

You can configure AAA to operate without a server by setting the switch to implement AAA in local mode. The switch then handles authentication and authorization. No accounting is available in this configuration.

Note
To secure the switch for HTTP access by using AAA methods, you must configure the switch with the **ip http authentication aaa** global configuration command. Configuring AAA authentication does not secure the switch for HTTP access by using AAA methods.

Follow these steps to configure AAA to operate without a server by setting the switch to implement AAA in local mode:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	Device# **configure terminal**	
Step 3	aaa new-model **Example:** Device(config)# **aaa new-model**	Enables AAA.
Step 4	aaa authentication login default local **Example:** Device(config)# **aaa authentication login default local**	Sets the login authentication to use the local username database. The default keyword applies the local user database authentication to all ports.
Step 5	aaa authorization exec default local **Example:** Device(config)# **aaa authorization exec default local**	Configures user AAA authorization, check the local database, and allow the user to run an EXEC shell.
Step 6	aaa authorization network default local **Example:** Device(config)# **aaa authorization network default local**	Configures user AAA authorization for all network-related service requests.
Step 7	username name [privilege level] {password encryption-type password} **Example:** Device(config)# **username your_user_name privilege 1 password 7 secret567**	Enters the local database, and establishes a username-based authentication system. Repeat this command for each user. • For name, specify the user ID as one word. Spaces and quotation marks are not allowed. • (Optional) For level, specify the privilege level the user has after gaining access. The range is 0 to 15. Level 15 gives privileged EXEC mode access. Level 0 gives user EXEC mode access. • For encryption-type, enter 0 to specify that an unencrypted password follows. Enter 7 to specify that a hidden password follows. • For password, specify the password the user must enter to gain access to the

	Command or Action	Purpose
		switch. The password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the username command.
Step 8	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 9	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 10	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Monitoring Local Authentication and Authorization

To display Local Authentication and Authorization configuration, use the show running-config privileged EXEC command.

Feature History for Local Authentication and Authorization

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	Local Authentication and Authorization	This feature helps AAA to operate without a server by setting the device to implement AAA in local mode.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

CHAPTER **7**

Configuring TACACS+

Prerequisites for TACACS+

The following are the prerequisites for set up and configuration of switch access with TACACS+ (must be performed in the order presented):

1. Configure the switches with the TACACS+ server addresses.

2. Set an authentication key.

3. Configure the key from Step 2 on the TACACS+ servers.

4. Enable authentication, authorization, and accounting (AAA).

5. Create a login authentication method list.

6. Apply the list to the terminal lines.

7. Create an authorization and accounting method list.

The following are the prerequisites for controlling switch access with TACACS+:

- You must have access to a configured TACACS+ server to configure TACACS+ features on your switch. Also, you must have access to TACACS+ services maintained in a database on a TACACS+ daemon typically running on a LINUX or Windows workstation.

- You need a system running the TACACS+ daemon software to use TACACS+ on your switch.

- To use TACACS+, it must be enabled.

- Authorization must be enabled on the switch to be used.

- Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

- To use any of the AAA commands listed in this section or elsewhere, you must first enable AAA with the aaa new-model command.

- At a minimum, you must identify the host or hosts maintaining the TACACS+ daemon and define the method lists for TACACS+ authentication. You can optionally define method lists for TACACS+ authorization and accounting.

- The method list defines the types of authentication to be performed and the sequence in which they are performed; it must be applied to a specific port before any of the defined authentication methods are performed. The only exception is the default method list (which, by coincidence, is named default). The default method list is automatically applied to all ports except those that have a named method list explicitly defined. A defined method list overrides the default method list.

- Use TACACS+ for privileged EXEC access authorization if authentication was performed by using TACACS+.

- Use the local database if authentication was not performed by using TACACS+.

Information About Controlling Switch Access with TACACS+

TACACS+ and Switch Access

This section describes TACACS+. TACACS+ provides detailed accounting information and flexible administrative control over the authentication and authorization processes. It is facilitated through authentication, authorization, accounting (AAA) and can be enabled only through AAA commands.

TACACS+ Overview

TACACS+ is a security application that provides centralized validation of users attempting to gain access to your switch.

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

The goal of TACACS+ is to provide a method for managing multiple network access points from a single management service. Your switch can be a network access server along with other Cisco routers and access servers.

Figure 9: Typical TACACS+ Network Configuration
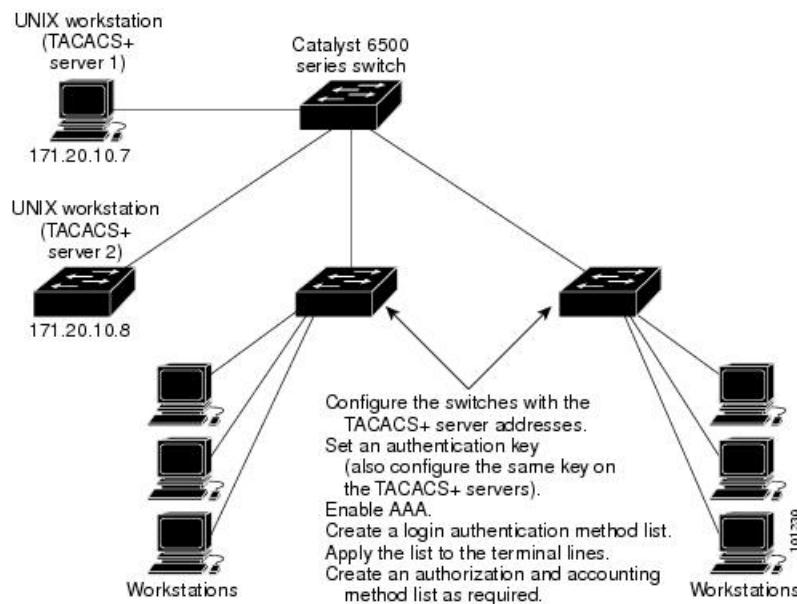
TACACS+, administered through the AAA security services, can provide these services:

- Authentication—Provides complete control of authentication through login and password dialog, challenge and response, and messaging support.

  The authentication facility can conduct a dialog with the user (for example, after a username and password are provided, to challenge a user with several questions, such as home address, mother's maiden name, service type, and social security number). The TACACS+ authentication service can also send messages to user screens. For example, a message could notify users that their passwords must be changed because of the company's password aging policy.

- Authorization—Provides fine-grained control over user capabilities for the duration of the user's session, including but not limited to setting autocommands, access control, session duration, or protocol support. You can also enforce restrictions on what commands a user can execute with the TACACS+ authorization feature.

- Accounting—Collects and sends information used for billing, auditing, and reporting to the TACACS+ daemon. Network managers can use the accounting facility to track user activity for a security audit or to provide information for user billing. Accounting records include user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

The TACACS+ protocol provides authentication between the switch and the TACACS+ daemon, and it ensures confidentiality because all protocol exchanges between the switch and the TACACS+ daemon are encrypted.

TACACS+ Operation

When a user attempts a simple ASCII login by authenticating to a switch using TACACS+, this process occurs:

1. When the connection is established, the switch contacts the TACACS+ daemon to obtain a username prompt to show to the user. The user enters a username, and the switch then contacts the TACACS+

daemon to obtain a password prompt. The switch displays the password prompt to the user, the user enters a password, and the password is then sent to the TACACS+ daemon.

TACACS+ allows a dialog between the daemon and the user until the daemon receives enough information to authenticate the user. The daemon prompts for a username and password combination, but can include other items, such as the user's mother's maiden name.

2. The switch eventually receives one of these responses from the TACACS+ daemon:

   - ACCEPT—The user is authenticated and service can begin. If the switch is configured to require authorization, authorization begins at this time.

   - REJECT—The user is not authenticated. The user can be denied access or is prompted to retry the login sequence, depending on the TACACS+ daemon.

   - ERROR—An error occurred at some time during authentication with the daemon or in the network connection between the daemon and the switch. If an ERROR response is received, the switch typically tries to use an alternative method for authenticating the user.

   - CONTINUE—The user is prompted for additional authentication information.

   After authentication, the user undergoes an additional authorization phase if authorization has been enabled on the switch. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

3. If TACACS+ authorization is required, the TACACS+ daemon is again contacted, and it returns an ACCEPT or REJECT authorization response. If an ACCEPT response is returned, the response contains data in the form of attributes that direct the EXEC or NETWORK session for that user and the services that the user can access:

   - Telnet, Secure Shell (SSH), rlogin, or privileged EXEC services

   - Connection parameters, including the host or client IP address, access list, and user timeouts

Method List

A method list defines the sequence and methods to be used to authenticate, to authorize, or to keep accounts on a user. You can use method lists to designate one or more security protocols to be used, thus ensuring a backup system if the initial method fails. The software uses the first method listed to authenticate, to authorize, or to keep accounts on users; if that method does not respond, the software selects the next method in the list. This process continues until there is successful communication with a listed method or the method list is exhausted.

If a method list is configured under VTY lines, the corresponding method list must be added to AAA. The following example shows how to configure a method list under a VTY line:

```
Device# configure terminal
Device(config)# line vty 0 4
Device(config)# authorization commands 15 auth1
```

The following example shows how to configure a method list in AAA:

```
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization commands 15 auth1 group tacacs+
```

If no method list is configured under VTY lines, the default method list must be added to AAA. The following example shows a VTY configuration without a method list:

```
Device# configure terminal
Device(config)# line vty 0 4
```

The following example shows how to configure the default method list:

```
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization commands 15 default group tacacs+
```

TACACS+ Configuration Options

You can configure the switch to use a single server or AAA server groups to group existing server hosts for authentication. You can group servers to select a subset of the configured server hosts and use them for a particular service. The server group is used with a global server-host list and contains the list of IP addresses of the selected server hosts.

TACACS+ Login Authentication

A method list describes the sequence and authentication methods to be queried to authenticate a user. You can designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops, and no other authentication methods are attempted.

TACACS+ Authorization for Privileged EXEC Access and Network Services

AAA authorization limits the services available to a user. When AAA authorization is enabled, the switch uses information retrieved from the user's profile, which is located either in the local user database or on the security server, to configure the user's session. The user is granted access to a requested service only if the information in the user profile allows it.

TACACS+ Accounting

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

Default TACACS+ Configuration

TACACS+ and AAA are disabled by default.

To prevent a lapse in security, you cannot configure TACACS+ through a network management application. When enabled, TACACS+ can authenticate users accessing the switch through the CLI.

| Note | Although TACACS+ configuration is performed through the CLI, the TACACS+ server authenticates HTTP connections that have been configured with a privilege level of 15. |

How to Configure Switch Access with TACACS+

This section describes how to configure your switch to support TACACS+.

Identifying the TACACS+ Server Host and Setting the Authentication Key

Follow these steps to identify the TACACS+ server host and set the authentication key:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	tacacs server server-name **Example:** `Device(config)# tacacs server yourserver`	Identifies the IP host or hosts maintaining a TACACS+ server. Enter this command multiple times to create a list of preferred hosts. The software searches for hosts in the order in which you specify them. For server-name, specify the server name.
Step 4	address {ipv4 | ipv6} ip address **Example:** `Device(config-server-tacacs)# address ipv4 10.0.1.12`	Configures the IP address for the TACACS server.
Step 5	exit **Example:**	Exits the TACACS server mode and enters the global configuration mode.

	Command or Action	Purpose
	Device(config-server-tacacs)# **exit**	
Step 6	aaa new-model **Example:** Device(config)# **aaa new-model**	Enables AAA.
Step 7	aaa group server tacacs+ group-name **Example:** Device(config)# **aaa group server tacacs+ your_server_group**	(Optional) Defines the AAA server-group with a group name, and enters server group configuration mode.
Step 8	server name server-name **Example:** Device(config-sg-tacacs)# **server name yourserver**	(Optional) Associates a particular TACACS+ server with the defined server group. Repeat this step for each TACACS+ server in the AAA server group. Each server in the group must be previously defined in Step 3.
Step 9	end **Example:** Device(config-sg-tacacs)# end	Exits server group configuration mode and returns to privileged EXEC mode.

Configuring TACACS+ Login Authentication

Follow these steps to configure TACACS+ login authentication:

Before you begin

To configure AAA authentication, you define a named list of authentication methods and then apply that list to various ports.

Note   To secure the for HTTP access by using AAA methods, you must configure the with the ip http authentication aaa global configuration command. Configuring AAA authentication does not secure the for HTTP access by using AAA methods.

For more information about the ip http authentication command, see the Cisco IOS Security Command Reference, Release 12.4.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	aaa new-model **Example:** Device(config)# **aaa new-model**	Enables AAA.
Step 4	aaa authentication login {default	list-name} method1 [method2...] **Example:** Device(config)# **aaa authentication login default tacacs+ local**	Creates a login authentication method list. • To create a default list that is used when a named list is not specified in the login authentication command, use the default keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all ports. • For list-name, specify a character string to name the list you are creating. • For method1..., specify the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. Select one of these methods: • enable—Use the enable password for authentication. Before you can use this authentication method, you must define an enable password by using the enable password global configuration command. • group tacacs+—Uses TACACS+ authentication. Before you can use this authentication method, you must configure the TACACS+ server.

	Command or Action	Purpose
		• line —Use the line password for authentication. Before you can use this authentication method, you must define a line password. Use the password password line configuration command.
		• local—Use the local username database for authentication. You must enter username information in the database. Use the username password global configuration command.
		• local-case—Use a case-sensitive local username database for authentication. You must enter username information in the database by using the username name password global configuration command.
		• none—Do not use any authentication for login.
Step 5	line [console | tty | vty] line-number [ending-line-number] **Example:** Device(config)# **line 2 4**	Enters line configuration mode, and configures the lines to which you want to apply the authentication list.
Step 6	login authentication {default | list-name} **Example:** Device(config-line)# **login authentication default**	Applies the authentication list to a line or set of lines. • If you specify default, use the default list created with the aaa authentication login command. • For list-name, specify the list created with the aaa authentication login command.
Step 7	end **Example:** Device(config-line)# **end**	Returns to privileged EXEC mode.
Step 8	show running-config **Example:** Device# **show running-config**	Verifies your entries.

	Command or Action	Purpose
Step 9	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring TACACS+ Authorization for Privileged EXEC Access and Network Services

You can use the aaa authorization global configuration command with the tacacs+ keyword to set parameters that restrict a user's network access to privileged EXEC mode.

> ✎
>
> **Note** Authorization is bypassed for authenticated users who log in through the CLI even if authorization has been configured.

Follow these steps to specify TACACS+ authorization for privileged EXEC access and network services:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	aaa authorization network tacacs+ **Example:** Device(config)# **aaa authorization network tacacs+**	Configures the switch for user TACACS+ authorization for all network-related service requests.
Step 4	aaa authorization exec tacacs+ **Example:** Device(config)# **aaa authorization exec**	Configures the switch for user TACACS+ authorization if the user has privileged EXEC access.

	Command or Action	**Purpose**
	`tacacs+`	The exec keyword might return user profile information (such as autocommand information).
Step 5	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 7	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Starting TACACS+ Accounting

Follow these steps to start TACACS+ Accounting:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	aaa accounting network start-stop tacacs+ **Example:** Device(config)# **aaa accounting network start-stop tacacs+**	Enables TACACS+ accounting for all network-related service requests.

	Command or Action	Purpose
Step 4	aaa accounting exec start-stop tacacs+ **Example:** Device(config)# **aaa accounting exec start-stop tacacs+**	Enables TACACS+ accounting to send a start-record accounting notice at the beginning of a privileged EXEC process and a stop-record at the end.
Step 5	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 7	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

What to do next

To establish a session with a router if the AAA server is unreachable, use the aaa accounting system guarantee-first command. It guarantees system accounting as the first record, which is the default condition. In some situations, users might be prevented from starting a session on the console or terminal connection until after the system reloads, which can take more than 3 minutes.

To establish a console or Telnet session with the router if the AAA server is unreachable when the router reloads, use the no aaa accounting system guarantee-first command.

Establishing a Session with a Router if the AAA Server is Unreachable

To establishing a session with a router if the AAA server is unreachable, use the aaa accounting system guarantee-first command. It guarantees system accounting as the first record, which is the default condition. In some situations, users might be prevented from starting a session on the console or terminal connection until after the system reloads, which can take more than 3 minutes.

To establish a console or Telnet session with the router if the AAA server is unreachable when the router reloads, use the no aaa accounting system guarantee-first command.

Monitoring TACACS+

Table 14: Commands for Displaying TACACS+ Information

Command	Purpose
show tacacs	Displays TACACS+ server statistics.

Additional References For Switch Access with TACACS+

Related Documents

Related Topic	Document Title
AAA configuration	Configuring Local Authentication and Authorization

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History for TACACS+

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	TACACS+	TACACS+ provides detailed accounting information and flexible administrative control over authentication and authorization processes. TACACS+ is facilitated through AAA and can be enabled only through AAA commands.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.
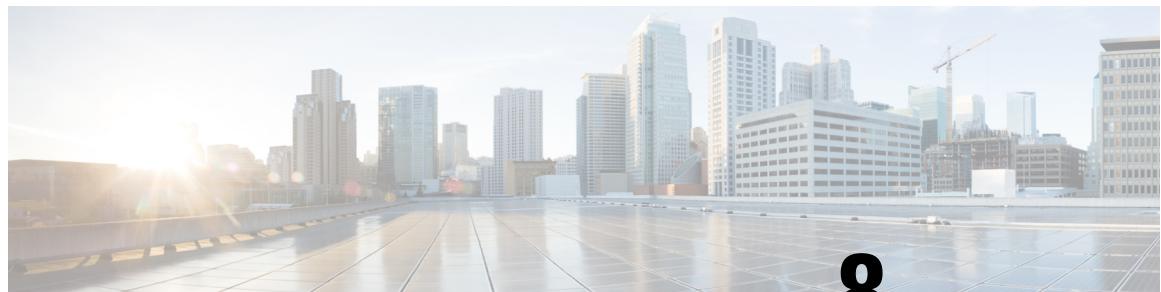
Configuring RADIUS

Prerequisites for Configuring RADIUS

This section lists the prerequisites for controlling Device access with RADIUS.

General:

- RADIUS and Authentication, Authorization, and Accounting (AAA) must be enabled to use any of the configuration commands in this chapter.

- RADIUS is facilitated through AAA and can be enabled only through AAA commands.

- Use the aaa new-model global configuration command to enable AAA.

- Use the aaa authentication global configuration command to define method lists for RADIUS authentication.

- Use line and interface commands to enable the defined method lists to be used.

- At a minimum, you must identify the host or hosts that run the RADIUS server software and define the method lists for RADIUS authentication. You can optionally define method lists for RADIUS authorization and accounting.

- You should have access to and should configure a RADIUS server before configuring RADIUS features on your Device.

- The RADIUS host is normally a multiuser system running RADIUS server software from Cisco (Cisco Secure Access Control Server Version 3.0), Livingston, Merit, Microsoft, or another software provider. For more information, see the RADIUS server documentation.

- To use the Change-of-Authorization (CoA) interface, a session must already exist on the switch. CoA can be used to identify a session and enforce a disconnect request. The update affects only the specified session.

For RADIUS operation:

- Users must first successfully complete RADIUS authentication before proceeding to RADIUS authorization, if it is enabled.

Restrictions for Configuring RADIUS

This topic covers restrictions for controlling Device access with RADIUS.

General:

- To prevent a lapse in security, you cannot configure RADIUS through a network management application.

RADIUS is not suitable in the following network security situations:

- Multiprotocol access environments. RADIUS does not support AppleTalk Remote Access (ARA), NetBIOS Frame Control Protocol (NBFCP), NetWare Asynchronous Services Interface (NASI), or X.25 PAD connections.

- Switch-to-switch or router-to-router situations. RADIUS does not provide two-way authentication. RADIUS can be used to authenticate from one device to a non-Cisco device if the non-Cisco device requires authentication.

- Networks using a variety of services. RADIUS generally binds a user to one service model.

Information about RADIUS

RADIUS and Switch Access

This section describes how to enable and configure RADIUS. RADIUS provides detailed accounting information and flexible administrative control over the authentication and authorization processes.

RADIUS Overview

RADIUS is a distributed client/server system that secures networks against unauthorized access. RADIUS clients run on supported Cisco routers and switches. Clients send authentication requests to a central RADIUS server, which contains all user authentication and network service access information.

Use RADIUS in these network environments that require access security:

- Networks with multiple-vendor access servers, each supporting RADIUS. For example, access servers from several vendors use a single RADIUS server-based security database. In an IP-based network with multiple vendors' access servers, dial-in users are authenticated through a RADIUS server that has been customized to work with the Kerberos security system.

- Turnkey network security environments in which applications support the RADIUS protocol, such as in an access environment that uses a smart card access control system. In one case, RADIUS has been used with Enigma's security cards to validates users and to grant access to network resources.

- Networks already using RADIUS. You can add a Cisco Device containing a RADIUS client to the network. This might be the first step when you make a transition to a TACACS+ server. See Figure: Transitioning from RADIUS to TACACS+ Services below.

- Network in which the user must only access a single service. Using RADIUS, you can control user access to a single host, to a single utility such as Telnet, or to the network through a protocol such as IEEE 802.1x. For more information about this protocol, see Configuring IEEE 802.1x Port-Based Authentication chapter.

- Networks that require resource accounting. You can use RADIUS accounting independently of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, showing the amount of resources (such as time, packets, bytes, and so forth) used during the session. An Internet service provider might use a freeware-based version of RADIUS access control and accounting software to meet special security and billing needs.

Figure 10: Transitioning from RADIUS to TACACS+ Services
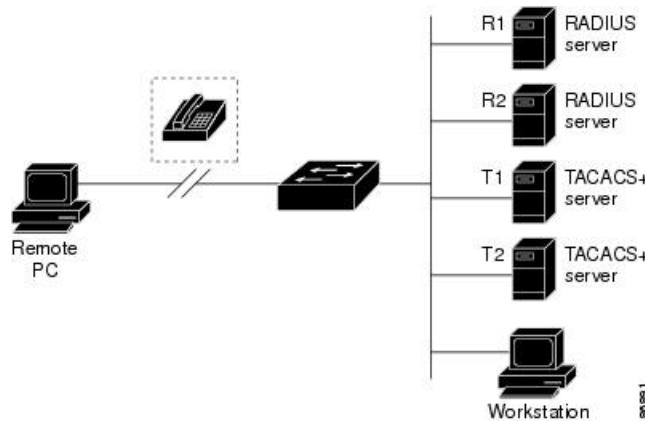
RADIUS Operation

When a user attempts to log in and authenticate to a Device that is access controlled by a RADIUS server, these events occur:

1. The user is prompted to enter 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 either not authenticated and is prompted to re-enter the username and password, or access is denied.

   - CHALLENGE—A challenge requires additional data from the user.

   - CHALLENGE PASSWORD—A response requests the user to select a new password.

The ACCEPT or REJECT response is bundled with additional data that is used for privileged EXEC or network authorization. The additional data included with the ACCEPT or REJECT packets includes these items:

• Telnet, SSH, rlogin, or privileged EXEC services

• Connection parameters, including the host or client IP address, access list, and user timeouts

RADIUS Change of Authorization

The RADIUS Change of Authorization (CoA) provides a mechanism to change the attributes of an authentication, authorization, and accounting (AAA) session after it is authenticated. When a policy changes for a user or user group in AAA, administrators can send RADIUS CoA packets from the AAA server such as a Cisco Secure Access Control Server (ACS) to reinitialize authentication and apply the new policy. This section provides an overview of the RADIUS interface including available primitives and how they are used during a CoA.

• Change-of-Authorization Requests

• CoA Request Response Code

• CoA Request Commands

• Session Reauthentication

• Stacking Guidelines for Session Termination

A standard RADIUS interface is typically used in a pulled model where the request originates from a network attached device and the response come from the queried servers. Catalyst support the RADIUS CoA extensions defined in RFC 5176 that are typically used in a pushed model and allow for the dynamic reconfiguring of sessions from external AAA or policy servers.

The supports these per-session CoA requests:

• Session reauthentication

• Session termination

• Session termination with port shutdown

• Session termination with port bounce

This feature is integrated with Cisco Secure Access Control Server (ACS) 5.1.

The RADIUS interface is enabled by default on Catalyst . However, some basic configuration is required for the following attributes:

• Security and Password—refer to the "Preventing Unauthorized Access to Your Switch" section in this guide.

• Accounting—refer to the "Starting RADIUS Accounting" section in the Configuring Switch-Based Authentication chapter in this guide.

Cisco IOS software supports the RADIUS CoA extensions defined in RFC 5176 that are typically used in a push model to allow the dynamic reconfiguring of sessions from external AAA or policy servers. Per-session CoA requests are supported for session identification, session termination, host reauthentication, port shutdown, and port bounce. This model comprises one request (CoA-Request) and two possible response codes:

• CoA acknowledgement (ACK) [CoA-ACK]

• CoA nonacknowledgement (NAK) [CoA-NAK]

The request is initiated from a CoA client (typically a AAA or policy server) and directed to the device that acts as a listener.

The table below shows the RADIUS CoA commands and vendor-specific attributes (VSAs) supported by Identity-Based Networking Services. All CoA commands must include the session identifier between the device and the CoA client.

Table 15: RADIUS CoA Commands Supported by Identity-Based Networking Services

CoA Command	Cisco VSA
Activate service	Cisco:Avpair="subscriber:command=activate-service"
	Cisco:Avpair="subscriber:service-name=<service-name>"
	Cisco:Avpair="subscriber:precedence=<precedence-number>"
	Cisco:Avpair="subscriber:activation-mode=replace-all"
Deactivate service	Cisco:Avpair="subscriber:command=deactivate-service"
	Cisco:Avpair="subscriber:service-name=<service-name>"
Bounce host port	Cisco:Avpair="subscriber:command=bounce-host-port"
Disable host port	Cisco:Avpair="subscriber:command=disable-host-port"
Session query	Cisco:Avpair="subscriber:command=session-query"
Session reauthenticate	Cisco:Avpair="subscriber:command=reauthenticate"
	Cisco:Avpair="subscriber:reauthenticate-type=last" or
	Cisco:Avpair="subscriber:reauthenticate-type=rerun"
Session terminate	This is a standard disconnect request and does not require a VSA.
Interface template	Cisco:AVpair="interface-template-name=<interfacetemplate>"

Change-of-Authorization Requests

Change of Authorization (CoA) requests, as described in RFC 5176, are used in a push model to allow for session identification, host reauthentication, and session termination. The model is comprised of one request (CoA-Request) and two possible response codes:

- CoA acknowledgment (ACK) [CoA-ACK]

- CoA non-acknowledgment (NAK) [CoA-NAK]

The request is initiated from a CoA client (typically a RADIUS or policy server) and directed to the switch that acts as a listener.

RFC 5176 Compliance

The Disconnect Request message, which is also referred to as Packet of Disconnect (POD), is supported by the switch for session termination.

This table shows the IETF attributes are supported for this feature.

Table 16: Supported IETF Attributes

Attribute Number	Attribute Name
24	State
31	Calling-Station-ID
44	Acct-Session-ID
80	Message-Authenticator
101	Error-Cause

This table shows the possible values for the Error-Cause attribute.

Table 17: Error-Cause Values

Value	Explanation
201	Residual Session Context Removed
202	Invalid EAP Packet (Ignored)
401	Unsupported Attribute
402	Missing Attribute
403	NAS Identification Mismatch
404	Invalid Request
405	Unsupported Service
406	Unsupported Extension
407	Invalid Attribute Value
501	Administratively Prohibited
502	Request Not Routable (Proxy)
503	Session Context Not Found
504	Session Context Not Removable
505	Other Proxy Processing Error
506	Resources Unavailable
507	Request Initiated
508	Multiple Session Selection Unsupported

CoA Request Response Code

The CoA Request response code can be used to convey a command to the switch.

The packet format for a CoA Request Response code as defined in RFC 5176 consists of the following fields: Code, Identifier, Length, Authenticator, and Attributes in the Type:Length:Value (TLV) format. The Attributes field is used to carry Cisco vendor-specific attributes (VSAs).

Session Identification

For disconnect and CoA requests targeted at a particular session, the switch locates the session based on one or more of the following attributes:

- Acct-Session-Id (IETF attribute #44)

- Audit-Session-Id (Cisco VSA)

- Calling-Station-Id (IETF attribute #31 which contains the host MAC address)

- IPv6 Attributes, which can be one of the following:

    - Framed-IPv6-Prefix (IETF attribute #97) and Framed-Interface-Id (IETF attribute #96), which together create a full IPv6 address per RFC 3162
    - Framed-IPv6-Address

- Plain IP Address (IETF attribute #8)

Unless all session identification attributes included in the CoA message match the session, the switch returns a Disconnect-NAK or CoA-NAK with the "Invalid Attribute Value" error-code attribute.

If more than one session identification attribute is included in the message, all the attributes must match the session or the switch returns a Disconnect- negative acknowledgment (NAK) or CoA-NAK with the error code "Invalid Attribute Value."

The packet format for a CoA Request code as defined in RFC 5176 consists of the fields: Code, Identifier, Length, Authenticator, and Attributes in Type:Length:Value (TLV) format.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Authenticator                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-
```
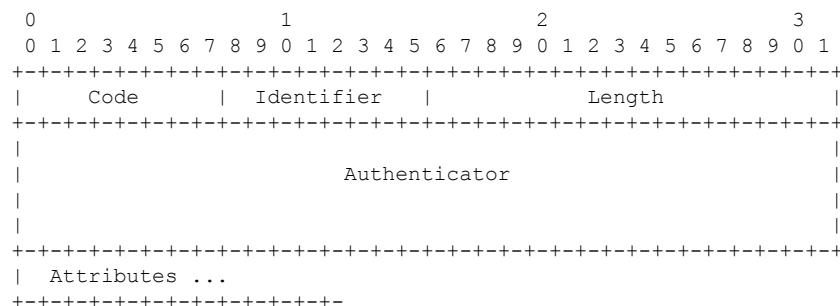
The attributes field is used to carry Cisco vendor-specific attributes (VSAs).

For CoA requests targeted at a particular enforcement policy, the device returns a CoA-NAK with the error code "Invalid Attribute Value" if any of the above session identification attributes are included in the message.

CoA ACK Response Code

If the authorization state is changed successfully, a positive acknowledgment (ACK) is sent. The attributes returned within CoA ACK will vary based on the CoA Request and are discussed in individual CoA Commands.

CoA NAK Response Code

A negative acknowledgment (NAK) indicates a failure to change the authorization state and can include attributes that indicate the reason for the failure. Use show commands to verify a successful CoA.

CoA Request Commands

Table 18: CoA Commands Supported on the

Command [1]	Cisco VSA
Reauthenticate host	Cisco:Avpair="subscriber:command=reauthenticate"
Terminate session	This is a standard disconnect request that does not require a VSA.
Bounce host port	Cisco:Avpair="subscriber:command=bounce-host-port"
Disable host port	Cisco:Avpair="subscriber:command=disable-host-port"

[1] All CoA commands must include the session identifier between the and the CoA client.

Session Reauthentication

The AAA server typically generates a session reauthentication request when a host with an unknown identity or posture joins the network and is associated with a restricted access authorization profile (such as a guest VLAN). A reauthentication request allows the host to be placed in the appropriate authorization group when its credentials are known.

To initiate session authentication, the AAA server sends a standard CoA-Request message which contains a Cisco VSA in this form: Cisco:Avpair="subscriber:command=reauthenticate" and one or more session identification attributes.

The current session state determines the switch response to the message. If the session is currently authenticated by IEEE 802.1x, the switch responds by sending an EAPoL (Extensible Authentication Protocol over Lan) -RequestId message to the server.

If the session is currently authenticated by MAC authentication bypass (MAB), the switch sends an access-request to the server, passing the same identity attributes used for the initial successful authentication.

If session authentication is in progress when the switch receives the command, the switch terminates the process, and restarts the authentication sequence, starting with the method configured to be attempted first.

If the session is not yet authorized, or is authorized via guest VLAN, or critical VLAN, or similar policies, the reauthentication message restarts the access control methods, beginning with the method configured to be attempted first. The current authorization of the session is maintained until the reauthentication leads to a different authorization result.

Session Reauthentication in a Switch Stack

When a switch stack receives a session reauthentication message:

- It checkpoints the need for a re-authentication before returning an acknowledgment (ACK).

- It initiates reauthentication for the appropriate session.

- If authentication completes with either success or failure, the signal that triggered the reauthentication is removed from the stack's member switch.

- If the stack's active switch fails before authentication completes, reauthentication is initiated after active switch changeover based on the original command (which is subsequently removed).

- If the active switch fails before sending an ACK, the new active switch treats the re-transmitted command as a new command.

Session Termination

There are three types of CoA requests that can trigger session termination. A CoA Disconnect-Request terminates the session, without disabling the host port. This command causes re-initialization of the authenticator state machine for the specified host, but does not restrict that host access to the network.

To restrict a host's access to the network, use a CoA Request with the Cisco:Avpair="subscriber:command=disable-host-port" VSA. This command is useful when a host is known to be causing problems on the network, and you need to immediately block network access for the host. When you want to restore network access on the port, re-enable it using a non-RADIUS mechanism.

When a device with no supplicant, such as a printer, needs to acquire a new IP address (for example, after a VLAN change), terminate the session on the host port with port-bounce (temporarily disable and then re-enable the port).

CoA Disconnect-Request

This command is a standard Disconnect-Request. If the session cannot be located, the switch returns a Disconnect-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the switch terminates the session. After the session has been completely removed, the switch returns a Disconnect-ACK.

If the switch fails-over to a standby switch before returning a Disconnect-ACK to the client, the process is repeated on the new active switch when the request is re-sent from the client. If the session is not found following re-sending, a Disconnect-ACK is sent with the "Session Context Not Found" error-code attribute.

CoA Request: Disable Host Port

The RADIUS server CoA disable port command administratively shuts down the authentication port that is hosting a session, resulting in session termination. This command is useful when a host is known to cause problems on the network and network access needs to be immediately blocked for the host. To restore network access on the port, reenable it using a non-RADIUS mechanism. This command is carried in a standard CoA-Request message that has this new vendor-specific attribute (VSA):

Cisco:Avpair="subscriber:command=disable-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification" section. If the session cannot be located, the switch returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the switch disables the hosting port and returns a CoA-ACK message.

If the switch fails before returning a CoA-ACK to the client, the process is repeated on the new active switch when the request is re-sent from the client. If the switch fails after returning a CoA-ACK message to the client but before the operation has completed, the operation is restarted on the new active switch.

Note   A Disconnect-Request failure following command re-sending could be the result of either a successful session termination before change-over (if the Disconnect-ACK was not sent) or a session termination by other means (for example, a link failure) that occurred after the original command was issued and before the standby switch became active.

CoA Request: Bounce-Port

A RADIUS server CoA bounce port sent from a RADIUS server can cause a link flap on an authentication port, which triggers DHCP renegotiation from one or more hosts connected to this port. This incident can occur when there is a VLAN change and the endpoint is a device (such as a printer) that does not have a mechanism to detect a change on this authentication port. The CoA bounce port is carried in a standard CoA-Request message that contains the following VSA:

Cisco:Avpair="subscriber:command=bounce-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes. If the session cannot be located, the switch returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the switch disables the hosting port for a period of 10 seconds, re-enables it (port-bounce), and returns a CoA-ACK.

If the switch fails before returning a CoA-ACK to the client, the process is repeated on the new active switch when the request is re-sent from the client. If the switch fails after returning a CoA-ACK message to the client but before the operation has completed, the operation is re-started on the new active switch.

Stacking Guidelines for Session Termination

No special handling is required for CoA Disconnect-Request messages in a switch stack.

Stacking Guidelines for CoA-Request Bounce-Port

Because the bounce-port command is targeted at a session, not a port, if the session is not found, the command cannot be executed.

When the Auth Manager command handler on the active switch receives a valid bounce-port command, it checkpoints the following information before returning a CoA-ACK message:

- the need for a port-bounce
- the port-id (found in the local session context)

The switch initiates a port-bounce (disables the port for 10 seconds, then re-enables it).

If the port-bounce is successful, the signal that triggered the port-bounce is removed from the standby switch.

If the active switch fails before the port-bounce completes, a port-bounce is initiated after an active switch changeover based on the original command (which is subsequently removed).

If the active switch fails before sending a CoA-ACK message, the new active switch treats the re-sent command as a new command.

Stacking Guidelines for CoA-Request Disable-Port

Because the disable-port command is targeted at a session, not a port, if the session is not found, the command cannot be executed.

When the Auth Manager command handler on the active switch receives a valid disable-port command, it verifies this information before returning a CoA-ACK message:

- the need for a port-disable
- the port-id (found in the local session context)

The switch attempts to disable the port.

If the port-disable operation is successful, the signal that triggered the port-disable is removed from the standby switch.

If the active switch fails before the port-disable operation completes, the port is disabled after an active switch changeover based on the original command (which is subsequently removed).

If the active switch fails before sending a CoA-ACK message, the new active switch treats the re-sent command as a new command.

Default RADIUS Configuration

RADIUS and AAA are disabled by default.

To prevent a lapse in security, you cannot configure RADIUS through a network management application. When enabled, RADIUS can authenticate users accessing the switch through the CLI.

RADIUS Server Host

Switch-to-RADIUS-server communication involves several components:

- Hostname or IP address
- Authentication destination port
- Accounting destination port
- Key string
- Timeout period
- Retransmission value

You identify RADIUS security servers by their hostname or IP address, hostname and specific UDP port numbers, or their IP address and specific UDP port numbers. The combination of the IP address and the UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. This unique identifier enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address.

If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as a fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the %RADIUS-4-RADIUS_DEAD message appears, and then the switch tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order that they are configured.)

A RADIUS server and the switch use a shared secret text string to encrypt passwords and exchange responses. To configure RADIUS to use the AAA security commands, you must specify the host running the RADIUS server daemon and a secret text (key) string that it shares with the switch.

The timeout, retransmission, and encryption key values can be configured globally for all RADIUS servers, on a per-server basis, or in some combination of global and per-server settings.

RADIUS Login Authentication

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

A method list describes the sequence and authentication methods to be queried to authenticate a user. You can designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops, and no other authentication methods are attempted.

AAA Server Groups

You can configure the switch to use AAA server groups to group existing server hosts for authentication. You select a subset of the configured server hosts and use them for a particular service. The server group is used with a global server-host list, which lists the IP addresses of the selected server hosts.

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

AAA Authorization

AAA authorization limits the services available to a user. When AAA authorization is enabled, the switch uses information retrieved from the user's profile, which is in the local user database or on the security server, to configure the user's session. The user is granted access to a requested service only if the information in the user profile allows it.

RADIUS Accounting

The AAA accounting feature tracks the services that users are using and the amount of network resources that they are consuming. When you enable AAA accounting, the switch reports user activity to the RADIUS security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. You can then analyze the data for network management, client billing, or auditing.

Vendor-Specific RADIUS Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the switch and the RADIUS server by using the vendor-specific attribute (attribute 26). Vendor-specific attributes (VSAs) allow vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option by using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named cisco-avpair. The value is a string with this format:

```
protocol : attribute sep value *
```

Protocol is a value of the Cisco protocol attribute for a particular type of authorization. Attribute and value are an appropriate attributevalue (AV) pair defined in the Cisco TACACS+ specification, and sep is = for mandatory attributes and is * for optional attributes. The full set of features available for TACACS+ authorization can then be used for RADIUS.

For example, the following AV pair causes Cisco's "multiple named IP address pools" feature to be activated during IP authorization (during PPP's Internet Protocol Control Protocol (IPCP) address assignment):

```
cisco-avpair= "ip:addr-pool=first"
```

If you insert an "*", the AV pair "ip:addr-pool=first" becomes optional. Note that any AV pair can be made optional:

```
cisco-avpair= "ip:addr-pool*first"
```

The following example shows how to cause a user logging in from a network access server to have immediate access to EXEC commands:

```
cisco-avpair= "shell:priv-lvl=15"
```

Other vendors have their own unique vendor-IDs, options, and associated VSAs. For more information about vendor-IDs and VSAs, see RFC 2138, "Remote Authentication Dial-In User Service (RADIUS)."

Attribute 26 contains the following three elements:

- Type

- Length

- String (also known as data)

    - Vendor-Id
    - Vendor-Type
    - Vendor-Length
    - Vendor-Data

The figure below shows the packet format for a VSA encapsulated "behind" attribute 26.

Figure 11: VSA Encapsulated Behind Attribute 26
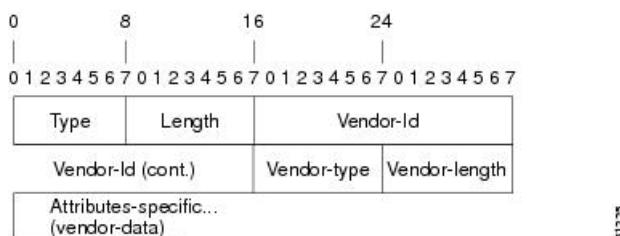
Note It is up to the vendor to specify the format of their VSA. The Attribute-Specific field (also known as Vendor-Data) is dependent on the vendor's definition of that attribute.

The table below describes significant fields listed in the Vendor-Specific RADIUS IETF Attributes table (second table below), which lists supported vendor-specific RADIUS attributes (IETF attribute 26).

Table 19: Vendor-Specific Attributes Table Field Descriptions

Field	Description
Number	All attributes listed in the following table are extensions of IETF attribute 26.
Vendor-Specific Command Codes	A defined code used to identify a particular vendor. Code 9 defines Cisco VSAs, 311 defines Microsoft VSAs, and 529 defines Ascend VSAs.
Sub-Type Number	The attribute ID number. This number is much like the ID numbers of IETF attributes, except it is a "second layer" ID number encapsulated behind attribute 26.
Attribute	The ASCII string name of the attribute.
Description	Description of the attribute.

Table 20: Vendor-Specific RADIUS IETF Attributes

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
MS-CHAP Attributes				
26	311	1	MSCHAP-Response	Contains the response value provided by a PPP MS-CHAP user in response to the challenge. It is only used in Access-Request packets. This attribute is identical to the PPP CHAP Identifier. (RFC 2548

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	311	11	MSCHAP-Challenge	Contains the challenge sent by a network access server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets. (RFC 2548)
VPDN Attributes				
26	9	1	l2tp-cm-local-window-size	Specifies the maximum receive window size for L2TP control messages. This value is advertised to the peer during tunnel establishment.
26	9	1	l2tp-drop-out-of-order	Respects sequence numbers on data packets by dropping those that are received out of order. This does not ensure that sequence numbers will be sent on data packets, just how to handle them if they are received.
26	9	1	l2tp-hello-interval	Specifies the number of seconds for the hello keepalive interval. Hello packets are sent when no data has been sent on a tunnel for the number of seconds configured here.
26	9	1	l2tp-hidden-avp	When enabled, sensitive AVPs in L2TP control messages are scrambled or hidden.
26	9	1	l2tp-nosession-timeout	Specifies the number of seconds that a tunnel will stay active with no sessions before timing out and shutting down.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	tunnel-tos-reflect	Copies the IP ToS field from the IP header of each payload packet to the IP header of the tunnel packet for packets entering the tunnel at the LNS.
26	9	1	l2tp-tunnel-authen	If this attribute is set, it performs L2TP tunnel authentication.
26	9	1	l2tp-tunnel-password	Shared secret used for L2TP tunnel authentication and AVP hiding.
26	9	1	l2tp-udp-checksum	This is an authorization attribute and defines whether L2TP should perform UDP checksums for data packets. Valid values are "yes" and "no." The default is no.
Store and Forward Fax Attributes				
26	9	3	Fax-Account-Id-Origin	Indicates the account ID origin as defined by system administrator for the mmoip aaa receive-id or the mmoip aaa send-id commands.
26	9	4	Fax-Msg-Id=	Indicates a unique fax message identification number assigned by Store and Forward Fax.
26	9	5	Fax-Pages	Indicates the number of pages transmitted or received during this fax session. This page count includes cover pages.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	6	Fax-Coverpage-Flag	Indicates whether or not a cover page was generated by the off-ramp gateway for this fax session. True indicates that a cover page was generated; false means that a cover page was not generated.
26	9	7	Fax-Modem-Time	Indicates the amount of time in seconds the modem sent fax data (x) and the amount of time in seconds of the total fax session (y), which includes both fax-mail and PSTN time, in the form x/y. For example, 10/15 means that the transfer time took 10 seconds, and the total fax session took 15 seconds.
26	9	8	Fax-Connect-Speed	Indicates the modem speed at which this fax-mail was initially transmitted or received. Possible values are 1200, 4800, 9600, and 14400.
26	9	9	Fax-Recipient-Count	Indicates the number of recipients for this fax transmission. Until e-mail servers support Session mode, the number should be 1.
26	9	10	Fax-Process-Abort-Flag	Indicates that the fax session was cancelled or successful. True means that the session was cancelled; false means that the session was successful.
26	9	11	Fax-Dsn-Address	Indicates the address to which DSNs will be sent.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	12	Fax-Dsn-Flag	Indicates whether or not DSN has been enabled. True indicates that DSN has been enabled; false means that DSN has not been enabled.
26	9	13	Fax-Mdn-Address	Indicates the address to which MDNs will be sent.
26	9	14	Fax-Mdn-Flag	Indicates whether or not message delivery notification (MDN) has been enabled. True indicates that MDN had been enabled; false means that MDN had not been enabled.
26	9	15	Fax-Auth-Status	Indicates whether or not authentication for this fax session was successful. Possible values for this field are success, failed, bypassed, or unknown.
26	9	16	Email-Server-Address	Indicates the IP address of the e-mail server handling the on-ramp fax-mail message.
26	9	17	Email-Server-Ack-Flag	Indicates that the on-ramp gateway has received a positive acknowledgment from the e-mail server accepting the fax-mail message.
26	9	18	Gateway-Id	Indicates the name of the gateway that processed the fax session. The name appears in the following format: hostname.domain-name.
26	9	19	Call-Type	Describes the type of fax activity: fax receive or fax send.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	20	Port-Used	Indicates the slot/port number of the Cisco AS5300 used to either transmit or receive this fax-mail.
26	9	21	Abort-Cause	If the fax session cancels, indicates the system component that signaled the cancel operation. Examples of system components that could trigger a cancel operation are FAP (Fax Application Process), TIFF (the TIFF reader or the TIFF writer), fax-mail client, fax-mail server, ESMTP client, or ESMTP server.
H323 Attributes				
26	9	23	Remote-Gateway-ID (h323-remote-address)	Indicates the IP address of the remote gateway.
26	9	24	Connection-ID (h323-conf-id)	Identifies the conference ID.
26	9	25	Setup-Time (h323-setup-time)	Indicates the setup time for this connection in Coordinated Universal Time (UTC) formerly known as Greenwich Mean Time (GMT) and Zulu time.
26	9	26	Call-Origin (h323-call-origin)	Indicates the origin of the call relative to the gateway. Possible values are originating and terminating (answer).
26	9	27	Call-Type (h323-call-type)	Indicates call leg type. Possible values are telephony and VoIP.
26	9	28	Connect-Time (h323-connect-time)	Indicates the connection time for this call leg in UTC.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	29	Disconnect-Time (h323-disconnect-time)	Indicates the time this call leg was disconnected in UTC.
26	9	30	Disconnect-Cause (h323-disconnect-cause)	Specifies the reason a connection was taken offline per Q.931 specification.
26	9	31	Voice-Quality (h323-voice-quality)	Specifies the impairment factor (ICPIF) affecting voice quality for a call.
26	9	33	Gateway-ID (h323-gw-id)	Indicates the name of the underlying gateway.
Large Scale Dialout Attributes				
26	9	1	callback-dialstring	Defines a dialing string to be used for callback.
26	9	1	data-service	No description available.
26	9	1	dial-number	Defines the number to dial.
26	9	1	force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available.
26	9	1	map-class	Allows the user profile to reference information configured in a map class of the same name on the network access server that dials out.
26	9	1	send-auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	send-name	PPP name authentication. To apply for PAP, do not configure the ppp pap sent-name password command on the interface. For PAP, "preauth:send-name" and "preauth:send-secret" will be used as the PAP username and PAP password for outbound authentication. For CHAP, "preauth:send-name" will be used not only for outbound authentication, but also for inbound authentication. For a CHAP inbound case, the NAS will use the name defined in "preauth:send-name" in the challenge packet to the caller box. **Note** The send-name attribute has changed over time: Initially, it performed the functions now provided by both the send-name and remote-name attributes. Because the remote-name attribute has been added, the send-name attribute is restricted to its current behavior.

Configuring RADIUS

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	send-secret	PPP password authentication. The vendor-specific attributes (VSAs) "preauth:send-name" and "preauth:send-secret" will be used as the PAP username and PAP password for outbound authentication. For a CHAP outbound case, both "preauth:send-name" and "preauth:send-secret" will be used in the response packet.
26	9	1	remote-name	Provides the name of the remote host for use in large-scale dial-out. Dialer checks that the large-scale dial-out remote name matches the authenticated name, to protect against accidental user RADIUS misconfiguration. (For example, dialing a valid phone number but connecting to the wrong device.)
Miscellaneous Attributes				

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	2	Cisco-NAS-Port	Specifies additional vendor specific attribute (VSA) information for NAS-Port accounting. To specify additional NAS-Port information in the form an Attribute-Value Pair (AVPair) string, use the radius-server vsa send global configuration command. **Note** This VSA is typically used in Accounting, but may also be used in Authentication (Access-Request) packets.
26	9	1	min-links	Sets the minimum number of links for MLP.
26	9	1	proxyacl#<n>	Allows users to configure the downloadable user profiles (dynamic ACLs) by using the authentication proxy feature so that users can have the configured authorization to permit traffic going through the configured interfaces.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	spi	Carries the authentication information needed by the home agent to authenticate a mobile node during registration. The information is in the same syntax as the ip mobile secure host <addr> configuration command. Basically it contains the rest of the configuration command that follows that string, verbatim. It provides the Security Parameter Index (SPI), key, authentication algorithm, authentication mode, and replay protection timestamp range.

Vendor-Proprietary RADIUS Server Communication

Although an IETF draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the switch and the RADIUS server, some vendors have extended the RADIUS attribute set in a unique way. Cisco IOS software supports a subset of vendor-proprietary RADIUS attributes.

As mentioned earlier, to configure RADIUS (whether vendor-proprietary or IETF draft-compliant), you must specify the host running the RADIUS server daemon and the secret text string it shares with the switch. You specify the RADIUS host and secret text string by using the radius server global configuration commands.

How to Configure RADIUS

Identifying the RADIUS Server Host

To apply these settings globally to all RADIUS servers communicating with the device, use the three unique global configuration commands: radius-server timeout, radius-server retransmit, and key string.

You can configure the device to use AAA server groups to group existing server hosts for authentication.

You also need to configure some settings on the RADIUS server. These settings include the IP address of the device and the key string to be shared by both the server and the device.

Follow these steps to configure per-server RADIUS server communication.

Before you begin

If you configure both global and per-server functions (timeout, retransmission, and key commands) on the device, the per-server timer, retransmission, and key value commands override global timer, retransmission, and key value commands.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	radius server server name **Example:** `Device(config)# radius server rsim`	Specifies the name for the RADIUS server configuration for Protected Access Credential (PAC) provisioning, and enters RADIUS server configuration mode.
Step 4	address {ipv4 | ipv6}ip address{ auth-port port number | acct-port port number} **Example:** `Device(config-radius-server)# address ipv4 124.2.2.12 auth-port 1612`	(Optional) Specifies the RADIUS server parameters. For auth-port port-number, specify the UDP destination port for authentication requests. The default is 1645. The range is 0 to 65536. For acct-port port-number, specify the UDP destination port for authentication requests. The default is 1646.
Step 5	key string **Example:** `Device(config-radius-server)#  key rad123`	(Optional) For key string, specify the authentication and encryption key used between the device and the RADIUS daemon running on the RADIUS server. **Note** The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the radius server command. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.

	Command or Action	**Purpose**
Step 6	retransmit value **Example:** Device(config-radius-server)# **retransmit 10**	(Optional) Specifies the number of times a RADIUS request is resent when the server is not responding or responding slowly. The range is 1 to 100. This setting overrides the radius-server retransmit global configuration command setting.
Step 7	timeout seconds **Example:** Device(config-radius-server)# **timeout 60**	(Optional) Specifies the time interval that the device waits for the RADIUS server to reply before sending a request again. The range is 1 to 1000. This setting overrides the radius-server timeout global configuration command setting.
Step 8	end **Example:** Device(config-radius-server)# **end**	Exits RADIUS server configuration mode and enters privileged EXEC mode.

Configuring RADIUS Login Authentication

Follow these steps to configure RADIUS login authentication:

Before you begin

To secure the device for HTTP access by using AAA methods, you must configure the device with the ip http authentication aaa global configuration command. Configuring AAA authentication does not secure the device for HTTP access by using AAA methods.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	aaa new-model **Example:**	Enables AAA.

	Command or Action	**Purpose**
	Device(config)# **aaa new-model**	

Step 4	aaa authentication login {default | list-name} method1 [method2...]
	
	Example:
	
	Device(config)# **aaa authentication login default local**
		Creates a login authentication method list.
		
		• To create a default list that is used when a named list is not specified in the login authentication command, use the default keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all ports.
		
		• For list-name, specify a character string to name the list you are creating.
		
		• For method1..., specify the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails.
		
		Select one of these methods:
		
		  • enable—Use the enable password for authentication. Before you can use this authentication method, you must define an enable password by using the enable password global configuration command.
		
		  • group radius—Use RADIUS authentication. Before you can use this authentication method, you must configure the RADIUS server.
		
		  • line—Use the line password for authentication. Before you can use this authentication method, you must define a line password. Use the password password line configuration command.
		
		  • local—Use the local username database for authentication. You must enter username information in the database. Use the username name password global configuration command.
		
		  • local-case—Use a case-sensitive local username database for authentication. You must enter username information

	Command or Action	**Purpose**
		in the database by using the username password global configuration command.
		• none—Do not use any authentication for login.
Step 5	line [console | tty | vty] line-number [ending-line-number] **Example:** Device(config)# **line 1 4**	Enters line configuration mode, and configure the lines to which you want to apply the authentication list.
Step 6	login authentication {default | list-name} **Example:** Device(config)# **login authentication default**	Applies the authentication list to a line or set of lines. • If you specify default, use the default list created with the aaa authentication login command. • For list-name, specify the list created with the aaa authentication login command.
Step 7	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 8	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 9	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Defining AAA Server Groups

You use the server group server configuration command to associate a particular server with a defined group server. You can either identify the server by its IP address or identify multiple host instances or entries by using the optional auth-port and acct-port keywords.

Follow these steps to define AAA server groups:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	radius server name **Example:** Device(config)# **radius server ISE**	Specifies the name of the RADIUS server configuration for Protected Access Credential (PAC) provisioning and enters RADIUS server configuration mode. The device also supports RADIUS for IPv6.
Step 4	address {ipv4 | ipv6} {ip-address | hostname} auth-port port-number acct-port port-number **Example:** Device(config-radius-server)# **address ipv4 10.1.1.1 auth-port 1645 acct-port 1646**	Configures the IPv4 address for the RADIUS server accounting and authentication parameters.
Step 5	key string **Example:** Device(config-radius-server)# **key cisco123**	Specifies the authentication and encryption key for all RADIUS communications between the device and the RADIUS server.
Step 6	end **Example:** Device(config-radius-server)# **end**	Exits RADIUS server configuration mode and returns to privileged EXEC mode.
Step 7	show running-config **Example:** Device# **show running-config**	Verifies your entries.

	Command or Action	Purpose
Step 8	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring RADIUS Authorization for User Privileged Access and Network Services

> **Note** Authorization is bypassed for authenticated users who log in through the CLI even if authorization has been configured.

Follow these steps to configure RADIUS authorization for user priviledged access and network services:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	aaa authorization network radius **Example:** Device(config)# **aaa authorization network radius**	Configures the device for user RADIUS authorization for all network-related service requests.
Step 4	aaa authorization exec radius **Example:** Device(config)# **aaa authorization exec radius**	Configures the device for user RADIUS authorization if the user has privileged EXEC access. The exec keyword might return user profile information (such as autocommand information).

	Command or Action	**Purpose**
Step 5	end	Returns to privileged EXEC mode.
	Example:	
	Device(config)# **end**	
Step 6	show running-config	Verifies your entries.
	Example:	
	Device# **show running-config**	
Step 7	copy running-config startup-config	(Optional) Saves your entries in the configuration file.
	Example:	
	Device# **copy running-config startup-config**	

What to do next

You can use the aaa authorization global configuration command with the radius keyword to set parameters that restrict a user's network access to privileged EXEC mode.

The aaa authorization exec radius local command sets these authorization parameters:

- Use RADIUS for privileged EXEC access authorization if authentication was performed by using RADIUS.
- Use the local database if authentication was not performed by using RADIUS.

Starting RADIUS Accounting

Follow these steps to start RADIUS accounting:

Procedure

	Command or Action	**Purpose**
Step 1	enable	Enables privileged EXEC mode.
	Example:	- Enter your password if prompted.
	Device> **enable**	
Step 2	configure terminal	Enters global configuration mode.
	Example:	

	Command or Action	**Purpose**
	Device# **configure terminal**	
Step 3	aaa accounting network start-stop radius **Example:** Device(config)# **aaa accounting network start-stop radius**	Enables RADIUS accounting for all network-related service requests.
Step 4	aaa accounting exec start-stop radius **Example:** Device(config)# **aaa accounting exec start-stop radius**	Enables RADIUS accounting to send a start-record accounting notice at the beginning of a privileged EXEC process and a stop-record at the end.
Step 5	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 7	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring Settings for All RADIUS Servers

Beginning in privileged EXEC mode, follow these steps to configure settings for all RADIUS servers:

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.

	Command or Action	**Purpose**
Step 2	**radius-server key string** **Example:** Device(config)# **radius-server key your_server_key** Device(config)# **key your_server_key**	Specifies the shared secret text string used between the switch and all RADIUS servers. **Note** The key is a text string that must match the encryption key used on the RADIUS server. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.
Step 3	**radius-server retransmit retries** **Example:** Device(config)# **radius-server retransmit 5**	Specifies the number of times the switch sends each RADIUS request to the server before giving up. The default is 3; the range 1 to 1000.
Step 4	**radius-server timeout seconds** **Example:** Device(config)# **radius-server timeout 3**	Specifies the number of seconds a switch waits for a reply to a RADIUS request before resending the request. The default is 5 seconds; the range is 1 to 1000.
Step 5	**radius-server deadtime minutes** **Example:** Device(config)# **radius-server deadtime 0**	When a RADIUS server is not responding to authentication requests, this command specifies a time to stop the request on that server. This avoids the wait for the request to timeout before trying the next configured server. The default is 0; the range is 1 to 1440 minutes.
Step 6	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

Command or Action	Purpose
Device# **copy running-config startup-config**	

Configuring the Device to Use Vendor-Specific RADIUS Attributes

Follow these steps to configure the device to use vendor-specific RADIUS attributes:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	radius-server vsa send [accounting | authentication] **Example:** Device(config)# **radius-server vsa send accounting**	Enables the device to recognize and use VSAs as defined by RADIUS IETF attribute 26. • (Optional) Use the accounting keyword to limit the set of recognized vendor-specific attributes to only accounting attributes. • (Optional) Use the authentication keyword to limit the set of recognized vendor-specific attributes to only authentication attributes. If you enter this command without keywords, both accounting and authentication vendor-specific attributes are used.
Step 4	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 5	show running-config **Example:**	Verifies your entries.

	Command or Action	Purpose
	Device# **show running-config**	
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring the Device for Vendor-Proprietary RADIUS Server Communication

Follow these steps to configure vendor-proprietary RADIUS server communication:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	radius server server name **Example:** Device(config)# **radius server rsim**	Specifies the name for the RADIUS server configuration for Protected Access Credential (PAC) provisioning, and enters RADIUS server configuration mode.
Step 4	address { ipv4 | ipv6 } ip address **Example:** Device(config-radius-server)# **address ipv4 172.24.25.10**	(Optional) Specifies the IP address of the RADIUS server.
Step 5	non-standard **Example:** Device(config-radius-server)# **non-standard**	Identifies that the RADIUS server using a vendor-proprietary implementation of RADIUS.
Step 6	key string **Example:**	Specifies the shared secret text string used between the device and the vendor-proprietary

	Command or Action	Purpose
	Device(config-radius-server)# **key rad123**	RADIUS server. The device and the RADIUS server use this text string to encrypt passwords and exchange responses.
Step 7	end **Example:** Device(config-radius-server)# **end**	Exits RADIUS server mode and enters privileged EXEC mode.

Configuring CoA on the Device

Follow these steps to configure CoA on a device. This procedure is required.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	aaa new-model **Example:** Device(config)# **aaa new-model**	Enables AAA.
Step 4	aaa server radius dynamic-author **Example:** Device(config)# **aaa server radius dynamic-author**	Configures the device as an authentication, authorization, and accounting (AAA) server to facilitate interaction with an external policy server.
Step 5	client {ip-address	name} [vrf vrfname] [server-key string]	Enters dynamic authorization local server configuration mode and specifies a RADIUS client from which a device will accept CoA and disconnect requests.

	Command or Action	Purpose
Step 6	server-key [0 | 7] string **Example:** Device(config-sg-radius)# **server-key your_server_key**	Configures the RADIUS key to be shared between a device and RADIUS clients.
Step 7	port port-number **Example:** Device(config-sg-radius)# **port 25**	Specifies the port on which a device listens for RADIUS requests from configured RADIUS clients.
Step 8	auth-type {any | all | session-key} **Example:** Device(config-sg-radius)# **auth-type any**	Specifies the type of authorization the device uses for RADIUS clients. The client must match all the configured attributes for authorization.
Step 9	ignore session-key	(Optional) Configures the device to ignore the session-key. For more information about the ignore command, see the Cisco IOS Intelligent Services Gateway Command Reference on Cisco.com.
Step 10	ignore server-key **Example:** Device(config-sg-radius)# **ignore server-key**	(Optional) Configures the device to ignore the server-key. For more information about the ignore command, see the Cisco IOS Intelligent Services Gateway Command Reference on Cisco.com.
Step 11	authentication command bounce-port ignore **Example:** Device(config-sg-radius)# **authentication command bounce-port ignore**	(Optional) Configures the device to ignore a CoA request to temporarily disable the port hosting a session. The purpose of temporarily disabling the port is to trigger a DHCP renegotiation from the host when a VLAN change occurs and there is no supplicant on the endpoint to detect the change.
Step 12	authentication command disable-port ignore **Example:** Device(config-sg-radius)# **authentication command disable-port ignore**	(Optional) Configures the device to ignore a nonstandard command requesting that the port hosting a session be administratively shut down. Shutting down the port results in termination of the session. Use standard CLI or SNMP commands to re-enable the port.

	Command or Action	Purpose
Step 13	end **Example:** Device(config-sg-radius)# **end**	Returns to privileged EXEC mode.
Step 14	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 15	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Monitoring CoA Functionality

Table 21: Privileged EXEC show Commands

Command	Purpose
show aaa attributes protocol radius	Displays AAA attributes of RADIUS commands.

Table 22: Global Troubleshooting Commands

Command	Purpose
debug radius	Displays information for troubleshooting RADIUS.
debug aaa coa	Displays information for troubleshooting CoA processing.
debug aaa pod	Displays information for troubleshooting POD packets.
debug aaa subsys	Displays information for troubleshooting POD packets.
debug cmdhd [detail	error	events]	Displays information for troubleshooting command headers.

For detailed information about the fields in these displays, see the command reference for this release.

Feature History for RADIUS

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	RADIUS	RADIUS is a distributed client/server system that secures networks against unauthorized access. RADIUS clients run on supported Cisco devices. Clients send authentication requests to a central RADIUS server, which contains all user authentication and network service access information.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

Configuring Kerberos

Prerequisites for Controlling Switch Access with Kerberos

The following are the prerequisites for controlling switch access with Kerberos.

- So that remote users can authenticate to network services, you must configure the hosts and the KDC in the Kerberos realm to communicate and mutually authenticate users and network services. To do this, you must identify them to each other. You add entries for the hosts to the Kerberos database on the KDC and add KEYTAB files generated by the KDC to all hosts in the Kerberos realm. You also create entries for the users in the KDC database.

- A Kerberos server can be a switch that is configured as a network security server and that can authenticate users by using the Kerberos protocol.

When you add or create entries for the hosts and users, follow these guidelines:

- The Kerberos principal name must be in all lowercase characters.

- The Kerberos instance name must be in all lowercase characters.

- The Kerberos realm name must be in all uppercase characters.

Information about Kerberos

This section provides Kerberos information.

Kerberos and Switch Access

This section describes how to enable and configure the Kerberos security system, which authenticates requests for network resources by using a trusted third party.

Note In the Kerberos configuration examples, the trusted third party can be any switch that supports Kerberos, that is configured as a network security server, and that can authenticate users by using the Kerberos protocol.

Kerberos Overview

Kerberos is a secret-key network authentication protocol, which was developed at the Massachusetts Institute of Technology (MIT). It uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication and authenticates requests for network resources. Kerberos uses the concept of a trusted third party to perform secure verification of users and services. This trusted third party is called the key distribution center (KDC).

Kerberos verifies that users are who they claim to be and the network services that they use are what the services claim to be. To do this, a KDC or trusted Kerberos server issues tickets to users. These tickets, which have a limited life span, are stored in user credential caches. The Kerberos server uses the tickets instead of user names and passwords to authenticate users and network services.

Note A Kerberos server can be any switch that is configured as a network security server and that can authenticate users by using the Kerberos protocol.

The Kerberos credential scheme uses a process called single logon. This process authenticates a user once and then allows secure authentication (without encrypting another password) wherever that user credential is accepted.

This software release supports Kerberos 5, which allows organizations that are already using Kerberos 5 to use the same Kerberos authentication database on the KDC that they are already using on their other network hosts (such as UNIX servers and PCs).

Kerberos supports these network services:

- Telnet
- rlogin
- rsh

This table lists the common Kerberos-related terms and definitions.

Table 23: Kerberos Terms

Term	Definition
Authentication	A process by which a user or service identifies itself to another service. For example, a client can authenticate to a switch or a switch can authenticate to another switch.
Authorization	A means by which the switch identifies what privileges the user has in a network or on the switch and what actions the user can perform.

Term	Definition
Credential	A general term that refers to authentication tickets, such as TGTs[2] and service credentials. Kerberos credentials verify the identity of a user or service. If a network service decides to trust the Kerberos server that issued a ticket, it can be used in place of re-entering a username and password. Credentials have a default life span of eight hours.
Instance	An authorization level label for Kerberos principals. Most Kerberos principals are of the form user@REALM (for example, smith@EXAMPLE.COM). A Kerberos principal with a Kerberos instance has the form user/instance@REALM (for example, smith/admin@EXAMPLE.COM). The Kerberos instance can be used to specify the authorization level for the user if authentication is successful. The server of each network service might implement and enforce the authorization mappings of Kerberos instances but is not required to do so. **Note** The Kerberos principal and instance names must be in all lowercase characters. **Note** The Kerberos realm name must be in all uppercase characters.
KDC[3]	Key distribution center that consists of a Kerberos server and database program that is running on a network host.
Kerberized	A term that describes applications and services that have been modified to support the Kerberos credential infrastructure.
Kerberos realm	A domain consisting of users, hosts, and network services that are registered to a Kerberos server. The Kerberos server is trusted to verify the identity of a user or network service to another user or network service. **Note** The Kerberos realm name must be in all uppercase characters.
Kerberos server	A daemon that is running on a network host. Users and network services register their identity with the Kerberos server. Network services query the Kerberos server to authenticate to other network services.
KEYTAB[4]	A password that a network service shares with the KDC. In Kerberos 5 and later Kerberos versions, the network service authenticates an encrypted service credential by using the KEYTAB to decrypt it. In Kerberos versions earlier than Kerberos 5, KEYTAB is referred to as SRVTAB[5].
Principal	Also known as a Kerberos identity, this is who you are or what a service is according to the Kerberos server. **Note** The Kerberos principal name must be in all lowercase characters.
Service credential	A credential for a network service. When issued from the KDC, this credential is encrypted with the password shared by the network service and the KDC. The password is also shared with the user TGT.
SRVTAB	A password that a network service shares with the KDC. In Kerberos 5 or later Kerberos versions, SRVTAB is referred to as KEYTAB.

Term	Definition
TGT	Ticket granting ticket that is a credential that the KDC issues to authenticated users. When users receive a TGT, they can authenticate to network services within the Kerberos realm represented by the KDC.

[2] ticket granting ticket
[3] key distribution center
[4] key table
[5] server table

Kerberos Operation

A Kerberos server can be a device that is configured as a network security server and that can authenticate remote users by using the Kerberos protocol. Although you can customize Kerberos in a number of ways, remote users attempting to access network services must pass through three layers of security before they can access network services.

To authenticate to network services by using a device as a Kerberos server, remote users must follow these steps:

Authenticating to a Boundary Switch

This section describes the first layer of security through which a remote user must pass. The user must first authenticate to the boundary switch. This process then occurs:

1. The user opens an un-Kerberized Telnet connection to the boundary switch.

2. The switch prompts the user for a username and password.

3. The switch requests a TGT from the KDC for this user.

4. The KDC sends an encrypted TGT that includes the user identity to the switch.

5. The switch attempts to decrypt the TGT by using the password that the user entered.

   • If the decryption is successful, the user is authenticated to the switch.

   • If the decryption is not successful, the user repeats Step 2 either by re-entering the username and password (noting if Caps Lock or Num Lock is on or off) or by entering a different username and password.

A remote user who initiates a un-Kerberized Telnet session and authenticates to a boundary switch is inside the firewall, but the user must still authenticate directly to the KDC before getting access to the network services. The user must authenticate to the KDC because the TGT that the KDC issues is stored on the switch and cannot be used for additional authentication until the user logs on to the switch.

Obtaining a TGT from a KDC

This section describes the second layer of security through which a remote user must pass. The user must now authenticate to a KDC and obtain a TGT from the KDC to access network services.

For instructions about how to authenticate to a KDC, see the "Obtaining a TGT from a KDC" section in the "Security Server Protocols" chapter of the Cisco IOS Security Configuration Guide, Release 12.4.

Authenticating to Network Services

This section describes the third layer of security through which a remote user must pass. The user with a TGT must now authenticate to the network services in a Kerberos realm.

For instructions about how to authenticate to a network service, see the "Authenticating to Network Services" section in the "Security Server Protocols" chapter of the Cisco IOS Security Configuration Guide, Release 12.4.

How to Configure Kerberos

To set up a Kerberos-authenticated server-client system, follow these steps:

- Configure the KDC by using Kerberos commands.

- Configure the switch to use the Kerberos protocol.

Monitoring the Kerberos Configuration

To display the Kerberos configuration, use the following commands:

- show running-config
- show kerberos creds: Lists the credentials in a current user's credentials cache.
- clear kerberos creds: Destroys all credentials in a current user's credentials cache, including those forwarded.

Feature History for Kerberos

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	Kerberos	Kerberos is a secret-key network authentication protocol, which was developed at the Massachusetts Institute of Technology (MIT). It uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication and authenticates requests for network resources. Kerberos uses the concept of a trusted third party to perform secure verification of users and services.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

Configuring Secure Shell

Prerequisites for Configuring Secure Shell

The following are the prerequisites for configuring the switch for secure shell (SSH):

- For SSH to work, the switch needs an Rivest, Shamir, and Adleman (RSA) public/private key pair. This is the same with Secure Copy Protocol (SCP), which relies on SSH for its secure transport.

- Before enabling SCP, you must correctly configure SSH, authentication, and authorization on the switch.

- Because SCP relies on SSH for its secure transport, the router must have an Rivest, Shamir, and Adelman (RSA) key pair.

- SCP relies on SSH for security.

- SCP requires that authentication, authorization, and accounting (AAA) authorization be configured so the router can determine whether the user has the correct privilege level.

- A user must have appropriate authorization to use SCP.

- A user who has appropriate authorization can use SCP to copy any file in the Cisco IOS File System (IFS) to and from a switch by using the copy command. An authorized administrator can also do this from a workstation.

- The Secure Shell (SSH) server requires an IPsec (Data Encryption Standard [DES] or 3DES) encryption software image; the SSH client requires an IPsec (DES or 3DES) encryption software image.)
- Configure a hostname and host domain for your device by using the hostname and ip domain-name commands in global configuration mode.

Restrictions for Configuring Secure Shell

The following are restrictions for configuring the device for secure shell.

- The switch supports Rivest, Shamir, and Adelman (RSA) authentication.

- SSH supports only the execution-shell application.

- The SSH server and the SSH client are supported only on Data Encryption Standard (DES) (56-bit) and 3DES (168-bit) data encryption software. In DES software images, DES is the only encryption algorithm available. In 3DES software images, both DES and 3DES encryption algorithms are available.

- The device supports the Advanced Encryption Standard (AES) encryption algorithm with a 128-bit key, 192-bit key, or 256-bit key. However, symmetric cipher AES to encrypt the keys is not supported.

- When using SCP, you cannot enter the password into the copy command. You must enter the password when prompted.

- The login banner is not supported in Secure Shell Version 1. It is supported in Secure Shell Version 2.

- The -l keyword and userid :{number} {ip-address} delimiter and arguments are mandatory when configuring the alternative method of Reverse SSH for console access.

- To authenticate clients with freeradius over RADSEC, you should generate an RSA key longer than 1024 bit. Use the crypto key generate rsa general-keys exportable label label-name command to achieve this.

Information About Configuring Secure Shell

Secure Shell (SSH) is a protocol that provides a secure, remote connection to a device. SSH provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated. This software release supports SSH Version 2 (SSHv2).

SSH And Switch Access

Secure Shell (SSH) is a protocol that provides a secure, remote connection to a device. SSH provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated. This software release supports SSH Version 2 (SSHv2).

SSH functions the same in IPv6 as in IPv4. For IPv6, SSH supports IPv6 addresses and enables secure, encrypted connections with remote IPv6 nodes over an IPv6 transport.

SSH Servers, Integrated Clients, and Supported Versions

The Secure Shell (SSH) Integrated Client feature is an application that runs over the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco device to make a secure, encrypted connection to another Cisco device or to any other device running the SSH server. This connection provides functionality similar to that of an outbound Telnet connection except that the connection is encrypted. With authentication and encryption, the SSH client allows for secure communication over an unsecured network.

The SSH server and SSH integrated client are applications that run on the switch. The SSH server works with the SSH client supported in this release and with non-Cisco SSH clients. The SSH client works with publicly

and commercially available SSH servers. The SSH client supports the ciphers of Data Encryption Standard (DES), 3DES, and password authentication.

Note The SSH client functionality is available only when the SSH server is enabled.

User authentication is performed like that in the Telnet session to the device. SSH also supports the following user authentication methods:

- TACACS+

- RADIUS

- Local authentication and authorization

SSH Configuration Guidelines

Follow these guidelines when configuring the switch as an SSH server or SSH client:

- An RSA key pair generated by a SSHv1 server can be used by an SSHv2 server, and the reverse.

- If the SSH server is running on an active switch and the active switch fails, the new active switch uses the RSA key pair generated by the previous active switch.

- If you get CLI error messages after entering the crypto key generate rsa global configuration command, an RSA key pair has not been generated. Reconfigure the hostname and domain, and then enter the crypto key generate rsa command.

- When generating the RSA key pair, the message No host name specified might appear. If it does, you must configure a hostname by using the hostname global configuration command.

- When generating the RSA key pair, the message No domain specified might appear. If it does, you must configure an IP domain name by using the ip domain-name global configuration command.

- When configuring the local authentication and authorization authentication method, make sure that AAA is disabled on the console.

Secure Copy Protocol Overview

The Secure Copy Protocol (SCP) feature provides a secure and authenticated method for copying switch configurations or switch image files. SCP relies on Secure Shell (SSH), an application and a protocol that provides a secure replacement for the Berkeley r-tools.

For SSH to work, the switch needs an RSA public/private key pair. This is the same with SCP, which relies on SSH for its secure transport.

Because SSH also relies on AAA authentication, and SCP relies further on AAA authorization, correct configuration is necessary.

- Before enabling SCP, you must correctly configure SSH, authentication, and authorization on the switch.

- Because SCP relies on SSH for its secure transport, the router must have an Rivest, Shamir, and Adelman (RSA) key pair.

Note	When using SCP, you cannot enter the password into the copy command. You must enter the password when prompted.

Secure Copy Protocol

The Secure Copy Protocol (SCP) feature provides a secure and authenticated method for copying device configurations or switch image files. The behavior of SCP is similar to that of remote copy (rcp), which comes from the Berkeley r-tools suite, except that SCP relies on SSH for security. SCP also requires that authentication, authorization, and accounting (AAA) authorization be configured so the device can determine whether the user has the correct privilege level. To configure the Secure Copy feature, you should understand the SCP concepts.

How to Configure Secure Shell

Setting Up the Device to Run SSH

Follow the procedure given below to set up your Device to run SSH:

Before you begin

Configure user authentication for local or remote access. This step is required. For more information, see Related Topics below.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** `Device> `**`enable`**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# `**`configure terminal`**	Enters global configuration mode.
Step 3	hostname hostname **Example:** `Device(config)# `**`hostname your_hostname`**	Configures a hostname and IP domain name for your Device. **Note** Follow this procedure only if you are configuring the Device as an SSH server.

	Command or Action	Purpose
Step 4	ip domain-name domain_name **Example:** Device(config)# **ip domain-name your_domain**	Configures a host domain for your Device.
Step 5	crypto key generate rsa **Example:** Device(config)# **crypto key generate rsa**	Enables the SSH server for local and remote authentication on the Device and generates an RSA key pair. Generating an RSA key pair for the Device automatically enables SSH. We recommend that a minimum modulus size of 1024 bits. When you generate RSA keys, you are prompted to enter a modulus length. A longer modulus length might be more secure, but it takes longer to generate and to use. **Note** Follow this procedure only if you are configuring the Device as an SSH server.
Step 6	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 7	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 8	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring the SSH Server

Follow these steps to configure the SSH server:

Note This procedure is only required if you are configuring the Device as an SSH server.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip ssh version [2] **Example:** Device(config)# **ip ssh version 2**	(Optional) Configures the Device to run SSH Version 2. If you do not enter this command or do not specify a keyword, the SSH server selects the latest SSH version supported by the SSH client.
Step 4	ip ssh {time-out seconds | authentication-retries number} **Example:** Device(config)# **ip ssh time-out 90** OR Device(config)# **ip ssh authentication-retries 2**	Configures the SSH control parameters: • time-out seconds: Specify the time-out value in seconds; the default is 120 seconds. The range is 0 to 120 seconds. This parameter applies to the SSH negotiation phase. After the connection is established, the Device uses the default time-out values of the CLI-based sessions. By default, up to five simultaneous, encrypted SSH connections for multiple CLI-based sessions over the network are available (session 0 to session 4). After the execution shell starts, the CLI-based session time-out value returns to the default of 10 minutes. • authentication-retries number: Specify the number of times that a client can re-authenticate to the server. The default is 3; the range is 0 to 5. Repeat this step when configuring both parameters.

	Command or Action	**Purpose**
Step 5	Use one or both of the following: • line vty line_number[ending_line_number] • transport input ssh **Example:** Device(config)# **line vty 1 10** or Device(config-line)# **transport input ssh**	(Optional) Configures the virtual terminal line settings. • Enters line configuration mode to configure the virtual terminal line settings. For the line_number and ending_line_number arguments, the range is from 0 to 15. • Specifies that the Device prevents non-SSH Telnet connections, limiting the device to only SSH connections.
Step 6	end **Example:** Device(config-line)# **end**	Exits line configuration mode and returns to privileged EXEC mode.
Step 7	Use one of the following: • show ip ssh • show ssh **Example:** Device# **show ip ssh** or Device# **show ssh**	• Shows the version and configuration information for your SSH server. • Shows the status of the SSH server connections on the Device.
Step 8	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 9	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Monitoring the SSH Configuration and Status

This table displays the SSH server configuration and status.

Table 24: Commands for Displaying the SSH Server Configuration and Status

Command	Purpose
show ip ssh	Shows the version and configuration information for the SSH server.
show ssh	Shows the status of the SSH server.

Feature Information for Secure Shell

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 25: Feature Information for Secure Shell

Feature Name	Releases	Feature Information
Secure Shell	Cisco IOS XE Everest 16.6.1	SSH is a protocol that provides a secure, remote connection to a device. SSH provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated. This software release supports SSHv2.

X.509v3 Certificates for SSH Authentication

X.509v3 Certificates for SSH Authentication

The X.509v3 Certificates for SSH Authentication feature uses the X.509v3 digital certificates in server and user authentication at the secure shell (SSH) server side.

This module describes how to configure server and user certificate profiles for a digital certificate.

Prerequisites for X.509v3 Certificates for SSH Authentication

- The X.509v3 Certificates for SSH Authentication feature introduces the ip ssh server algorithm authentication command to replace the ip ssh server authenticate user command. If you use the ip ssh server authenticate user command, the following deprecation message is displayed.

```
Warning: SSH command accepted but this CLI will be deprecated soon.
Please move to new CLI "ip ssh server algorithm authentication".
Please configure "default ip ssh server authenticate user" to make the CLI ineffective.
```

  - Use the default ip ssh server authenticate user command to remove the ip ssh server authenticate user command from effect. The IOS secure shell (SSH) server then starts using the ip ssh server algorithm authentication command.

Restrictions for X.509v3 Certificates for SSH Authentication

- The X.509v3 Certificates for SSH Authentication feature implementation is applicable only on the Cisco IOS XE secure shell (SSH) server side.

- The SSH server supports only the x509v3-ssh-rsa algorithm-based certificate for server and user authentication.

Information About X.509v3 Certificates for SSH Authentication

The following section provides information about digital certificates, and server and user authentication.

Digital Certificates

The validity of the authentication depends upon the strength of the linkage between the public signing key and the identity of the signer. Digital certificates in the X.509v3 format (RFC5280) are used to provide identity management. A chain of signatures by a trusted root certification authority and its intermediate certificate authorities binds a given public signing key to a given digital identity.

Public key infrastructure (PKI) trustpoint helps manage the digital certificates. The association between the certificate and the trustpoint helps track the certificate. The trustpoint contains information about the certificate authority (CA), different identity parameters, and the digital certificate. Multiple trustpoints can be created to associate with different certificates.

Server and User Authentication using X.509v3

For server authentication, the Cisco IOS XE secure shell (SSH) server sends its own certificate to the SSH client for verification. This server certificate is associated with the trustpoint configured in the server certificate profile (ssh-server-cert-profile-server configuration mode).

For user authentication, the SSH client sends the user's certificate to the SSH server for verification. The SSH server validates the incoming user certificate using public key infrastructure (PKI) trustpoints configured in the server certificate profile (ssh-server-cert-profile-user configuration mode).

By default, certificate-based authentication is enabled for server and user at the SSH server end.

How to Configure X.509v3 Certificates for SSH Authentication

The following section provides information about how to configure X.509v3 Certificates for SSH Authentication.

Configuring the SSH Server to Use Digital Certificates for Server Authentication

To configure the SSH server to use digital certificates for server authentication, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device> `**`enable`**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:**	Enters global configuration mode.

	Command or Action	**Purpose**
	Device# **configure terminal**	

Step 3	ip ssh server algorithm hostkey {x509v3-ssh-rsa [ssh-rsa] | ssh-rsa [x509v3-ssh-rsa]}
	
	Example:
	
	Device(config)# **ip ssh server algorithm hostkey x509v3-ssh-rsa**
		Defines the order of host key algorithms. Only the configured algorithm is negotiated with the secure shell (SSH) client.
		
		Note The IOS SSH server must have at least one configured host key algorithm:
		
		        • ssh-rsa – public key based authentication
		
		        • x509v3-ssh-rsa – certificate-based authentication

Step 4	ip ssh server certificate profile **Example:** Device(config)# **ip ssh server certificate profile**	Configures server certificate profile and user certificate profile and enters SSH certificate profile configuration mode.
Step 5	server **Example:** Device(ssh-server-cert-profile)# **server**	Configures server certificate profile and enters SSH server certificate profile server configuration mode.
Step 6	trustpoint sign PKI-trustpoint-name **Example:** Device(ssh-server-cert-profile-server)# **trustpoint sign trust1**	Attaches the public key infrastructure (PKI) trustpoint to the server certificate profile. The SSH server uses the certificate associated with this PKI trustpoint for server authentication.
Step 7	ocsp-response include **Example:** Device(ssh-server-cert-profile-server)# **ocsp-response include**	(Optional) Sends the Online Certificate Status Protocol (OCSP) response or OCSP stapling along with the server certificate. **Note** By default the no form of this command is configured and no OCSP response is sent along with the server certificate.
Step 8	end **Example:** Device(ssh-server-cert-profile-server)# **end**	Exits SSH server certificate profile server configuration mode and returns to privileged EXEC mode.

Configuring the SSH Server to Verify Digital Certificates for User Authentication

To configure the SSH Server to use digital certificates for user authentication, perform this procedure:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip ssh server algorithm authentication {publickey | keyboard | password} **Example:** Device(config)# **ip ssh server algorithm authentication publickey**	Defines the order of user authentication algorithms. Only the configured algorithm is negotiated with the secure shell (SSH) client. **Note** The SSH server must have at least one configured user authentication algorithm. **Note** To use the certificate method for user authentication, the publickey keyword must be configured. **Note** The ip ssh server algorithm authentication command replaces the ip ssh server authenticate user command.
Step 4	ip ssh server algorithm publickey {x509v3-ssh-rsa [ssh-rsa] | ssh-rsa [x509v3-ssh-rsa]} **Example:** Device(config)# **ip ssh server algorithm publickey x509v3-ssh-rsa**	Defines the order of public key algorithms. Only the configured algorithm is accepted by the SSH client for user authentication. **Note** The SSH client must have at least one configured public key algorithm: • ssh-rsa – public-key-based authentication • x509v3-ssh-rsa – certificate-based authentication
Step 5	ip ssh server certificate profile **Example:** Device(config)# **ip ssh server certificate profile**	Configures server certificate profile and user certificate profile and enters SSH certificate profile configuration mode.
Step 6	user **Example:** Device(ssh-server-cert-profile)# **user**	Configures user certificate profile and enters SSH server certificate profile user configuration mode.

	Command or Action	Purpose
Step 7	trustpoint verify PKI-trustpoint-name **Example:** Device(ssh-server-cert-profile-user)# **trustpoint verify trust2**	Configures the public key infrastructure (PKI) trustpoint that is used to verify the incoming user certificate. **Note** Configure multiple trustpoints by executing the same command multiple times. A maximum of 10 trustpoints can be configured.
Step 8	ocsp-response required **Example:** Device(ssh-server-cert-profile-user)# **ocsp-response required**	(Optional) Mandates the presence of the Online Certificate Status Protocol (OCSP) response with the incoming user certificate. **Note** By default the no form of this command is configured and the user certificate is accepted without an OCSP response.
Step 9	end **Example:** Device(ssh-server-cert-profile-user)# **end**	Exits SSH server certificate profile user configuration mode and returns to privileged EXEC mode.

Verifying Configuration for Server and User Authentication Using Digital Certificates

To verify configuration for server and user Authentication using digital certificates, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	show ip ssh **Example:** Device# **show ip ssh** SSH Enabled - version 1.99 Authentication methods:publickey,keyboard-interactive,password Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa Hostkey Algorithms:x509v3-ssh-rsa,ssh-rsa Authentication timeout: 120 secs; Authentication retries: 3 Minimum expected Diffie Hellman key size	Displays the currently configured authentication methods. To confirm the use of certificate-based authentication, ensure that the x509v3-ssh-rsa algorithm is the configured host key algorithm.

Command or Action	Purpose
: 1024 bits	

Configuration Examples for X.509v3 Certificates for SSH Authentication

The following section provides examples for user and server authentication using digital certificates.

Example: Configuring the SSH Server to Use Digital Certificates for Server Authentication

This example shows how to configure the SSH Server to use digital certificates for server authentication.

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa
Device(config)# ip ssh server certificate profile
Device(ssh-server-cert-profile)# server
Device(ssh-server-cert-profile-server)# trustpoint sign trust1
Device(ssh-server-cert-profile-server)# end
```

Example: Configuring the SSH Server to Verify Digital Certificates for User Authentication

This example shows how to configure the SSH server to verify user's digital certificate for user authentication.

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm authentication publickey
Device(config)# ip ssh server algorithm publickey x509v3-ssh-rsa
Device(config)# ip ssh server certificate profile
Device(ssh-server-cert-profile)# user
Device(ssh-server-cert-profile-user)# trustpoint verify trust2
Device(ssh-server-cert-profile-user)# end
```

Feature History for X.509v3 Certificates for SSH Authentication

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	X.509v3 Certificates for SSH Authentication	The X.509v3 Certificates for SSH Authentication feature uses the X.509v3 digital certificates in server and user authentication at the SSH server side.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

CHAPTER 12

SSH Algorithms for Common Criteria Certification

Information About SSH Algorithms for Common Criteria Certification

This section provides information about the Secure Shell (SSH) Algorithms for Common Criteria Certification, the Cisco IOS SSH Server Algorithms and Cisco IOS SSH Client Algorithms.

SSH Algorithms for Common Criteria Certification

A Secure Shell (SSH) configuration enables a Cisco IOS SSH server and client to authorize the negotiation of only those algorithms that are configured from the allowed list. If a remote party tries to negotiate using only those algorithms that are not part of the allowed list, the request is rejected and the session is not established.

Cisco IOS SSH Server Algorithms

Cisco IOS secure shell (SSH) servers support the encryption algorithms (Advanced Encryption Standard Counter Mode [AES-CTR], AES Cipher Block Chaining [AES-CBC], Triple Data Encryption Standard [3DES]) in the following order:

Supported Default Encryption Order:

1. aes128-ctr

2. aes192-ctr

3. aes256-ctr

Supported Non-Default Encrytion Order:

1. aes128-cbc

2. aes192-cbc

3. aes256-cbc

4. 3des

Cisco IOS SSH clients support the Message Authentication Code (MAC) algorithms in the following order:

Supported Default HMAC order:

1. hmac-sha2-256

2. hmac-sha2-512

3. hmac-sha1

4. hmac-sha1-96

Cisco IOS SSH clients support only one host key algorithm and do not need a CLI configuration.

Supported Default Host Key order:

1. x509v3-ssh-rsa

2. ssh-rsa

Cisco IOS SSH Client Algorithms

Cisco IOS secure shell (SSH) clients support the encryption algorithms (Advanced Encryption Standard counter mode [AES-CTR], AES Cipher Block Chaining [AES-CBC], Triple Data Encryption Standard [3DES]) in the following order:

Supported Default Encryption Order:

1. aes128-ctr

2. aes192-ctr

3. aes256-ctr

Supported Non-Default Encrytion Order:

1. aes128-cbc

2. aes192-cbc

3. aes256-cbc

4. 3des

Cisco IOS SSH clients support the Message Authentication Code (MAC) algorithms in the following order:

Supported Default HMAC order:

1. hmac-sha2-256

2. hmac-sha2-512

3. hmac-sha1

4. hmac-sha1-96

Cisco IOS SSH clients support only one host key algorithm and do not need a CLI configuration.

Supported Default Host Key order:

1. x509v3-ssh-rsa

2. ssh-rsa

How to Configure SSH Algorithms for Common Criteria Certification

This section provides information on how to configure and troubleshoot:

- Encryption key algorithm for a Cisco IOS SSH server and client

- MAC algorithm for a Cisco IOS SSH server and client

- Host Key algorithm for a Cisco IOS SSH server

Configuring an Encryption Key Algorithm for a Cisco IOS SSH Server and Client

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	ip ssh {server	client} algorithm encryption {aes128-ctr	aes192-ctr	aes256-ctr	aes128-cbc	aes192-cbc	aes256-cbc	3des-cbc } **Example:** `Device(config)# ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr aes128-cbc 3des-cbc aes192-cbc aes256-cbc`	Defines the order of encryption algorithms in the SSH server and client. This order is presented during algorithm negotiation. **Note** The Cisco IOS SSH server and client must have at least one configured encryption algorithm.

Command or Action	Purpose	
Device(config)# **ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr aes128-cbc 3des-cbc aes192-cbc aes256-cbc**	Note	To disable one algorithm from the previously configured algorithm list, use the no form of this command. To disable more than one algorithm, use the no form of this command multiple times with different algorithm names.
	Note	For a default configuration, use the default form of this command as shown below: Device(config)# **ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr aes128-cbc 3des-cbc aes192-cbc aes256-cbc**
Step 4	end **Example:** Device(config)# **end**	Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

If you try to disable the last encryption algorithm in the configuration, the following message is displayed and the command is rejected:

```
% SSH command rejected: All encryption algorithms cannot be disabled
```

Configuring a MAC Algorithm for a Cisco IOS SSH Server and Client

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose

Step 3	ip ssh {server	client} algorithm mac {hmac-sha2-256	hmac-sha2-512	hmac-sha1	hmac-sha1-96}
					
					Example:
					
					`Device(config)# ip ssh server algorithm mac hmac-sha2-256 hmac-sha2-512 hmac-sha1 hmac-sha1-96`
					
					`Device(config)# ip ssh client algorithm mac hmac-sha2-256 hmac-sha2-512 hmac-sha1 hmac-sha1-96`
						Defines the order of MAC (Message Authentication Code) algorithms in the SSH server and client. This order is presented during algorithm negotiation.
						
						Note The Cisco IOS SSH server and client must have at least one configured Hashed Message Authentication Code (HMAC) algorithm.
						
						Note To disable one algorithm from the previously configured algorithm list, use the no form of this command. To disable more than one algorithm, use the no form of this command multiple times with different algorithm names.
						
						Note For default configuration, use the default form of this command as shown below:
						
						`Device(config)# ip ssh server algorithm mac hmac-sha2-256 hmac-sha2-512 hmac-sha1 hmac-sha1-96`

Step 4	end **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

If you try to disable the last MAC algorithm in the configuration, the following message is displayed and the command is rejected:

```
% SSH command rejected: All mac algorithms cannot be disabled
```

Configuring a Host Key Algorithm for a Cisco IOS SSH Server

Procedure

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Device> **enable**	Enter your password if prompted.
Step 2	configure terminal Example: Device# **configure terminal**	Enters global configuration mode.

Step 3	ip ssh server algorithm hostkey {x509v3-ssh-rsa | ssh-rsa}
	Example:
	
	Device(config)# **ip ssh server algorithm hostkey x509v3-ssh-rsa ssh-rsa**
		Defines the order of host key algorithms. Only the configured algorithm is negotiated with the Cisco IOS secure shell (SSH) client.
		
		Note The Cisco IOS SSH server must have at least one configured host key algorithm:
		
		    • x509v3-ssh-rsa—X.509v3 certificate-based authentication
		
		    • ssh-rsa—Public-key-based authentication
		
		Note To disable one algorithm from the previously configured algorithm list, use the no form of this command. To disable more than one algorithm, use the no form of this command multiple times with different algorithm names.
		
		Note For default configuration, use the default form of this command as shown below:
		
		Device(config)# **ip ssh server algorithm hostkey x509v3-ssh-rsa ssh-rsa**

Step 4	end Example: Device(config)# **end**	Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

If you try to disable the last host key algorithm in the configuration, the following message is displayed and the command is rejected:

```
% SSH command rejected: All hostkey algorithms cannot be disabled
```

Configuration Examples For SSH Algorithms for Common Criteria Certification

This section provides configuration examples for SSH algorithms for common certification.

Example: Configuring Encryption Key Algorithms for a Cisco IOS SSH Server

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr aes128-cbc
 aes192-cbc aes256-cbc 3des
Device(config)# end
```

Example: Configuring Encryption Key Algorithms for a Cisco IOS SSH Client

```
Device> enable
Device# configure terminal
Device(config)# ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr aes128-cbc
 aes192-cbc aes256-cbc 3des

Device(config)# end
```

Example: Configuring MAC Algorithms for a Cisco IOS SSH Server

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm mac hmac-sha2-256, hmac-sha2-512, hmac-sha1,
hmac-sha1-96
Device(config)# end
```

Example: Configuring Host Key Algorithms for a Cisco IOS SSH Server

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa ssh-rsa
Device(config)# end
```

Verifying SSH Algorithms for Common Criteria Certification

Procedure

Step 1   enable

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2   show ip ssh

Displays configured Secure Shell (SSH) encryption, host key, and Message Authentication Code (MAC) algorithms.

Example:

The following sample output from the show ip ssh command shows the encryption algorithms configured in the default order:

```
Device# show ip ssh

Encryption Algorithms: aes128-ctr, aes192-ctr, aes256-ctr, aes128-cbc, aes192-cbc, aes256-cbc,
 3des
```

The following sample output from the show ip ssh command shows the MAC algorithms configured in the default order:

```
Device# show ip ssh

MAC Algorithms: hmac-sha2-256,hmac-sha2-512, hmac-sha1, hmac-sha1-96
```

The following sample output from the show ip ssh command shows the host key algorithms configured in the default order:

```
Device# show ip ssh

Hostkey Algorithms: x509v3-ssh-rsa,ssh-rsa
```

Feature Information for Secure Shell Algorithms for Common Criteria Certification

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	Secure Shell Algorithms for Common Criteria Certification	The SSH Algorithms for Common Criteria Certification feature provides the list and order of the algorithms that are allowed for Common Criteria Certification. This module describes how to configure the encryption, Message Authentication Code (MAC), and host key algorithms for a secure shell (SSH) server and client so that SSH connections can be limited on the basis of the allowed algorithms list.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

CHAPTER 13

Configuring Secure Socket Layer HTTP

Information about Secure Socket Layer HTTP

Secure HTTP Servers and Clients Overview

On a secure HTTP connection, data to and from an HTTP server is encrypted before being sent over the Internet. HTTP with SSL encryption provides a secure connection to allow such functions as configuring a switch from a Web browser. Cisco's implementation of the secure HTTP server and secure HTTP client uses an implementation of SSL Version 3.0 with application-layer encryption. HTTP over SSL is abbreviated as HTTPS; the URL of a secure connection begins with https:// instead of http://.

Note SSL evolved into Transport Layer Security (TLS) in 1999, but is still used in this particular context.

The primary role of the HTTP secure server (the switch) is to listen for HTTPS requests on a designated port (the default HTTPS port is 443) and pass the request to the HTTP 1.1 Web server. The HTTP 1.1 server processes requests and passes responses (pages) back to the HTTP secure server, which, in turn, responds to the original request.

The primary role of the HTTP secure client (the web browser) is to respond to Cisco IOS application requests for HTTPS User Agent services, perform HTTPS User Agent services for the application, and pass the response back to the application.

Certificate Authority Trustpoints

Certificate authorities (CAs) manage certificate requests and issue certificates to participating network devices. These services provide centralized security key and certificate management for the participating devices. Specific CA servers are referred to as trustpoints.

When a connection attempt is made, the HTTPS server provides a secure connection by issuing a certified X.509v3 certificate, obtained from a specified CA trustpoint, to the client. The client (usually a Web browser), in turn, has a public key that allows it to authenticate the certificate.

For secure HTTP connections, we highly recommend that you configure a CA trustpoint. If a CA trustpoint is not configured for the device running the HTTPS server, the server certifies itself and generates the needed RSA key pair. Because a self-certified (self-signed) certificate does not provide adequate security, the connecting client generates a notification that the certificate is self-certified, and the user has the opportunity to accept or reject the connection. This option is useful for internal network topologies (such as testing).

If you do not configure a CA trustpoint, when you enable a secure HTTP connection, either a temporary or a persistent self-signed certificate for the secure HTTP server (or client) is automatically generated.

  • If the switch is not configured with a hostname and a domain name, a temporary self-signed certificate is generated. If the switch reboots, any temporary self-signed certificate is lost, and a new temporary new self-signed certificate is assigned.

  • If the switch has been configured with a host and domain name, a persistent self-signed certificate is generated. This certificate remains active if you reboot the switch or if you disable the secure HTTP server so that it will be there the next time you re-enable a secure HTTP connection.

Note   The certificate authorities and trustpoints must be configured on each device individually. Copying them from other devices makes them invalid on the switch.

When a new certificate is enrolled, the new configuration change is not applied to the HTTPS server until the server is restarted. You can restart the server using either the CLI or by physical reboot. On restarting the server, the switch starts using the new certificate.

If a self-signed certificate has been generated, this information is included in the output of the show running-config privileged EXEC command. This is a partial sample output from that command displaying a self-signed certificate.

```
Device# show running-config
Building configuration...

<output truncated>

crypto pki trustpoint TP-self-signed-3080755072
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-3080755072
 revocation-check none
 rsakeypair TP-self-signed-3080755072
!
!
crypto ca certificate chain TP-self-signed-3080755072
 certificate self-signed 01
  3082029F 30820208 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  59312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 33303830 37353530 37323126 30240609 2A864886 F70D0109
  02161743 45322D33 3535302D 31332E73 756D6D30 342D3335 3530301E 170D3933
  30333031 30303030 35395A17 0D323030 31303130 30303030 305A3059 312F302D

<output truncated>
```

You can remove this self-signed certificate by disabling the secure HTTP server and entering the no crypto pki trustpoint TP-self-signed-30890755072 global configuration command. If you later re-enable a secure HTTP server, a new self-signed certificate is generated.

Note The values that follow TP self-signed depend on the serial number of the device.

You can use an optional command (ip http secure-client-auth) to allow the HTTPS server to request an X.509v3 certificate from the client. Authenticating the client provides more security than server authentication by itself.

CipherSuites

A CipherSuite specifies the encryption algorithm and the digest algorithm to use on a SSL connection. When connecting to the HTTPS server, the client Web browser offers a list of supported CipherSuites, and the client and server negotiate the best encryption algorithm to use from those on the list that are supported by both. For example, Netscape Communicator 4.76 supports U.S. security with RSA Public Key Cryptography, MD2, MD5, RC2-CBC, RC4, DES-CBC, and DES-EDE3-CBC.

For the best possible encryption, you should use a client browser that supports 128-bit encryption, such as Microsoft Internet Explorer Version 5.5 (or later) or Netscape Communicator Version 4.76 (or later). The SSL_RSA_WITH_DES_CBC_SHA CipherSuite provides less security than the other CipherSuites, as it does not offer 128-bit encryption.

The more secure and more complex CipherSuites require slightly more processing time. This list defines the CipherSuites supported by the switch and ranks them from fastest to slowest in terms of router processing load (speed):

1. SSL_RSA_WITH_DES_CBC_SHA—RSA key exchange (RSA Public Key Cryptography) with DES-CBC for message encryption and SHA for message digest

2. SSL_RSA_WITH_NULL_SHA key exchange with NULL for message encryption and SHA for message digest (only for SSL 3.0).

3. SSL_RSA_WITH_NULL_MD5 key exchange with NULL for message encryption and MD5 for message digest (only for SSL 3.0).

4. SSL_RSA_WITH_RC4_128_MD5—RSA key exchange with RC4 128-bit encryption and MD5 for message digest

5. SSL_RSA_WITH_RC4_128_SHA—RSA key exchange with RC4 128-bit encryption and SHA for message digest

6. SSL_RSA_WITH_3DES_EDE_CBC_SHA—RSA key exchange with 3DES and DES-EDE3-CBC for message encryption and SHA for message digest

7. SSL_RSA_WITH_AES_128_CBC_SHA—RSA key exchange with AES 128-bit encryption and SHA for message digest (only for SSL 3.0).

8. SSL_RSA_WITH_AES_256_CBC_SHA—RSA key exchange with AES 256-bit encryption and SHA for message digest (only for SSL 3.0).

9. SSL_RSA_WITH_DHE_AES_128_CBC_SHA—RSA key exchange with AES 128-bit encryption and SHA for message digest (only for SSL 3.0).

10. SSL_RSA_WITH_DHE_AES_256_CBC_SHA—RSA key exchange with AES 256-bit encryption and SHA for message digest (only for SSL 3.0).

Note The latest versions of Chrome do not support the four original cipher suites, thus disallowing access to both web GUI and guest portals.

RSA (in conjunction with the specified encryption and digest algorithm combinations) is used for both key generation and authentication on SSL connections. This usage is independent of whether or not a CA trustpoint is configured.

Default SSL Configuration

The standard HTTP server is enabled.

SSL is enabled.

No CA trustpoints are configured.

No self-signed certificates are generated.

SSL Configuration Guidelines

When SSL is used in a switch cluster, the SSL session terminates at the cluster commander. Cluster member switches must run standard HTTP.

Before you configure a CA trustpoint, you should ensure that the system clock is set. If the clock is not set, the certificate is rejected due to an incorrect date.

In a switch stack, the SSL session terminates at the active switch.

How to Configure Secure Socket Layer HTTP

Configuring a CA Trustpoint

For secure HTTP connections, we recommend that you configure an official CA trustpoint. A CA trustpoint is more secure than a self-signed certificate.

Beginning in privileged EXEC mode, follow these steps to configure a CA Trustpoint:

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.

	Command or Action	**Purpose**
Step 2	hostname hostname	Specifies the hostname of the switch (required only if you have not previously configured a hostname). The hostname is required for security keys and certificates.
	Example:	
	Device(config)# **hostname your_hostname**	
Step 3	ip domain-name domain-name	Specifies the IP domain name of the switch (required only if you have not previously configured an IP domain name). The domain name is required for security keys and certificates.
	Example:	
	Device(config)# **ip domain-name your_domain**	
Step 4	crypto key generate rsa	(Optional) Generates an RSA key pair. RSA key pairs are required before you can obtain a certificate for the switch. RSA key pairs are generated automatically. You can use this command to regenerate the keys, if needed.
	Example:	
	Device(config)# **crypto key generate rsa**	
Step 5	crypto ca trustpoint name	Specifies a local configuration name for the CA trustpoint and enter CA trustpoint configuration mode.
	Example:	
	Device(config)# **crypto ca trustpoint your_trustpoint**	
Step 6	enrollment url url	Specifies the URL to which the switch should send certificate requests.
	Example:	
	Device(ca-trustpoint)# **enrollment url http://your_server:80**	
Step 7	enrollment http-proxy host-name port-number	(Optional) Configures the switch to obtain certificates from the CA through an HTTP proxy server.
	Example:	• For host-name , specify the proxy server used to get the CA.
	Device(ca-trustpoint)# **enrollment http-proxy your_host 49**	• For port-number, specify the port number used to access the CA.
Step 8	crl query url	Configures the switch to request a certificate revocation list (CRL) to ensure that the certificate of the peer has not been revoked.
	Example:	
	Device(ca-trustpoint)# **crl query ldap://your_host:49**	

	Command or Action	Purpose
Step 9	primary name **Example:** Device(ca-trustpoint)# **primary your_trustpoint**	(Optional) Specifies that the trustpoint should be used as the primary (default) trustpoint for CA requests. • For name, specify the trustpoint that you just configured.
Step 10	exit **Example:** Device(ca-trustpoint)# **exit**	Exits CA trustpoint configuration mode and return to global configuration mode.
Step 11	crypto ca authentication name **Example:** Device(config)# **crypto ca authentication your_trustpoint**	Authenticates the CA by getting the public key of the CA. Use the same name used in Step 5.
Step 12	crypto ca enroll name **Example:** Device(config)# **crypto ca enroll your_trustpoint**	Obtains the certificate from the specified CA trustpoint. This command requests a signed certificate for each RSA key pair.
Step 13	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

Configuring the Secure HTTP Server

Beginning in privileged EXEC mode, follow these steps to configure a secure HTTP server:

Before you begin

If you are using a certificate authority for certification, you should use the previous procedure to configure the CA trustpoint on the switch before enabling the HTTP server. If you have not configured a CA trustpoint, a self-signed certificate is generated the first time that you enable the secure HTTP server. After you have configured the server, you can configure options (path, access list to apply, maximum number of connections, or timeout policy) that apply to both standard and secure HTTP servers.

To verify the secure HTTP connection by using a Web browser, enter https://URL, where the URL is the IP address or hostname of the server switch. If you configure a port other than the default port, you must also specify the port number after the URL. For example:

Note   AES256_SHA2 is not supported.

```
https://209.165.129:1026
```

or

```
https://host.domain.com:1026
```

The existing ip http access-class access-list-number command for specifying the access-list(Only IPv4 ACLs) is going to be deprecated. You can still use this command to specify an access list to allow access to the HTTP server. Two new commands have been introduced to enable support for specifying IPv4 and IPv6 ACLs. These are ip http access-class ipv4 access-list-name | access-list-number for specifying IPv4 ACLs and ip http access-class ipv6 access-list-name for specifying IPv6 ACLs. We recommend using the new CLI to avoid receiving warning messages.

Note the following considerations for specifying access-lists:

- If you specify an access-list that does not exist, the configuration takes place but you receive the below warning message:

  ```
  ACL being attached does not exist, please configure it
  ```

- If you use the ip http access-class command for specifying an access-list for the HTTP server, the below warning message appears:

  ```
  This CLI will be deprecated soon, Please use new CLI ip http
  access-class ipv4/ipv6 <access-list-name>| <access-list-number>
  ```

- If you use ip http access-class ipv4 access-list-name | access-list-number or ip http access-class ipv6 access-list-name , and an access-list was already configured using ip http access-class , the below warning message appears:

  ```
  Removing ip http access-class <access-list-number>
  ```

ip http access-class access-list-number and ip http access-class ipv4 access-list-name | access-list-number share the same functionality. Each command overrides the configuration of the previous command. The following combinations between the configuration of the two commands explain the effect on the running configuration:

- If ip http access-class access-list-number is already configured and you try to configure using ip http access-class ipv4 access-list-number command, the configuration of ip http access-class access-list-number will be removed and the configuration of ip http access-class ipv4 access-list-number will be added to the running configuration.

- If ip http access-class access-list-number is already configured and you try to configure using ip http access-class ipv4 access-list-name command, the configuration of ip http access-class access-list-number will be removed and the configuration of ip http access-class ipv4 access-list-name will be added to the running configuration.

- If ip http access-class ipv4 access-list-number is already configured and you try to configure using ip http access-class access-list-name, the configuration of ip http access-class ipv4 access-list-number will be removed from configuration and the configuration of ip http access-class access-list-name will be added to the running configuration.

- If ip http access-class ipv4 access-list-name is already configured and you try to configure using ip http access-class access-list-number, the configuration of ip http access-class ipv4 access-list-name will be removed from the configuration and the configuration of ip http access-class access-list-number will be added to the running configuration.

Procedure

	Command or Action	**Purpose**
Step 1	show ip http server status **Example:** Device# **show ip http server status**	(Optional) Displays the status of the HTTP server to determine if the secure HTTP server feature is supported in the software. You should see one of these lines in the output: HTTP secure server capability: Present or HTTP secure server capability: Not present
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip http secure-server **Example:** Device(config)# **ip http secure-server**	Enables the HTTPS server if it has been disabled. The HTTPS server is enabled by default.
Step 4	ip http secure-port port-number **Example:** Device(config)# **ip http secure-port 443**	(Optional) Specifies the port number to be used for the HTTPS server. The default port number is 443. Valid options are 443 or any number in the range 1025 to 65535.
Step 5	ip http secure-ciphersuite {[3des-ede-cbc-sha] [rc4-128-md5] [rc4-128-sha] [des-cbc-sha]} **Example:** Device(config)# **ip http secure-ciphersuite rc4-128-md5**	(Optional) Specifies the CipherSuites (encryption algorithms) to be used for encryption over the HTTPS connection. If you do not have a reason to specify a particularly CipherSuite, you should allow the server and client to negotiate a CipherSuite that they both support. This is the default.
Step 6	ip http secure-client-auth **Example:**	(Optional) Configures the HTTP server to request an X.509v3 certificate from the client

	Command or Action	**Purpose**
	Device(config)# **ip http secure-client-auth**	for authentication during the connection process. The default is for the client to request a certificate from the server, but the server does not attempt to authenticate the client.
Step 7	ip http secure-trustpoint name **Example:** Device(config)# **ip http secure-trustpoint your_trustpoint**	Specifies the CA trustpoint to use to get an X.509v3 security certificate and to authenticate the client certificate connection. **Note** Use of this command assumes you have already configured a CA trustpoint according to the previous procedure.
Step 8	ip http path path-name **Example:** Device(config)# **ip http path /your_server:80**	(Optional) Sets a base HTTP path for HTML files. The path specifies the location of the HTTP server files on the local system (usually located in system flash memory).
Step 9	ip http access-class access-list-number **Example:** Device(config)# **ip http access-class 2**	(Optional) Specifies an access list to use to allow access to the HTTP server.
Step 10	ip http access-class { ipv4 {access-list-number | access-list-name} | ipv6 {access-list-name} } **Example:** Device(config)# **ip http access-class ipv4 4**	(Optional)Specifies an access list to use to allow access to the HTTP server.
Step 11	ip http max-connections value **Example:** Device(config)# **ip http max-connections 4**	(Optional) Sets the maximum number of concurrent connections that are allowed to the HTTP server. We recommend that the value be at least 10 and not less. This is required for the UI to function as expected.
Step 12	ip http timeout-policy idle seconds life seconds requests value **Example:** Device(config)# **ip http timeout-policy idle 120 life 240 requests 1**	(Optional) Specifies how long a connection to the HTTP server can remain open under the defined circumstances: • idle—the maximum time period when no data is received or response data cannot be sent. The range is 1 to 600 seconds. The default is 180 seconds (3 minutes).

	Command or Action	Purpose
		• life—the maximum time period from the time that the connection is established. The range is 1 to 86400 seconds (24 hours). The default is 180 seconds. • requests—the maximum number of requests processed on a persistent connection. The maximum value is 86400. The default is 1.
Step 13	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

Configuring the Secure HTTP Client

Beginning in privileged EXEC mode, follow these steps to configure a secure HTTP client:

Before you begin

The standard HTTP client and secure HTTP client are always enabled. A certificate authority is required for secure HTTP client certification. This procedure assumes that you have previously configured a CA trustpoint on the switch. If a CA trustpoint is not configured and the remote HTTPS server requires client authentication, connections to the secure HTTP client fail.

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	ip http client secure-trustpoint name **Example:** Device(config)# **ip http client secure-trustpoint your_trustpoint**	(Optional) Specifies the CA trustpoint to be used if the remote HTTP server requests client authentication. Using this command assumes that you have already configured a CA trustpoint by using the previous procedure. The command is optional if client authentication is not needed or if a primary trustpoint has been configured.
Step 3	ip http client secure-ciphersuite {[3des-ede-cbc-sha] [rc4-128-md5] [rc4-128-sha] [des-cbc-sha]}	(Optional) Specifies the CipherSuites (encryption algorithms) to be used for encryption over the HTTPS connection. If you

	Command or Action	Purpose

	Example:
	
	`Device(config)# ip http client`
	`secure-ciphersuite rc4-128-md5`
		do not have a reason to specify a particular CipherSuite, you should allow the server and client to negotiate a CipherSuite that they both support. This is the default.

Step 4	end **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Monitoring Secure HTTP Server and Client Status

To monitor the SSL secure server and client status, use the privileged EXEC commands in the following table.

Table 26: Commands for Displaying the SSL Secure Server and Client Status

Command	Purpose
show ip http client secure status	Shows the HTTP secure client configuration.
show ip http server secure status	Shows the HTTP secure server configuration.
show running-config	Shows the generated self-signed certificate for secure HTTP connections.

Additional References for Secure Socket Layer HTTP

Related Documents

Related Topic	Document Title
Certification Authority	Configuring Certification Authority Interoperability

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History for Secure Socket Layer HTTP

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	Secure Socket Layer HTTP	Cisco's implementation of the secure HTTP server and secure HTTP client uses an implementation of SSL Version 3.0 with application-layer encryption. On a secure HTTP connection, data to and from an HTTP server is encrypted before being sent over the Internet.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.
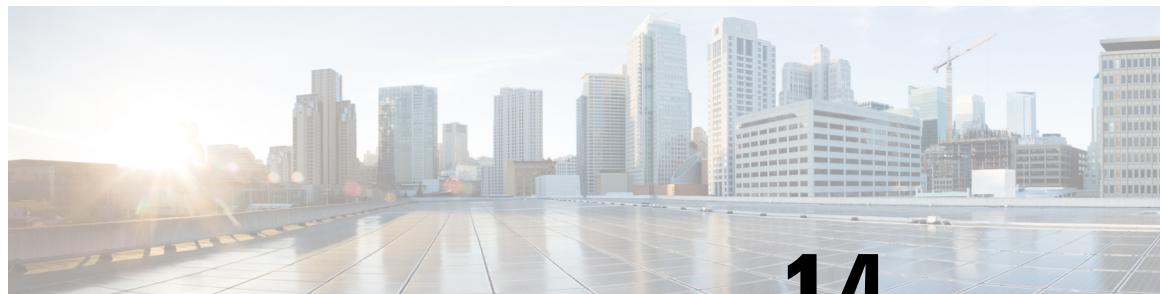
IPv4 ACLs

Restrictions for Configuring IPv4 Access Control Lists

General Network Security

The following are restrictions for configuring network security with ACLs:

- Not all commands that accept a numbered ACL accept a named ACL. ACLs for packet filters and route filters on interfaces can use a name. VLAN maps also accept a name.

- A standard ACL and an extended ACL cannot have the same name.

- Though visible in the command-line help strings, appletalk is not supported as a matching condition for the deny and permit MAC access-list configuration mode commands.

- ACL wildcard is not supported in downstream client policy.

- If a downloadable ACL contains any type of duplicate entries, the entries are not auto merged. As a result, the 802.1X session authorization fails. Ensure that the downloadable ACL is optimized without any duplicate entries, for example port-based and name-based entries for the same port.

IPv4 ACL Network Interfaces

The following restrictions apply to IPv4 ACLs to network interfaces:

- When controlling access to an interface, you can use a named or numbered ACL.

- If you apply an ACL to a Layer 2 interface that is a member of a VLAN, the Layer 2 (port) ACL takes precedence over an input Layer 3 ACL applied to the VLAN interface or a VLAN map applied to the VLAN.

- If you apply an ACL to a Layer 3 interface and routing is not enabled on the switch, the ACL only filters packets that are intended for the CPU, such as SNMP, Telnet, or web traffic.

- If the preauth_ipv4_acl ACL is configured to filter packets, the ACL is removed after authentication.

- You do not have to enable routing to apply ACLs to Layer 2 interfaces.

MAC ACLs on a Layer 2 Interface

After you create a MAC ACL, you can apply it to a Layer 2 interface to filter non-IP traffic coming in that interface. When you apply the MAC ACL, consider these guidelines:

- You can apply no more than one IP access list and one MAC access list to the same Layer 2 interface. The IP access list filters only IP packets, and the MAC access list filters non-IP packets.

- A Layer 2 interface can have only one MAC access list. If you apply a MAC access list to a Layer 2 interface that has a MAC ACL configured, the new ACL replaces the previously configured one.

Note The mac access-group interface configuration command is only valid when applied to a physical Layer 2 interface. You cannot use the command on EtherChannel port channels.

IP Access List Entry Sequence Numbering

- This feature does not support dynamic, reflexive, or firewall access lists.

Information about Network Security with ACLs

This chapter describes how to configure network security on the switch by using access control lists (ACLs), which in commands and tables are also referred to as access lists.

ACL Overview

Packet filtering can help limit network traffic and restrict network use by certain users or devices. ACLs filter traffic as it passes through a router or switch and permit or deny packets crossing specified interfaces. An ACL is a sequential collection of permit and deny conditions that apply to packets. When a packet is received on an interface, the switch compares the fields in the packet against any applied ACLs to verify that the packet has the required permissions to be forwarded, based on the criteria specified in the access lists. One by one, it tests packets against the conditions in an access list. The first match decides whether the switch accepts or rejects the packets. Because the switch stops testing after the first match, the order of conditions in the list is critical. If no conditions match, the switch rejects the packet. If there are no restrictions, the switch forwards the packet; otherwise, the switch drops the packet. The switch can use ACLs on all packets it forwards.

You configure access lists on a router or Layer 3 switch to provide basic security for your network. If you do not configure ACLs, all packets passing through the switch could be allowed onto all parts of the network. You can use ACLs to control which hosts can access different parts of a network or to decide which types of traffic are forwarded or blocked at router interfaces. For example, you can allow e-mail traffic to be forwarded but not Telnet traffic.

Access Control Entries

An ACL contains an ordered list of access control entries (ACEs). Each ACE specifies permit or deny and a set of conditions the packet must satisfy in order to match the ACE. The meaning of permit or deny depends on the context in which the ACL is used.

ACL Supported Types

The switch supports IP ACLs and Ethernet (MAC) ACLs:

- IP ACLs filter IPv4 traffic, including TCP, User Datagram Protocol (UDP), Internet Group Management Protocol (IGMP), and Internet Control Message Protocol (ICMP).

- Ethernet ACLs filter non-IP traffic.

This switch also supports quality of service (QoS) classification ACLs.

Supported ACLs

The switch supports three types of ACLs to filter traffic:

- Port ACLs access-control traffic entering a Layer 2 interface. You can apply port ACLs to a Layer 2 interface in each direction to each access list type — IPv4 and MAC.

- Router ACLs access-control routed traffic between VLANs and are applied to Layer 3 interfaces in a specific direction (inbound or outbound).

ACL Precedence

When Port ACLs, and router ACLs are configured on the same switch, the filtering precedence, from greatest to least for ingress traffic is port ACL, and then router ACL. For egress traffic, the filtering precedence is router ACL, and then port ACL.

The following examples describe simple use cases:

- When an input router ACL and input port ACL exist in a switch virtual interface (SVI), incoming packets received on ports to which a port ACL is applied are filtered by the port ACL. Incoming routed IP packets received on other ports are filtered by the router ACL. Other packets are not filtered.

- When an output router ACL and input port ACL exist in an SVI, incoming packets received on the ports to which a port ACL is applied are filtered by the port ACL. Outgoing routed IP packets are filtered by the router ACL. Other packets are not filtered.

Port ACLs

Port ACLs are ACLs that are applied to Layer 2 interfaces on a switch. Port ACLs are supported on physical interfaces and EtherChannel interfaces but not supported on the EtherChannel member interfaces. Port ACLs can be applied to the interface in outbound and inbound direction. The following access lists are supported:

- Standard IP access lists using source addresses

- Extended IP access lists using source and destination addresses and optional protocol type information

- MAC extended access lists using source and destination MAC addresses and optional protocol type information

The switch examines ACLs on an interface and permits or denies packet forwarding based on how the packet matches the entries in the ACL. In this way, ACLs control access to a network or to part of a network.

Figure 12: Using ACLs to Control Traffic in a Network
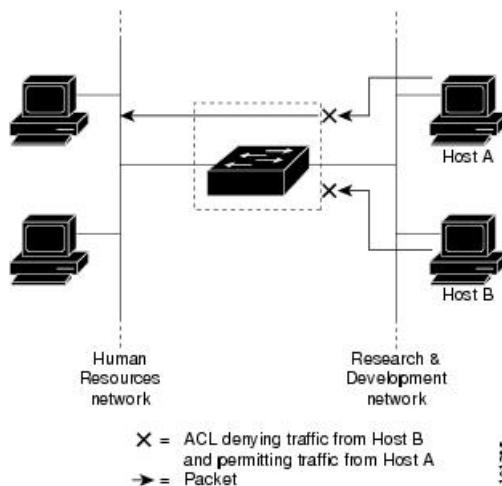
This is an example of using port ACLs to control access to a network when all workstations are in the same VLAN. ACLs applied at the Layer 2 input would allow Host A to access the Human Resources network, but prevent Host B from accessing the same network. Port ACLs can only be applied to Layer 2 interfaces in the inbound direction.

When you apply a port ACL to a trunk port, the ACL filters traffic on all VLANs present on the trunk port. When you apply a port ACL to a port with voice VLAN, the ACL filters traffic on both data and voice VLANs.

With port ACLs, you can filter IP traffic by using IP access lists and non-IP traffic by using MAC addresses. You can filter both IP and non-IP traffic on the same Layer 2 interface by applying both an IP access list and a MAC access list to the interface.

Note   You can't apply more than one IP access list and one MAC access list to a Layer 2 interface. If an IP access list or MAC access list is already configured on a Layer 2 interface and you apply a new IP access list or MAC access list to the interface, the new ACL replaces the previously configured one.

Router ACLs

You can apply router ACLs on switch virtual interfaces (SVIs), which are Layer 3 interfaces to VLANs; on physical Layer 3 interfaces; and on Layer 3 EtherChannel interfaces. You apply router ACLs on interfaces for specific directions (inbound or outbound). You can apply one router ACL in each direction on an interface.

The switch supports these access lists for IPv4 traffic:

- Standard IP access lists use source addresses for matching operations.

- Extended IP access lists use source and destination addresses and optional protocol type information for matching operations.

As with port ACLs, the switch examines ACLs associated with features configured on a given interface. As packets enter the switch on an interface, ACLs associated with all inbound features configured on that interface are examined. After packets are routed and before they are forwarded to the next hop, all ACLs associated with outbound features configured on the egress interface are examined.

ACLs permit or deny packet forwarding based on how the packet matches the entries in the ACL, and can be used to control access to a network or to part of a network.

VLAN Maps

VLAN ACLs or VLAN maps are used to control network traffic within a VLAN. You can apply VLAN maps to all packets that are bridged within a VLAN in the switch or switch stack. VACLs are strictly for security packet filtering and for redirecting traffic to specific physical interfaces. VACLs are not defined by direction (ingress or egress).

All non-IP protocols are access-controlled through MAC addresses and Ethertype using MAC VLAN maps. (IP traffic is not access controlled by MAC VLAN maps.) You can enforce VLAN maps only on packets going through the switch; you cannot enforce VLAN maps on traffic between hosts on a hub or on another switch connected to this switch.

With VLAN maps, forwarding of packets is permitted or denied, based on the action specified in the map.

Figure 13: Using VLAN Maps to Control Traffic

This figure shows how a VLAN map is applied to prevent a specific type of traffic from Host A in VLAN 10 from being forwarded. You can apply only one VLAN map to a VLAN.
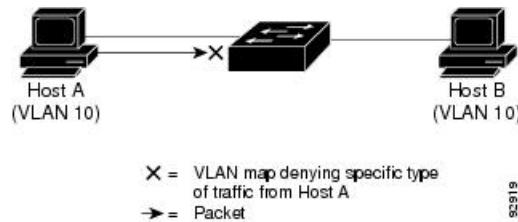
ACEs and Fragmented and Unfragmented Traffic

IP packets can be fragmented as they cross the network. When this happens, only the fragment containing the beginning of the packet contains the Layer 4 information, such as TCP or UDP port numbers, ICMP type and code, and so on. All other fragments are missing this information.

Some access control entries (ACEs) do not check Layer 4 information and therefore can be applied to all packet fragments. ACEs that do test Layer 4 information cannot be applied in the standard manner to most of the fragments in a fragmented IP packet. When the fragment contains no Layer 4 information and the ACE tests some Layer 4 information, the matching rules are modified:

- Permit ACEs that check the Layer 3 information in the fragment (including protocol type, such as TCP, UDP, and so on) are considered to match the fragment regardless of what the missing Layer 4 information might have been.

| **Note** | For TCP ACEs with L4 Ops, the fragmented packets will be dropped per RFC 1858. |

- Deny ACEs that check Layer 4 information never match a fragment unless the fragment contains Layer 4 information.

ACEs and Fragmented and Unfragmented Traffic Examples

Consider access list 102, configured with these commands, applied to three fragmented packets:

```
Device(config)# access-list 102 permit tcp any host 10.1.1.1 eq smtp
Device(config)# access-list 102 deny tcp any host 10.1.1.2 eq telnet
Device(config)# access-list 102 permit tcp any host 10.1.1.2
Device(config)# access-list 102 deny tcp any any
```

Note In the first and second ACEs in the examples, the eq keyword after the destination address means to test for the TCP-destination-port well-known numbers equaling Simple Mail Transfer Protocol (SMTP) and Telnet, respectively.

- Packet A is a TCP packet from host 10.2.2.2., port 65000, going to host 10.1.1.1 on the SMTP port. If this packet is fragmented, the first fragment matches the first ACE (a permit) as if it were a complete packet because all Layer 4 information is present. The remaining fragments also match the first ACE, even though they do not contain the SMTP port information, because the first ACE only checks Layer 3 information when applied to fragments. The information in this example is that the packet is TCP and that the destination is 10.1.1.1.

- Packet B is from host 10.2.2.2, port 65001, going to host 10.1.1.2 on the Telnet port. If this packet is fragmented, the first fragment matches the second ACE (a deny) because all Layer 3 and Layer 4 information is present. The remaining fragments in the packet do not match the second ACE because they are missing Layer 4 information. Instead, they match the third ACE (a permit).

  Because the first fragment was denied, host 10.1.1.2 cannot reassemble a complete packet, so packet B is effectively denied. However, the later fragments that are permitted will consume bandwidth on the network and resources of host 10.1.1.2 as it tries to reassemble the packet.

- Fragmented packet C is from host 10.2.2.2, port 65001, going to host 10.1.1.3, port ftp. If this packet is fragmented, the first fragment matches the fourth ACE (a deny). All other fragments also match the fourth ACE because that ACE does not check any Layer 4 information and because Layer 3 information in all fragments shows that they are being sent to host 10.1.1.3, and the earlier permit ACEs were checking different hosts.

ACLs and Switch Stacks

ACL support is the same for a switch stack as for a standalone switch. ACL configuration information is propagated to all switches in the stack. All switches in the stack, including the active switch, process the information and program their hardware.

Active Switch and ACL Functions

The active switch performs these ACL functions:

- It processes the ACL configuration and propagates the information to all stack members.

- It distributes the ACL information to any switch that joins the stack.

- If packets must be forwarded by software for any reason (for example, not enough hardware resources), the active switch forwards the packets only after applying ACLs on the packets.

- It programs its hardware with the ACL information it processes.

Stack Member and ACL Functions

Stack members perform these ACL functions:

- They receive the ACL information from the active switch and program their hardware.

- A stack member configured as a standby switch, performs the functions of the active switch in the event the active switch fails.

Active Switch Failure and ACLs

Both the active and standby switches have the ACL information. When the active switch fails, the standby takes over. The new active switch distributes the ACL information to all stack members.

Standard and Extended IPv4 ACLs

This section describes IP ACLs.

An ACL is a sequential collection of permit and deny conditions. One by one, the switch tests packets against the conditions in an access list. The first match determines whether the switch accepts or rejects the packet. Because the switch stops testing after the first match, the order of the conditions is critical. If no conditions match, the switch denies the packet.

The software supports these types of ACLs or access lists for IPv4:

- Standard IP access lists use source addresses for matching operations.

- Extended IP access lists use source and destination addresses for matching operations and optional protocol-type information for finer granularity of control.

IPv4 ACL Switch Unsupported Features

Configuring IPv4 ACLs on the switch is the same as configuring IPv4 ACLs on other Cisco switches and routers.

The following ACL-related features are not supported:

- Non-IP protocol ACLs

- IP accounting

- Reflexive ACLs and dynamic ACLs are not supported.

Access List Numbers

The number you use to denote your ACL shows the type of access list that you are creating.

This lists the access-list number and corresponding access list type and shows whether or not they are supported in the switch. The switch supports IPv4 standard and extended access lists, numbers 1 to 199 and 1300 to 2699.

Table 27: Access List Numbers

Access List Number	Type	Supported
1–99	IP standard access list	Yes
100–199	IP extended access list	Yes
200–299	Protocol type-code access list	No
300–399	DECnet access list	No
400–499	XNS standard access list	No
500–599	XNS extended access list	No
600–699	AppleTalk access list	No
700–799	48-bit MAC address access list	No
800–899	IPX standard access list	No
900–999	IPX extended access list	No
1000–1099	IPX SAP access list	No
1100–1199	Extended 48-bit MAC address access list	No
1200–1299	IPX summary address access list	No
1300–1999	IP standard access list (expanded range)	Yes
2000–2699	IP extended access list (expanded range)	Yes

In addition to numbered standard and extended ACLs, you can also create standard and extended named IP ACLs by using the supported numbers. That is, the name of a standard IP ACL can be 1 to 99; the name of an extended IP ACL can be 100 to 199. The advantage of using named ACLs instead of numbered lists is that you can delete individual entries from a named list.

Numbered Standard IPv4 ACLs

When creating an ACL, remember that, by default, the end of the ACL contains an implicit deny statement for all packets that it did not find a match for before reaching the end. With standard access lists, if you omit the mask from an associated IP host address ACL specification, 0.0.0.0 is assumed to be the mask.

The switch always rewrites the order of standard access lists so that entries with host matches and entries with matches having a don't care mask of 0.0.0.0 are moved to the top of the list, above any entries with non-zero

don't care masks. Therefore, in show command output and in the configuration file, the ACEs do not necessarily appear in the order in which they were entered.

After creating a numbered standard IPv4 ACL, you can apply it to , to terminal lines, or to interfaces.

Numbered Extended IPv4 ACLs

Although standard ACLs use only source addresses for matching, you can use extended ACL source and destination addresses for matching operations and optional protocol type information for finer granularity of control. When you are creating ACEs in numbered extended access lists, remember that after you create the ACL, any additions are placed at the end of the list. You cannot reorder the list or selectively add or remove ACEs from a numbered list.

The switch does not support dynamic or reflexive access lists. It also does not support filtering based on the type of service (ToS) minimize-monetary-cost bit.

Some protocols also have specific parameters and keywords that apply to that protocol.

You can define an extended TCP, UDP, ICMP, IGMP, or other IP ACL. The switch also supports these IP protocols:

These IP protocols are supported:

- Authentication Header Protocol (ahp)

- Encapsulation Security Payload (esp)

- Enhanced Interior Gateway Routing Protocol (eigrp)

- generic routing encapsulation (gre)

- Internet Control Message Protocol (icmp)

- Internet Group Management Protocol (igmp)

- any Interior Protocol (ip)

- IP in IP tunneling (ipinip)

- KA9Q NOS-compatible IP over IP tunneling (nos)

- Open Shortest Path First routing (ospf)

- Payload Compression Protocol (pcp)

- Protocol-Independent Multicast (pim)

- Transmission Control Protocol (tcp)

- User Datagram Protocol (udp)

Named IPv4 ACLs

You can identify IPv4 ACLs with an alphanumeric string (a name) rather than a number. You can use named ACLs to configure more IPv4 access lists in a router than if you were to use numbered access lists. If you identify your access list with a name rather than a number, the mode and command syntax are slightly different. However, not all commands that use IP access lists accept a named access list.

Note The name you give to a standard or extended ACL can also be a number in the supported range of access list numbers. That is, the name of a standard IP ACL can be 1 to 99 and . The advantage of using named ACLs instead of numbered lists is that you can delete individual entries from a named list.

Consider these guidelines before configuring named ACLs:

- Numbered ACLs are also available.

- A standard ACL and an extended ACL cannot have the same name.

ACL Logging

The switch software can provide logging messages about packets permitted or denied by a standard IP access list. That is, any packet that matches the ACL causes an informational logging message about the packet to be sent to the console. The level of messages logged to the console is controlled by the logging console commands controlling the syslog messages.

Note ACL logging is only supported for RACL.

Note Because routing is done in hardware and logging is done in software, if a large number of packets match a permit or deny ACE containing a log keyword, the software might not be able to match the hardware processing rate, and not all packets will be logged.

The first packet that triggers the ACL causes a logging message right away, and subsequent packets are collected over 5-minute intervals before they appear or logged. The logging message includes the access list number, whether the packet was permitted or denied, the source IP address of the packet, and the number of packets from that source permitted or denied in the prior 5-minute interval.

Note The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from crashing due to too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list.

Hardware and Software Treatment of IP ACLs

ACL processing is performed in hardware. If the hardware reaches its capacity to store ACL configurations, all packets on that interface are dropped.

Note If an ACL configuration cannot be implemented in hardware due to an out-of-resource condition on a device or stack member, then only the traffic in that VLAN arriving on that device is affected.

For router ACLs, other factors can cause packets to be sent to the CPU:

- Using the log keyword

- Generating ICMP unreachable messages

When you enter the show ip access-lists privileged EXEC command, the match count displayed does not account for packets that are access controlled in hardware. Use the show platform software fed switch { switch_num | active | standby } acl counters hardware privileged EXEC command to obtain some basic hardware ACL statistics for switched and routed packets.

Router ACLs function as follows:

- The hardware controls permit and deny actions of standard and extended ACLs (input and output) for security access control.

- If log has not been specified, the flows that match a deny statement in a security ACL are dropped by the hardware if ip unreachables is disabled. The flows matching a permit statement are switched in hardware.

- Adding the log keyword to an ACE in a router ACL causes a copy of the packet to be sent to the CPU for logging only. If the ACE is a permit statement, the packet is still switched and routed in hardware.

VLAN Map Configuration Guidelines

VLAN maps are the only way to control filtering within a VLAN. VLAN maps have no direction. To filter traffic in a specific direction by using a VLAN map, you need to include an ACL with specific source or destination addresses. If there is a match clause for that type of packet (IP or MAC) in the VLAN map, the default action is to drop the packet if the packet does not match any of the entries within the map. If there is no match clause for that type of packet, the default is to forward the packet.

The following are the VLAN map configuration guidelines:

- If there is no ACL configured to deny traffic on an interface and no VLAN map is configured, all traffic is permitted.

- Each VLAN map consists of a series of entries. The order of entries in an VLAN map is important. A packet that comes into the switch is tested against the first entry in the VLAN map. If it matches, the action specified for that part of the VLAN map is taken. If there is no match, the packet is tested against the next entry in the map.

- If the VLAN map has at least one match clause for the type of packet (IP or MAC) and the packet does not match any of these match clauses, the default is to drop the packet. If there is no match clause for that type of packet in the VLAN map, the default is to forward the packet.

- Logging is not supported for VLAN maps.

- When a switch has an IP access list or MAC access list applied to a Layer 2 interface, and you apply a VLAN map to a VLAN that the port belongs to, the port ACL takes precedence over the VLAN map.

- If a VLAN map configuration cannot be applied in hardware, all packets in that VLAN are dropped.

VLAN Maps with Router ACLs

To access control both bridged and routed traffic, you can use VLAN maps only or a combination of router ACLs and VLAN maps. You can define router ACLs on both input and output routed VLAN interfaces, and you can define a VLAN map to access control the bridged traffic.

If a packet flow matches a VLAN-map deny clause in the ACL, regardless of the router ACL configuration, the packet flow is denied.

Note When you use router ACLs with VLAN maps, packets that require logging on the router ACLs are not logged if they are denied by a VLAN map.

If the VLAN map has a match clause for the type of packet (IP or MAC) and the packet does not match the type, the default is to drop the packet. If there is no match clause in the VLAN map, and no action specified, the packet is forwarded if it does not match any VLAN map entry.

VLAN Maps and Router ACL Configuration Guidelines

These guidelines are for configurations where you need to have an router ACL and a VLAN map on the same VLAN. These guidelines do not apply to configurations where you are mapping router ACLs and VLAN maps on different VLANs.

If you must configure a router ACL and a VLAN map on the same VLAN, use these guidelines for both router ACL and VLAN map configuration:

- You can configure only one VLAN map and one router ACL in each direction (input/output) on a VLAN interface.

- Whenever possible, try to write the ACL with all entries having a single action except for the final, default action of the other type. That is, write the ACL using one of these two forms:

  permit... permit... permit... deny ip any any

  or

  deny... deny... deny... permit ip any any

- To define multiple actions in an ACL (permit, deny), group each action type together to reduce the number of entries.

- Avoid including Layer 4 information in an ACL; adding this information complicates the merging process. The best merge results are obtained if the ACLs are filtered based on IP addresses (source and destination) and not on the full flow (source IP address, destination IP address, protocol, and protocol ports). It is also helpful to use don't care bits in the IP address, whenever possible.

  If you need to specify the full-flow mode and the ACL contains both IP ACEs and TCP/UDP/ICMP ACEs with Layer 4 information, put the Layer 4 ACEs at the end of the list. This gives priority to the filtering of traffic based on IP addresses.

Time Ranges for ACLs

You can selectively apply extended ACLs based on the time of day and the week by using the time-range global configuration command. First, define a time-range name and set the times and the dates or the days of

the week in the time range. Then enter the time-range name when applying an ACL to set restrictions to the access list. You can use the time range to define when the permit or deny statements in the ACL are in effect, for example, during a specified time period or on specified days of the week. The time-range keyword and argument are referenced in the named and numbered extended ACL task tables.

Time-based access lists trigger CPU activity because the new configuration of the access list must be merged with other features and the combined configuration loaded into the hardware memory. For this reason, you should be careful not to have several access lists configured to take affect in close succession (within a small number of minutes of each other.)

Note   The time range relies on the switch system clock; therefore, you need a reliable clock source. We recommend that you use Network Time Protocol (NTP) to synchronize the switch clock.

IPv4 ACL Interface Considerations

For inbound ACLs, after receiving a packet, the switch checks the packet against the ACL. If the ACL permits the packet, the switch continues to process the packet. If the ACL rejects the packet, the switch discards the packet.

For outbound ACLs, after receiving and routing a packet to a controlled interface, the switch checks the packet against the ACL. If the ACL permits the packet, the switch sends the packet. If the ACL rejects the packet, the switch discards the packet.

When you apply an undefined ACL to an interface, the switch acts as if the ACL has not been applied to the interface and permits all packets. Remember this behavior if you use undefined ACLs for network security.

How to Configure ACLs

Configuring IPv4 ACLs

Follow the procedure given below to use IP ACLs on the switch:

Procedure

Step 1   Create an ACL by specifying an access list number or name and the access conditions.

Step 2   Apply the ACL to interfaces or terminal lines. You can also apply standard and extended IP ACLs to VLAN maps.

Creating a Numbered Standard ACL (CLI)

Follow the procedure given below to create a numbered standard ACL:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	access-list access-list-number {deny | permit} source source-wildcard] **Example:** Device(config)# **access-list 2 deny your_host**	Defines a standard IPv4 access list by using a source address and wildcard. The access-list-number is a decimal number from 1 to 99 or 1300 to 1999. Enter deny or permit to specify whether to deny or permit access if conditions are matched. The source is the source address of the network or host from which the packet is being sent specified as: • The 32-bit quantity in dotted-decimal format. • The keyword any as an abbreviation for source and source-wildcard of 0.0.0.0 255.255.255.255. You do not need to enter a source-wildcard. • The keyword host as an abbreviation for source and source-wildcard of source 0.0.0.0. (Optional) The source-wildcard applies wildcard bits to the source. **Note** Logging is supported only on ACLs attached to Layer 3 interfaces.
Step 4	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

	Command or Action	**Purpose**
Step 5	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Creating a Numbered Extended ACL (CLI)

Follow the procedure given below to create a numbered extended ACL:

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	access-list access-list-number {deny | permit} protocol source source-wildcard destination destination-wildcard [precedence precedence] [tos tos] [fragments] [time-range time-range-name] [dscp dscp] **Example:** Device(config)# **access-list 101 permit ip host 10.1.1.2 any precedence 0 tos 0 log**	Defines an extended IPv4 access list and the access conditions. The access-list-number is a decimal number from 100 to 199 or 2000 to 2699. Enter deny or permit to specify whether to deny or permit the packet if conditions are matched. For protocol, enter the name or number of an P protocol: ahp, eigrp, esp, gre, icmp, igmp, igrp, ip, ipinip, nos, ospf, pcp, pim, tcp, or udp, or an integer in the range 0 to 255 representing an IP protocol number. To match any Internet protocol (including ICMP, TCP, and UDP), use the keyword ip. **Note** This step includes options for most IP protocols. For additional specific parameters for TCP, UDP, ICMP, and IGMP, see the following steps. The source is the number of the network or host from which the packet is sent.

Command or Action	Purpose
	The source-wildcard applies wildcard bits to the source.
	The destination is the network or host number to which the packet is sent.
	The destination-wildcard applies wildcard bits to the destination.
	Source, source-wildcard, destination, and destination-wildcard can be specified as:
	• The 32-bit quantity in dotted-decimal format.
	• The keyword any for 0.0.0.0 255.255.255.255 (any host).
	• The keyword host for a single host 0.0.0.0.
	The other keywords are optional and have these meanings:
	• precedence—Enter to match packets with a precedence level specified as a number from 0 to 7 or by name: routine (0), priority (1), immediate (2), flash (3), flash-override (4), critical (5), internet (6), network (7).
	• fragments—Enter to check non-initial fragments.
	• tos—Enter to match by type of service level, specified by a number from 0 to 15 or a name: normal (0), max-reliability (2), max-throughput (4), min-delay (8).
	• time-range—Specify the time-range name.
	• dscp—Enter to match packets with the DSCP value specified by a number from 0 to 63, or use the question mark (?) to see a list of available values.

	Command or Action	**Purpose**
		Note Your controller must support the ability to:
		• Mark DCSP
		• Mark UP
		• Map DSCP and UP
		For more information on DSCP-to-UP Mapping, see:
		https://tools.ietf.org/html/draft-ietf-tsvwg-ieee-802-11-01
		Note If you enter a dscp value, you cannot enter tos or precedence. You can enter both a tos and a precedence value with no dscp.
Step 3	access-list access-list-number {deny | permit} tcp source source-wildcard [operator port] destination destination-wildcard [operator port] [established] [precedence precedence] [tos tos] [fragments] [time-range time-range-name] [dscp dscp] [flag] **Example:** `Device(config)# `**`access-list 101 permit tcp any any eq 500`**	Defines an extended TCP access list and the access conditions. The parameters are the same as those described for an extended IPv4 ACL, with these exceptions: (Optional) Enter an operator and port to compare source (if positioned after source source-wildcard) or destination (if positioned after destination destination-wildcard) port. Possible operators include eq (equal), gt (greater than), lt (less than), neq (not equal), and range (inclusive range). Operators require a port number (range requires two port numbers separated by a space). Enter the port number as a decimal number (from 0 to 65535) or the name of a TCP port. Use only TCP port numbers or names when filtering TCP. The other optional keywords have these meanings: • established—Enter to match an established connection. This has the same function as matching on the ack or rst flag. • flag—Enter one of these flags to match by the specified TCP header bits: ack (acknowledge), fin (finish), psh (push), rst (reset), syn (synchronize), or urg (urgent).

		Command or Action	**Purpose**
Step 4		access-list access-list-number {deny | permit} udp source source-wildcard [operator port] destination destination-wildcard [operator port] [precedence precedence] [tos tos] [fragments] [time-range time-range-name] [dscp dscp] **Example:** Device(config)# **access-list 101 permit udp any any eq 100**	(Optional) Defines an extended UDP access list and the access conditions. The UDP parameters are the same as those described for TCP except that the [operator [port]] port number or name must be a UDP port number or name, and the flag and established keywords are not valid for UDP.
Step 5		access-list access-list-number {deny | permit} icmp source source-wildcard destination destination-wildcard [icmp-type | [[icmp-type icmp-code] | [icmp-message]] [precedence precedence] [tos tos] [fragments] [time-range time-range-name] [dscp dscp] **Example:** Device(config)# **access-list 101 permit icmp any any 200**	Defines an extended ICMP access list and the access conditions. The ICMP parameters are the same as those described for most IP protocols in an extended IPv4 ACL, with the addition of the ICMP message type and code parameters. These optional keywords have these meanings: • icmp-type—Enter to filter by ICMP message type, a number from 0 to 255. • icmp-code—Enter to filter ICMP packets that are filtered by the ICMP message code type, a number from 0 to 255. • icmp-message—Enter to filter ICMP packets by the ICMP message type name or the ICMP message type and code name.
Step 6		access-list access-list-number {deny | permit} igmp source source-wildcard destination destination-wildcard [igmp-type] [precedence precedence] [tos tos] [fragments] [time-range time-range-name] [dscp dscp] **Example:** Device(config)# **access-list 101 permit igmp any any 14**	(Optional) Defines an extended IGMP access list and the access conditions. The IGMP parameters are the same as those described for most IP protocols in an extended IPv4 ACL, with this optional parameter. igmp-type—To match IGMP message type, enter a number from 0 to 15, or enter the message name: dvmrp, host-query, host-report, pim, or trace.
Step 7		end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

Creating Named Standard ACLs

Follow the procedure given below to create a standard ACL using names:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip access-list standard name **Example:** Device(config)# **ip access-list standard 20**	Defines a standard IPv4 access list using a name, and enter access-list configuration mode. The name can be a number from 1 to 99.
Step 4	Use one of the following: - deny {source [source-wildcard] | host source | any} [log] - permit {source [source-wildcard] | host source | any} [log] **Example:** Device(config-std-nacl)# **deny 192.168.0.0 0.0.255.255 255.255.0.0 0.0.255.255** or Device(config-std-nacl)# **permit 10.108.0.0 0.0.0.0 255.255.255.0 0.0.0.0**	In access-list configuration mode, specify one or more conditions denied or permitted to decide if the packet is forwarded or dropped. - host source—A source and source wildcard of source 0.0.0.0. - any—A source and source wildcard of 0.0.0.0 255.255.255.255.
Step 5	end **Example:** Device(config-std-nacl)# **end**	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	show running-config	Verifies your entries.
	Example:	
	Device# **show running-config**	
Step 7	copy running-config startup-config	(Optional) Saves your entries in the configuration file.
	Example:	
	Device# **copy running-config startup-config**	

Creating Extended Named ACLs

Follow the procedure given below to create an extended ACL using names:

Procedure

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. Enter your password if prompted.
	Example:	
	Device> **enable**	
Step 2	configure terminal	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	ip access-list extended name	Defines an extended IPv4 access list using a name, and enter access-list configuration mode.
	Example:	The name can be a number from 100 to 199.
	Device(config)# **ip access-list extended 150**	
Step 4	{deny	permit} protocol {source [source-wildcard]	host source	any} {destination [destination-wildcard]	host destination	any} [precedence precedence] [tos tos] [established] [log] [time-range time-range-name]	In access-list configuration mode, specify the conditions allowed or denied. Use the log keyword to get access list logging messages, including violations.
	Example:	• host source—A source and source wildcard of source 0.0.0.0.

	Command or Action	Purpose
	`Device(config-ext-nacl)# permit 0 any any`	• host destintation—A destination and destination wildcard of destination 0.0.0.0. • any—A source and source wildcard or destination and destination wildcard of 0.0.0.0 255.255.255.255.
Step 5	end **Example:** `Device(config-ext-nacl)# end`	Returns to privileged EXEC mode.
Step 6	show running-config **Example:** `Device# show running-config`	Verifies your entries.
Step 7	copy running-config startup-config **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

When you are creating extended ACLs, remember that, by default, the end of the ACL contains an implicit deny statement for everything if it did not find a match before reaching the end. For standard ACLs, if you omit the mask from an associated IP host address access list specification, 0.0.0.0 is assumed to be the mask.

After you create an ACL, any additions are placed at the end of the list. You cannot selectively add ACL entries to a specific ACL. However, you can use no permit and no deny access-list configuration mode commands to remove entries from a named ACL.

Being able to selectively remove lines from a named ACL is one reason you might use named ACLs instead of numbered ACLs.

What to do next

After creating a named ACL, you can apply it to interfaces or to VLANs .

Configuring Time Ranges for ACLs

Follow these steps to configure a time-range parameter for an ACL:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device(config)# **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	time-range time-range-name **Example:** Device(config)# **time-range workhours**	Assigns a meaningful name (for example, workhours) to the time range to be created, and enter time-range configuration mode. The name cannot contain a space or quotation mark and must begin with a letter.

Step 4	Use one of the following:
	
	   • absolute [start time date] [end time date]
	   • periodic day-of-the-week hh:mm to [day-of-the-week] hh:mm
	   • periodic {weekdays | weekend | daily} hh:mm to hh:mm
	
	Example:
	
	Device(config-time-range)# **absolute start 00:00 1 Jan 2006 end 23:59 1 Jan 2006**
	
	or
	
	Device(config-time-range)# **periodic weekdays 8:00 to 12:00**
		Specifies when the function it will be applied to is operational.
		
		   • You can use only one absolute statement in the time range. If you configure more than one absolute statement, only the one configured last is executed.
		
		   • You can enter multiple periodic statements. For example, you could configure different hours for weekdays and weekends.
		
		See the example configurations.

Step 5	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	show running-config **Example:** Device# **show running-config**	Verifies your entries.

	Command or Action	**Purpose**
Step 7	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

What to do next

Repeat the steps if you have multiple items that you want in effect at different times.

Applying an IPv4 ACL to a Terminal Line

You can use numbered ACLs to control access to one or more terminal lines. You cannot apply named ACLs to lines. You must set identical restrictions on all the virtual terminal lines because a user can attempt to connect to any of them.

Follow these steps to restrict incoming and outgoing connections between a virtual terminal line and the addresses in an ACL:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device(config)# **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	line [console | vty] line-number **Example:** Device(config)# **line console 0**	Identifies a specific line to configure, and enter in-line configuration mode. • console—Specifies the console terminal line. The console port is DCE. • vty—Specifies a virtual terminal for remote console access. The line-number is the first line number in a contiguous group that you want to configure when the line type is specified. The range is from 0 to 16.

	Command or Action	Purpose
Step 4	access-class access-list-number {in	out} **Example:** Device(config-line)# **access-class 10 in**	Restricts incoming and outgoing connections between a particular virtual terminal line (into a device) and the addresses in an access list.
Step 5	end **Example:** Device(config-line)# **end**	Returns to privileged EXEC mode.
Step 6	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 7	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Applying an IPv4 ACL to an Interface (CLI)

This section describes how to apply IPv4 ACLs to network interfaces.

Beginning in privileged EXEC mode, follow the procedure given below to control access to an interface:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet1/0/1**	Identifies a specific interface for configuration, and enter interface configuration mode. The interface can be a Layer 2 interface (port ACL), or a Layer 3 interface (router ACL).

	Command or Action	**Purpose**
Step 3	ip access-group {access-list-number | name} {in | out} **Example:** Device(config-if)# **ip access-group 2 in**	Controls access to the specified interface.
Step 4	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.
Step 5	show running-config **Example:** Device# **show running-config**	Displays the access list configuration.
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Creating Named MAC Extended ACLs

You can filter non-IPv4 traffic on a VLAN or on a Layer 2 interface by using MAC addresses and named MAC extended ACLs. The procedure is similar to that of configuring other extended named ACLs.

Follow these steps to create a named MAC extended ACL:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 3	mac access-list extended name **Example:** Device(config)# **mac access-list extended mac1**	Defines an extended MAC access list using a name.
Step 4	{deny | permit} {any | host source MAC address | source MAC address mask} {any | host destination MAC address | destination MAC address mask} [type mask | lsap lsap mask | aarp | amber | dec-spanning | decnet-iv | diagnostic | dsm | etype-6000 | etype-8042 | lat | lavc-sca | mop-console | mop-dump | msdos | mumps | netbios | vines-echo | vines-ip | xns-idp | 0-65535] [cos cos] **Example:** Device(config-ext-macl)# **deny any any decnet-iv** or Device(config-ext-macl)# **permit any any**	In extended MAC access-list configuration mode, specifies to permit or deny any source MAC address, a source MAC address with a mask, or a specific host source MAC address and any destination MAC address, destination MAC address with a mask, or a specific destination MAC address. (Optional) You can also enter these options: • type mask—An arbitrary EtherType number of a packet with Ethernet II or SNAP encapsulation in decimal, hexadecimal, or octal with optional mask of don't care bits applied to the EtherType before testing for a match. • lsap lsap mask—An LSAP number of a packet with IEEE 802.2 encapsulation in decimal, hexadecimal, or octal with optional mask of don't care bits. • aarp | amber | dec-spanning | decnet-iv | diagnostic | dsm | etype-6000 | etype-8042 | lat | lavc-sca | mop-console | mop-dump | msdos | mumps | netbios | vines-echo | vines-ip | xns-idp—A non-IP protocol. • cos cos—An IEEE 802.1Q cost of service number from 0 to 7 used to set priority.
Step 5	end **Example:** Device(config-ext-macl)# **end**	Returns to privileged EXEC mode.
Step 6	show running-config **Example:** Device# **show running-config**	Verifies your entries.

	Command or Action	**Purpose**
Step 7	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Applying a MAC ACL to a Layer 2 Interface

Follow these steps to apply a MAC access list to control access to a Layer 2 interface:

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	interface interface-id **Example:** Device(config)# **interface gigabitethernet1/0/2**	Identifies a specific interface, and enter interface configuration mode. The interface must be a physical Layer 2 interface (port ACL).
Step 4	mac access-group {name} {in } **Example:** Device(config-if)# **mac access-group mac1 in**	Controls access to the specified interface by using the MAC access list. Port ACLs are supported in the inbound directions .
Step 5	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	show mac access-group [interface interface-id] **Example:** Device# **show mac access-group interface gigabitethernet1/0/2**	Displays the MAC access list applied to the interface or all Layer 2 interfaces.
Step 7	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 8	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.

After receiving a packet, the switch checks it against the inbound ACL. If the ACL permits it, the switch continues to process the packet. If the ACL rejects the packet, the switch discards it. When you apply an undefined ACL to an interface, the switch acts as if the ACL has not been applied and permits all packets. Remember this behavior if you use undefined ACLs for network security.

Configuring VLAN Maps

Follow the procedure given below to create a VLAN map and apply it to one or more VLANs:

Before you begin

Create the standard or extended IPv4 ACLs or named MAC extended ACLs that you want to apply to the VLAN.

Procedure

	Command or Action	Purpose
Step 1	vlan access-map name [number] **Example:** Device(config)# **vlan access-map map_1 20**	Creates a VLAN map, and give it a name and (optionally) a number. The number is the sequence number of the entry within the map. When you create VLAN maps with the same name, numbers are assigned sequentially in increments of 10. When modifying or deleting maps, you can enter the number of the map entry that you want to modify or delete. VLAN maps do not use the specific permit or deny keywords. To deny a packet by using VLAN maps, create an ACL that would match

	Command or Action	**Purpose**
		the packet, and set the action to drop. A permit in the ACL counts as a match. A deny in the ACL means no match.
		Entering this command changes to access-map configuration mode.
Step 2	match {ip | mac} address {name | number} [name | number] **Example:** Device(config-access-map)# **match ip address ip2**	Match the packet (using either the IP or MAC address) against one or more standard or extended access lists. Note that packets are only matched against access lists of the correct protocol type. IP packets are matched against standard or extended IP access lists. Non-IP packets are only matched against named MAC extended access lists.
		Note If the VLAN map is configured with a match clause for a type of packet (IP or MAC) and the map action is drop, all packets that match the type are dropped. If the VLAN map has no match clause, and the configured action is drop, all IP and Layer 2 packets are dropped.

Step 3	Enter one of the following commands to specify an IP packet or a non-IP packet (with only a known MAC address) and to match the packet against one or more ACLs (standard or extended):
	
	• action { forward}
	
	  Device(config-access-map)# **action forward**
	
	• action { drop}
	
	  Device(config-access-map)# **action drop**
		Sets the action for the map entry.

Step 4	vlan filter mapname vlan-list list **Example:** Device(config)# **vlan filter map 1 vlan-list 20-22**	Applies the VLAN map to one or more VLAN IDs. The list can be a single VLAN ID (22), a consecutive list (10-22), or a string of VLAN IDs (12, 22, 30). Spaces around the comma and hyphen are optional.

Creating a VLAN Map

Each VLAN map consists of an ordered series of entries. Beginning in privileged EXEC mode, follow these steps to create, add to, or delete a VLAN map entry:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	vlan access-map name [number] **Example:** Device(config)# **vlan access-map map_1 20**	Creates a VLAN map, and give it a name and (optionally) a number. The number is the sequence number of the entry within the map. When you create VLAN maps with the same name, numbers are assigned sequentially in increments of 10. When modifying or deleting maps, you can enter the number of the map entry that you want to modify or delete. VLAN maps do not use the specific permit or deny keywords. To deny a packet by using VLAN maps, create an ACL that would match the packet, and set the action to drop. A permit in the ACL counts as a match. A deny in the ACL means no match. Entering this command changes to access-map configuration mode.
Step 3	match {ip | mac} address {name | number} [name | number] **Example:** Device(config-access-map)# **match ip address ip2**	Match the packet (using either the IP or MAC address) against one or more standard or extended access lists. Note that packets are only matched against access lists of the correct protocol type. IP packets are matched against standard or extended IP access lists. Non-IP packets are only matched against named MAC extended access lists.
Step 4	action {drop | forward} **Example:** Device(config-access-map)# **action forward**	(Optional) Sets the action for the map entry. The default is to forward.
Step 5	end **Example:**	Returns to global configuration mode.

	Command or Action	Purpose
	Device(config-access-map)# **end**	
Step 6	show running-config	Displays the access list configuration.
	Example:	
	Device# **show running-config**	
Step 7	copy running-config startup-config	(Optional) Saves your entries in the configuration file.
	Example:	
	Device# **copy running-config startup-config**	

Applying a VLAN Map to a VLAN

To apply a VLAN map to one or more VLANs, perform these steps.

Procedure

	Command or Action	Purpose
Step 1		
Step 2	configure terminal	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	vlan filter mapname vlan-list list	Applies the VLAN map to one or more VLAN IDs.
	Example:	The list can be a single VLAN ID (22), a consecutive list (10-22), or a string of VLAN IDs (12, 22, 30). Spaces around the comma and hyphen are optional.
	Device(config)# **vlan filter map 1 vlan-list 20-22**	
Step 4	end	Returns to privileged EXEC mode.
	Example:	
	Device(config)# **end**	
Step 5	show running-config	Displays the access list configuration.
	Example:	

	Command or Action	Purpose
	Device# **show running-config**	
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Monitoring IPv4 ACLs

You can monitor IPv4 ACLs by displaying the ACLs that are configured on the switch, and displaying the ACLs that have been applied to interfaces and VLANs.

When you use the ip access-group interface configuration command to apply ACLs to a Layer 2 or 3 interface, you can display the access groups on the interface. You can also display the MAC ACLs applied to a Layer 2 interface. You can use the privileged EXEC commands as described in this table to display this information.

Table 28: Commands for Displaying Access Lists and Access Groups

Command	Purpose
show access-lists [number | name]	Displays the contents of one or all current IP and MAC address access lists or a specific access list (numbered or named).
show ip access-lists [number | name]	Displays the contents of all current IP access lists or a specific IP access list (numbered or named).
show ip interface interface-id	Displays detailed configuration and status of an interface. If IP is enabled on the interface and ACLs have been applied by using the ip access-group interface configuration command, the access groups are included in the display.
show running-config [interface interface-id]	Displays the contents of the configuration file for the switch or the specified interface, including all configured MAC and IP access lists and which access groups are applied to an interface.
show mac access-group [interface interface-id]	Displays MAC access lists applied to all Layer 2 interfaces or the specified Layer 2 interface.

Configuration Examples for ACLs

Examples: Using Time Ranges with ACLs

This example shows how to verify after you configure time ranges for workhours and to configure January 1, 2006, as a company holiday.

```
Device# show time-range
time-range entry: new_year_day_2003 (inactive)
   absolute start 00:00 01 January 2006 end 23:59 01 January 2006
time-range entry: workhours (inactive)
   periodic weekdays 8:00 to 12:00
   periodic weekdays 13:00 to 17:00
```

To apply a time range, enter the time-range name in an extended ACL that can implement time ranges. This example shows how to create and verify extended access list 188 that denies TCP traffic from any source to any destination during the defined holiday times and permits all TCP traffic during work hours.

```
Device(config)# access-list 188 deny tcp any any time-range new_year_day_2006
Device(config)# access-list 188 permit tcp any any time-range workhours
Device(config)# end
Device# show access-lists
Extended IP access list 188
   10 deny tcp any any time-range new_year_day_2006 (inactive)
   20 permit tcp any any time-range workhours (inactive)
```

This example uses named ACLs to permit and deny the same traffic.

```
Device(config)# ip access-list extended deny_access
Device(config-ext-nacl)# deny tcp any any time-range new_year_day_2006
Device(config-ext-nacl)# exit
Device(config)# ip access-list extended may_access
Device(config-ext-nacl)# permit tcp any any time-range workhours
Device(config-ext-nacl)# end
Device# show ip access-lists
Extended IP access list lpip_default
    10 permit ip any any
Extended IP access list deny_access
    10 deny tcp any any time-range new_year_day_2006 (inactive)
Extended IP access list may_access
    10 permit tcp any any time-range workhours (inactive)
```

Examples: Including Comments in ACLs

You can use the remark keyword to include comments (remarks) about entries in any IP standard or extended ACL. The remarks make the ACL easier for you to understand and scan. Each remark line is limited to 100 characters.

The remark can go before or after a permit or deny statement. You should be consistent about where you put the remark so that it is clear which remark describes which permit or deny statement. For example, it would

be confusing to have some remarks before the associated permit or deny statements and some remarks after the associated statements.

To include a comment for IP numbered standard or extended ACLs, use the access-list access-list number remark remark global configuration command. To remove the remark, use the no form of this command.

In this example, the workstation that belongs to Jones is allowed access, and the workstation that belongs to Smith is not allowed access:

```
Device(config)# access-list 1 remark Permit only Jones workstation through
Device(config)# access-list 1 permit 171.69.2.88
Device(config)# access-list 1 remark Do not allow Smith through
Device(config)# access-list 1 deny 171.69.3.13
```

For an entry in a named IP ACL, use the remark access-list configuration command. To remove the remark, use the no form of this command.

In this example, the Jones subnet is not allowed to use outbound Telnet:

```
Device(config)# ip access-list extended telnetting
Device(config-ext-nacl)# remark Do not allow Jones subnet to telnet out
Device(config-ext-nacl)# deny tcp host 171.69.2.88 any eq telnet
```

IPv4 ACL Configuration Examples

This section provides examples of configuring and applying IPv4 ACLs. For detailed information about compiling ACLs, see the Cisco IOS Security Configuration Guide, Release 12.4 and to the Configuring IP Services" section in the "IP Addressing and Services" chapter of the Cisco IOS IP Configuration Guide, Release 12.4.

ACLs in a Small Networked Office

Figure 14: Using Router ACLs to Control Traffic

This shows a small networked office environment with routed Port 2 connected to Server A, containing benefits and other information that all employees can access, and routed Port 1 connected to Server B, containing

confidential payroll data. All users can access Server A, but Server B has restricted
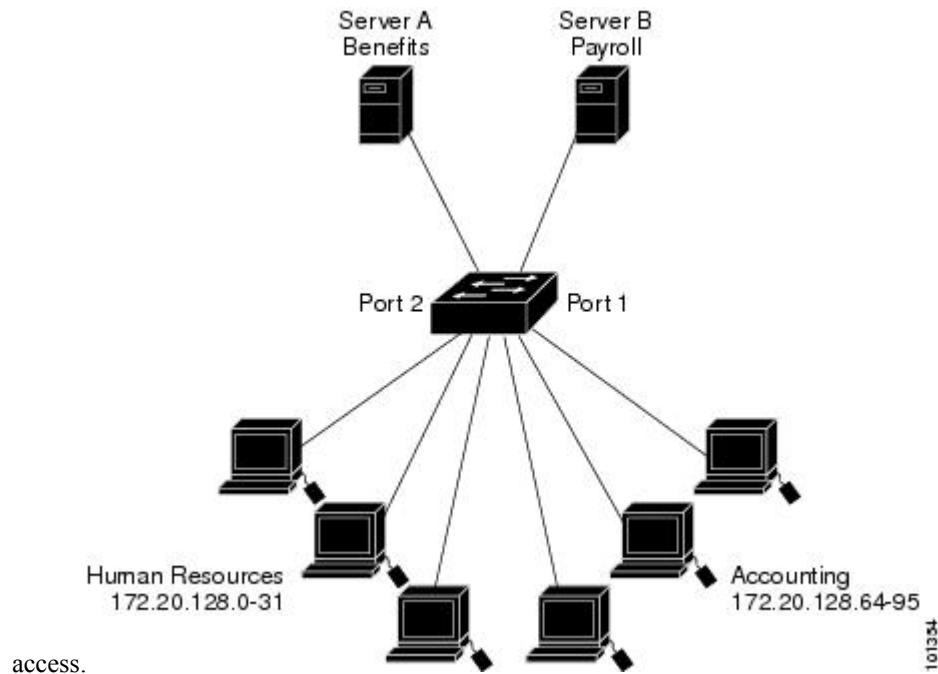
access.

Use router ACLs to do this in one of two ways:

- Create a standard ACL, and filter traffic coming to the server from Port 1.

- Create an extended ACL, and filter traffic coming from the server into Port 1.

Examples: ACLs in a Small Networked Office

This example uses a standard ACL to filter traffic coming into Server B from a port, permitting traffic only from Accounting's source addresses 172.20.128.64 to 172.20.128.95. The ACL is applied to traffic coming out of routed Port 1 from the specified source address.

```
Device(config)# access-list 6 permit 172.20.128.64 0.0.0.31
Device(config)# end
Device# how access-lists
Standard IP access list 6
    10 permit 172.20.128.64, wildcard bits 0.0.0.31
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 6 out
```

This example uses an extended ACL to filter traffic coming from Server B into a port, permitting traffic from any source address (in this case Server B) to only the Accounting destination addresses 172.20.128.64 to 172.20.128.95. The ACL is applied to traffic going into routed Port 1, permitting it to go only to the specified destination addresses. Note that with extended ACLs, you must enter the protocol (IP) before the source and destination information.

```
Device(config)# access-list 106 permit ip any 172.20.128.64 0.0.0.31
Device(config)# end
Device# show access-lists
Extended IP access list 106
```

```
      10 permit ip any 172.20.128.64 0.0.0.31
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 106 in
```

Example: Numbered ACLs

In this example, network 10.0.0.0 is a Class A network whose second octet specifies a subnet; that is, its subnet mask is 255.255.0.0. The third and fourth octets of a network 10.0.0.0 address specify a particular host. Using access list 2, the switch accepts one address on subnet 48 and reject all others on that subnet. The last line of the list shows that the switch accepts addresses on all other network 10.0.0.0 subnets. The ACL is applied to packets entering a port.

```
Device(config)# access-list 2 permit 10.48.0.3
Device(config)# access-list 2 deny 10.48.0.0 0.0.255.255
Device(config)# access-list 2 permit 10.0.0.0 0.255.255.255
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# ip access-group 2 in
```

Examples: Extended ACLs

In this example, the first line permits any incoming TCP connections with destination ports greater than 1023. The second line permits incoming TCP connections to the Simple Mail Transfer Protocol (SMTP) port of host 128.88.1.2. The third line permits incoming ICMP messages for error feedback.

```
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 gt 1023
Device(config)# access-list 102 permit tcp any host 128.88.1.2 eq 25
Device(config)# access-list 102 permit icmp any any
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# ip access-group 102 in
```

In this example, suppose that you have a network connected to the Internet, and you want any host on the network to be able to form TCP connections to any host on the Internet. However, you do not want IP hosts to be able to form TCP connections to hosts on your network, except to the mail (SMTP) port of a dedicated mail host.

SMTP uses TCP port 25 on one end of the connection and a random port number on the other end. The same port numbers are used throughout the life of the connection. Mail packets coming in from the Internet have a destination port of 25. Because the secure system of the network always accepts mail connections on port 25, the incoming are separately controlled.

```
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 eq 23
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 eq 25
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 102 in
```

In this example, the network is a Class B network with the address 128.88.0.0, and the mail host address is 128.88.1.2. The established keyword is used only for the TCP to show an established connection. A match occurs if the TCP datagram has the ACK or RST bits set, which show that the packet belongs to an existing connection. Gigabit Ethernet interface 1 on stack member 1 is the interface that connects the router to the Internet.

```
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 established
```

```
Device(config)# access-list 102 permit tcp any host 128.88.1.2 eq 25
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 102 in
```

Examples: Named ACLs

Creating named standard and extended ACLs

This example creates a standard ACL named internet_filter and an extended ACL named marketing_group. The internet_filter ACL allows all traffic from the source address 1.2.3.4.

```
Device(config)# ip access-list standard Internet_filter
Device(config-ext-nacl)# permit 1.2.3.4
Device(config-ext-nacl)# exit
```

The marketing_group ACL allows any TCP Telnet traffic to the destination address and wildcard 171.69.0.0 0.0.255.255 and denies any other TCP traffic. It permits ICMP traffic, denies UDP traffic from any source to the destination address range 171.69.0.0 through 179.69.255.255 with a destination port less than 1024, denies any other IP traffic, and provides a log of the result.

```
Device(config)# ip access-list extended marketing_group
Device(config-ext-nacl)# permit tcp any 171.69.0.0 0.0.255.255 eq telnet
Device(config-ext-nacl)# deny tcp any any
Device(config-ext-nacl)# permit icmp any any
Device(config-ext-nacl)# deny udp any 171.69.0.0 0.0.255.255 lt 1024
Device(config-ext-nacl)# deny ip any any log
Device(config-ext-nacl)# exit
```

The Internet_filter ACL is applied to outgoing traffic and the marketing_group ACL is applied to incoming traffic on a Layer 3 port.

```
Device(config)# interface gigabitethernet3/0/1
Device(config-if)# no switchport
Device(config-if)# ip address 2.0.5.1 255.255.255.0
Device(config-if)# ip access-group Internet_filter out
Device(config-if)# ip access-group marketing_group in
```

Deleting individual ACEs from named ACLs

This example shows how you can delete individual ACEs from the named access list border-list:

```
Device(config)# ip access-list extended border-list
Device(config-ext-nacl)# no permit ip host 10.1.1.3 any
```

Examples: Time Range Applied to an IP ACL

This example denies HTTP traffic on IP on Monday through Friday between the hours of 8:00 a.m. and 6:00 p.m (18:00). The example allows UDP traffic only on Saturday and Sunday from noon to 8:00 p.m. (20:00).

```
Device(config)# time-range no-http
Device(config)# periodic weekdays 8:00 to 18:00
```

```
!
Device(config)# time-range udp-yes
Device(config)# periodic weekend 12:00 to 20:00
!
Device(config)# ip access-list extended strict
Device(config-ext-nacl)# deny tcp any any eq www time-range no-http
Device(config-ext-nacl)# permit udp any any time-range udp-yes
!
Device(config-ext-nacl)# exit
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# ip access-group strict in
```

Examples: Configuring Commented IP ACL Entries

In this example of a numbered ACL, the workstation that belongs to Jones is allowed access, and the workstation that belongs to Smith is not allowed access:

```
Device(config)# access-list 1 remark Permit only Jones workstation through
Device(config)# access-list 1 permit 171.69.2.88
Device(config)# access-list 1 remark Do not allow Smith workstation through
Device(config)# access-list 1 deny 171.69.3.13
```

In this example of a numbered ACL, the Winter and Smith workstations are not allowed to browse the web:

```
Device(config)# access-list 100 remark Do not allow Winter to browse the web
Device(config)# access-list 100 deny host 171.69.3.85 any eq www
Device(config)# access-list 100 remark Do not allow Smith to browse the web
Device(config)# access-list 100 deny host 171.69.3.13 any eq www
```

In this example of a named ACL, the Jones subnet is not allowed access:

```
Device(config)# ip access-list standard prevention
Device(config-std-nacl)# remark Do not allow Jones subnet through
Device(config-std-nacl)# deny 171.69.0.0 0.0.255.255
```

In this example of a named ACL, the Jones subnet is not allowed to use outbound Telnet:

```
Device(config)# ip access-list extended telnetting
Device(config-ext-nacl)# remark Do not allow Jones subnet to telnet out
Device(config-ext-nacl)# deny tcp 171.69.0.0 0.0.255.255 any eq telnet
```

Examples: ACL Logging

Two variations of logging are supported on ACLs. The log keyword sends an informational logging message to the console about the packet that matches the entry; the log-input keyword includes the input interface in the log entry.

In this example, standard named access list stan1 denies traffic from 10.1.1.0 0.0.0.255, allows traffic from all other sources, and includes the log keyword.

```
Device(config)# ip access-list standard stan1
Device(config-std-nacl)# deny 10.1.1.0 0.0.0.255 log
Device(config-std-nacl)# permit any log
```

```
Device(config-std-nacl)# exit
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group stan1 in
Device(config-if)# end
Device# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: level debugging, 37 messages logged
    Monitor logging: level debugging, 0 messages logged
    Buffer logging: level debugging, 37 messages logged
    File logging: disabled
    Trap logging: level debugging, 39 message lines logged

Log Buffer (4096 bytes):

00:00:48: NTP: authentication delay calculation problems

<output truncated>

00:09:34:%SEC-6-IPACCESSLOGS:list stan1 permitted 0.0.0.0 1 packet
00:09:59:%SEC-6-IPACCESSLOGS:list stan1 denied 10.1.1.15 1 packet
00:10:11:%SEC-6-IPACCESSLOGS:list stan1 permitted 0.0.0.0 1 packet
```

This example is a named extended access list ext1 that permits ICMP packets from any source to 10.1.1.0 0.0.0.255 and denies all UDP packets.

```
Device(config)# ip access-list extended ext1
Device(config-ext-nacl)# permit icmp any 10.1.1.0 0.0.0.255 log
Device(config-ext-nacl)# deny udp any any log
Device(config-std-nacl)# exit
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip access-group ext1 in
```

This is a an example of a log for an extended ACL:

```
01:24:23:%SEC-6-IPACCESSLOGDP:list ext1 permitted icmp 10.1.1.15 -> 10.1.1.61 (0/0), 1
packet
01:25:14:%SEC-6-IPACCESSLOGDP:list ext1 permitted icmp 10.1.1.15 -> 10.1.1.61 (0/0), 7
packets
01:26:12:%SEC-6-IPACCESSLOGP:list ext1 denied udp 0.0.0.0(0) -> 255.255.255.255(0), 1 packet
01:31:33:%SEC-6-IPACCESSLOGP:list ext1 denied udp 0.0.0.0(0) -> 255.255.255.255(0), 8 packets
```

Note that all logging entries for IP ACLs start with %SEC-6-IPACCESSLOG with minor variations in format depending on the kind of ACL and the access entry that has been matched.

This is an example of an output message when the log-input keyword is entered:

```
00:04:21:%SEC-6-IPACCESSLOGDP:list inputlog permitted icmp 10.1.1.10 (Vlan1 0001.42ef.a400)
 ->
10.1.1.61 (0/0), 1 packet
```

A log message for the same sort of packet using the log keyword does not include the input interface information:

```
00:05:47:%SEC-6-IPACCESSLOGDP:list inputlog permitted icmp 10.1.1.10 -> 10.1.1.61 (0/0), 1
 packet
```

Configuration Examples for ACLs and VLAN Maps

Example: Creating an ACL and a VLAN Map to Deny a Packet

This example shows how to create an ACL and a VLAN map to deny a packet. In the first map, any packets that match the ip1 ACL (TCP packets) would be dropped. You first create the ip1 ACL to permit any TCP packet and no other packets. Because there is a match clause for IP packets in the VLAN map, the default action is to drop any IP packet that does not match any of the match clauses.

```
Device(config)# ip access-list extended ip1
Device(config-ext-nacl)# permit tcp any any
Device(config-ext-nacl)# exit
Device(config)# vlan access-map map_1 10
Device(config-access-map)# match ip address ip1
Device(config-access-map)# action drop
```

Example: Creating an ACL and a VLAN Map to Permit a Packet

This example shows how to create a VLAN map to permit a packet. ACL ip2 permits UDP packets and any packets that match the ip2 ACL are forwarded. In this map, any IP packets that did not match any of the previous ACLs (that is, packets that are not TCP packets or UDP packets) would get dropped.

```
Device(config)# ip access-list extended ip2
Device(config-ext-nacl)# permit udp any any
Device(config-ext-nacl)# exit
Device(config)# vlan access-map map_1 20
Device(config-access-map)# match ip address ip2
Device(config-access-map)# action forward
```

Example: Default Action of Dropping IP Packets and Forwarding MAC Packets

In this example, the VLAN map has a default action of drop for IP packets and a default action of forward for MAC packets. Used with standard ACL 101 and extended named access lists igmp-match and tcp-match, the map will have the following results:

- Forward all UDP packets

- Drop all IGMP packets

- Forward all TCP packets

- Drop all other IP packets

- Forward all non-IP packets

```
Device(config)# access-list 101 permit udp any any
Device(config)# ip access-list extended igmp-match
Device(config-ext-nacl)# permit igmp any any

Device(config-ext-nacl)# permit tcp any any
Device(config-ext-nacl)# exit
Device(config)# vlan access-map drop-ip-default 10
Device(config-access-map)# match ip address 101
Device(config-access-map)# action forward
Device(config-access-map)# exit
```

```
Device(config)# vlan access-map drop-ip-default 20
Device(config-access-map)# match ip address igmp-match
Device(config-access-map)# action drop
Device(config-access-map)# exit
Device(config)# vlan access-map drop-ip-default 30
Device(config-access-map)# match ip address tcp-match
Device(config-access-map)# action forward
```

Example: Default Action of Dropping MAC Packets and Forwarding IP Packets

In this example, the VLAN map has a default action of drop for MAC packets and a default action of forward for IP packets. Used with MAC extended access lists good-hosts and good-protocols, the map will have the following results:

- Forward MAC packets from hosts 0000.0c00.0111 and 0000.0c00.0211

- Forward MAC packets with decnet-iv or vines-ip protocols

- Drop all other non-IP packets

- Forward all IP packets

Example: Default Action of Dropping All Packets

In this example, the VLAN map has a default action of drop for all packets (IP and non-IP). Used with access lists tcp-match and good-hosts from Examples 2 and 3, the map will have the following results:

- Forward all TCP packets

- Forward MAC packets from hosts 0000.0c00.0111 and 0000.0c00.0211

- Drop all other IP packets

- Drop all other MAC packets

```
Device(config)# vlan access-map drop-all-default 10
Device(config-access-map)# match ip address tcp-match
Device(config-access-map)# action forward
Device(config-access-map)# exit
Device(config)# vlan access-map drop-all-default 20
Device(config-access-map)# match mac address good-hosts
Device(config-access-map)# action forward
```

Configuration Examples for Using VLAN Maps in Your Network

Example: Wiring Closet Configuration

Figure 15: Wiring Closet Configuration

In a wiring closet configuration, routing might not be enabled on the switch. In this configuration, the switch can still support a VLAN map and a QoS classification ACL. Assume that Host X and Host Y are in different VLANs and are connected to wiring closet switches A and C. Traffic from Host X to Host Y is eventually being routed by Switch B, a Layer 3 switch with routing enabled. Traffic from Host X to Host Y can be

access-controlled at the traffic entry point,
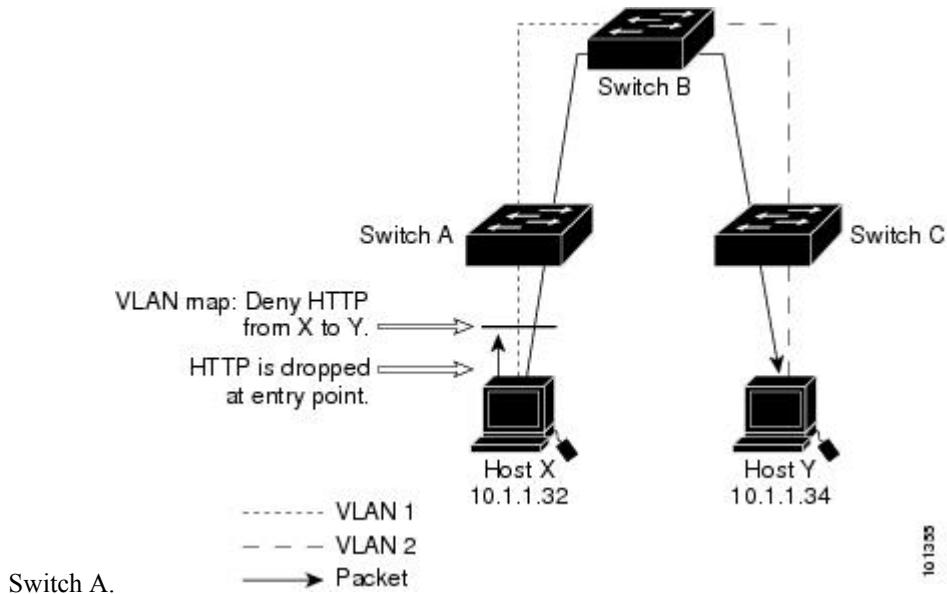
Switch A.

If you do not want HTTP traffic switched from Host X to Host Y, you can configure a VLAN map on Switch A to drop all HTTP traffic from Host X (IP address 10.1.1.32) to Host Y (IP address 10.1.1.34) at Switch A and not bridge it to Switch B.

First, define the IP access list http that permits (matches) any TCP traffic on the HTTP port.

```
Device(config)# ip access-list extended http
Device(config-ext-nacl)# permit tcp host 10.1.1.32 host 10.1.1.34 eq www
Device(config-ext-nacl)# exit
```

Next, create VLAN access map map2 so that traffic that matches the http access list is dropped and all other IP traffic is forwarded.

```
Device(config)# vlan access-map map2 10
Device(config-access-map)# match ip address http
Device(config-access-map)# action drop
Device(config-access-map)# exit
Device(config)# ip access-list extended match_all
Device(config-ext-nacl)# permit ip any any
Device(config-ext-nacl)# exit
Device(config)# vlan access-map map2 20
Device(config-access-map)# match ip address match_all
Device(config-access-map)# action forward
```

Then, apply VLAN access map map2 to VLAN 1.

```
Device(config)# vlan filter map2 vlan 1
```

Example: Restricting Access to a Server on Another VLAN

Figure 16: Restricting Access to a Server on Another VLAN

You can restrict access to a server on another VLAN. For example, server 10.1.1.100 in VLAN 10 needs to have access denied to these hosts:

   • Hosts in subnet 10.1.2.0/8 in VLAN 20 should not have access.

   • Hosts 10.1.1.4 and 10.1.1.8 in VLAN 10 should not have access.
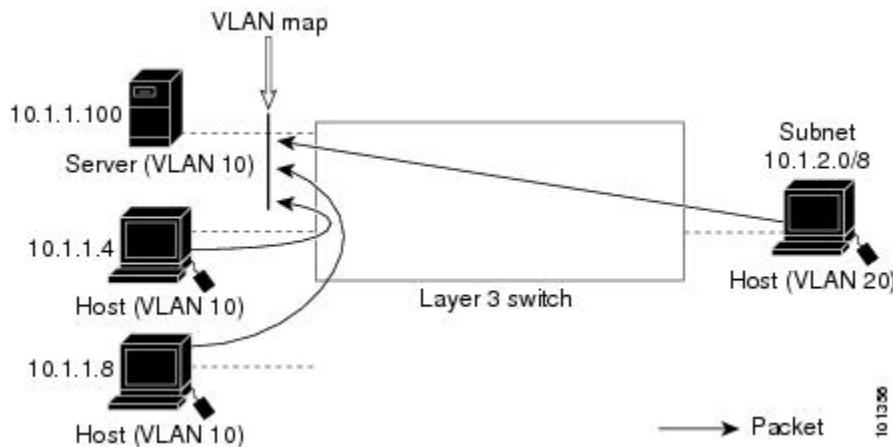
Example: Denying Access to a Server on Another VLAN

This example shows how to deny access to a server on another VLAN by creating the VLAN map SERVER 1 that denies access to hosts in subnet 10.1.2.0.8, host 10.1.1.4, and host 10.1.1.8 and permits other IP traffic. The final step is to apply the map SERVER1 to VLAN 10.

Define the IP ACL that will match the correct packets.

```
Device(config)# ip access-list extended SERVER1_ACL
Device(config-ext-nacl))# permit ip 10.1.2.0 0.0.0.255 host 10.1.1.100
Device(config-ext-nacl))# permit ip host 10.1.1.4 host 10.1.1.100
Device(config-ext-nacl))# permit ip host 10.1.1.8 host 10.1.1.100
Device(config-ext-nacl))# exit
```

Define a VLAN map using this ACL that will drop IP packets that match SERVER1_ACL and forward IP packets that do not match the ACL.

```
Device(config)# vlan access-map SERVER1_MAP
Device(config-access-map)# match ip address SERVER1_ACL
Device(config-access-map)# action drop
Device(config)# vlan access-map SERVER1_MAP 20
Device(config-access-map)# action forward
Device(config-access-map)# exit
```

Apply the VLAN map to VLAN 10.

```
Device(config)# vlan filter SERVER1_MAP vlan-list 10
```

Configuration Examples of Router ACLs and VLAN Maps Applied to VLANs

This section gives examples of applying router ACLs and VLAN maps to a VLAN for switched, bridged, routed, and multicast packets. Although the following illustrations show packets being forwarded to their destination, each time the packet's path crosses a line indicating a VLAN map or an ACL, it is also possible that the packet might be dropped, rather than forwarded.

Example: ACLs and Switched Packets

Figure 17: Applying ACLs on Switched Packets

This example shows how an ACL is applied on packets that are switched within a VLAN. Packets switched within the VLAN without being routed or forwarded by fallback bridging are only subject to the VLAN map of the input VLAN.
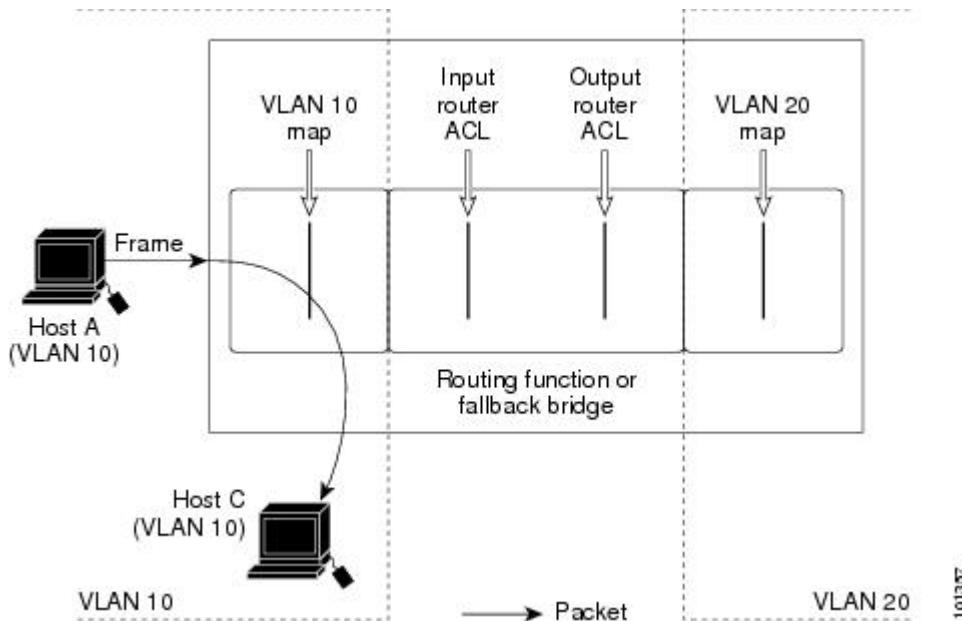
Example: ACLs and Bridged Packets

Figure 18: Applying ACLs on Bridged Packets

This example shows how an ACL is applied on fallback-bridged packets. For bridged packets, only Layer 2 ACLs are applied to the input VLAN. Only non-IP, non-ARP packets can be fallback-bridged.
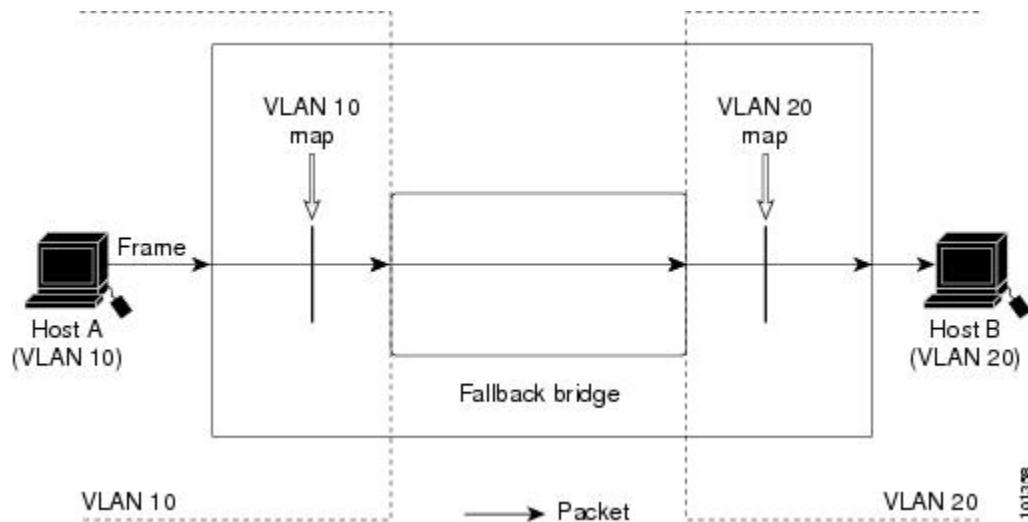
Example: ACLs and Routed Packets

Figure 19: Applying ACLs on Routed Packets

This example shows how ACLs are applied on routed packets. The ACLs are applied in this order:

1. VLAN map for input VLAN

2. Input router ACL

3. Output router ACL

4. VLAN map for output VLAN
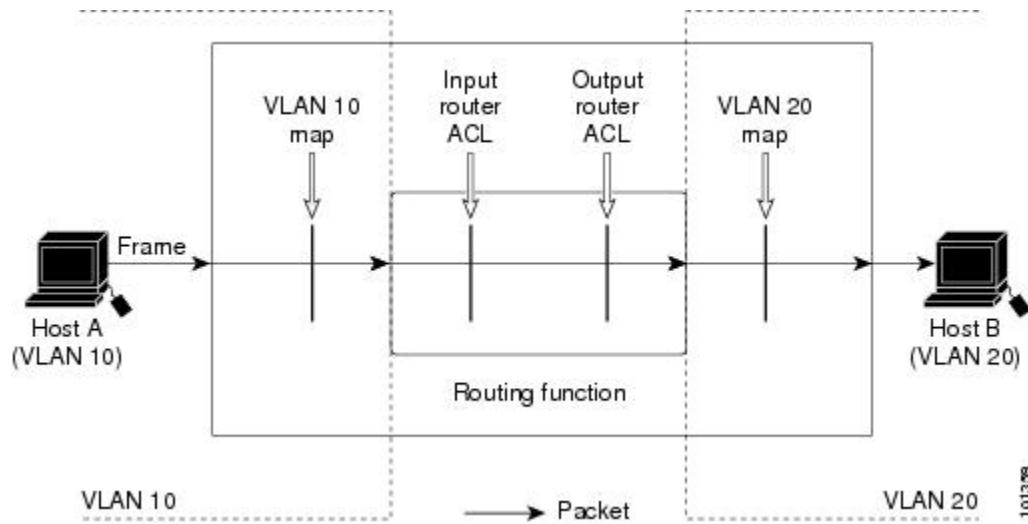
Example: ACLs and Multicast Packets

Figure 20: Applying ACLs on Multicast Packets

This example shows how ACLs are applied on packets that are replicated for IP multicasting. A multicast packet being routed has two different kinds of filters applied: one for destinations that are other ports in the input VLAN and another for each of the destinations that are in other VLANs to which the packet has been routed. The packet might be routed to more than one output VLAN, in which case a different router output ACL and VLAN map would apply for each destination VLAN. The final result is that the packet might be permitted in some of the output VLANs and not in others. A copy of the packet is forwarded to those destinations where it is permitted. However, if the input VLAN map drops the packet, no destination receives a copy of the packet.
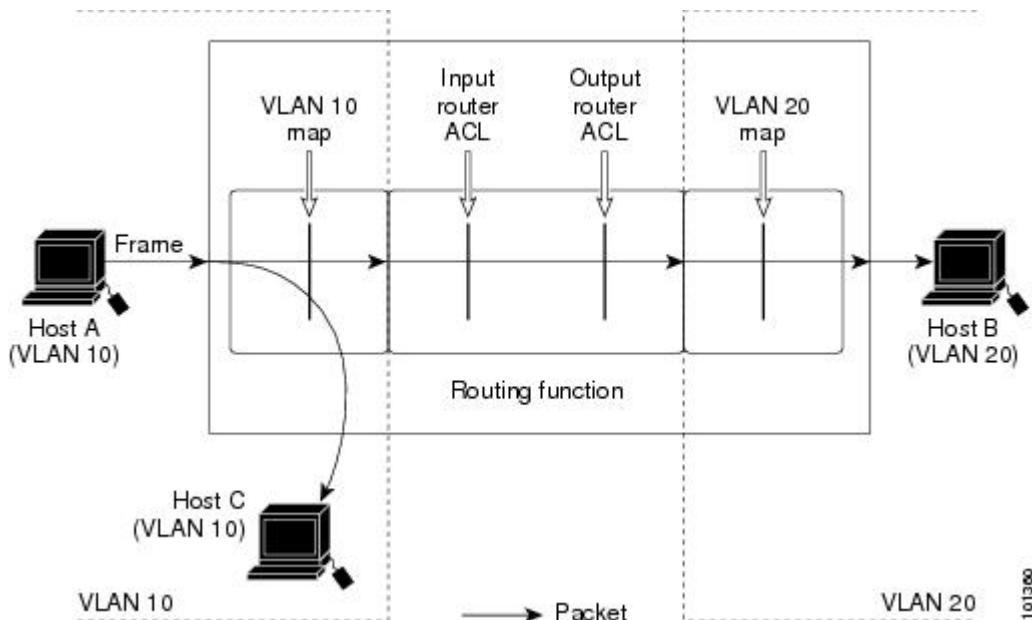
Feature History for IPv4 Access Control Lists

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	IPv4 Access Control Lists	This chapter describes how to configure network security on the switch by using ACLs. Packet filtering can help limit network traffic and restrict network use by certain users or devices. ACLs filter traffic as it passes through device and permit or deny packets crossing specified interfaces.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.
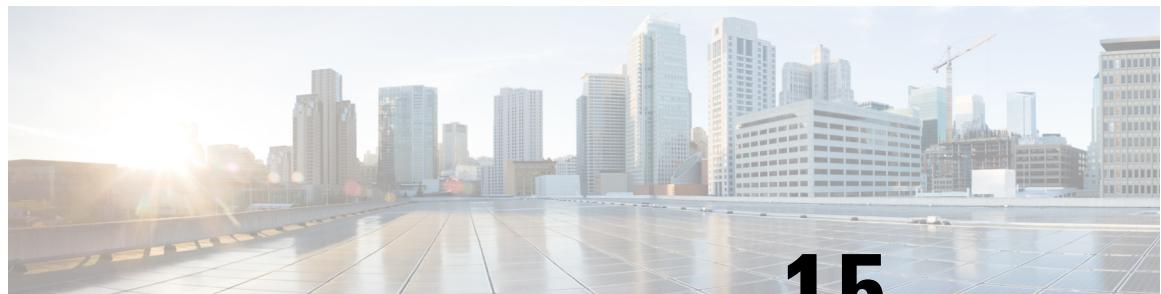
IPv6 ACLs

IPv6 ACLs Overview

You can filter IP Version 6 (IPv6) traffic by creating IPv6 access control lists (ACLs) and applying them to interfaces similar to how you create and apply IP Version 4 (IPv4) named ACLs. You can also create and apply input router ACLs to filter Layer 3 management traffic when the switch is running IP base and LAN base feature sets.

A switch supports three types of IPv6 ACLs:

- IPv6 router ACLs are supported on outbound or inbound traffic on Layer 3 interfaces, which can be routed ports, switch virtual interfaces (SVIs), or Layer 3 EtherChannels. IPv6 router ACLs apply only to IPv6 packets that are routed.

- IPv6 port ACLs are supported on outbound and inbound Layer 2 interfaces. IPv6 port ACLs are applied to all IPv6 packets entering the interface.

- VLAN ACLs or VLAN maps access-control all packets in a VLAN. You can use VLAN maps to filter traffic between devices in the same VLAN. ACL VLAN maps are applied on L2 VLANs. VLAN maps are configured to provide access control based on Layer 3 addresses for IPv6. Unsupported protocols are access-controlled through MAC addresses using Ethernet ACEs. After a VLAN map is applied to a VLAN, all packets entering the VLAN are checked against the VLAN map.

The switch supports VLAN ACLs (VLAN maps) for IPv6 traffic.

You can apply both IPv4 and IPv6 ACLs to an interface. As with IPv4 ACLs, IPv6 port ACLs take precedence over router ACLs.

Switch Stacks and IPv6 ACLs

The active switch supports IPv6 ACLs in hardware and distributes the IPv6 ACLs to the stack members.

If a standby switch takes over as the active switch, it distributes the ACL configuration to all stack members. The member switches sync up the configuration distributed by the new active switch and flush out entries that are not required.

When an ACL is modified, attached to, or detached from an interface, the active switch distributes the change to all stack members.

ACL Precedence

When Port ACLs, and router ACLs are configured on the same switch, the filtering precedence, from greatest to least for ingress traffic is port ACL, and then router ACL. For egress traffic, the filtering precedence is router ACL, and then port ACL.

The following examples describe simple use cases:

- When an input router ACL and input port ACL exist in a switch virtual interface (SVI), incoming packets received on ports to which a port ACL is applied are filtered by the port ACL. Incoming routed IP packets received on other ports are filtered by the router ACL. Other packets are not filtered.

- When an output router ACL and input port ACL exist in an SVI, incoming packets received on the ports to which a port ACL is applied are filtered by the port ACL. Outgoing routed IP packets are filtered by the router ACL. Other packets are not filtered.

VLAN Maps

VLAN ACLs or VLAN maps are used to control network traffic within a VLAN. You can apply VLAN maps to all packets that are bridged within a VLAN in the switch or switch stack. VACLs are strictly for security packet filtering and for redirecting traffic to specific physical interfaces. VACLs are not defined by direction (ingress or egress).

All non-IP protocols are access-controlled through MAC addresses and Ethertype using MAC VLAN maps. (IP traffic is not access controlled by MAC VLAN maps.) You can enforce VLAN maps only on packets going through the switch; you cannot enforce VLAN maps on traffic between hosts on a hub or on another switch connected to this switch.

With VLAN maps, forwarding of packets is permitted or denied, based on the action specified in the map.

Figure 21: Using VLAN Maps to Control Traffic

This figure shows how a VLAN map is applied to prevent a specific type of traffic from Host A in VLAN 10 from being forwarded. You can apply only one VLAN map to a VLAN.

X = VLAN map denying specific type of traffic from Host A
→ = Packet
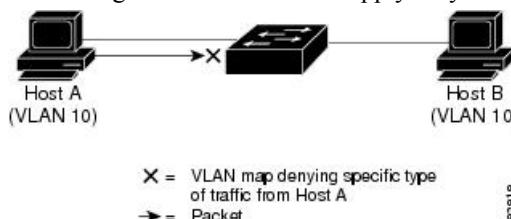
Interactions with Other Features and Switches

- If an IPv6 router ACL is configured to deny a packet, the packet is not routed. A copy of the packet is sent to the Internet Control Message Protocol (ICMP) queue to generate an ICMP unreachable message for the frame.

- If a bridged frame is to be dropped due to a port ACL, the frame is not bridged.

- You can create both IPv4 and IPv6 ACLs on a switch or switch stack, and you can apply both IPv4 and IPv6 ACLs to the same interface. Each ACL must have a unique name; an error message appears if you try to use a name that is already configured.

  You use different commands to create IPv4 and IPv6 ACLs and to attach IPv4 or IPv6 ACLs to the same Layer 2 or Layer 3 interface. If you use the wrong command to attach an ACL (for example, an IPv4 command to attach an IPv6 ACL), you receive an error message.

- You cannot use MAC ACLs to filter IPv6 frames. MAC ACLs can only filter non-IP frames.

- If the hardware memory is full, packets are dropped on the interface and an unload error message is logged.

Restrictions for IPv6 ACLs

With IPv4, you can configure standard and extended numbered IP ACLs, named IP ACLs, and MAC ACLs. IPv6 supports only named ACLs.

The switch supports most Cisco IOS-supported IPv6 ACLs with some exceptions:

- The switch does not support matching on these keywords: routing header, and undetermined-transport.

- The switch does not support reflexive ACLs (the reflect keyword).

- This release supports port ACLs, router ACLs and VLAN ACLs (VLAN maps) for IPv6.

- The switch does not apply MAC-based ACLs on IPv6 frames.

- When configuring an ACL, there is no restriction on keywords entered in the ACL, regardless of whether or not they are supported on the platform. When you apply the ACL to an interface that requires hardware forwarding (physical ports or SVIs), the switch checks to determine whether or not the ACL can be supported on the interface. If not, attaching the ACL is rejected.

- If an ACL is applied to an interface and you attempt to add an access control entry (ACE) with an unsupported keyword, the switch does not allow the ACE to be added to the ACL that is currently attached to the interface.

IPv6 ACLs on the switch have these characteristics:

- Fragmented frames (the fragments keyword as in IPv4) are supported

- The same statistics supported in IPv4 are supported for IPv6 ACLs.

- If the switch runs out of hardware space, the packets associated with the ACL are dropped on the interface.

- Logging is supported for router ACLs, but not for port ACLs.

- The switch supports IPv6 address-matching for a full range of prefix-lengths.

Default Configuration for IPv6 ACLs

The default IPv6 ACL configuration is as follows:

```
Switch# show access-lists preauth_ipv6_acl
IPv6 access list preauth_ipv6_acl (per-user)
permit udp any any eq domain sequence 10
permit tcp any any eq domain sequence 20
permit icmp any any nd-ns sequence 30
permit icmp any any nd-na sequence 40
permit icmp any any router-solicitation sequence 50
permit icmp any any router-advertisement sequence 60
permit icmp any any redirect sequence 70
permit udp any eq 547 any eq 546 sequence 80
permit udp any eq 546 any eq 547 sequence 90
deny ipv6 any any sequence 100
```

Configuring IPv6 ACLs

To filter IPv6 traffic, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	{ipv6 access-list list-name **Example:** Device(config)# **ipv6 access-list example_acl_list**	Defines an IPv6 ACL name, and enters IPv6 access list configuration mode.
Step 4	{deny | permit} protocol {source-ipv6-prefix/ |prefix-length |any| host source-ipv6-address} [operator [port-number]] { destination-ipv6-prefix/ prefix-length | any | host destination-ipv6-address} [operator [port-number]][dscp value] [fragments] [log] [log-input][sequence value] [time-range name]	Enter deny or permit to specify whether to deny or permit the packet if conditions are matched. These are the conditions: • For protocol, enter the name or number of an IP: ahp, esp, icmp, ipv6, pcp, stcp, tcp, or udp, or an integer in the range 0 to 255 representing an IPv6 protocol number.

Command or Action	Purpose
	• The source-ipv6-prefix/prefix-length or destination-ipv6-prefix/ prefix-length is the source or destination IPv6 network or class of networks for which to set deny or permit conditions, specified in hexadecimal and using 16-bit values between colons (see RFC 2373).
	• Enter any as an abbreviation for the IPv6 prefix ::/0.
	• For host source-ipv6-address or destination-ipv6-address, enter the source or destination IPv6 host address for which to set deny or permit conditions, specified in hexadecimal using 16-bit values between colons.
	• (Optional) For operator, specify an operand that compares the source or destination ports of the specified protocol. Operands are lt (less than), gt (greater than), eq (equal), neq (not equal), and range.
	If the operator follows the source-ipv6-prefix/prefix-length argument, it must match the source port. If the operator follows the destination-ipv6- prefix/prefix-length argument, it must match the destination port.
	• (Optional) The port-number is a decimal number from 0 to 65535 or the name of a TCP or UDP port. You can use TCP port names only when filtering TCP. You can use UDP port names only when filtering UDP.
	• (Optional) Enter dscp value to match a differentiated services code point value against the traffic class value in the Traffic Class field of each IPv6 packet header. The acceptable range is from 0 to 63.
	• (Optional) Enter fragments to check noninitial fragments. This keyword is visible only if the protocol is ipv6.
	• (Optional) Enter log to cause an logging message to be sent to the console about

	Command or Action	Purpose
		the packet that matches the entry. Enter log-input to include the input interface in the log entry. Logging is supported only for router ACLs.
		• (Optional) Enter sequence value to specify the sequence number for the access list statement. The acceptable range is from 1 to 4,294,967,295.
		• (Optional) Enter time-range name to specify the time range that applies to the deny or permit statement.
Step 5	{deny | permit} tcp {source-ipv6-prefix/prefix-length | any | host source-ipv6-address} [operator [port-number]] {destination-ipv6- prefix/prefix-length | any | host destination-ipv6-address} [operator [port-number]] [ack] [dscp value [established] [fin] [log] [log-input] [neq {port | protocol}] [psh] [range {port | protocol}] [rst] [sequence value] [syn] [time-range name] [urg]	(Optional) Define a TCP access list and the access conditions. Enter tcp for Transmission Control Protocol. The parameters are the same as those described in Step 3a, with these additional optional parameters: • ack: Acknowledgment bit set. • established: An established connection. A match occurs if the TCP datagram has the ACK or RST bits set. • fin: Finished bit set; no more data from sender. • neq { port |protocol}: Matches only packets that are not on a given port number. • psh—Push function bit set. • range { port |protocol}: Matches only packets in the port number range. • rst: Reset bit set. • syn: Synchronize bit set. • urg: Urgent pointer bit set.
Step 6	{deny | permit} udp {source-ipv6-prefix/prefix-length | any | host source-ipv6-address} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address} [operator [port-number]] [dscp value] [log] [log-input] [neq {port | protocol}] [range {port | protocol}] [sequence value] [time-range name]]	(Optional) Define a UDP access list and the access conditions. Enter udp for the User Datagram Protocol. The UDP parameters are the same as those described for TCP, except that the [operator [port]] port number or name must be a UDP

	Command or Action	**Purpose**
		port number or name, and the established parameter is not valid for UDP.
Step 7	{deny | permit} icmp {source-ipv6-prefix/prefix-length | any | host source-ipv6-address} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address} [operator [port-number]] [icmp-type [icmp-code] | icmp-message] [dscp value] [log] [log-input] [sequence value] [time-range name]	(Optional) Define an ICMP access list and the access conditions. Enter icmp for Internet Control Message Protocol. The ICMP parameters are the same as those described for most IP protocols in Step 1, with the addition of the ICMP message type and code parameters. These optional keywords have these meanings: - icmp-type: Enter to filter by ICMP message type, a number from 0 to 255. - icmp-code: Enter to filter ICMP packets that are filtered by the ICMP message code type, a number from 0 to 255. - icmp-message: Enter to filter ICMP packets by the ICMP message type name or the ICMP message type and code name. To see a list of ICMP message type names and code names, use the ? key or see command reference for this release.
Step 8	end	Return to privileged EXEC mode.
Step 9	show ipv6 access-list	Verify the access list configuration.
Step 10	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 11	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Attaching an IPv6 ACL to an Interface

You can apply an ACL to outbound or inbound traffic on Layer 3 interfaces, or to inbound traffic on Layer 2 interfaces. You can also apply ACLs only to inbound management traffic on Layer 3 interfaces.

Follow these steps to control access to an interface.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	interface interface-id	Identify a Layer 2 interface (for port ACLs) or Layer 3 interface (for router ACLs) on which to apply an access list, and enter interface configuration mode.
Step 4	no switchport	If applying a router ACL, this changes the interface from Layer 2 mode (the default) to Layer 3 mode.
Step 5	ipv6 address ipv6-address	Configure an IPv6 address on a Layer 3 interface (for router ACLs).
Step 6	ipv6 traffic-filter access-list-name {in | out}	Apply the access list to incoming or outgoing traffic on the interface.
Step 7	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 8	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 9	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring a VLAN Map

To create a VLAN map and apply it to one or more VLANs, perform these steps:

Before you begin

Create the IPv6 ACL that you want to apply to the VLAN.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	vlan access-map name [number] **Example:** Device(config)# **vlan access-map map_1 20**	Creates a VLAN map, and give it a name and (optionally) a number. The number is the sequence number of the entry within the map. When you create VLAN maps with the same name, numbers are assigned sequentially in increments of 10. When modifying or deleting maps, you can enter the number of the map entry that you want to modify or delete. VLAN maps do not use the specific permit or deny keywords. To deny a packet by using VLAN maps, create an ACL that would match the packet, and set the action to drop. A permit in the ACL counts as a match. A deny in the ACL means no match. Entering this command changes to access-map configuration mode.
Step 4	match {ip	ipv6	mac} address {name	number} [name	number] **Example:** Device(config-access-map)# **match ipv6 address ip_net**	Match the packet against one or more access lists. Note that packets are only matched against access lists of the correct protocol type. IP packets are matched against IP access lists. Non-IP packets are only matched against named MAC access lists.

	Command or Action	Purpose
		Note If the VLAN map is configured with a match clause for a type of packet (IP or MAC) and the map action is drop, all packets that match the type are dropped. If the VLAN map has no match clause, and the configured action is drop, all IP and Layer 2 packets are dropped.
Step 5	Enter one of the following commands to specify an IP packet or a non-IP packet (with only a known MAC address) and to match the packet against one or more ACLs: • action { forward} Device(config-access-map)# action forward • action { drop} Device(config-access-map)# action drop	Sets the action for the map entry.
Step 6	vlan filter mapname vlan-list list **Example:** Device(config)# **vlan filter map 1 vlan-list 20-22**	Applies the VLAN map to one or more VLAN IDs. The list can be a single VLAN ID (22), a consecutive list (10-22), or a string of VLAN IDs (12, 22, 30). Spaces around the comma and hyphen are optional.

Applying a VLAN Map to a VLAN

To apply a VLAN map to one or more VLANs, perform these steps.

Procedure

	Command or Action	Purpose
Step 1		
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	vlan filter mapname vlan-list list **Example:**	Applies the VLAN map to one or more VLAN IDs.

	Command or Action	Purpose
	Device(config)# **vlan filter map 1 vlan-list 20-22**	The list can be a single VLAN ID (22), a consecutive list (10-22), or a string of VLAN IDs (12, 22, 30). Spaces around the comma and hyphen are optional.
Step 4	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 5	show running-config **Example:** Device# **show running-config**	Displays the access list configuration.
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Monitoring IPv6 ACLs

You can display information about all configured access lists, all IPv6 access lists, or a specific access list by using one or more of the privileged EXEC commands shown in the table below:

Table 29: show ACL commands

Command	Purpose
show access-lists	Displays all access lists configured on the switch.
show ipv6 access-list [access-list-name]	Displays all configured IPv6 access lists or the access list specified by name.
show vlan access-map [map-name]	Displays VLAN access map configuration.
show vlan filter [access-map access-map | vlan vlan-id]	Displays the mapping between VACLs and VLANs.

This is an example of the output from the show access-lists privileged EXEC command. The output shows all access lists that are configured on the switch or switch stack.

```
Switch # show access-lists
Extended IP access list hello
    10 permit ip any any
```

```
IPv6 access list ipv6
    permit ipv6 any any sequence 10
```

This is an example of the output from the show ipv6 access-list privileged EXEC command. The output shows only IPv6 access lists configured on the switch or switch stack

```
Switch# show ipv6 access-list
IPv6 access list inbound
    permit tcp any any eq bgp (8 matches) sequence 10
    permit tcp any any eq telnet (15 matches) sequence 20
    permit udp any any sequence 30
IPv6 access list outbound
    deny udp any any sequence 10
    deny tcp any any eq telnet sequence 20
```

This is an example of the output from the show vlan access-map privileged EXEC command. The output shows VLAN access map information.

```
Switch# show vlan access-map
Vlan access-map "m1"  10
  Match clauses:
    ipv6 address: ip2
  Action: drop
```

Feature History for IPv6 ACLs

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	IPv6 ACLs	You can filter IPv6 traffic by creating IPv6 ACLs and applying them to interfaces similar to how you create and apply IPv4 named ACLs. You can also create and apply input router ACLs to filter Layer 3 management traffic.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.
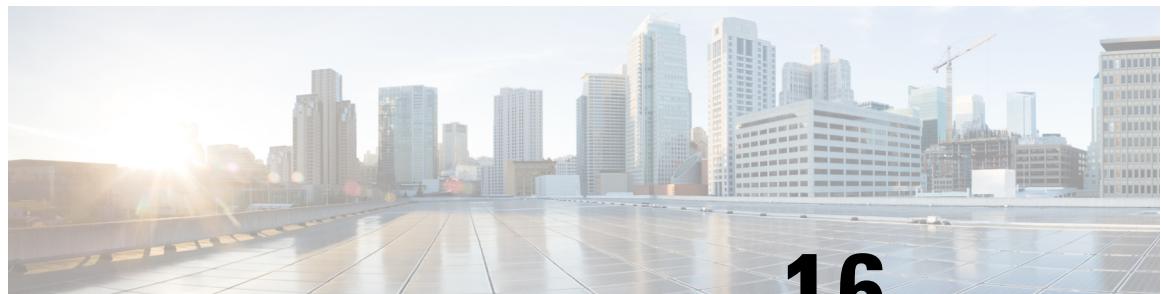
Configuring DHCP

Restrictions for Configuring DHCP

We recommend that you do not use transmit (Tx) Switched Port Analyzer (SPAN) or egress SPAN that supports DHCP Snooping, DHCP Relay Agent. If SPAN at Tx is required, avoid using VLAN ports that are in the forwarding path for DHCP packets.

Information About DHCP

DHCP Server

The DHCP server assigns IP addresses from specified address pools on a switch or router to DHCP clients and manages them. If the DHCP server cannot give the DHCP client the requested configuration parameters from its database, it forwards the request to one or more secondary DHCP servers defined by the network administrator. The switch can act as a DHCP server.

DHCP Relay Agent

A DHCP relay agent is a Layer 3 device that forwards DHCP packets between clients and servers. Relay agents forward requests and replies between clients and servers when they are not on the same physical subnet. Relay agent forwarding is different from the normal Layer 2 forwarding, in which IP datagrams are switched transparently between networks. Relay agents receive DHCP messages and generate new DHCP messages to send on output interfaces.

DHCP Snooping

DHCP snooping is a DHCP security feature that provides network security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding database, also referred to as a DHCP snooping binding table.

DHCP snooping acts like a firewall between untrusted hosts and DHCP servers. You use DHCP snooping to differentiate between untrusted interfaces connected to the end user and trusted interfaces connected to the DHCP server or another switch.

> **Note** For DHCP snooping to function properly, all DHCP servers must be connected to the switch through trusted interfaces.

An untrusted DHCP message is a message that is received through an untrusted interface. By default, the switch considers all interfaces untrusted. So, the switch must be configured to trust some interfaces to use DHCP Snooping. When you use DHCP snooping in a service-provider environment, an untrusted message is sent from a device that is not in the service-provider network, such as a customer's switch. Messages from unknown devices are untrusted because they can be sources of traffic attacks.

The DHCP snooping binding database has the MAC address, the IP address, the lease time, the binding type, the VLAN number, and the interface information that corresponds to the local untrusted interfaces of a switch. It does not have information regarding hosts interconnected with a trusted interface.

In a service-provider network, an example of an interface you might configure as trusted is one connected to a port on a device in the same network. An example of an untrusted interface is one that is connected to an untrusted interface in the network or to an interface on a device that is not in the network.

When a switch receives a packet on an untrusted interface and the interface belongs to a VLAN in which DHCP snooping is enabled, the switch compares the source MAC address and the DHCP client hardware address. If the addresses match (the default), the switch forwards the packet. If the addresses do not match, the switch drops the packet.

The switch drops a DHCP packet when one of these situations occurs:

- A packet from a DHCP server, such as a DHCPOFFER, DHCPACK, DHCPNAK, or DHCPLEASEQUERY packet, is received from outside the network or firewall.

- A packet is received on an untrusted interface, and the source MAC address and the DHCP client hardware address do not match.

- The switch receives a DHCPRELEASE or DHCPDECLINE broadcast message that has a MAC address in the DHCP snooping binding database, but the interface information in the binding database does not match the interface on which the message was received.

- A DHCP relay agent forwards a DHCP packet that includes a relay-agent IP address that is not 0.0.0.0, or the relay agent forwards a packet that includes option-82 information to an untrusted port.

- The maximum snooping queue size of 1000 is exceeded when DHCP snooping is enabled.

If the switch is an aggregation switch supporting DHCP snooping and is connected to an edge switch that is inserting DHCP option-82 information, the switch drops packets with option-82 information when packets are received on an untrusted interface. If DHCP snooping is enabled and packets are received on a trusted port, the aggregation switch does not learn the DHCP snooping bindings for connected devices and cannot build a complete DHCP snooping binding database.

When an aggregation switch can be connected to an edge switch through an untrusted interface and you enter the ip dhcp snooping information option allow-untrusted global configuration command, the aggregation switch accepts packets with option-82 information from the edge switch. The aggregation switch learns the bindings for hosts connected through an untrusted switch interface. The DHCP security features, such as dynamic ARP inspection or IP source guard, can still be enabled on the aggregation switch while the switch receives packets with option-82 information on untrusted input interfaces to which hosts are connected. The port on the edge switch that connects to the aggregation switch must be configured as a trusted interface.

Option-82 Data Insertion

In residential, metropolitan Ethernet-access environments, DHCP can centrally manage the IP address assignments for a large number of subscribers. When the DHCP option-82 feature is enabled on the switch, a subscriber device is identified by the switch port through which it connects to the network (in addition to its MAC address). Multiple hosts on the subscriber LAN can be connected to the same port on the access switch and are uniquely identified.

Note The DHCP option-82 feature is supported only when DHCP snooping is globally enabled on the VLANs to which subscriber devices using option-82 are assigned.

The following illustration shows a metropolitan Ethernet network in which a centralized DHCP server assigns IP addresses to subscribers connected to the switch at the access layer. Because the DHCP clients and their associated DHCP server do not reside on the same IP network or subnet, a DHCP relay agent (the Catalyst switch) is configured with a helper address to enable broadcast forwarding and to transfer DHCP messages between the clients and the server.

Figure 22: DHCP Relay Agent in a Metropolitan Ethernet Network
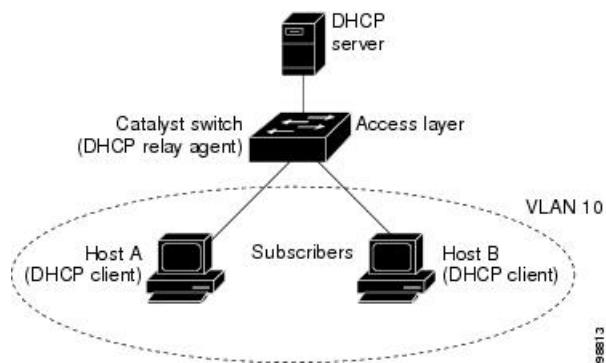
When you enable the DHCP snooping information option 82 on the switch, the following sequence of events occurs:

- The host (DHCP client) generates a DHCP request and broadcasts it on the network.

- When the switch receives the DHCP request, it adds the option-82 information in the packet. By default, the remote-ID suboption is the switch MAC address, and the circuit-ID suboption is the port identifier, vlan-mod-port, from which the packet is received.You can configure the remote ID and circuit ID.

- If the IP address of the relay agent is configured, the switch adds this IP address in the DHCP packet.

- The switch forwards the DHCP request that includes the option-82 field to the DHCP server.

- The DHCP server receives the packet. If the server is option-82-capable, it can use the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. Then the DHCP server echoes the option-82 field in the DHCP reply.

- The DHCP server unicasts the reply to the switch if the request was relayed to the server by the switch. The switch verifies that it originally inserted the option-82 data by inspecting the remote ID and possibly the circuit ID fields. The switch removes the option-82 field and forwards the packet to the switch port that connects to the DHCP client that sent the DHCP request.

In the default suboption configuration, when the described sequence of events occurs, the values in these fields do not change (see the illustration,Suboption Packet Formats):

- Circuit-ID suboption fields

    - Suboption type

    - Length of the suboption type

    - Circuit-ID type

    - Length of the circuit-ID type

- Remote-ID suboption fields

    - Suboption type

    - Length of the suboption type

    - Remote-ID type

    - Length of the remote-ID type

In the port field of the circuit ID suboption, the port numbers start at 3. For example, on a switch with 24 10/100/1000 ports and four small form-factor pluggable (SFP) module slots, port 3 is the Gigabit Ethernet 1/0/1 port, port 4 is the Gigabit Ethernet 1/0/2 port, and so forth. Port 27 is the SFP module slot Gigabit Ethernet1/0/25, and so forth.

The illustration, Suboption Packet Formats. shows the packet formats for the remote-ID suboption and the circuit-ID suboption when the default suboption configuration is used. For the circuit-ID suboption, the module number corresponds to the switch number in the stack. The switch uses the packet formats when you globally enable DHCP snooping and enter the ip dhcp snooping information option global configuration command.

Figure 23: Suboption Packet Formats
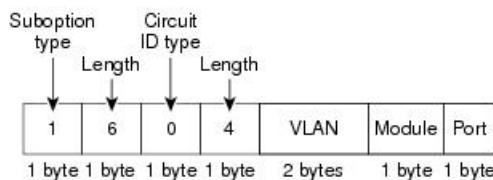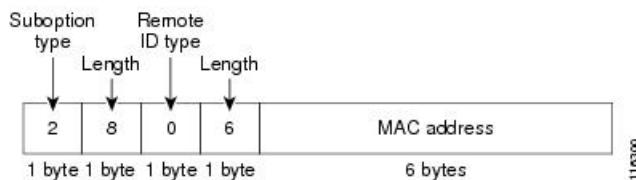
The illustration, User-Configured Suboption Packet Formats, shows the packet formats for user-configured remote-ID and circuit-ID suboptions The switch uses these packet formats when DHCP snooping is globally enabled and when the ip dhcp snooping information option format remote-id global configuration command and theip dhcp snooping vlan information option format-type circuit-id string interface configuration command are entered.

The values for these fields in the packets change from the default values when you configure the remote-ID and circuit-ID suboptions:

- Circuit-ID suboption fields

  - The circuit-ID type is 1.

  - The length values are variable, depending on the length of the string that you configure.

- Remote-ID suboption fields

  - The remote-ID type is 1.

  - The length values are variable, depending on the length of the string that you configure.

Figure 24: User-Configured Suboption Packet Formats

Circuit ID Suboption Frame Format (for user-configured string):
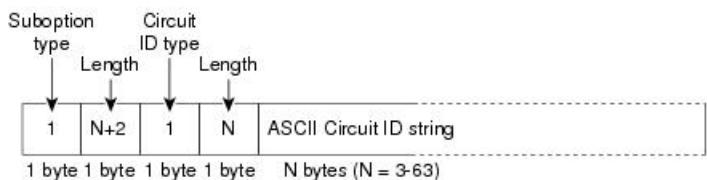
Remote ID Suboption Frame Format (for user-configured string):
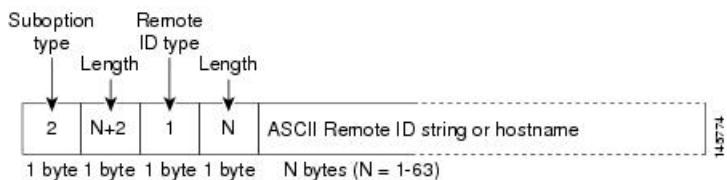
Cisco IOS DHCP Server Database

During the DHCP-based autoconfiguration process, the designated DHCP server uses the Cisco IOS DHCP server database. It has IP addresses, address bindings, and configuration parameters, such as the boot file.

An address binding is a mapping between an IP address and a MAC address of a host in the Cisco IOS DHCP server database. You can manually assign the client IP address, or the DHCP server can allocate an IP address from a DHCP address pool. For more information about manual and automatic address bindings, see the "Configuring DHCP" chapter of the Cisco IOS IP Configuration Guide, Release 12.4.

For procedures to enable and configure the Cisco IOS DHCP server database, see the "DHCP Configuration Task List" section in the "Configuring DHCP" chapter of the Cisco IOS IP Configuration Guide, Release 12.4.

DHCP Snooping Binding Database

When DHCP snooping is enabled, the switch uses the DHCP snooping binding database to store information about untrusted interfaces. The database can have up to 64,000 bindings.

Each database entry (binding) has an IP address, an associated MAC address, the lease time (in hexadecimal format), the interface to which the binding applies, and the VLAN to which the interface belongs. The database agent stores the bindings in a file at a configured location. At the end of each entry is a checksum that accounts for all the bytes from the start of the file through all the bytes associated with the entry. Each entry is 72 bytes, followed by a space and then the checksum value.

To keep the bindings when the switch reloads, you must use the DHCP snooping database agent. If the agent is disabled, dynamic ARP inspection or IP source guard is enabled, and the DHCP snooping binding database has dynamic bindings, the switch loses its connectivity. If the agent is disabled and only DHCP snooping is enabled, the switch does not lose its connectivity, but DHCP snooping might not prevent DHCP spoofing attacks.

When reloading, the switch reads the binding file to build the DHCP snooping binding database. The switch updates the file when the database changes.

When a switch learns of new bindings or when it loses bindings, the switch immediately updates the entries in the database. The switch also updates the entries in the binding file. The frequency at which the file is

updated is based on a configurable delay, and the updates are batched. If the file is not updated in a specified time (set by the write-delay and cancel-timeout values), the update stops.

This is the format of the file with bindings:

```
<initial-checksum>
TYPE DHCP-SNOOPING
VERSION 1
BEGIN
<entry-1> <checksum-1>
<entry-2> <checksum-1-2>
...
...
<entry-n> <checksum-1-2-..-n>
END
```

Each entry in the file is tagged with a checksum value that the switch uses to verify the entries when it reads the file. The initial-checksum entry on the first line distinguishes entries associated with the latest file update from entries associated with a previous file update.

This is an example of a binding file:

```
2bb4c2a1
TYPE DHCP-SNOOPING
VERSION 1
BEGIN
192.1.168.1 3 0003.47d8.c91f 2BB6488E Gi1/0/4 21ae5fbb
192.1.168.3 3 0003.44d6.c52f 2BB648EB Gi1/0/4 1bdb223f
192.1.168.2 3 0003.47d9.c8f1 2BB648AB Gi1/0/4 584a38f0
END
```

When the switch starts and the calculated checksum value equals the stored checksum value, the switch reads entries from the binding file and adds the bindings to its DHCP snooping binding database. The switch ignores an entry when one of these situations occurs:

- The switch reads the entry and the calculated checksum value does not equal the stored checksum value. The entry and the ones following it are ignored.

- An entry has an expired lease time (the switch might not remove a binding entry when the lease time expires).

- The interface in the entry no longer exists on the system.

- The interface is a routed interface or a DHCP snooping-trusted interface.

DHCP Snooping and Switch Stacks

DHCP snooping is managed on the active switch. When a new switch joins the stack, the switch receives the DHCP snooping configuration from the active switch. When a member switch leaves the stack, all DHCP snooping address bindings associated with the switch age out.

All snooping statistics are generated on the active switch. If a new active switch is elected, the statistics counters reset.

When a stack merge occurs, all DHCP snooping bindings in the active switch are lost if it is no longer the active switch. With a stack partition, the existing active switch is unchanged, and the bindings belonging to

the partitioned switches age out. The new active switch of the partitioned stack begins processing the new incoming DHCP packets.

How to Configure DHCP Features

Default DHCP Snooping Configuration

Table 30: Default DHCP Configuration

Feature	Default Setting
DHCP server	Enabled in Cisco IOS software, requires configuration[6]
DHCP relay agent	Enabled[7]
DHCP packet forwarding address	None configured
Checking the relay agent information	Enabled (invalid messages are dropped)
DHCP relay agent forwarding policy	Replace the existing relay agent information
DHCP snooping enabled globally	Disabled
DHCP snooping information option	Enabled
DHCP snooping option to accept packets on untrusted input interfaces[8]	Disabled
DHCP snooping limit rate	None configured
DHCP snooping trust	Untrusted
DHCP snooping VLAN	Disabled
DHCP snooping MAC address verification	Enabled
Cisco IOS DHCP server binding database	Enabled in Cisco IOS software, requires configuration. **Note** The switch gets network addresses and configuration parameters only from a device configured as a DHCP server.
DHCP snooping binding database agent	Enabled in Cisco IOS software, requires configuration. This feature is operational only when a destination is configured.

[6] The switch responds to DHCP requests only if it is configured as a DHCP server.
[7] The switch relays DHCP packets only if the IP address of the DHCP server is configured on the SVI of the DHCP client.

[8] Use this feature when the switch is an aggregation switch that receives packets with option-82 information from an edge switch.

DHCP Snooping Configuration Guidelines

- If a switch port is connected to a DHCP server, configure a port as trusted by entering the ip dhcp snooping trust interface configuration command.

- If a switch port is connected to a DHCP client, configure a port as untrusted by entering the no ip dhcp snooping trust interface configuration command.

- You can display DHCP snooping statistics by entering the show ip dhcp snooping statistics user EXEC command, and you can clear the snooping statistics counters by entering the clear ip dhcp snooping statistics privileged EXEC command.

Configuring the DHCP Server

The switch can act as a DHCP server.

For procedures to configure the switch as a DHCP server, see the "Configuring DHCP" section of the "IP addressing and Services" section of the Cisco IOS IP Configuration Guide, Release 12.4.

DHCP Server and Switch Stacks

The DHCP binding database is managed on the stack's active switch. When a new active switch is assigned, the new active switch downloads the saved binding database from the TFTP server. When a switchover happens, the new active switch stack will use its database file that has been synced from the old active switch stack using the SSO function. The IP addresses associated with the lost bindings are released. You should configure an automatic backup by using the ip dhcp database url [timeout seconds | write-delay seconds] global configuration command.

Configuring the DHCP Relay Agent

Follow these steps to enable the DHCP relay agent on the switch:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.

	Command or Action	**Purpose**
Step 3	service dhcp **Example:** Device(config)# **service dhcp**	Enables the DHCP server and relay agent on your switch. By default, this feature is enabled.
Step 4	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 5	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

What to do next

- Checking (validating) the relay agent information

- Configuring the relay agent forwarding policy

Specifying the Packet Forwarding Address

If the DHCP server and the DHCP clients are on different networks or subnets, you must configure the switch with the ip helper-address address interface configuration command. The general rule is to configure the command on the Layer 3 interface closest to the client. The address used in the ip helper-address command can be a specific DHCP server IP address, or it can be the network address if other DHCP servers are on the destination network segment. Using the network address enables any DHCP server to respond to requests.

Beginning in privileged EXEC mode, follow these steps to specify the packet forwarding address:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. - Enter your password if prompted.

	Command or Action	**Purpose**
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	interface vlan vlan-id **Example:** Device(config)# interface vlan 1	Creates a switch virtual interface by entering a VLAN ID, and enter interface configuration mode.
Step 4	ip address ip-address subnet-mask **Example:** Device(config-if)# ip address 192.108.1.27 255.255.255.0	Configures the interface with an IP address and an IP subnet.
Step 5	ip helper-address address **Example:** Device(config-if)# ip helper-address 172.16.1.2	Specifies the DHCP packet forwarding address. The helper address can be a specific DHCP server address, or it can be the network address if other DHCP servers are on the destination network segment. Using the network address enables other servers to respond to DHCP requests. If you have multiple servers, you can configure one helper address for each server.
Step 6	end **Example:** Device(config-if)# end	Returns to global configuration mode.

Step 7	Use one of the following:
	
	    • interface range port-range
	    • interface interface-id
	
	Example:
	
	Device(config)# interface
	gigabitethernet1/0/2
		Configures multiple physical ports that are connected to the DHCP clients, and enter interface range configuration mode.
		
		or
		
		Configures a single physical port that is connected to the DHCP client, and enter interface configuration mode.

Step 8	switchport mode access **Example:** Device(config-if)# switchport mode access	Defines the VLAN membership mode for the port.

	Command or Action	**Purpose**
Step 9	switchport access vlan vlan-id **Example:** Device(config-if)# switchport access vlan 1	Assigns the ports to the same VLAN as configured in Step 2.
Step 10	end **Example:** Device(config-if)# end	Returns to privileged EXEC mode.
Step 11	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 12	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Prerequisites for Configuring DHCP Snooping and Option 82

The prerequisites for DHCP Snooping and Option 82 are as follows:

- You must globally enable DHCP snooping on the switch.

- Before globally enabling DHCP snooping on the switch, make sure that the devices acting as the DHCP server and the DHCP relay agent are configured and enabled.

- If you want the switch to respond to DHCP requests, it must be configured as a DHCP server.

- Before configuring the DHCP snooping information option on your switch, be sure to configure the device that is acting as the DHCP server. You must specify the IP addresses that the DHCP server can assign or exclude, or you must configure DHCP options for these devices.

- For DHCP snooping to function properly, all DHCP servers must be connected to the switch through trusted interfaces. In a service-provider network, a trusted interface is connected to a port on a device in the same network.

- You must configure the switch to use the Cisco IOS DHCP server binding database to use it for DHCP snooping.

- To use the DHCP snooping option of accepting packets on untrusted inputs, the switch must be an aggregation switch that receives packets with option-82 information from an edge switch.

- The following prerequisites apply to DHCP snooping binding database configuration:

• You must configure a destination on the DHCP snooping binding database to use the switch for DHCP snooping.

• Because both NVRAM and the flash memory have limited storage capacity, we recommend that you store the binding file on a TFTP server.

• For network-based URLs (such as TFTP and FTP), you must create an empty file at the configured URL before the switch can write bindings to the binding file at that URL. See the documentation for your TFTP server to determine whether you must first create an empty file on the server; some TFTP servers cannot be configured this way.

• To ensure that the lease time in the database is accurate, we recommend that you enable and configure Network Time Protocol (NTP).

• If NTP is configured, the switch writes binding changes to the binding file only when the switch system clock is synchronized with NTP.

• Before configuring the DHCP relay agent on your switch, make sure to configure the device that is acting as the DHCP server. You must specify the IP addresses that the DHCP server can assign or exclude, configure DHCP options for devices, or set up the DHCP database agent.

• If you want the switch to relay DHCP packets, the IP address of the DHCP server must be configured on the switch virtual interface (SVI) of the DHCP client.

• If a switch port is connected to a DHCP server, configure a port as trusted by entering the ip dhcp snooping trust interface configuration command.

• If a switch port is connected to a DHCP client, configure a port as untrusted by entering the no ip dhcp snooping trust interface configuration command.

Enabling the Cisco IOS DHCP Server Database

For procedures to enable and configure the Cisco IOS DHCP server database, see the "DHCP Configuration Task List" section in the "Configuring DHCP" chapter of the Cisco IOS IP Configuration Guide, Release 12.4

Monitoring DHCP Snooping Information

Table 31: Commands for Displaying DHCP Information

show ip dhcp snooping	Displays the DHCP snooping configuration for a switch
show ip dhcp snooping binding	Displays only the dynamically configured bindings in the DHCP snooping binding database, also referred to as a binding table.
show ip dhcp snooping database	Displays the DHCP snooping binding database status and statistics.
show ip dhcp snooping statistics	Displays the DHCP snooping statistics in summary or detail form.

show ip source binding	Display the dynamically and statically configured bindings.

Note   If DHCP snooping is enabled and an interface changes to the down state, the switch does not delete the statically configured bindings.

Configuring DHCP Server Port-Based Address Allocation

DHCP Server Port-Based Address Allocation

DHCP server port-based address allocation is a feature that enables DHCP to maintain the same IP address on an Ethernet switch port regardless of the attached device client identifier or client hardware address.

When Ethernet switches are deployed in the network, they offer connectivity to the directly connected devices. In some environments, such as on a factory floor, if a device fails, the replacement device must be working immediately in the existing network. With the current DHCP implementation, there is no guarantee that DHCP would offer the same IP address to the replacement device. Control, monitoring, and other software expect a stable IP address associated with each device. If a device is replaced, the address assignment should remain stable even though the DHCP client has changed.

When configured, the DHCP server port-based address allocation feature ensures that the same IP address is always offered to the same connected port even as the client identifier or client hardware address changes in the DHCP messages received on that port. The DHCP protocol recognizes DHCP clients by the client identifier option in the DHCP packet. Clients that do not include the client identifier option are identified by the client hardware address. When you configure this feature, the port name of the interface overrides the client identifier or hardware address and the actual point of connection, the switch port, becomes the client identifier.

In all cases, by connecting the Ethernet cable to the same port, the same IP address is allocated through DHCP to the attached device.

The DHCP server port-based address allocation feature is only supported on a Cisco IOS DHCP server and not a third-party server.

Default Port-Based Address Allocation Configuration

By default, DHCP server port-based address allocation is disabled.

Port-Based Address Allocation Configuration Guidelines

• By default, DHCP server port-based address allocation is disabled.

• To restrict assignments from the DHCP pool to preconfigured reservations (unreserved addresses are not offered to the client and other clients are not served by the pool), you can enter the reserved-only DHCP pool configuration command.

Enabling the DHCP Snooping Binding Database Agent

Beginning in privileged EXEC mode, follow these steps to enable and configure the DHCP snooping binding database agent on the switch:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip dhcp snooping database {flash[number]:/filename | ftp://user:password@host/filename | http://[[username:password]@]{hostname | host-ip}[/directory] /image-name.tar | rcp://user@host/filename}| tftp://host/filename **Example:** Device(config)# **ip dhcp snooping database tftp://10.90.90.90/snooping-rp2**	Specifies the URL for the database agent or the binding file by using one of these forms: • flash[number]:/filename (Optional) Use the number parameter to specify the stack member number of the active switch. The range for number is 1 to 9. • ftp://user:password@host/filename • http://[[username:password]@]{hostname | host-ip}[/directory] /image-name.tar • rcp://user@host/filename • tftp://host/filename
Step 4	ip dhcp snooping database timeout seconds **Example:** Device(config)# **ip dhcp snooping database timeout 300**	Specifies (in seconds) how long to wait for the database transfer process to finish before stopping the process. The default is 300 seconds. The range is 0 to 86400. Use 0 to define an infinite duration, which means to continue trying the transfer indefinitely.
Step 5	ip dhcp snooping database write-delay seconds **Example:** Device(config)# **ip dhcp snooping database write-delay 15**	Specifies the duration for which the transfer should be delayed after the binding database changes. The range is from 15 to 86400 seconds. The default is 300 seconds (5 minutes).

	Command or Action	**Purpose**
Step 6	end	Returns to privileged EXEC mode.
	Example:	
	Device(config)# **end**	
Step 7	ip dhcp snooping binding mac-address vlan vlan-id ip-address interface interface-id expiry seconds	(Optional) Adds binding entries to the DHCP snooping binding database. The vlan-id range is from 1 to 4904. The seconds range is from 1 to 4294967295.
	Example:	Enter this command for each entry that you add.
	Device# **ip dhcp snooping binding 0001.1234.1234 vlan 1 172.20.50.5 interface gi1/1 expiry 1000**	Use this command when you are testing or debugging the switch.
Step 8	show ip dhcp snooping database [detail]	Displays the status and statistics of the DHCP snooping binding database agent.
	Example:	
	Device# show ip dhcp snooping database detail	
Step 9	show running-config	Verifies your entries.
	Example:	
	Device# **show running-config**	
Step 10	copy running-config startup-config	(Optional) Saves your entries in the configuration file.
	Example:	
	Device# **copy running-config startup-config**	

Enabling DHCP Server Port-Based Address Allocation

Follow these steps to globally enable port-based address allocation and to automatically generate a subscriber identifier on an interface.

Procedure

	Command or Action	**Purpose**
Step 1	enable	Enables privileged EXEC mode.
	Example:	• Enter your password if prompted.

	Command or Action	Purpose
	Device> **enable**	
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip dhcp use subscriber-id client-id **Example:** Device(config)# **ip dhcp use subscriber-id client-id**	Configures the DHCP server to globally use the subscriber identifier as the client identifier on all incoming DHCP messages.
Step 4	ip dhcp subscriber-id interface-name **Example:** Device(config)# **ip dhcp subscriber-id interface-name**	Automatically generates a subscriber identifier based on the short name of the interface. A subscriber identifier configured on a specific interface takes precedence over this command.
Step 5	interface interface-id **Example:** Device(config)# **interface gigabitethernet1/0/1**	Specifies the interface to be configured, and enter interface configuration mode.
Step 6	ip dhcp server use subscriber-id client-id **Example:** Device(config-if)# **ip dhcp server use subscriber-id client-id**	Configures the DHCP server to use the subscriber identifier as the client identifier on all incoming DHCP messages on the interface.
Step 7	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 8	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 9	copy running-config startup-config **Example:** Device# **copy running-config**	(Optional) Saves your entries in the configuration file.

Command or Action	Purpose
`startup-config`	

What to do next

After enabling DHCP port-based address allocation on the switch, use the ip dhcp pool global configuration command to preassign IP addresses and to associate them to clients.

Monitoring DHCP Server Port-Based Address Allocation

Table 32: Commands for Displaying DHCP Port-Based Address Allocation Information

Command	Purpose
show interface interface id	Displays the status and configuration of a specific interface.
show ip dhcp pool	Displays the DHCP address pools.
show ip dhcp binding	Displays address bindings on the Cisco IOS DHCP server.

Feature Information for DHCP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 33: Feature Information for DHCP

Feature Name	Releases	Feature Information
DHCP	Cisco IOS XE Everest 16.6.1	DHCP provides configuration parameters to Internet hosts. DHCP consists of two components: a protocol for delivering host-specific configuration parameters from a DHCP Server to a host and a mechanism for allocating network addresses to hosts. DHCP is built on a client/server model, where designated DHCP Server hosts allocate network addresses and deliver configuration parameters to dynamically configured hosts.

Feature Name	Releases	Feature Information
DHCP Client Option 12	Cisco IOS XE Fuji 16.8.1a	The DHCP Client Option 12 feature specifies the hostname of the client. While acquiring an IP address for an interface from the Dynamic Host Configuration Protocol (DHCP) server, if the client device receives the DHCP Hostname option inside the response, the hostname from that option is set. DHCP is used by DHCP clients to obtain configuration information for operation in an IP network.
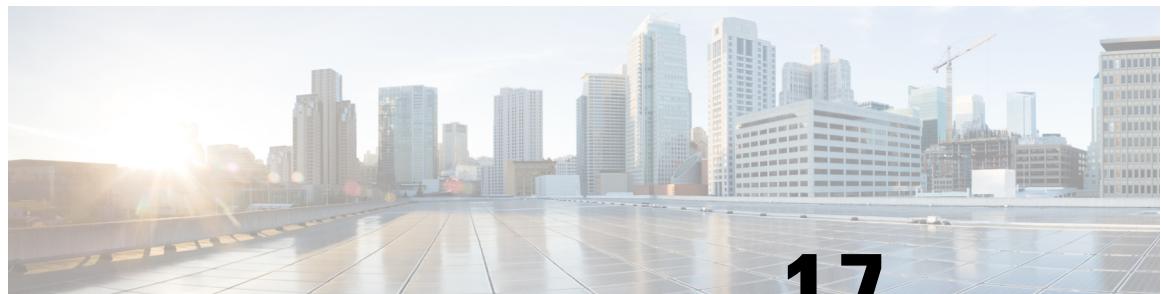
Configuring IP Source Guard

Information About IP Source Guard

IP Source Guard

You can use IP source guard to prevent traffic attacks if a host tries to use the IP address of its neighbor and you can enable IP source guard when DHCP snooping is enabled on an untrusted interface.

After IPSG is enabled on an interface, the switch blocks all IP traffic received on the interface except for DHCP packets allowed by DHCP snooping.

The switch uses a source IP lookup table in hardware to bind IP addresses to ports. For IP and MAC filtering, a combination of source IP and source MAC lookups are used. IP traffic with a source IP address in the binding table is allowed, all other traffic is denied.

The IP source binding table has bindings that are learned by DHCP snooping or are manually configured (static IP source bindings). An entry in this table has an IP address, its associated MAC address, and its associated VLAN number. The switch uses the IP source binding table only when IP source guard is enabled.

IPSG is supported only on Layer 2 ports, including access and trunk ports. You can configure IPSG with source IP address filtering or with source IP and MAC address filtering.

IP Source Guard for Static Hosts

Note Do not use IPSG (IP source guard) for static hosts on uplink ports or trunk ports.

IPSG for static hosts extends the IPSG capability to non-DHCP and static environments. The previous IPSG used the entries created by DHCP snooping to validate the hosts connected to a switch. Any traffic received from a host without a valid DHCP binding entry is dropped. This security feature restricts IP traffic on nonrouted Layer 2 interfaces. It filters traffic based on the DHCP snooping binding database and on manually

configured IP source bindings. The previous version of IPSG required a DHCP environment for IPSG to work.

IPSG for static hosts allows IPSG to work without DHCP. IPSG for static hosts relies on IP device tracking-table entries to install port ACLs. The switch creates static entries based on ARP requests or other IP packets to maintain the list of valid hosts for a given port. You can also specify the number of hosts allowed to send traffic to a given port. This is equivalent to port security at Layer 3.

IPSG for static hosts also supports dynamic hosts. If a dynamic host receives a DHCP-assigned IP address that is available in the IP DHCP snooping table, the same entry is learned by the IP device tracking table. In a stacked environment, when the active switch failover occurs, the IP source guard entries for static hosts attached to member ports are retained. When you enter the show device-tracking databaseEXEC command, the IP device tracking table displays the entries as ACTIVE.

> **Note** Some IP hosts with multiple network interfaces can inject some invalid packets into a network interface. The invalid packets contain the IP or MAC address for another network interface of the host as the source address. The invalid packets can cause IPSG for static hosts to connect to the host, to learn the invalid IP or MAC address bindings, and to reject the valid bindings. Consult the vender of the corresponding operating system and the network interface to prevent the host from injecting invalid packets.

IPSG for static hosts initially learns IP or MAC bindings dynamically through an ACL-based snooping mechanism. IP or MAC bindings are learned from static hosts by ARP and IP packets. They are stored in the device tracking database. When the number of IP addresses that have been dynamically learned or statically configured on a given port reaches a maximum, the hardware drops any packet with a new IP address. To resolve hosts that have moved or gone away for any reason, IPSG for static hosts leverages IP device tracking to age out dynamically learned IP address bindings. This feature can be used with DHCP snooping. Multiple bindings are established on a port that is connected to both DHCP and static hosts. For example, bindings are stored in both the device tracking database as well as in the DHCP snooping binding database.

IP Source Guard Configuration Guidelines

- You can configure static IP bindings only on nonrouted ports. If you enter the ip source binding mac-address vlan vlan-id ip-address interface interface-id global configuration command on a routed interface, this error message appears:

  ```
  Static IP source binding can only be configured on switch port.
  ```

- When IP source guard with source IP filtering is enabled on an interface, DHCP snooping must be enabled on the access VLAN for that interface.

- If you are enabling IP source guard on a trunk interface with multiple VLANs and DHCP snooping is enabled on all the VLANs, the source IP address filter is applied on all the VLANs.

  > **Note** If IP source guard is enabled and you enable or disable DHCP snooping on a VLAN on the trunk interface, the switch might not properly filter traffic.

- You can enable this feature when 802.1x port-based authentication is enabled.

How to Configure IP Source Guard

Enabling IP Source Guard

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	interface interface-id **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specifies the interface to be configured, and enters interface configuration mode.
Step 4	ip verify source [mac-check] **Example:** Device(config-if)# **ip verify source**	Enables IP source guard with source IP address filtering. (Optional) mac-check—Enables IP Source Guard with source IP address and MAC address filtering.
Step 5	exit **Example:** Device(config-if)# **exit**	Returns to global configuration mode.
Step 6	ip source binding mac-address vlan vlan-id ip-address interface interface-id **Example:** Device(config)# **ip source binding 0100.0230.0002 vlan 11 10.0.0.4 interface gigabitethernet1/0/1**	Adds a static IP source binding. Enter this command for each static binding.

	Command or Action	Purpose
Step 7	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 8	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 9	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring IP Source Guard for Static Hosts on a Layer 2 Access Port

You must configure the ip device tracking maximum limit-number interface configuration command globally for IPSG for static hosts to work. If you only configure this command on a port without enabling IP device tracking globally or by setting an IP device tracking maximum on that interface, IPSG with static hosts rejects all the IP traffic from that interface.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip device tracking **Example:** Device(config)# **ip device tracking**	Turns on the IP host table, and globally enables IP device tracking.

		Command or Action	**Purpose**
Step 4		interface interface-id	Enters interface configuration mode.
		Example:	
		Device(config)# **interface gigabitethernet 1/0/1**	
Step 5		switchport mode access	Configures a port as access.
		Example:	
		Device(config-if)# **switchport mode access**	
Step 6		switchport access vlan vlan-id	Configures the VLAN for this port.
		Example:	
		Device(config-if)# **switchport access vlan 10**	
Step 7		ip device tracking maximum number	Establishes a maximum limit for the number of static IPs that the IP device tracking table allows on the port. The range is 1to 10. The maximum number is 10.
		Example:	
		Device(config-if)# **ip device tracking maximum 8**	**Note** You must configure the ip device tracking maximum limit-number interface configuration command.
Step 8		end	Returns to privileged EXEC mode.
		Example:	
		Device(config)# **end**	

Monitoring IP Source Guard

Table 34: Privileged EXEC show Commands

Command	**Purpose**
show ip verify source [interface interface-id]	Displays the IP source guard configuration on the switch or on a specific interface.
show ip device tracking { all | interface interface-id | ip ip-address | mac mac-address}	Displays information about the entries in the IP device tracking table.

Table 35: Interface Configuration Commands

Command	Purpose
ip verify source tracking	Verifies the data source.

For detailed information about the fields in these displays, see the command reference for this release.

Feature History for IP Source Guard

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	IP Source Guard	You can use IP source guard to prevent traffic attacks if a host tries to use the IP address of its neighbor and you can enable IP source guard when DHCP snooping is enabled on an untrusted interface.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.
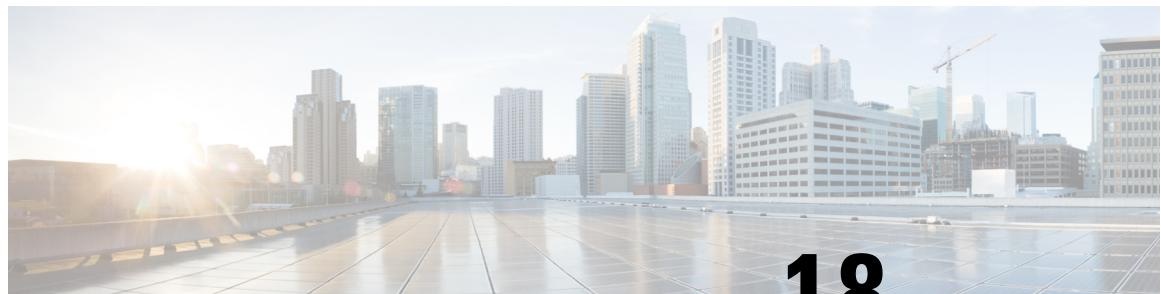
CHAPTER 18

Configuring Dynamic ARP Inspection

Restrictions for Dynamic ARP Inspection

This section lists the restrictions and guidelines for configuring Dynamic ARP Inspection on the switch.

- Dynamic ARP inspection is an ingress security feature; it does not perform any egress checking.

- Dynamic ARP inspection is not effective for hosts connected to switches that do not support dynamic ARP inspection or that do not have this feature enabled. Because man-in-the-middle attacks are limited to a single Layer 2 broadcast domain, separate the domain with dynamic ARP inspection checks from the one with no checking. This action secures the ARP caches of hosts in the domain enabled for dynamic ARP inspection.

- Dynamic ARP inspection depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically assigned IP addresses.

  When DHCP snooping is disabled or in non-DHCP environments, use ARP ACLs to permit or to deny packets.

- Dynamic ARP inspection is supported on access ports, trunk ports, and EtherChannel ports.

Note	Do not enable Dynamic ARP inspection on RSPAN VLANs. If Dynamic ARP inspection is enabled on RSPAN VLANs, Dynamic ARP inspection packets might not reach the RSPAN destination port.

- A physical port can join an EtherChannel port channel only when the trust state of the physical port and the channel port match. Otherwise, the physical port remains suspended in the port channel. A port channel inherits its trust state from the first physical port that joins the channel. Consequently, the trust state of the first physical port need not match the trust state of the channel.

  Conversely, when you change the trust state on the port channel, the switch configures a new trust state on all the physical ports that comprise the channel.

- The rate limit is calculated separately on each switch in a switch stack. For a cross-stack EtherChannel, this means that the actual rate limit might be higher than the configured value. For example, if you set the rate limit to 30 pps on an EtherChannel that has one port on switch 1 and one port on switch 2, each port can receive packets at 29 pps without causing the EtherChannel to become error-disabled.

- The operating rate for the port channel is cumulative across all the physical ports within the channel. For example, if you configure the port channel with an ARP rate-limit of 400 pps, all the interfaces combined on the channel receive an aggregate 400 pps. The rate of incoming ARP packets on EtherChannel ports is equal to the sum of the incoming rate of packets from all the channel members. Configure the rate limit for EtherChannel ports only after examining the rate of incoming ARP packets on the channel-port members.

  The rate of incoming packets on a physical port is checked against the port-channel configuration rather than the physical-ports configuration. The rate-limit configuration on a port channel is independent of the configuration on its physical ports.

  If the EtherChannel receives more ARP packets than the configured rate, the channel (including all physical ports) is placed in the error-disabled state.

- Make sure to limit the rate of ARP packets on incoming trunk ports. Configure trunk ports with higher rates to reflect their aggregation and to handle packets across multiple dynamic ARP inspection-enabled VLANs. You also can use the ip arp inspection limit none interface configuration command to make the rate unlimited. A high rate-limit on one VLAN can cause a denial-of-service attack to other VLANs when the software places the port in the error-disabled state.

- When you enable dynamic ARP inspection on the switch, policers that were configured to police ARP traffic are no longer effective. The result is that all ARP traffic is sent to the CPU.

Understanding Dynamic ARP Inspection

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, Host B wants to send information to Host A but does not have the MAC address of Host A in its ARP cache. Host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of Host A. All hosts within the broadcast domain receive the ARP request, and Host A responds with its MAC address. However, because ARP allows a gratuitous reply from a host even if an ARP request was not received, an ARP spoofing attack and the poisoning of ARP caches can occur. After the attack, all traffic from the device under attack flows through the attacker's computer and then to the router, switch, or host.

A malicious user can attack hosts, switches, and routers connected to your Layer 2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting traffic intended for other hosts on the subnet. Figure 26-1 shows an example of ARP cache poisoning.

Figure 25: ARP Cache Poisoning
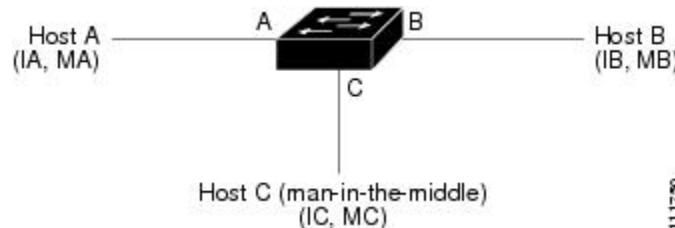
Hosts A, B, and C are connected to the switch on interfaces A, B and C, all of which are on the same subnet. Their IP and MAC addresses are shown in parentheses; for example, Host A uses IP address IA and MAC address MA. When Host A needs to communicate to Host B at the IP layer, it broadcasts an ARP request for the MAC address associated with IP address IB. When the switch and Host B receive the ARP request, they populate their ARP caches with an ARP binding for a host with the IP address IA and a MAC address MA; for example, IP address IA is bound to MAC address MA. When Host B responds, the switch and Host A populate their ARP caches with a binding for a host with the IP address IB and the MAC address MB.

Host C can poison the ARP caches of the switch, Host A, and Host B by broadcasting forged ARP responses with bindings for a host with an IP address of IA (or IB) and a MAC address of MC. Hosts with poisoned ARP caches use the MAC address MC as the destination MAC address for traffic intended for IA or IB. This means that Host C intercepts that traffic. Because Host C knows the true MAC addresses associated with IA and IB, it can forward the intercepted traffic to those hosts by using the correct MAC address as the destination. Host C has inserted itself into the traffic stream from Host A to Host B, the classic man-in-the middleattack.

Dynamic ARP inspection is a security feature that validates ARP packets in a network. It intercepts, logs,and discards ARP packets with invalid IP-to-MAC address bindings. This capability protects the network from certain man-in-the-middle attacks.

Dynamic ARP inspection ensures that only valid ARP requests and responses are relayed. The switch performs these activities:

- Intercepts all ARP requests and responses on untrusted ports
- Verifies that each of these intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the appropriate destination
- Drops invalid ARP packets

Dynamic ARP inspection determines the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a trusted database, the DHCP snooping binding database. This database is built by DHCP snooping if DHCP snooping is enabled on the VLANs and on the switch. If the ARP packet is received on a trusted interface, the switch forwards the packet without any checks. On untrusted interfaces, the switch forwards the packet only if it is valid.

You enable dynamic ARP inspection on a per-VLAN basis by using the ip arp inspection vlan vlan-range global configuration command.

In non-DHCP environments, dynamic ARP inspection can validate ARP packets against user-configured ARP access control lists (ACLs) for hosts with statically configured IP addresses. You define an ARP ACL by using the arp access-list acl-name global configuration command.

You can configure dynamic ARP inspection to drop ARP packets when the IP addresses in the packets are invalid or when the MAC addresses in the body of the ARP packets do not match the addresses specified in the Ethernet header. Use the ip arp inspection validate {[src-mac] [dst-mac] [ip]} global configuration command.

Interface Trust States and Network Security

Dynamic ARP inspection associates a trust state with each interface on the switch. Packets arriving on trusted interfaces bypass all dynamic ARP inspection validation checks, and those arriving on untrusted interfaces undergo the dynamic ARP inspection validation process.

In a typical network configuration, you configure all switch ports connected to host ports as untrusted and configure all switch ports connected to switches as trusted. With this configuration, all ARP packets entering the network from a given switch bypass the security check. No other validation is needed at any other place in the VLAN or in the network. You configure the trust setting by using theip arp inspection trust interface configuration command.

⚠️

Caution Use the trust state configuration carefully. Configuring interfaces as untrusted when they should betrusted can result in a loss of connectivity.

In the following figure, assume that both Switch A and Switch B are running dynamic ARP inspection on the VLAN that includes Host 1 and Host 2. If Host 1 and Host 2 acquire their IP addresses from the DHCP server connected to Switch A, only Switch A binds the IP-to-MAC address of Host 1. Therefore, if the interface between Switch A and Switch B is untrusted, the ARP packets from Host 1 are dropped by Switch B. Connectivity between Host 1 and Host 2 is lost.

Figure 26: ARP Packet Validation on a VLAN Enabled for Dynamic ARP Inspection
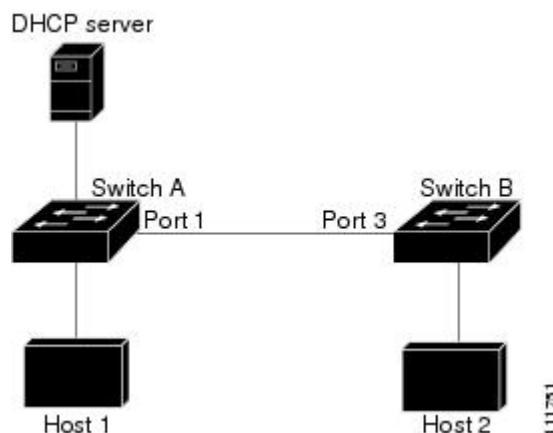
Configuring interfaces to be trusted when they are actually untrusted leaves a security hole in the network. If Switch A is not running dynamic ARP inspection, Host 1 can easily poison the ARP cache of Switch B (and Host 2, if the link between the switches is configured as trusted). This condition can occur even though Switch B is running dynamic ARP inspection.

Dynamic ARP inspection ensures that hosts (on untrusted interfaces) connected to a switch running dynamic ARP inspection do not poison the ARP caches of other hosts in the network. However, dynamic ARP inspection does not prevent hosts in other portions of the network from poisoning the caches of the hosts that are connected to a switch running dynamic ARP inspection.

In cases in which some switches in a VLAN run dynamic ARP inspection and other switches do not, configure the interfaces connecting such switches as untrusted. However, to validate the bindings of packets from nondynamic ARP inspection switches, configure the switch running dynamic ARP inspection with ARP ACLs. When you cannot determine such bindings, at Layer 3, isolate switches running dynamic ARP inspection from switches not running dynamic ARP inspection switches.

Note   Depending on the setup of the DHCP server and the network, it might not be possible to validate a given ARP packet on all switches in the VLAN.

Rate Limiting of ARP Packets

The switch CPU performs dynamic ARP inspection validation checks; therefore, the number of incoming ARP packets is rate-limited to prevent a denial-of-service attack. By default, the rate for untrusted interfaces is 15 packets per second (pps). Trusted interfaces are not rate-limited. You can change this setting by using the ip arp inspection limit interface configuration command.

When the rate of incoming ARP packets exceeds the configured limit, the switch places the port in the error-disabled state. The port remains in that state until you intervene. You can use the errdisable recovery global configuration command to enable error disable recovery so that ports automatically emerge from this state after a specified timeout period.

Note   The rate limit for an EtherChannel is applied separately to each switch in a stack. For example, if a limit of 20 pps is configured on the EtherChannel, each switch with ports in the EtherChannel can carry up to 20 pps. If any switch exceeds the limit, the entire EtherChannel is placed into the error-disabled state.

Relative Priority of ARP ACLs and DHCP Snooping Entries

Dynamic ARP inspection uses the DHCP snooping binding database for the list of valid IP-to-MAC address bindings.

ARP ACLs take precedence over entries in the DHCP snooping binding database. The switch uses ACLs only if you configure them by using the ip arp inspection filter vlan global configuration command. The switch first compares ARP packets to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, the switch also denies the packet even if a valid binding exists in the database populated by DHCP snooping.

Logging of Dropped Packets

When the switch drops a packet, it places an entry in the log buffer and then generates system messages on a rate-controlled basis. After the message is generated, the switch clears the entry from the log buffer. Each log entry contains flow information, such as the receiving VLAN, the port number, the source and destination IP addresses, and the source and destination MAC addresses.

You use the ip arp inspection log-buffer global configuration command to configure the number of entries in the buffer and the number of entries needed in the specified interval to generate system messages. You specify the type of packets that are logged by using the ip arp inspection vlan logging global configuration command.

Default Dynamic ARP Inspection Configuration

Feature	Default Settings
Dynamic ARP inspection	Disabled on all VLANs.
Interface trust state	All interfaces are untrusted.
Rate limit of incoming ARP packets	The rate is 15 pps on untrusted interfaces, assuming that the network is a switched network with a host connecting to as many as 15 new hosts per second. The rate is unlimited on all trusted interfaces. The burst interval is 1 second.
ARP ACLs for non-DHCP environments	No ARP ACLs are defined.
Validation checks	No checks are performed.
Log buffer	When dynamic ARP inspection is enabled, all denied or dropped ARP packets are logged. The number of entries in the log is 32. The number of system messages is limited to 5 per second. The logging-rate interval is 1 second.
Per-VLAN logging	All denied or dropped ARP packets are logged.

Relative Priority of ARP ACLs and DHCP Snooping Entries

Dynamic ARP inspection uses the DHCP snooping binding database for the list of valid IP-to-MAC address bindings.

ARP ACLs take precedence over entries in the DHCP snooping binding database. The switch uses ACLs only if you configure them by using the ip arp inspection filter vlan global configuration command. The switch first compares ARP packets to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, the switch also denies the packet even if a valid binding exists in the database populated by DHCP snooping.

Configuring ARP ACLs for Non-DHCP Environments

This procedure shows how to configure dynamic ARP inspection when Switch B shown in Figure 2 does not support dynamic ARP inspection or DHCP snooping.

If you configure port 1 on Switch A as trusted, a security hole is created because both Switch A and Host 1 could be attacked by either Switch B or Host 2. To prevent this possibility, you must configure port 1 on Switch A as untrusted. To permit ARP packets from Host 2, you must set up an ARP ACL and apply it to VLAN 1. If the IP address of Host 2 is not static (it is impossible to apply the ACL configuration on Switch A) you must separate Switch A from Switch B at Layer 3 and use a router to route packets between them.

Follow these steps to configure an ARP ACL on Switch A. This procedure is required in non-DHCP environments.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	arp access-list acl-name	Defines an ARP ACL, and enters ARP access-list configuration mode. By default, no ARP access lists are defined. **Note** At the end of the ARP access list, there is an implicit deny ip any mac any command.
Step 4	permit ip host sender-ip mac host sender-mac	Permits ARP packets from the specified host (Host 2). • Forsender-ip, enter the IP address of Host 2. • For sender-mac, enter the MAC address of Host 2.
Step 5	exit	Returns to global configuration mode.
Step 6	ip arp inspection filter arp-acl-name vlan vlan-range [static]	Applies ARP ACL to the VLAN. By default, no defined ARP ACLs are applied to any VLAN. • For arp-acl-name, specify the name of the ACL created in Step 2. • For vlan-range, specify the VLAN that the switches and hosts are in. You can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094.

	Command or Action	**Purpose**
		• (Optional) Specify static to treat implicit denies in the ARP ACL as explicit denies and to drop packets that do not match any previous clauses in the ACL. DHCP bindings are not used.
		If you do not specify this keyword, it means that there is no explicit deny in the ACL that denies the packet, and DHCP bindings determine whether a packet is permitted or denied if the packet does not match any clauses in the ACL.
		ARP packets containing only IP-to-MAC address bindings are compared against the ACL. Packets are permitted only if the access list permits them.
Step 7	interface interface-id	Specifies Switch A interface that is connected to Switch B, and enters the interface configuration mode.
Step 8	no ip arp inspection trust	Configures Switch A interface that is connected to Switch B as untrusted.
		By default, all interfaces are untrusted.
		For untrusted interfaces, the switch intercepts all ARP requests and responses. It verifies that the intercepted packets have valid IP-to-MAC address bindings before updating the local cache and before forwarding the packet to the appropriate destination. The switch drops invalid packets and logs them in the log buffer according to the logging configuration specified with the ip arp inspection vlan logging global configuration command.
Step 9	end	Returns to privileged EXEC mode.
Step 10	Use the following show commands: • show arp access-list acl-name • show ip arp inspection vlan vlan-range • show ip arp inspection interfaces	Verifies your entries.
Step 11	show running-config **Example:** Device# **show running-config**	Verifies your entries.

	Command or Action	Purpose
Step 12	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring Dynamic ARP Inspection in DHCP Environments

Before you begin

This procedure shows how to configure dynamic ARP inspection when two switches support this feature. Host 1 is connected to Switch A, and Host 2 is connected to Switch B. Both switches are running dynamic ARP inspection on VLAN 1 where the hosts are located. A DHCP server is connected to Switch A. Both hosts acquire their IP addresses from the same DHCP server. Therefore, Switch A has the bindings for Host 1 and Host 2, and Switch B has the binding for Host 2.

Note Dynamic ARP inspection depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically assigned IP addresses.

Follow these steps to configure dynamic ARP inspection. You must perform this procedure on both switches. This procedure is required.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	show cdp neighbors **Example:** Device(config-if)#**show cdp neighbors**	Verify the connection between the switches.
Step 3	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.

	Command or Action	**Purpose**
Step 4	ip arp inspection vlan vlan-range **Example:** Device(config)# `ip arp inspection vlan 1`	Enable dynamic ARP inspection on a per-VLAN basis. By default, dynamic ARP inspection is disabled on all VLANs. For vlan-range, specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. Specify the same VLAN ID for both switches.
Step 5	Interfaceinterface-id **Example:** Device(config)# `interface gigabitethernet1/0/1`	Specifies the interface connected to the other switch, and enter interface configuration mode.
Step 6	ip arp inspection trust **Example:** Device(config-if)#`ip arp inspection trust`	Configures the connection between the switches as trusted. By default, all interfaces are untrusted. The switch does not check ARP packets that it receives from the other switch on the trusted interface. It simply forwards the packets. For untrusted interfaces, the switch intercepts all ARP requests and responses. It verifies that the intercepted packets have valid IP-to-MAC address bindings before updating the local cache and before forwarding the packet to the appropriate destination. The switch drops invalid packets and logs them in the log buffer according to the logging configuration specified with the ip arp inspection vlan logging global configuration command.
Step 7	end **Example:** Device(config-if)#`end`	Returns to privileged EXEC mode.
Step 8	show ip arp inspection interfaces **Example:**	Verifies the dynamic ARP inspection configuration on interfaces.
Step 9	show ip arp inspection vlan vlan-range **Example:** Device(config-if)#`show ip arp inspection vlan 1`	Verifies the dynamic ARP inspection configuration on VLAN.
Step 10	show ip dhcp snooping binding **Example:** Device(config-if)#`show ip dhcp snooping binding`	Verifies the DHCP bindings.

	Command or Action	**Purpose**
Step 11	show ip arp inspection statistics vlan vlan-range **Example:** Device(config-if)#**show ip arp inspection statistics vlan 1**	Checks the dynamic ARP inspection statistics on VLAN.
Step 12	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 13	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.

Limiting the Rate of Incoming ARP Packets

The switch CPU performs dynamic ARP inspection validation checks; therefore, the number of incoming ARP packets is rate-limited to prevent a denial- of-service attack.

When the rate of incoming ARP packets exceeds the configured limit, the switch places the port in the error-disabled state. The port remains in that state until you enable error-disabled recovery so that ports automatically emerge from this state after a specified timeout period.

Note Unless you configure a rate limit on an interface, changing the trust state of the interface also changes its rate limit to the default value for that trust state. After you configure the rate limit, the interface retains the rate limit even when its trust state is changed. If you enter the no ip arp inspection limit interface configuration command, the interface reverts to its default rate limit.

Follow these steps to limit the rate of incoming ARP packets. This procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	interface interface-id	Specifies the interface to be rate-limited, and enter interface configuration mode.
Step 4	ip arp inspection limit {rate pps [burst interval seconds] | none}	Limits the rate of incoming ARP requests and responses on the interface. The default rate is 15 pps on untrusted interfaces and unlimited on trusted interfaces. The burst interval is 1 second. The keywords have these meanings: • For ratepps, specify an upper limit for the number of incoming packets processed per second. The range is 0 to 2048 pps. • (Optional) For burst intervalseconds, specify the consecutive interval in seconds, over which the interface is monitored for a high rate of ARP packets. The range is 1 to 15. • For rate none, specify no upper limit for the rate of incoming ARP packets that can be processed.
Step 5	exit	Returns to global configuration mode.
Step 6	Use the following commands: • errdisable detect cause arp-inspection • errdisable recovery cause arp-inspection • errdisable recovery interval interval	(Optional) Enables error recovery from the dynamic ARP inspection error-disabled state, and configure the dynamic ARP inspection recover mechanism variables. By default, recovery is disabled, and the recovery interval is 300 seconds. For interval interval, specify the time in seconds to recover from the error-disabled state. The range is 30 to 86400.
Step 7	exit	Returns to privileged EXEC mode.
Step 8	Use the following show commands: • show ip arp inspection interfaces • show errdisable recovery	Verifies your settings.

	Command or Action	Purpose
Step 9	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 10	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Performing Dynamic ARP Inspection Validation Checks

Dynamic ARP inspection intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. You can configure the switch to perform additional checks on the destination MAC address, the sender and target IP addresses, and the source MAC address.

Follow these steps to perform specific checks on incoming ARP packets. This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip arp inspection validate {[src-mac] [dst-mac] [ip]}	Performs a specific check on incoming ARP packets. By default, no checks are performed. The keywords have these meanings: • For src-mac, check the source MAC address in the Ethernet header against the sender MAC address in the ARP body. This check is performed on both ARP requests and responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

	Command or Action	**Purpose**
		• For dst-mac, check the destination MAC address in the Ethernet header against the target MAC address in ARP body. This check is performed for ARP responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.
		• For ip, check the ARP body for invalid and unexpected IP addresses. Addresses include 0.0.0.0, 255.255.255.255, and all IP multicast addresses. Sender IP addresses are checked in all ARP requests and responses, and target IP addresses are checked only in ARP responses.
		You must specify at least one of the keywords. Each command overrides the configuration of the previous command; that is, if a command enables src and dst mac validations, and a second command enables IP validation only, the src and dst mac validations are disabled as a result of the second command.
Step 4	exit	Returns to privileged EXEC mode.
Step 5	show ip arp inspection vlan vlan-range	Verifies your settings.
Step 6	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 7	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Monitoring DAI

To monitor DAI, use the following commands:

Command	**Description**
clear ip arp inspection statistics	Clears dynamic ARP inspection statistics.

Command	Description
show ip arp inspection statistics [vlan vlan-range]	Displays statistics for forwarded, dropped, MAC validation failure, IP validation failure, ACL permitted and denied, and DHCP permitted and denied packets for the specified VLAN. If no VLANs are specified or if a range is specified, displays information only for VLANs with dynamic ARP inspection enabled (active).
clear ip arp inspection log	Clears the dynamic ARP inspection log buffer.
show ip arp inspection log	Displays the configuration and contents of the dynamic ARP inspection log buffer.

For the show ip arp inspection statistics command, the switch increments the number of forwarded packets for each ARP request and response packet on a trusted dynamic ARP inspection port. The switch increments the number of ACL or DHCP permitted packets for each packet that is denied by source MAC, destination MAC, or IP validation checks, and the switch increments the appropriate.

Verifying the DAI Configuration

To display and verify the DAI configuration, use the following commands:

Command	Description
show arp access-list [acl-name]	Displays detailed information about ARP ACLs.
show ip arp inspection interfaces [interface-id]	Displays the trust state and the rate limit of ARP packets for the specified interface or all interfaces.
show ip arp inspection vlan vlan-range	Displays the configuration and the operating state of dynamic ARP inspection for the specified VLAN. If no VLANs are specified or if a range is specified, displays information only for VLANs with dynamic ARP inspection enabled (active).

Feature Information for Dynamic ARP Inspection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 36: Feature Information for Dynamic ARP Inspection

Feature Name	Releases	Feature Information
Dynamic ARP Inspection	Cisco IOS XE Everest 16.6.1	ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. Dynamic ARP inspection is a security feature that validates ARP packets in a network. It intercepts, logs,and discards ARP packets with invalid IP-to-MAC address bindings. This capability protects the network from certain man-in-the-middle attacks.
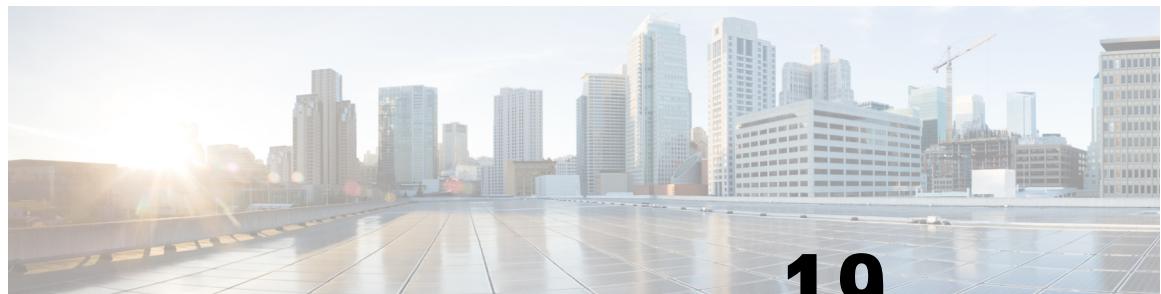
Configuring IPv6 First Hop Security

Prerequisites for First Hop Security in IPv6

- You have configured the necessary IPv6 enabled SDM template.

- You should be familiar with the IPv6 neighbor discovery feature.

Restrictions for First Hop Security in IPv6

- The following restrictions apply when applying FHS policies to EtherChannel interfaces (Port Channels):

  - A physical port with an FHS policy attached cannot join an EtherChannel group.

  - An FHS policy cannot be attached to an physical port when it is a member of an EtherChannel group.

- By default, a snooping policy has a security-level of guard. When such a snooping policy is configured on an access switch, external IPv6 Router Advertisement (RA) or Dynamic Host Configuration Protocol

server packets are blocked, even though the uplink port facing the router or DHCP server/relay is configured as a trusted port. To allow IPv6 RA or DHCPv6 server messages, do the following:

- Apply an IPv6 RA-guard policy (for RA) or IPv6 DHCP-guard policy (for DHCP server messages) on the uplink port.

- Configure a snooping policy with a lower security-level, for example glean or inspect. However; configuring a lower security level is not recommended with such a snooping policy, because benefits of First Hop security features are not effective.

Information about First Hop Security in IPv6

First Hop Security in IPv6 (FHS IPv6) is a set of IPv6 security features, the policies of which can be attached to a physical interface, or a VLAN. An IPv6 software policy database service stores and accesses these policies. When a policy is configured or modified, the attributes of the policy are stored or updated in the software policy database, then applied as was specified. The following IPv6 policies are currently supported:

- IPv6 Snooping Policy—IPv6 Snooping Policy acts as a container policy that enables most of the features available with FHS in IPv6.

- IPv6 FHS Binding Table Content—A database table of IPv6 neighbors connected to the switch is created from information sources such as Neighbor Discovery (ND) protocol snooping. This database, or binding, table is used by various IPv6 guard features (such as IPv6 ND Inspection) to validate the link-layer address (LLA), the IPv4 or IPv6 address, and prefix binding of the neighbors to prevent spoofing and redirect attacks.
- IPv6 Neighbor Discovery Inspection—IPv6 ND inspection learns and secures bindings for stateless autoconfiguration addresses in Layer 2 neighbor tables. IPv6 ND inspection analyzes neighbor discovery messages in order to build a trusted binding table database and IPv6 neighbor discovery messages that do not conform are dropped. An ND message is considered trustworthy if its IPv6-to-Media Access Control (MAC) mapping is verifiable.

This feature mitigates some of the inherent vulnerabilities of the ND mechanism, such as attacks on DAD, address resolution, router discovery, and the neighbor cache.

> **Note** Effective Cisco IOS XE Release 16.3.1, ND Inspection functionality, IPv6 Snooping Policy, and IPv6 FHS Binding Table Content are supported through Switch Integrated Security Feature (SISF)-based Device Tracking. For more information, see Configuring SISF based device tracking section of the Software Configuration Guide.

- IPv6 Router Advertisement Guard—The IPv6 Router Advertisement (RA) guard feature enables the network administrator to block or reject unwanted or rogue RA guard messages that arrive at the network switch platform. RAs are used by routers to announce themselves on the link. The RA Guard feature analyzes the RAs and filters out bogus RAs sent by unauthorized routers. In host mode, all router advertisement and router redirect messages are disallowed on the port. The RA guard feature compares configuration information on the Layer 2 device with the information found in the received RA frame. Once the Layer 2 device has validated the content of the RA frame and router redirect frame against the configuration, it forwards the RA to its unicast or multicast destination. If the RA frame content is not validated, the RA is dropped.

- IPv6 DHCP Guard—The IPv6 DHCP Guard feature blocks reply and advertisement messages that come from unauthorized DHCPv6 servers and relay agents. IPv6 DHCP guard can prevent forged messages from being entered in the binding table and block DHCPv6 server messages when they are received on ports that are not explicitly configured as facing a DHCPv6 server or DHCP relay. To use this feature, configure a policy and attach it to an interface or a VLAN. To debug DHCP guard packets, use the **debug ipv6 snooping dhcp-guard** privileged EXEC command.

How to Configure an IPv6 Snooping Policy

The IPv6 Snooping Policy feature has been deprecated. Although the commands are visible on the CLI and you can configure them, we recommend that you use the Switch Integrated Security Feature (SISF)-based Device Tracking feature instead.

Beginning in privileged EXEC mode, follow these steps to configure IPv6 Snooping Policy :

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 2	ipv6 snooping policy policy-name **Example:** Device(config)# **ipv6 snooping policy example_policy**	Creates a snooping policy and enters IPv6 Snooping Policy Configuration mode.
Step 3	{[default] | [device-role {node | switch}] | [limit address-count value] | [no] | [protocol {dhcp | ndp}] | [security-level {glean | guard | inspect}] | [tracking {disable [stale-lifetime [seconds | infinite] | enable [reachable-lifetime [seconds | infinite] }] | [trusted-port] } **Example:** Device**(config-ipv6-snooping)# security-level inspect** **Example:** Device**(config-ipv6-snooping)# trusted-port**	Enables data address gleaning, validates messages against various criteria, specifies the security level for messages. • (Optional) default—Sets all to default options. • (Optional) device-role{node] | switch}—Specifies the role of the device attached to the port. Default is node. • (Optional) limit address-count value—Limits the number of addresses allowed per target. • (Optional) no—Negates a command or sets it to defaults. • (Optional) protocol{dhcp | ndp}—Specifies which protocol should be redirected to the snooping feature for analysis. The default, is dhcp and ndp. To

	Command or Action	Purpose
		change the default, use the no protocol command.
		• (Optional) security-level{glean|guard|inspect}—Specifies the level of security enforced by the feature. Default is guard.
		glean—Gleans addresses from messages and populates the binding table without any verification.
		guard—Gleans addresses and inspects messages. In addition, it rejects RA and DHCP server messages. This is the default option.
		inspect—Gleans addresses, validates messages for consistency and conformance, and enforces address ownership.
		• (Optional) tracking {disable | enable}—Overrides the default tracking behavior and specifies a tracking option.
		• (Optional) trusted-port—Sets up a trusted port. It disables the guard on applicable targets. Bindings learned through a trusted port have preference over bindings learned through any other port. A trusted port is given preference in case of a collision while making an entry in the table.
Step 4	end **Example:** Device(config-ipv6-snooping)# **exit**	Exits configuration modes to Privileged EXEC mode.
Step 5	show ipv6 snooping policy policy-name **Example:** Device#**show ipv6 snooping policy example_policy**	Displays the snooping policy configuration.

What to do next

Attach an IPv6 Snooping policy to interfaces or VLANs.

How to Attach an IPv6 Snooping Policy to an Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Snooping policy on an interface or VLAN:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 2	interface Interface_type stack/module/port **Example:** Device(config)# **interface gigabitethernet 1/1/4**	Specifies an interface type and identifier; enters the interface configuration mode.
Step 3	switchport **Example:** Device(config-if)# **switchport**	Enters the Switchport mode. **Note** To configure Layer 2 parameters, if the interface is in Layer 3 mode, you must enter the switchport interface configuration command without any parameters to put the interface into Layer 2 mode. This shuts down the interface and then re-enables it, which might generate messages on the device to which the interface is connected. When you put an interface that is in Layer 3 mode into Layer 2 mode, the previous configuration information related to the affected interface might be lost, and the interface is returned to its default configuration. The command prompt displays as (config-if)# in Switchport configuration mode.
Step 4	ipv6 snooping [attach-policy policy_name [vlan {vlan_id | add vlan_ids | exceptvlan_ids | none | remove vlan_ids}] | vlan {vlan_id | add vlan_ids | exceptvlan_ids | none | remove vlan_ids | all}] **Example:** Device(config-if)# **ipv6 snooping** or Device(config-if)# **ipv6 snooping**	Attaches a custom ipv6 snooping policy to the interface or the specified VLANs on the interface. To attach the default policy to the interface, use the ipv6 snooping command without the attach-policy keyword. To attach the default policy to VLANs on the interface, use the ipv6 snooping vlan command. The default policy is, security-level guard, device-role node, protocol ndp and dhcp.

	Command or Action	Purpose
	attach-policy example_policy or Device(config-if)# **ipv6 snooping vlan 111,112** or Device(config-if)# **ipv6 snooping attach-policy example_policy vlan 111,112**	
Step 5	do show running-config **Example:** Device#(config-if)# **do show running-config**	Verifies that the policy is attached to the specified interface without exiting the interface configuration mode.

How to Attach an IPv6 Snooping Policy to a Layer 2 EtherChannel Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Snooping policy on an EtherChannel interface or VLAN:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 2	interface range Interface_name **Example:** Device(config)# **interface range Po11**	Specify the port-channel interface name assigned when the EtherChannel was created. Enters the interface range configuration mode. **Tip**　　Enter the do show interfaces summary command for quick reference to interface names and types.
Step 3	ipv6 snooping [attach-policy policy_name [vlan {vlan_ids | add vlan_ids | except vlan_ids | none | remove vlan_ids | all}] | vlan [{vlan_ids | add vlan_ids | exceptvlan_ids | none | remove vlan_ids | all}] **Example:** Device(config-if-range)# **ipv6 snooping attach-policy example_policy**	Attaches the IPv6 Snooping policy to the interface or the specified VLANs on that interface. The default policy is attached if the attach-policy option is not used.

	Command or Action	Purpose

	`or`
	
	`Device(config-if-range)# ipv6 snooping`
	`attach-policy example_policy vlan`
	`222,223,224`
	
	`or`
	
	`Device(config-if-range)#ipv6 snooping`
	`vlan 222, 223,224`	

Step 4	do show running-config
	interfaceportchannel_interface_name
	
	Example:
	
	`Device#(config-if-range)#  do show`
	`running-config int po11`
		Confirms that the policy is attached to the specified interface without exiting the configuration mode.

How to Attach an IPv6 Snooping Policy to VLANs Globally

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Snooping Policy to VLANs across multiple interfaces:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 2	vlan configuration vlan_list **Example:** `Device(config)#  vlan configuration 333`	Specifies the VLANs to which the IPv6 Snooping policy will be attached ; enters the VLAN interface configuration mode.

Step 3	ipv6 snooping [attach-policy policy_name]
	
	Example:
	
	`Device(config-vlan-config)#ipv6 snooping`
	`attach-policy example_policy`
		Attaches the IPv6 Snooping policy to the specified VLANs across all switch and stack interfaces. The default policy is attached if the attach-policy option is not used. The default policy is, security-level guard, device-role node, protocol ndp and dhcp.

Step 4	do show running-config
	
	Example:
	
	`Device#(config-if)#  do show`
	`running-config`
		Verifies that the policy is attached to the specified VLANs without exiting the interface configuration mode.

How to Configure the IPv6 Binding Table Content

Beginning in privileged EXEC mode, follow these steps to configure IPv6 Binding Table Content :

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 2	[no] ipv6 neighbor binding [vlan vlan-id {ipv6-address interface interface_type stack/module/port hw_address [reachable-lifetimevalue [seconds | default | infinite] | [tracking{ [default | disable] [reachable-lifetimevalue [seconds | default | infinite] | [enable [reachable-lifetimevalue [seconds | default | infinite] | [retry-interval {seconds| default [reachable-lifetimevalue [seconds | default | infinite] }] **Example:** Device(config)# **ipv6 neighbor binding**	Adds a static entry to the binding table database. **Note** Switch adds small variance to configured reachable-time value to improve system stability during timer expiry of binding entries.
Step 3	[no] ipv6 neighbor binding max-entries number [mac-limit number | port-limit number [mac-limit number] | vlan-limit number [[mac-limit number] | [port-limit number [mac-limitnumber]]]] **Example:** Device(config)# **ipv6 neighbor binding max-entries 30000**	Specifies the maximum number of entries that are allowed to be inserted in the binding table cache.
Step 4	ipv6 neighbor binding logging **Example:** Device(config)# **ipv6 neighbor binding logging**	Enables the logging of binding table main events.
Step 5	exit **Example:** Device(config)# **exit**	Exits global configuration mode, and places the router in privileged EXEC mode.
Step 6	show ipv6 neighbor binding **Example:** Device# **show ipv6 neighbor binding**	Displays contents of a binding table.

How to Configure an IPv6 Neighbor Discovery Inspection Policy

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 ND Inspection Policy:

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 2	[no]ipv6 nd inspection policy policy-name **Example:** Device(config)# **ipv6 nd inspection policy example_policy**	Specifies the ND inspection policy name and enters ND Inspection Policy configuration mode.
Step 3	device-role {host	switch} **Example:** Device(config-nd-inspection)# **device-role switch**	Specifies the role of the device attached to the port. The default is host.
Step 4	limit address-count value **Example:** Device(config-nd-inspection)# **limit address-count 1000**	Enter 1–10,000.
Step 5	tracking {enable [reachable-lifetime {value	infinite}]	disable [stale-lifetime {value	infinite}]} **Example:** Device(config-nd-inspection)# **tracking disable stale-lifetime infinite**	Overrides the default tracking policy on a port.
Step 6	trusted-port **Example:** Device(config-nd-inspection)# **trusted-port**	Configures a port to become a trusted port.
Step 7	validate source-mac **Example:** Device(config-nd-inspection)# **validate source-mac**	Checks the source media access control (MAC) address against the link-layer address.
Step 8	no {device-role	limit address-count	tracking	trusted-port	validate source-mac} **Example:**	Remove the current configuration of a parameter with the no form of the command.

	Command or Action	Purpose
	Device(config-nd-inspection)# **no validate source-mac**	
Step 9	default {device-role | limit address-count | tracking | trusted-port | validate source-mac}	Restores configuration to the default values.
	Example:	
	Device(config-nd-inspection)# **default limit address-count**	
Step 10	do show ipv6 nd inspection policy policy_name	Verifies the ND Inspection Configuration without exiting ND inspection configuration mode.
	Example:	
	Device(config-nd-inspection)# **do show ipv6 nd inspection policy example_policy**	

How to Attach an IPv6 Neighbor Discovery Inspection Policy to an Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 ND Inspection policy to an interface or VLANs on an interface :

Procedure

	Command or Action	Purpose
Step 1	configure terminal	Enters the global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 2	interface Interface_type stack/module/port	Specifies an interface type and identifier; enters the interface configuration mode.
	Example:	
	Device(config)# **interface gigabitethernet 1/1/4**	
Step 3	ipv6 nd inspection [attach-policy policy_name [vlan {vlan_ids | add vlan_ids | except vlan_ids | none | remove vlan_ids | all}] | vlan [{vlan_ids | add vlan_ids | exceptvlan_ids | none | remove vlan_ids | all}]	Attaches the Neighbor Discovery Inspection policy to the interface or the specified VLANs on that interface. The default policy is attached if the attach-policy option is not used.
	Example:	
	Device(config-if)# **ipv6 nd inspection attach-policy example_policy** or Device(config-if)# **ipv6 nd inspection attach-policy example_policy vlan 222,223,224** **or**	

	Command or Action	Purpose
	Device(config-if)# **ipv6 nd inspection vlan 222, 223,224**	
Step 4	do show running-config **Example:** Device#(config-if)# **do show running-config**	Verifies that the policy is attached to the specified interface without exiting the interface configuration mode.

How to Attach an IPv6 Neighbor Discovery Inspection Policy to a Layer 2 EtherChannel Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Neighbor Discovery Inspection policy on an EtherChannel interface or VLAN:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 2	interface range Interface_name **Example:** Device(config)# **interface Po11**	Specify the port-channel interface name assigned when the EtherChannel was created. Enters the interface range configuration mode. **Tip** Enter the do show interfaces summary command for quick reference to interface names and types.
Step 3	ipv6 nd inspection [attach-policy policy_name [vlan {vlan_ids | add vlan_ids | except vlan_ids | none | remove vlan_ids | all}] | vlan [{vlan_ids | add vlan_ids | exceptvlan_ids | none | remove vlan_ids | all}] **Example:** Device(config-if-range)# **ipv6 nd inspection attach-policy example_policy** or Device(config-if-range)# **ipv6 nd inspection attach-policy example_policy vlan 222,223,224** **or**	Attaches the ND Inspection policy to the interface or the specified VLANs on that interface. The default policy is attached if the attach-policy option is not used.

	Command or Action	Purpose
	Device(config-if-range)#**ipv6 nd inspection vlan 222, 223,224**	
Step 4	do show running-config interfaceportchannel_interface_name **Example:** Device#(config-if-range)# **do show running-config int po11**	Confirms that the policy is attached to the specified interface without exiting the configuration mode.

How to Attach an IPv6 Neighbor Discovery Inspection Policy to VLANs Globally

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 ND Inspection policy to VLANs across multiple interfaces:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 2	vlan configuration vlan_list **Example:** Device(config)# **vlan configuration 334**	Specifies the VLANs to which the IPv6 Snooping policy will be attached ; enters the VLAN interface configuration mode.
Step 3	ipv6 nd inspection [attach-policy policy_name] **Example:** Device(config-vlan-config)#**ipv6 nd inspection attach-policy example_policy**	Attaches the IPv6 Neighbor Discovery policy to the specified VLANs across all switch and stack interfaces. The default policy is attached if the attach-policy option is not used. The default policy is, device-role host, no drop-unsecure, limit address-count disabled, sec-level minimum is disabled, tracking is disabled, no trusted-port, no validate source-mac.
Step 4	do show running-config **Example:** Device#(config-if)# **do show running-config**	Confirms that the policy is attached to the specified VLANs without exiting the configuration mode.

How to Configure an IPv6 Router Advertisement Guard Policy

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 Router Advertisement policy :

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** `Device# `**`configure terminal`**	Enters the global configuration mode.
Step 2	[no]ipv6 nd raguard policy policy-name **Example:** `Device(config)# `**`ipv6 nd raguard policy example_policy`**	Specifies the RA Guard policy name and enters RA Guard Policy configuration mode.
Step 3	[no]device-role {host | monitor | router | switch} **Example:** `Device(config-nd-raguard)# `**`device-role switch`**	Specifies the role of the device attached to the port. The default is host. **Note** For a network with both host-facing ports and router-facing ports, along with a RA guard policy configured with device-role host on host-facing ports or vlan, it is mandatory to configure a RA guard policy with device-role router on router-facing ports to allow the RA Guard feature to work properly.
Step 4	[no]hop-limit {maximum | minimum} value **Example:** `Device(config-nd-raguard)# `**`hop-limit maximum 33`**	(1–255) Range for Maximum and Minimum Hop Limit values. Enables filtering of Router Advertisement messages by the Hop Limit value. A rogue RA message may have a low Hop Limit value (equivalent to the IPv4 Time to Live) that when accepted by the host, prevents the host from generating traffic to destinations beyond the rogue RA message generator. An RA message with an unspecified Hop Limit value is blocked. If not configured, this filter is disabled. Configure minimum to block RA messages with Hop Limit values lower than the value you specify. Configure maximumto block RA messages with Hop Limit values greater than the value you specify.

	Command or Action	Purpose
Step 5	[no]managed-config-flag {off | on} **Example:** Device(config-nd-raguard)# **managed-config-flag on**	Enables filtering of Router Advertisement messages by the Managed Address Configuration, or "M" flag field. A rouge RA message with an M field of 1 can cause a host to use a rogue DHCPv6 server. If not configured, this filter is disabled. On—Accepts and forwards RA messages with an M value of 1, blocks those with 0. Off—Accepts and forwards RA messages with an M value of 0, blocks those with 1.
Step 6	[no]match {ipv6 access-list list | ra prefix-list list} **Example:** Device(config-nd-raguard)# **match ipv6 access-list example_list**	Matches a specified prefix list or access list.
Step 7	[no]other-config-flag {on | off} **Example:** Device(config-nd-raguard)# **other-config-flag on**	Enables filtering of Router Advertisement messages by the Other Configuration, or "O" flag field. A rouge RA message with an O field of 1 can cause a host to use a rogue DHCPv6 server. If not configured, this filter is disabled. On—Accepts and forwards RA messages with an O value of 1, blocks those with 0. Off—Accepts and forwards RA messages with an O value of 0, blocks those with 1.
Step 8	[no]router-preference maximum {high | medium | low} **Example:** Device(config-nd-raguard)# **router-preference maximum high**	Enables filtering of Router Advertisement messages by the Router Preference flag. If not configured, this filter is disabled. • high—Accepts RA messages with the Router Preference set to high, medium, or low. • medium—Blocks RA messages with the Router Preference set to high. • low—Blocks RA messages with the Router Preference set to medium and high.
Step 9	[no]trusted-port **Example:** Device(config-nd-raguard)# **trusted-port**	When configured as a trusted port, all attached devices are trusted, and no further message verification is performed.
Step 10	default {device-role | hop-limit {maximum | minimum} | managed-config-flag | match	Restores a command to its default value.

	Command or Action	Purpose
	{ipv6 access-list | ra prefix-list } | other-config-flag | router-preference maximum| trusted-port} **Example:** Device(config-nd-raguard)# **default hop-limit**	
Step 11	do show ipv6 nd raguard policy policy_name **Example:** Device(config-nd-raguard)# **do show ipv6 nd raguard policy example_policy**	(Optional)—Displays the ND Guard Policy configuration without exiting the RA Guard policy configuration mode.

How to Attach an IPv6 Router Advertisement Guard Policy to an Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Router Advertisement policy to an interface or to VLANs on the interface :

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 2	interface Interface_type stack/module/port **Example:** Device(config)# **interface gigabitethernet 1/1/4**	Specifies an interface type and identifier; enters the interface configuration mode.
Step 3	ipv6 nd raguard [attach-policy policy_name [vlan {vlan_ids | add vlan_ids | except vlan_ids | none | remove vlan_ids | all}] | vlan [{vlan_ids | add vlan_ids | exceptvlan_ids | none | remove vlan_ids | all}] **Example:** Device(config-if)# **ipv6 nd raguard attach-policy example_policy** or Device(config-if)# **ipv6 nd raguard attach-policy example_policy vlan 222,223,224** **or** Device(config-if)# **ipv6 nd raguard vlan 222, 223,224**	Attaches the Neighbor Discovery Inspection policy to the interface or the specified VLANs on that interface. The default policy is attached if the attach-policy option is not used.

	Command or Action	Purpose
Step 4	do show running-config **Example:** Device#(config-if)# **do show running-config**	Confirms that the policy is attached to the specified interface without exiting the configuration mode.

How to Attach an IPv6 Router Advertisement Guard Policy to a Layer 2 EtherChannel Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Router Advertisement Guard Policy on an EtherChannel interface or VLAN:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 2	interface range Interface_name **Example:** Device(config)# **interface Po11**	Specify the port-channel interface name assigned when the EtherChannel was created. Enters the interface range configuration mode. **Tip** Enter the do show interfaces summary command for quick reference to interface names and types.
Step 3	ipv6 nd raguard [attach-policy policy_name [vlan {vlan_ids | add vlan_ids | except vlan_ids | none | remove vlan_ids | all}] | vlan [{vlan_ids | add vlan_ids | exceptvlan_ids | none | remove vlan_ids | all}] **Example:** Device(config-if-range)# **ipv6 nd raguard attach-policy example_policy** or Device(config-if-range)# **ipv6 nd raguard attach-policy example_policy vlan 222,223,224** **or** Device(config-if-range)#**ipv6 nd raguard vlan 222, 223,224**	Attaches the RA Guard policy to the interface or the specified VLANs on that interface. The default policy is attached if the attach-policy option is not used.

	Command or Action	Purpose
Step 4	do show running-config interfaceportchannel_interface_name **Example:** Device#(config-if-range)# **do show running-config int po11**	Confirms that the policy is attached to the specified interface without exiting the configuration mode.

How to Attach an IPv6 Router Advertisement Guard Policy to VLANs Globally

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Router Advertisement policy to VLANs regardless of interface:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	vlan configuration vlan_list **Example:** Device(config)# **vlan configuration 335**	Specifies the VLANs to which the IPv6 RA Guard policy will be attached ; enters the VLAN interface configuration mode.
Step 3	ipv6 dhcp guard [attach-policy policy_name] **Example:** Device(config-vlan-config)#**ipv6 nd raguard attach-policy example_policy**	Attaches the IPv6 RA Guard policy to the specified VLANs across all switch and stack interfaces. The default policy is attached if the attach-policy option is not used.
Step 4	do show running-config **Example:** Device#(config-if)# **do show running-config**	Confirms that the policy is attached to the specified VLANs without exiting the configuration mode.

How to Configure an IPv6 DHCP Guard Policy

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 DHCP (DHCPv6) Guard policy:

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 2	[no]ipv6 dhcp guard policy policy-name **Example:** Device(config)# **ipv6 dhcp guard policy example_policy**	Specifies the DHCPv6 Guard policy name and enters DHCPv6 Guard Policy configuration mode.
Step 3	[no]device-role {client | server} **Example:** Device(config-dhcp-guard)# **device-role server**	(Optional) Filters out DHCPv6 replies and DHCPv6 advertisements on the port that are not from a device of the specified role. Default is client. • client—Default value, specifies that the attached device is a client. Server messages are dropped on this port. • server—Specifies that the attached device is a DHCPv6 server. Server messages are allowed on this port.
Step 4	[no] match server access-list ipv6-access-list-name **Example:** ;;Assume a preconfigured IPv6 Access List as follows: Device(config)# **ipv6 access-list my_acls** Device(config-ipv6-acl)# **permit host FE80::A8BB:CCFF:FE01:F700 any** ;;configure DCHPv6 Guard to match approved access list. Device(config-dhcp-guard)# **match server access-list my_acls**	(Optional). Enables verification that the advertised DHCPv6 server or relay address is from an authorized server access list (The destination address in the access list is 'any'). If not configured, this check will be bypassed. An empty access list is treated as a permit all.
Step 5	[no] match reply prefix-list ipv6-prefix-list-name **Example:** ;;Assume a preconfigured IPv6 prefix list as follows: Device(config)# **ipv6 prefix-list my_prefix permit 2001:0DB8::/64 le 128** ;; Configure DCHPv6 Guard to match prefix Device(config-dhcp-guard)# **match reply prefix-list my_prefix**	(Optional) Enables verification of the advertised prefixes in DHCPv6 reply messages from the configured authorized prefix list. If not configured, this check will be bypassed. An empty prefix list is treated as a permit.

	Command or Action	**Purpose**
Step 6	[no]preference{ max limit	min limit } **Example:** Device(config-dhcp-guard)# **preference max 250** Device(config-dhcp-guard)#**preference min 150**	Configure max and min when device-role is serverto filter DCHPv6 server advertisements by the server preference value. The defaults permit all advertisements. max limit—(0 to 255) (Optional) Enables verification that the advertised preference (in preference option) is less than the specified limit. Default is 255. If not specified, this check will be bypassed. min limit—(0 to 255) (Optional) Enables verification that the advertised preference (in preference option) is greater than the specified limit. Default is 0. If not specified, this check will be bypassed.
Step 7	[no] trusted-port **Example:** Device(config-dhcp-guard)# **trusted-port**	(Optional) trusted-port—Sets the port to a trusted mode. No further policing takes place on the port. **Note** If you configure a trusted port then the device-role option is not available.
Step 8	default {device-role	trusted-port} **Example:** Device(config-dhcp-guard)# **default device-role**	(Optional) default—Sets a command to its defaults.
Step 9	do show ipv6 dhcp guard policy policy_name **Example:** Device(config-dhcp-guard)# **do show ipv6 dhcp guard policy example_policy**	(Optional) Displays the configuration of the IPv6 DHCP guard policy without leaving the configuration submode. Omitting the policy_name variable displays all DHCPv6 policies.

Example of DHCPv6 Guard Configuration

```
enable
configure terminal
ipv6 access-list acl1
 permit host FE80::A8BB:CCFF:FE01:F700 any
ipv6 prefix-list abc permit 2001:0DB8::/64 le 128
ipv6 dhcp guard policy pol1
 device-role server
 match server access-list acl1
 match reply prefix-list abc
 preference min 0
 preference max 255
 trusted-port
interface GigabitEthernet 0/2/0
 switchport
```

```
    ipv6 dhcp guard attach-policy pol1 vlan add 1
    vlan 1
      ipv6 dhcp guard attach-policy pol1
    show ipv6 dhcp guard policy pol1
```

How to Attach an IPv6 DHCP Guard Policy to an Interface or a VLAN on an Interface

Beginning in privileged EXEC mode, follow these steps to configure IPv6 Binding Table Content :

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 2	interface Interface_type stack/module/port **Example:** Device(config)# **interface gigabitethernet 1/1/4**	Specifies an interface type and identifier; enters the interface configuration mode.
Step 3	ipv6 dhcp guard [attach-policy policy_name [vlan {vlan_ids | add vlan_ids | except vlan_ids | none | remove vlan_ids | all}]] | vlan [{vlan_ids | add vlan_ids | exceptvlan_ids | none | remove vlan_ids | all}] **Example:** Device(config-if)# **ipv6 dhcp guard attach-policy example_policy** or Device(config-if)# **ipv6 dhcp guard attach-policy example_policy vlan 222,223,224** **or** Device(config-if)# **ipv6 dhcp guard vlan 222, 223,224**	Attaches the DHCP Guard policy to the interface or the specified VLANs on that interface. The default policy is attached if the attach-policy option is not used.
Step 4	do show running-config interface Interface_type stack/module/port **Example:** Device#(config-if)# **do show running-config gig 1/1/4**	Confirms that the policy is attached to the specified interface without exiting the configuration mode.

How to Attach an IPv6 DHCP Guard Policy to a Layer 2 EtherChannel Interface

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 DHCP Guard policy on an EtherChannel interface or VLAN:

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 2	interface range Interface_name **Example:** Device(config)# **interface Po11**	Specify the port-channel interface name assigned when the EtherChannel was created. Enters the interface range configuration mode. **Tip** Enter the do show interfaces summary command for quick reference to interface names and types.
Step 3	ipv6 dhcp guard [attach-policy policy_name [vlan {vlan_ids | add vlan_ids | except vlan_ids | none | remove vlan_ids | all}] | vlan [{vlan_ids | add vlan_ids | exceptvlan_ids | none | remove vlan_ids | all}] **Example:** Device(config-if-range)# **ipv6 dhcp guard attach-policy example_policy** or Device(config-if-range)# **ipv6 dhcp guard attach-policy example_policy vlan 222,223,224** **or** Device(config-if-range)#**ipv6 dhcp guard vlan 222, 223,224**	Attaches the DHCP Guard policy to the interface or the specified VLANs on that interface. The default policy is attached if the attach-policy option is not used.
Step 4	do show running-config interfaceportchannel_interface_name **Example:** Device#(config-if-range)# **do show running-config int po11**	Confirms that the policy is attached to the specified interface without exiting the configuration mode.

How to Attach an IPv6 DHCP Guard Policy to VLANs Globally

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 DHCP Guard policy to VLANs across multiple interfaces:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 2	vlan configuration vlan_list **Example:** `Device(config)# vlan configuration 334`	Specifies the VLANs to which the IPv6 Snooping policy will be attached ; enters the VLAN interface configuration mode.
Step 3	ipv6 dhcp guard [attach-policy policy_name] **Example:** `Device(config-vlan-config)#ipv6 dhcp guard attach-policy example_policy`	Attaches the IPv6 Neighbor Discovery policy to the specified VLANs across all switch and stack interfaces. The default policy is attached if the attach-policy option is not used. The default policy is, device-role client, no trusted-port.
Step 4	do show running-config **Example:** `Device#(config-if)#  do show running-config`	Confirms that the policy is attached to the specified VLANs without exiting the configuration mode.

How to Configure IPv6 Source Guard

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters the global configuration mode.

	Command or Action	**Purpose**
Step 3	[no] ipv6 source-guard policy policy_name **Example:** Device(config)# **ipv6 source-guard policy example_policy**	Specifies the IPv6 Source Guard policy name and enters IPv6 Source Guard policy configuration mode.
Step 4	[deny global-autoconf] [permit link-local] [default{. . . }] [exit] [no{. . . }] **Example:** Device(config-sisf-sourceguard)# **deny global-autoconf**	(Optional) Defines the IPv6 Source Guard policy. • deny global-autoconf—Denies data traffic from auto-configured global addresses. This is useful when all global addresses on a link are DHCP-assigned and the administrator wants to block hosts with self-configured addresses to send traffic. • permit link-local—Allows all data traffic that is sourced by a link-local address. **Note** Trusted option under source guard policy is not supported.
Step 5	end **Example:** Device(config-sisf-sourceguard)# **end**	Exits out of IPv6 Source Guard policy configuration mode.
Step 6	show ipv6 source-guard policy policy_name **Example:** Device# **show ipv6 source-guard policy example_policy**	Shows the policy configuration and all the interfaces where the policy is applied.

What to do next

Apply the IPv6 Source Guard policy to an interface.

How to Attach an IPv6 Source Guard Policy to an Interface

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:**	Enters the global configuration mode.

	Command or Action	**Purpose**
	Device# **configure terminal**	
Step 3	interface Interface_type stack/module/port **Example:** Device(config)# **interface gigabitethernet 1/1/4**	Specifies an interface type and identifier; enters the interface configuration mode.
Step 4	ipv6 source-guard [attach-policy \<policy_name>] **Example:** Device(config-if)# **ipv6 source-guard attach-policy example_policy**	Attaches the IPv6 Source Guard policy to the interface. The default policy is attached if the attach-policy option is not used.
Step 5	show ipv6 source-guard policy policy_name **Example:** Device#(config-if)# **show ipv6 source-guard policy example_policy**	Shows the policy configuration and all the interfaces where the policy is applied.

How to attach an IPv6 Source Guard Policy to a Layer 2 EtherChannel Interface

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 3	interface port-channel port-channel-number **Example:** Device (config)# **interface Po4**	Specifies an interface type and port number and places the switch in the port channel configuration mode.
Step 4	ipv6 source-guard [attach-policy \<policy_name>] **Example:** Device(config-if) # **ipv6 source-guard attach-policy example_policy**	Attaches the IPv6 Source Guard policy to the interface. The default policy is attached if the attach-policy option is not used.

	Command or Action	Purpose
Step 5	show ipv6 source-guard policy policy_name **Example:** Device(config-if) #**show ipv6 source-guard policy example_policy**	Shows the policy configuration and all the interfaces where the policy is applied.

How to Configure IPv6 Prefix Guard

Note   To allow routing protocol control packets sourced by a link-local address when prefix guard is applied, enable the permit link-local command in the source-guard policy configuration mode.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	[no] ipv6 source-guard policy source-guard-policy **Example:** Device(config)# **ipv6 source-guard policy my_snooping_policy**	Defines an IPv6 source-guard policy name and enters switch integrated security features source-guard policy configuration mode.
Step 4	[no] validate address **Example:** Device(config-sisf-sourceguard)# **no validate address**	Disables the validate address feature and enables the IPv6 prefix guard feature to be configured.
Step 5	validate prefix **Example:** Device(config-sisf-sourceguard)# **validate prefix**	Enables IPv6 source guard to perform the IPv6 prefix-guard operation.

	Command or Action	Purpose
Step 6	exit **Example:** Device(config-sisf-sourceguard)# **exit**	Exits switch integrated security features source-guard policy configuration mode and returns to privileged EXEC mode.
Step 7	show ipv6 source-guard policy [source-guard-policy] **Example:** Device# **show ipv6 source-guard policy policy1**	Displays the IPv6 source-guard policy configuration.

How to Attach an IPv6 Prefix Guard Policy to an Interface

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 3	interface Interface_type stack/module/port **Example:** Device(config)# **interface gigabitethernet 1/1/4**	Specifies an interface type and identifier; enters the interface configuration mode.
Step 4	ipv6 source-guard attach-policy policy_name **Example:** Device(config-if)# **ipv6 source-guard attach-policy example_policy**	Attaches the IPv6 Source Guard policy to the interface. The default policy is attached if the attach-policy option is not used.
Step 5	show ipv6 source-guard policy policy_name **Example:** Device(config-if)# **show ipv6 source-guard policy example_policy**	Shows the policy configuration and all the interfaces where the policy is applied.

How to attach an IPv6 Prefix Guard Policy to a Layer 2 EtherChannel Interface

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters the global configuration mode.
Step 3	interface port-channel port-channel-number **Example:** Device (config)# **interface Po4**	Specifies an interface type and port number and places the switch in the port channel configuration mode.
Step 4	ipv6 source-guard [attach-policy <policy_name>] **Example:** Device(config-if)# **ipv6 source-guard attach-policy example_policy**	Attaches the IPv6 Source Guard policy to the interface. The default policy is attached if the attach-policy option is not used.
Step 5	show ipv6 source-guard policy policy_name **Example:** Device(config-if)# **show ipv6 source-guard policy example_policy**	Shows the policy configuration and all the interfaces where the policy is applied.

Configuration Examples for IPv6 First Hop Security

Examples: How to attach an IPv6 Source Guard Policy to a Layer 2 EtherChannel Interface

The following example shows how to attach an IPv6 Source Guard Policy to a Layer 2 EtherChannel Interface:

```
Switch# configure terminal
Switch(config)# ipv6 source-guard policy POL
Switch(config-sisf-sourceguard) # validate address
switch(config-sisf-sourceguard)# exit
Switch(config)# interface Po4
Switch(config)# ipv6 snooping
Switch(config-if)# ipv6 source-guard attach-policy POL
Switch(config-if)# exit
switch(config)#
```

Examples: How to attach an IPv6 Prefix Guard Policy to a Layer 2 EtherChannel Interface

The following example shows how to attach an IPv6 Prefix Guard Policy to a Layer 2 EtherChannel Interface:

```
Switch# configure terminal
Switch(config)# ipv6 source-guard policy POL
Switch (config-sisf-sourceguard)# no validate address
Switch((config-sisf-sourceguard)# validate prefix
Switch(config)# interface Po4
Switch(config-if)# ipv6 snooping
Switch(config-if)# ipv6 source-guard attach-policy POL
```

Feature History for IPv6 First Hop Security

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	IPv6 First Hop Security	First Hop Security in IPv6 is a set of IPv6 security features, the policies of which can be attached to a physical interface, an EtherChannel interface, or a VLAN. An IPv6 software policy database service stores and accesses these policies. When a policy is configured or modified, the attributes of the policy are stored or updated in the software policy database, then applied as was specified.
		The IPv6 Snooping Policy feature has been deprecated. Although the commands are visible on the CLI and you can configure them, we recommend that you use the Switch Integrated Security Feature (SISF)-based Device Tracking feature instead.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.
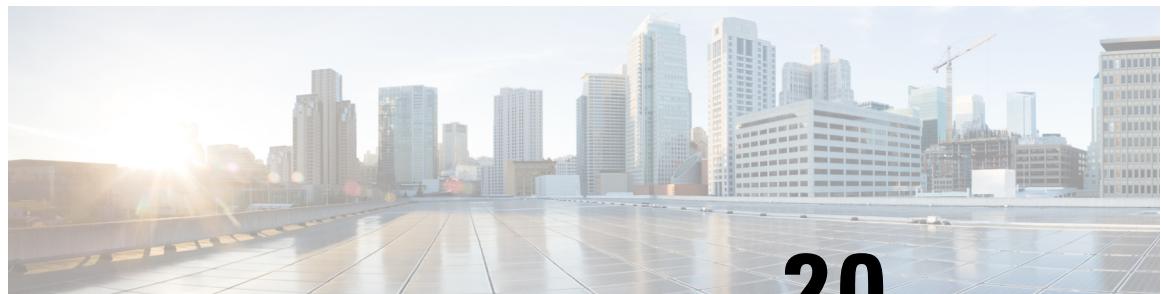
CHAPTER 20

Configuring SISF-Based Device Tracking

Information About SISF-Based Device Tracking

Overview of SISF-Based Device Tracking

The Switch Integrated Security Features based (SISF-based) device tracking feature is part of the suite of first-hop security features.

The main role of the feature is to track the presence, location, and movement of end-nodes in the network. SISF snoops traffic received by the switch, extracts device identity (MAC and IP address), and stores them in a binding table. Many features, such as, Cisco TrustSec, IEEE 802.1X, LISP, and web authentication depend on the accuracy of this information to operate properly.

SISF-based device tracking supports both IPv4 and IPv6.

Even with the introduction of SISF-based device tracking, the legacy device tracking CLI (IP Device Tracking (IPDT) and IPv6 Snooping CLI) continues to be available. When you bootup the switch, the set of commands that is available depends on existing configuration, and only one of the following is available:

- SISF-based device tracking CLI, or

- IPDT and IPv6 Snooping CLI

Note   The IPDT and IPv6 Snooping commands are deprecated, but continue to be available. We recommend that you upgrade to SISF-based device tracking.

If you are using the IPDT and IPv6 Snooping CLI and want to migrate to SISF-based device tracking, see Migrating from legacy IPDT and IPv6 Snooping to SISF-Based Device Tracking, for more information.

SISF-based device tracking can be enabled manually (by using device-tracking commands), or programmatically (which is the case when providing device tracking services to other features).

Options to Enable SISF-Based Device Tracking

SISF-Based device tracking is disabled by default.

You can enable it by defining a device tracking policy and attaching the policy to a specific target.

Note The target could be an interface or a VLAN.

Manually Enabling SISF-Based Device Tracking

- Option 1: Apply the default device tracking policy to a target.

  Enter the device-tracking command in the interface configuration mode or in the VLAN configuration mode. The system then attaches the default policy it to the interface or VLAN.

  Note The default policy is a built-in policy with default settings; you cannot change any of the attributes of the default policy. In order to be able to configure device tracking policy attributes you must create a custom policy. See Option 2: Create a custom policy with custom settings.

- Option 2: Create a custom policy with custom settings.

  Enter the device-tracking policy command in global configuration mode and enter a custom policy name. The system creates a policy with the name you specify. You can then configure the available settings, in the device tracking configuration mode (config-device-tracking), and attach the policy to a specified target.

Programmatically Enabling SISF-Based Device Tracking

Some features rely on device tracking and utilize the trusted database of binding entries that SISF-based device tracking builds and maintains. These features, also called device tracking clients, enable device tracking programmatically (create and attach the device tracking policy).

Note The exceptions here are IEEE 802.1X, web authentication, Cisco TrustSec, and IP Source Guard (IPSG) - they also rely on device tracking, but they do not enable it. For these device tracking clients, you must enter the ip dhcp snooping vlan vlan command, to programmatically enable device tracking on a particular target.

Note the following about programmatically enabling SISF-based device tracking:

- A device tracking client requires device tracking to be enabled.

  There are several device tracking clients, therefore, multiple programmatic policies could be created. The settings of each policy differ depending on the device tracking client that creates the policy.

- The policy that is created, and its settings, are system-defined.

Configurable policy attributes are available in the device tracking configuration mode (config-device-tracking) and vary from one release to another. If you try to modify an attribute that is not configurable, the configuration change is rejected and an error message is displayed.

For release-specific information about programmatically created policies, see Programmatically Enabling SISF-Based Device Tracking in Cisco IOS XE <release name> <release number> in the required version of the document.

Migrating from Legacy IPDT and IPv6 Snooping to SISF-Based Device Tracking

Based on the legacy configuration that exists on your device, the device-tracking upgrade-cli command upgrades your CLI differently. Consider the following configuration scenarios and the corresponding migration results before you migrate your existing configuration.

Note You cannot configure a mix of the old IPDT and IPv6 snooping CLI with the SISF-based device tracking CLI.

Only IPDT Configuration Exists

If your device has only IPDT configuration, running the device-tracking upgrade-cli command converts the configuration to use the new SISF policy that is created and attached to the interface. You can then update this SISF policy.

If you continue to use the legacy commands, this restricts you to operate in a legacy mode where only the legacy IPDT and IPv6 snooping commands are available on the device.

Only IPv6 Snooping Configuration Exists

On a device with existing IPv6 snooping configuration, the old IPv6 Snooping commands are available for further configuration. The following options are available:

- (Recommended) Use the device-tracking upgrade-cli command to convert all your legacy configuration to the new SISF-based device tracking commands. After conversion, only the new device tracking commands will work on your device.

- Use the legacy IPv6 Snooping commands for your future configuration and do not run the device-tracking upgrade-cli command. With this option, only the legacy IPv6 Snooping commands are available on your device, and you cannot use the new SISF-based device tracking CLI commands.

Both IPDT and IPv6 Snooping Configuration Exist

On a device that has both legacy IPDT configuration and IPv6 snooping configuration, you can convert legacy commands to the SISF-based device tracking CLI commands. However, note that only one snooping policy can be attached to an interface, and the IPv6 snooping policy parameters override the IPDT settings.

> **Note**
>
> If you do not migrate to the new SISF-based commands and continue to use the legacy IPv6 snooping or IPDT commands, your IPv4 device tracking configuration information may be displayed in the IPv6 snooping commands, as the SISF-based device tracking feature handles both IPv4 and IPv6 configuration. To avoid this, we recommend that you convert your legacy configuration to SISF-based device tracking commands.

No IPDT or IPv6 Snooping Configuration Exists

If your device has no legacy IP Device Tracking or IPv6 Snooping configurations, you can use only the new SISF-based device tracking commands for all your future configuration. The legacy IPDT commands and IPv6 snooping commands are not available.

How to Configure SISF-Based Device Tracking

Manually Enabling SISF-Based Device Tracking

Applying the Default Device Tracking Policy to a Target

Beginning in privileged EXEC mode, follow these steps to apply the default device tracking policy to an interface or VLAN:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Specify an interface or a VLAN
	
	• interface interface
	• vlan configuration vlan_list
	
	Example:
	`Device(config)# interface gigabitethernet 1/1/4`
	OR
	`Device(config)# vlan configuration 333`
		interface type number—Specifies the interface and enters interface configuration mode. The device tracking policy will be attached to the specified interface.
		
		vlan configuration vlan_list—Specifies the VLANs and enters VLAN feature configuration mode. The device tracking policy will be attached to the specified VLAN.

	Command or Action	**Purpose**
Step 4	device-tracking **Example:** Device(config-if)# **device-tracking** OR Device(config-vlan-config)# **device-tracking**	Enables SISF-based device tracking and attaches the default policy it to the interface or VLAN. The default policy is a built-in policy with default settings; none of the attributes of the default policy can be changed.
Step 5	end **Example:** Device(config-if)# **end** OR Device(config-vlan-config)# **end**	Exits interface configuration mode and returns to privileged EXEC mode. Exits VLAN feature configuration mode and returns to privileged EXEC mode.
Step 6	show device-tracking policy policy-name **Example:** Device# **show device-tracking policy default**	Displays device-tracking policy configuration, and all the targets it is applied to.

Creating a Custom Device Tracking Policy with Custom Settings

Beginning in privileged EXEC mode, follow these steps to create and configure a device tracking policy:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	[no] device-tracking policy policy-name **Example:** Device(config)# **device-tracking policy example_policy**	Creates the policy and enters device-tracking configuration mode.
Step 4	[data-glean	default	destination-glean	device-role	distribution-switch	exit	limit	no	prefix-glean	protocol	security-level	tracking	trusted-port	vpc] **Example:** Device(config-device-tracking)# **destination-glean log-only**	Enter the question mark (?) at the system prompt to obtain a list of available options in this mode. You can configure the following for both IPv4 and IPv6: • (Optional) data-glean—Enables learning of addresses from a data packet snooped from a source inside the network and populates the binding table with the data

Command or Action	Purpose
	traffic source address. Enter one of these options:
	• log-only—Generates a syslog message upon data packet notification
	• recovery—Uses a protocol to enable binding table recovery. Enter NDP or DHCP.
	• (Optional) default—Sets the policy attribute to its default value. You can set these policy attributes to their default values: data-glean, destination-glean, device-role, limit, prefix-glean, protocol, security-level, tracking, trusted-port.
	• (Optional) destination-glean—Populates the binding table by gleaning data traffic destination address. Enter one of these options:
	• log-only—Generates a syslog message upon data packet notification
	• recovery—Uses a protocol to enable binding table recovery. Enter DHCP.
	• (Optional) device-role—Sets the role of the device attached to the port. It can be a node or a switch. Enter one of these options:
	• node—Configures the attached device as a node. This is the default option.
	• switch—Configures the attached device as a switch.
	• (Optional) distribution-switch—Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect.
	• exit—Exits the device-tracking policy configuration mode.
	• limit address-count—Specifies an address count limit per port. The range is 1 to 32000.
	• no—Negates the command or sets it to defaults.

Command or Action	Purpose
	• (Optional) prefix-glean—Enables learning of prefixes from either IPv6 Router Advertisements or from DHCP-PD. You have the following option:
	• (Optional) only—Gleans only prefixes and not host addresses.
	• (Optional) protocol—Sets the protocol to glean; by default, all are gleaned. Enter one of these options:
	• arp [prefix-list name]—Gleans addresses in ARP packets. Optionally, enter the name of prefix-list that is to be matched.
	• dhcp4 [prefix-list name]—Glean addresses in DHCPv4 packets. Optionally, enter the name of prefix-list that is to be matched.
	• dhcp6 [prefix-list name]—Glean addresses in DHCPv6 packets. Optionally, enter the name of prefix-list that is to be matched.
	• ndp [prefix-list name]—Glean addresses in NDP packets. Optionally, enter the name of prefix-list that is to be matched.
	• udp [prefix-list name]—Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect.
	• (Optional) security-level—Specifies the level of security enforced by the feature. Enter one of these options:
	• glean—Gleans addresses passively.
	• guard—Inspects and drops un-authorized messages. This is the default.
	• inspect—Gleans and validates messages.
	• (Optional) tracking—Specfies a tracking option. Enter one of these options:

	Command or Action	**Purpose**
		• disable ［stale-lifetime ［1-86400-seconds｜infinite］ ］ —Turns of device-tracking. Optionally, you can enter the duration for which the entry is kept inactive before deletion, or keep it permanently inactive.
		• enable ［reachable-lifetime ［1-86400-seconds｜infinite］ ］ —Turns on device-tracking. Optionally, you can enter the duration for which the entry is kept reachable, or keep it permanently reachable.
		• (Optional) trusted-port—Sets up a trusted port. Disables the guard on applicable targets. Bindings learned through a trusted port have preference over bindings learned through any other port. A trusted port is given preference in case of a collision while making an entry in the table.
		• (Optional) vpc—Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect.
Step 5	end **Example:** Device(config-device-tracking)# **end**	Exits device-tracking configuration mode and returns to privileged EXEC mode.
Step 6	show device-tracking policy policy-name **Example:** Device# **show device-tracking policy example_policy**	Displays the device-tracking policy configuration.

What to do next

Attach the policy to an interface or VLAN.

Attaching a Device Tracking Policy to an Interface

Beginning in privileged EXEC mode, follow these steps to attach a device tracking policy to an interface:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	interface interface **Example:** Device(config-if)# **interface gigabitethernet 1/1/4**	Specifies an interface and enters interface configuration mode.
Step 4	device-tracking attach-policy policy name **Example:** Device(config-if)# **device-tracking attach-policy example_policy**	Attaches the device tracking policy to the interface. Device tracking is also supported on EtherChannels. **Note** SISF based device-tracking policies can be disabled only if they are custom policies. Programmatically created policies can be removed only if the corresponding device-tracking client feature configuration is removed.
Step 5	end **Example:** Device(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	show device-tracking policies [interface interface] **Example:** Device# **show device-tracking policies interface gigabitethernet 1/1/4**	Displays policies that match the specified interface type and number.

Attaching a Device Tracking Policy to a VLAN

Beginning in privileged EXEC mode, follow these steps to attach a device-tracking policy to VLANs across multiple interfaces:

Procedure

	Command or Action	**Purpose**
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	**Purpose**
	Example: Device> **enable**	• Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	vlan configuration vlan_list **Example:** Device(config)# **vlan configuration 333**	Specifies the VLANs to which the device tracking policy will be attached; enters the VLAN interface configuration mode.
Step 4	device-tracking attach-policy policy_name **Example:** Device(config-vlan-config)# **device-tracking attach-policy** **example_policy**	Attaches the device tracking policy to the specified VLANs across all switch interfaces. **Note** SISF based device-tracking policies can be disabled only if they are custom policies. Programmatically created policies can be removed only if the corresponding device-tracking client feature configuration is removed.
Step 5	do show device-tracking policies vlan vlan-ID **Example:** Device(config-vlan-config)# **do show** **device-tracking policies vlan 333**	Verifies that the policy is attached to the specified VLAN, without exiting the VLAN interface configuration mode.
Step 6	end **Example:** Device(config-vlan-config)# **end**	Exits VLAN feature configuration mode and returns to privileged EXEC mode.

Programmatically Enabling SISF-Based Device Tracking in Cisco IOS XE Everest 16.6.x

Table 37: Programmatically Enabling SISF-Based Device Tracking in Cisco IOS XE Everest 16.6.x

Device tracking client features that can enable SISF-based device tracking	In this release, you can programmatically enable SISF-based device tracking for these features: • IEEE 802.1X, web authentication, Cisco TrustSec, and IPSG features: enter the ip dhcp snooping vlan vlan command. • Cisco Locator/ID Separation Protocol. **Note** LISP settings are effective, if there is more than one programmatically created policy. This does not adversely affect the way, any of the other device tracking client features work. For example, if you have configured the ip dhcp snooping vlan vlan command for IEEE 802.1X, and also enabled SISF-based device tracking by configuring LISP, the IEEE 802.1X feature continues to work as expected.
Policy Name	DT-PROGRAMMATIC Although there is more than one device tracking client feature, the system-generated policy is one and the same; the list of settings may differ with each programmatically created policy. See the examples for more information.

User Options	• Only one policy can be attached to the same interface or VLAN.
	• The policy cannot be replaced by another policy.
	• The policy cannot be removed unless the device tracking client feature configuration is removed.
	• You can change these setting of of a programmatically created policy (device-tracking policy command, in the device tracking configuration mode (config-device-tracking)):
	• data-glean
	• default
	• device-role
	• destination-glean
	• exit
	• limit
	• no
	• prefix-glean
	• protocol
	• security-level
	• tracking
	• trusted-port
	• The distribution-switch and vpc are visible on the CLI, but any configuration changes are not effective.
	• The address count limit per MAC setting cannot be changed (This refers to the limit address-count for IPv4 per mac and limit address-count for IPv6 per mac commands), but the address count limit per port or interface can be changed.
	• When a device-tracking policy is attached to an interface under a VLAN, the policy settings on the interface take precedence over those on its VLAN; exceptions here are the values for limit address-count for IPv4 per mac and limit address-count for IPv6 per mac, which are aggregated from the policy on both the interface and VLAN.

Configuring a Multi-Switch Network to Stop Creating Binding Entries from a Trunk Port

In a multi-switch network, SISF-based device tracking provides the capability to distribute binding table entries between switches running the feature. Binding entries are only created on the switches where the host appears on an access port. No entry is created for a host that appears over a trunk port. This is achieved by configuring a policy with the trusted-port and device-role switch options, and attaching it to the trunk port.

☞

Important	Both, the trusted-port, and device-role switch options, must be configured in the policy.
	Further, we recommended that you apply such a policy on a port facing a device, which also has SISF-based device tracking enabled.

Complete the following steps:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	device-tracking policy policy-name **Example:** Device(config)# **device-tracking policy example_trusted_policy**	Enters device-tracking policy configuration mode, for the specified policy.
Step 4	device-role switch **Example:** Device(config-device-tracking)# **device-role switch**	Specifies the role of the device attached to the port. Default is node. Enter the device-role switch option to stop the creation of binding entries for the port.
Step 5	trusted-port **Example:** Device(config-device-tracking)# **trusted-port**	Sets up a trusted port. Disables the guard on applicable targets. Bindings learned through a trusted port have preference over bindings learned through any other port. A trusted port is given preference in case of a collision while making an entry in the table.
Step 6	exit **Example:** Device(config-device-tracking)# **exit**	Exits device-tracking policy configuration mode and returns to global configuration mode
Step 7	interface interface **Example:** Device(config)# **interface gigabitethernet 1/0/25**	Specifies an interface and enters interface configuration mode.

	Command or Action	**Purpose**
Step 8	device-tracking attach-policy policy-name **Example:** Device(config-if)# **device-tracking attach-policy example_trusted_policy**	Attaches a device tracking policy to the interface or the specified VLANs on the interface.
Step 9	end **Example:** Device(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for SISF-Based Device Tracking

These examples show sample device-tracking configuration and other recommended or related configuration for certain situations.

Example: Programmatically Enabling SISF-Based Device Tracking in Cisco IOS XE Everest 16.6.x

The sample output in the examples show the different settings of programmatically created policies.

Device tracking client: LISP

The LISP configuration here is only meant to serve as an example.

After you configure LISP, enter the show device-tracking policy command in privileged EXEC mode, to display the DT-PROGRAMMATIC policy that is created and the corresponding settings:

```
Device(config)# router lisp
<output truncated>
Device(config-router-lisp)# instance-id 3
Device(config-router-lisp-instance)# service ethernet
Device(config-router-lisp-instance-service)# eid-table vlan 10
Device(config-router-lisp-instance-dynamic-eid)# database-mapping 10.1.1.0/24 locator-set
set1
Device(config-router-lisp-instance-service)# exit-service-ethernet
Device(config-router-lisp-instance)# exit-instance-id
Device(config-router-lisp)# exit-router-lisp

Device# show device-tracking policy DT-PROGRAMMATIC
Policy DT-PROGRAMMATIC configuration:
  security-level guard (*)
  device-role node
  gleaning from Neighbor Discovery
  gleaning from DHCP
  gleaning from ARP
  gleaning from DHCP4
  NOT gleaning from protocol unkn
  limit address-count for IPv4 per mac 1 (*)
  limit address-count for IPv6 per mac 8 (*)
  tracking enable
Policy DT-PROGRAMMATIC is applied on the following targets:
```

```
Target       Type  Policy            Feature           Target range
vlan 10      VLAN  DT-PROGRAMMATIC   Device-tracking   vlan all
 note:
 Binding entry Down timer: 10 minutes (*)
 Binding entry Stale timer:  30 minutes (*)
```

Device tracking clients: IEEE 802.1X, Web Authentication, Cisco TrustSec, IPSG

Configure the ip dhcp snooping vlan vlan command in global configuration mode to enable device-tracking for the IEEE 802.1X, web authentication, Cisco TrustSec, and IPSG features. Enter the show device-tracking policy command in privileged EXEC mode, to display the DT-PROGRMMATIC policy that is created and the corresponding settings that are made:

```
Device# configure terminal
Device(config)# ip dhcp snooping vlan 10
Device(config)# end

Device# show device-tracking policy DT-PROGRAMMATIC
Policy DT-PROGRAMMATIC configuration:
  security-level glean (*)
  device-role node
  gleaning from Neighbor Discovery
  gleaning from DHCP
  gleaning from ARP
  gleaning from DHCP4
  NOT gleaning from protocol unkn
  limit address-count for IPv4 per mac 1 (*)

  tracking enable
Policy DT-PROGRAMMATIC is applied on the following targets:
Target    Type    Policy            Feature           Target range
vlan 10   VLAN    DT-PROGRAMMATIC   Device-tracking   vlan all
  note:
  Binding entry Down timer: 24 hours (*)
  Binding entry Stale timer:   24 hours (*)
```

Example: Disabling IPv6 Device Tracking on a Target

By default, SISF-based device tracking supports both IPv4 and IPv6. The following configuration examples show how you can disable IPv6 device tracking if you have to:

Disabling IPv6 device tracking when the target is attached to a custom policy:

```
Device(config)# device-tracking policy example-policy
 Device(config-device-tracking)# no protocol ndp
 Device(config-device-tracking)# no protocol dhcp6
 Device(config-device-tracking)# end
```

Disabling IPv6 device tracking when the target is attached to a programmatic policy:

In Cisco IOS XE Everest 16.6.x you can disable IPv6 device tracking by changing the programmatic policy:

```
Device(config)# device-tracking policy DT-PROGRAMMATIC
Device(config-device-tracking)# no protocol ndp
Device(config-device-tracking)# no protocol dhcp6
Device(config-device-tracking)# end
```

Example: Enabling IPv6 for SVI on VLAN (To Mitigate the Duplicate Address Problem)

When IPv6 is enabled in the network and a switched virtual interface (SVI) is configured on a VLAN, we recommend that you add the following to the SVI configuration. This enables the SVI to acquire a link-local address automatically; this address is used as the source IP address of the SISF probe, thus preventing the duplicate IP address issue.

```
Device> enable
Device# configure terminal
Device(config)# interface vlan 10
Device(config-if)# ipv6 enable
Device(config-if)# end
```

Example: Mitigating the IPv4 Duplicate Address Problem

This example shows how you can tackle the `Duplicate IP Address 0.0.0.0 error message` problem encountered by clients that run Microsoft Windows:

Configure the device-tracking tracking auto-source command in global configuration mode. This command determines the source IP and MAC address used in the Address Resolution Packet (ARP) request sent by the switch to probe a client, in order to maintain its entry in the device-tracking table. The purpose, is to avoid using 0.0.0.0 as source IP address.

> **Note** Configure the device-tracking tracking auto-source command when a switch virtual interface (SVI) is not configured. You do not have to configure it when a SVI is configured with an IPv4 address on the VLAN.

Command	Action (In order to select source IP and MAC address for device tracking ARP probe)	Notes
device-tracking tracking auto-source	• Set source to VLAN SVI if present. • Look for IP and MAC binding in device-tracking table from same subnet. • Use 0.0.0.0	We recommend that you disable device-tracking on all trunk ports to avoid MAC flapping.
device-tracking tracking auto-source override	• Set source to VLAN SVI if present • Use 0.0.0.0	Not recommended when there is no SVI.

Command	Action **(In order to select source IP and MAC address for device tracking ARP probe)**	Notes
ip device tracking probe auto-source fallback 0.0.0.X 255.255.255.0	• Set source to VLAN SVI if present. • Look for IP and MAC binding in device-tracking table from same subnet. • Compute source IP from client IP using host bit and mask provided. Source MAC is taken from the MAC address of the switchport facing the client*.	We recommend that you disable device-tracking on all trunk ports to avoid MAC flapping. The computed IPv4 address must not be assigned to any client or network device.
device-tracking tracking auto-source fallback 0.0.0.X 255.255.255.0 override	• Set source to VLAN SVI if present. Compute source IP from client IP using host bit and mask provided*. Source MAC is taken from the MAC address of the switchport facing the client*.	

* Depending on the client IP address, an IPv4 address has to be reserved for the source IP.

A reserved source IPv4 address = (host-ip and mask) | client-ip

- Client IP = 192.0.2.25

- Source IP = (192.0.2.25 and 255.255.255.0) | (0.0.0.1) = 192.0.2.1

IP address 192.0.2.1 should not be assigned to any client or network device.

Example: Avoiding a Short Device-Tracking Binding Reachable Time

When migrating from an older release, the following configuration may be present:

```
device-tracking binding reachable-time 10
```

Remove this by entering the no version of the command.

```
Device> enable
Device# configure terminal
Device(config)# no device-tracking binding reachable-time 10
Device(config)# end
```

Feature History for SISF-Based Device Tracking

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	SISF-Based Device Tracking	Tracks the presence, location, and movement of end-nodes in the network. The feature snoops traffic received by the switch, extracts device identity (MAC and IP address), and stores them in a binding table. Other features (device tracking clients) depend on the accuracy of this information to operate properly. Both IPv4 and IPv6 are supported.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com.
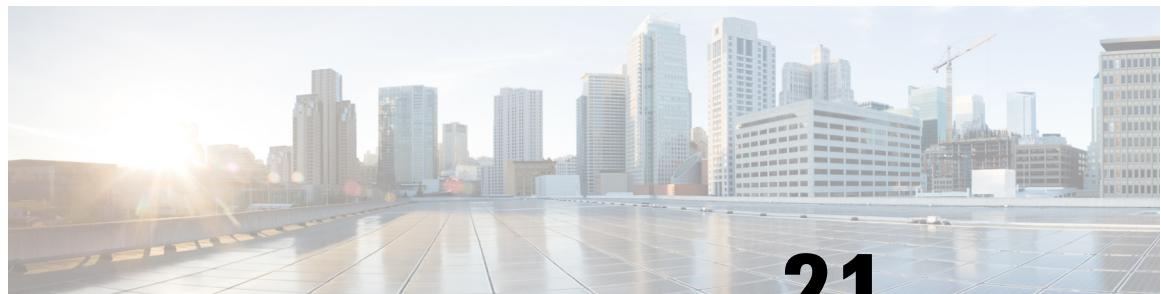
CHAPTER 21

Configuring IEEE 802.1x Port-Based Authentication

This chapter describes how to configure IEEE 802.1x port-based authentication. IEEE 802.1x authentication prevents unauthorized devices (clients) from gaining access to the network. Unless otherwise noted, the term switch refers to a standalone switch or a switch stack.

Restrictions for IEEE 802.1x Port-Based Authentication

- Switchports are always unauthorized when used with private VLANs. Dynamic VLANs pushed from the Authentication, Authorization, and Accounting (AAA) server is not supported on private VLAN ports. The data client session is expected to authorize on the secondary VLAN of the private VLAN dot1x port.

- Only interface-configured private VLAN-based authorization and dynamic VLAN on a normal access VLAN port is supported.

- If the dot1q tag vlan native command is configured globally, the dot1x reauthentication will fail on trunk ports.

- Do not configure the same VLAN ID for both voice VLAN and access VLAN at the same time, because it may cause authentication failures.

- Port security is not supported with IEEE 802.1x port-based authentication.

Information About 802.1x Port-Based Authentication

The 802.1x standard defines a client-server-based access control and authentication protocol that prevents unauthorized clients from connecting to a LAN through publicly accessible ports unless they are properly authenticated. The authentication server authenticates each client connected to a switch port before making available any services offered by the switch or the LAN.

Note TACACS is not supported with 802.1x authentication.

Until the client is authenticated, 802.1x access control allows only Extensible Authentication Protocol over LAN (EAPOL), Cisco Discovery Protocol (CDP), and Spanning Tree Protocol (STP) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

Client session	Maximum sessions supported
Maximum dot1x or MAB client sessions	2000
Maximum web-based authentication sessions	2000
Maximum dot1x sessions with critical-auth VLAN enabled and server re-initialized	2000
Maximum MAB sessions with various session features applied	2000
Maximum dot1x sessions with service templates or session features applied	2000

Port-Based Authentication Process

To configure IEEE 802.1X port-based authentication, you must enable authentication, authorization, and accounting (AAA) and specify the authentication method list. A method list describes the sequence and authentication method to be queried to authenticate a user.

The AAA process begins with authentication. When 802.1x port-based authentication is enabled and the client supports 802.1x-compliant client software, these events occur:

- If the client identity is valid and the 802.1x authentication succeeds, the switch grants the client access to the network.

- If 802.1x authentication times out while waiting for an EAPOL message exchange and MAC authentication bypass is enabled, the switch can use the client MAC address for authorization. If the client MAC address is valid and the authorization succeeds, the switch grants the client access to the network. If the client MAC address is invalid and the authorization fails, the switch assigns the client to a guest VLAN that provides limited services if a guest VLAN is configured.

- If the switch gets an invalid identity from an 802.1x-capable client and a restricted VLAN is specified, the switch can assign the client to a restricted VLAN that provides limited services.

- If the RADIUS authentication server is unavailable (down) and inaccessible authentication bypass is enabled, the switch grants the client access to the network by putting the port in the critical-authentication state in the RADIUS-configured or the user-specified access VLAN.

Note Inaccessible authentication bypass is also referred to as critical authentication or the AAA fail policy.

If Multi Domain Authentication (MDA) is enabled on a port, this flow can be used with some exceptions that are applicable to voice authorization.

Figure 27: Authentication Flowchart

This figure shows the authentication process.
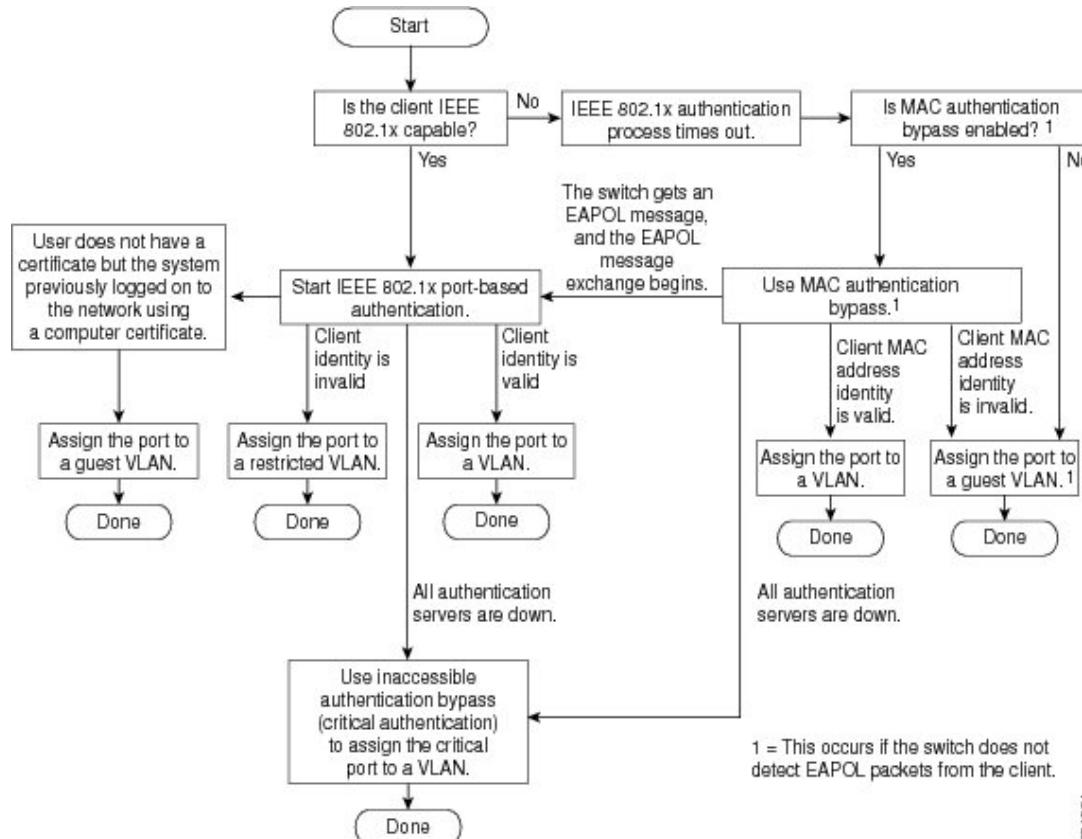
The switch re-authenticates a client when one of these situations occurs:

- Periodic re-authentication is enabled, and the re-authentication timer expires.

  You can configure the re-authentication timer to use a switch-specific value or to be based on values from the RADIUS server.

  After 802.1x authentication using a RADIUS server is configured, the switch uses timers based on the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]).

  The Session-Timeout RADIUS attribute (Attribute[27]) specifies the time after which re-authentication occurs. The range is 1 to 65535 seconds.

  The Termination-Action RADIUS attribute (Attribute [29]) specifies the action to take during re-authentication. The actions are Initialize and ReAuthenticate. When the Initialize action is set (the attribute value is DEFAULT), the 802.1x session ends, and connectivity is lost during re-authentication. When the ReAuthenticate action is set (the attribute value is RADIUS-Request), the session is not affected during re-authentication.

- You manually re-authenticate the client by entering the dot1x re-authenticate interface interface-id privileged EXEC command.

Port-Based Authentication Initiation and Message Exchange

During 802.1x authentication, the switch or the client can initiate authentication. If you enable authentication on a port by using the authentication port-control auto interface configuration command, the switch initiates authentication when the link state changes from down to up or periodically as long as the port remains up and unauthenticated. The switch sends an EAP-request/identity frame to the client to request its identity. Upon receipt of the frame, the client responds with an EAP-response/identity frame.

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

Note

If 802.1x authentication is not enabled or supported on the network access device, any EAPOL frames from the client are dropped. If the client does not receive an EAP-request/identity frame after three attempts to start authentication, the client sends frames as if the port is in the authorized state. A port in the authorized state effectively means that the client has been successfully authenticated.

When the client supplies its identity, the switch begins its role as the intermediary, passing EAP frames between the client and the authentication server until authentication succeeds or fails. If the authentication succeeds, the switch port becomes authorized. If the authentication fails, authentication can be retried, the port might be assigned to a VLAN that provides limited services, or network access is not granted.

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

Figure 28: Message Exchange

This figure shows a message exchange initiated by the client when the client uses the One-Time-Password (OTP) authentication method with a RADIUS server.
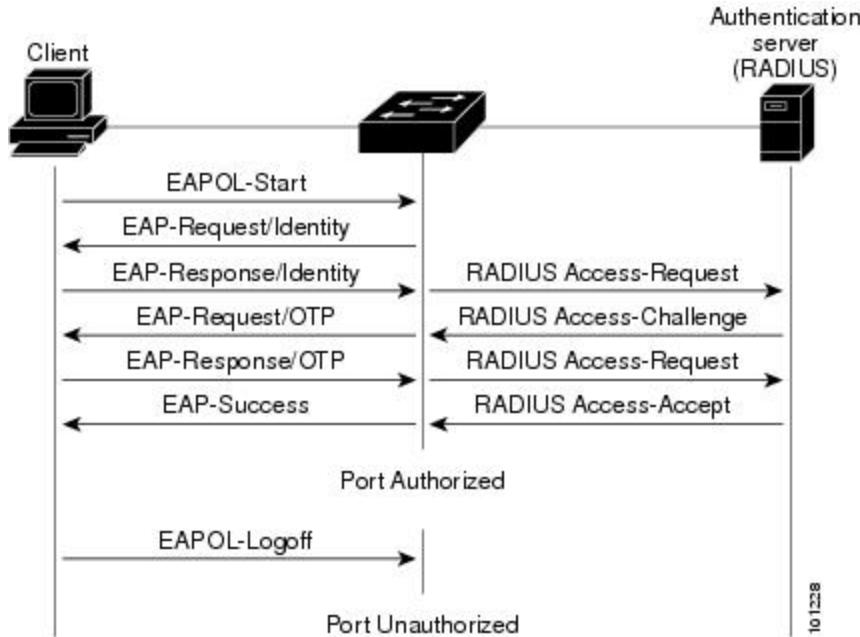
If 802.1x authentication times out while waiting for an EAPOL message exchange and MAC authentication bypass is enabled, the switch can authorize the client when the switch detects an Ethernet packet from the

client. The switch uses the MAC address of the client as its identity and includes this information in the RADIUS-access/request frame that is sent to the RADIUS server. After the server sends the switch the RADIUS-access/accept frame (authorization is successful), the port becomes authorized. If authorization fails and a guest VLAN is specified, the switch assigns the port to the guest VLAN. If the switch detects an EAPOL packet while waiting for an Ethernet packet, the switch stops the MAC authentication bypass process and starts 802.1x authentication.

Figure 29: Message Exchange During MAC Authentication Bypass

This figure shows the message exchange during MAC authentication bypass.
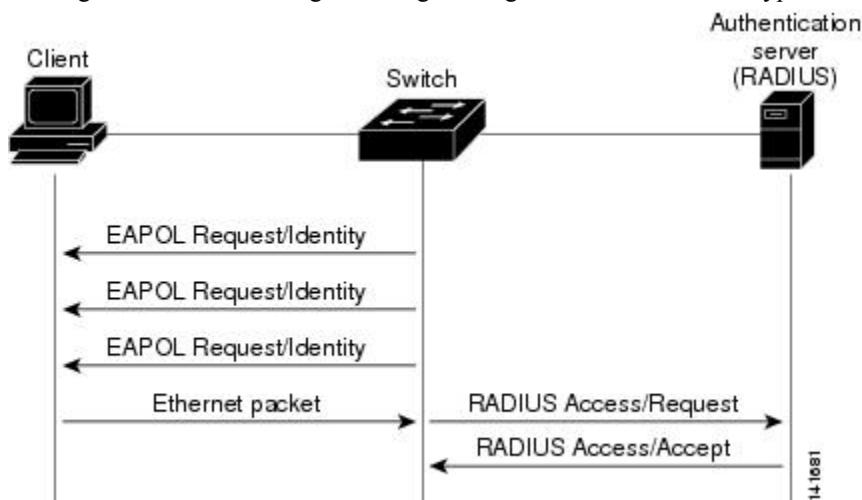
Authentication Manager for Port-Based Authentication

Port-Based Authentication Methods

Table 38: 802.1x Features

Authentication method	Mode			
	Single host	**Multiple host**	**MDA**	**Multiple Authentication**
802.1x	VLAN assignment Per-user ACL Filter-ID attribute Downloadable ACL Redirect URL	VLAN assignment	VLAN assignment Per-user ACL Filter-Id attribute Downloadable ACL Redirect URL	VLAN assignment Per-user ACL Filter-Id attribute Downloadable ACL Redirect URL

Authentication method	Mode			
	Single host	**Multiple host**	**MDA**	**Multiple Authentication**
MAC authentication bypass	VLAN assignment Per-user ACL Filter-ID attribute Downloadable ACL Redirect URL	VLAN assignment	VLAN assignment Per-user ACL Filter-Id attribute Downloadable ACL Redirect URL	VLAN assignment Per-user ACL Filter-Id attribute Downloadable ACL Redirect URL
Standalone web authentication	Proxy ACL, Filter-Id attribute, downloadable ACL			
NAC Layer 2 IP validation	Filter-Id attribute Downloadable ACL Redirect URL	Filter-Id attribute Downloadable ACL Redirect URL	Filter-Id attribute Downloadable ACL Redirect URL	Filter-Id attribute Downloadable ACL Redirect URL
Web authentication as fallback method	Proxy ACL Filter-Id attribute Downloadable ACL	Proxy ACL Filter-Id attribute Downloadable ACL	Proxy ACL Filter-Id attribute Downloadable ACL	Proxy ACL Filter-Id attribute Downloadable ACL

[9]   Supported in Cisco IOS Release 12.2(50)SE and later.
[10]   For clients that do not support 802.1x authentication.

Per-User ACLs and Filter-Ids

Note   Using role-based ACLs as Filter-Id is not recommended.

More than one host can be authenticated on MDA-enabled and multiauth ports. The ACL policy applied for one host does not effect the traffic of another host. If only one host is authenticated on a multi-host port, and the other hosts gain network access without authentication, the ACL policy for the first host can be applied to the other connected hosts by specifying any in the source address.

Port-Based Authentication Manager CLI Commands

The authentication-manager interface-configuration commands control all the authentication methods, such as 802.1x, MAC authentication bypass, and web authentication. The authentication manager commands determine the priority and order of authentication methods applied to a connected host.

The authentication manager commands control generic authentication features, such as host-mode, violation mode, and the authentication timer. Generic authentication commands include the authentication host-mode, authentication violation, and authentication timer interface configuration commands.

802.1x-specific commands begin with the dot1x keyword. For example, the authentication port-control auto interface configuration command enables authentication on an interface.

To disable dot1x on a switch, remove the configuration globally by using the no dot1x system-auth-control command, and also remove it from all configured interfaces.

Note If 802.1x authentication is globally disabled, other authentication methods are still enabled on that port, such as web authentication.

The authentication manager commands provide the same functionality as earlier 802.1x commands.

When filtering out verbose system messages generated by the authentication manager, the filtered content typically relates to authentication success. You can also filter verbose messages for 802.1x authentication and MAB authentication. There is a separate command for each authentication method:

- The no authentication logging verbose global configuration command filters verbose messages from the authentication manager.

- The no dot1x logging verbose global configuration command filters 802.1x authentication verbose messages.

- The no mab logging verbose global configuration command filters MAC authentication bypass (MAB) verbose messages

Table 39: Authentication Manager Commands and Earlier 802.1x Commands

The authentication manager commands in Cisco IOS Release 12.2(50)SE or later	The equivalent 802.1x commands in Cisco IOS Release 12.2(46)SE and earlier	Description
authentication control-direction {both | in}	dot1x control-direction {both | in}	Enable 802.1x authentication with the wake-on-LAN (WoL) feature, and configure the port control as unidirectional or bidirectional.
authentication event	dot1x auth-fail vlan dot1x critical (interface configuration) dot1x guest-vlan6	Enable the restricted VLAN on a port. Enable the inaccessible-authentication-bypass feature. Specify an active VLAN as an 802.1x guest VLAN.
authentication fallback fallback-profile	dot1x fallback fallback-profile	Configure a port to use web authentication as a fallback method for clients that do not support 802.1x authentication.
authentication host-mode [multi-auth | multi-domain | multi-host | single-host]	dot1x host-mode {single-host | multi-host | multi-domain}	Allow a single host (client) or multiple hosts on an 802.1x-authorized port.

The authentication manager commands in Cisco IOS Release 12.2(50)SE or later	The equivalent 802.1x commands in Cisco IOS Release 12.2(46)SE and earlier	Description
authentication order	mab	Provides the flexibility to define the order of authentication methods to be used.
authentication periodic	dot1x reauthentication	Enable periodic re-authentication of the client.
authentication port-control {auto | force-authorized | force-un authorized}	dot1x port-control {auto | force-authorized | force-unauthorized}	Enable manual control of the authorization state of the port.
authentication timer	dot1x timeout	Set the 802.1x timers.
authentication violation {protect | restrict | shutdown}	dot1x violation-mode {shutdown | restrict | protect}	Configure the violation modes that occur when a new device connects to a port or when a new device connects to a port after the maximum number of devices are connected to that port.

Ports in Authorized and Unauthorized States

During 802.1x authentication, depending on the switch port state, the switch can grant a client access to the network. The port starts in the unauthorized state. While in this state, the port that is not configured as a voice VLAN port disallows all ingress and egress traffic except for 802.1x authentication, CDP, and STP packets. When a client is successfully authenticated, the port changes to the authorized state, allowing all traffic for the client to flow normally. If the port is configured as a voice VLAN port, the port allows VoIP traffic and 802.1x protocol packets before the client is successfully authenticated.

Note CDP bypass is not supported and may cause a port to go into err-disabled state.

If a client that does not support 802.1x authentication connects to an unauthorized 802.1x port, the switch 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 standard, 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.

You control the port authorization state by using the authentication port-control interface configuration command and these keywords:

- force-authorized—disables 802.1x authentication and causes the port to change to the authorized state without any authentication exchange required. The port sends and receives normal traffic without 802.1x-based authentication of the client. This is the default setting.

- force-unauthorized—causes the port to remain in the unauthorized state, ignoring all attempts by the client to authenticate. The switch cannot provide authentication services to the client through the port.

- auto—enables 802.1x 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 changes from down to up or when an EAPOL-start frame is received. The switch requests the identity of the client and begins relaying authentication messages between the client and the authentication server. Each client attempting to access the network is uniquely identified by the switch by using the client MAC address.

If the client is successfully authenticated (receives an Accept frame from the authentication server), the port state changes to authorized, and all frames from the authenticated client 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 switch can resend the request. If no response is received from the server after the specified number of attempts, authentication fails, and network access is not granted.

When a client logs off, it sends an EAPOL-logoff message, causing the switch port to change to the unauthorized state.

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

Port-Based Authentication and Switch Stacks

If a switch is added to or removed from a switch stack, 802.1x authentication is not affected as long as the IP connectivity between the RADIUS server and the stack remains intact. This statement also applies if the stack's active switch is removed from the switch stack. Note that if the active switch fails, a stack member becomes the new active switch of the stack by using the election process, and the 802.1x authentication process continues as usual.

If IP connectivity to the RADIUS server is interrupted because the switch that was connected to the server is removed or fails, these events occur:

- Ports that are already authenticated and that do not have periodic re-authentication enabled remain in the authenticated state. Communication with the RADIUS server is not required.

- Ports that are already authenticated and that have periodic re-authentication enabled (with the dot1x re-authentication global configuration command) fail the authentication process when the re-authentication occurs. Ports return to the unauthenticated state during the re-authentication process. Communication with the RADIUS server is required.

  For an ongoing authentication, the authentication fails immediately because there is no server connectivity.

If the switch that failed comes up and rejoins the switch stack, the authentications might or might not fail depending on the boot-up time and whether the connectivity to the RADIUS server is re-established by the time the authentication is attempted.

To avoid loss of connectivity to the RADIUS server, you should ensure that there is a redundant connection to it. For example, you can have a redundant connection to the stack's active switch and another to a stack member, and if the active switch fails, the switch stack still has connectivity to the RADIUS server.

802.1x Host Mode

You can configure an 802.1x port for single-host or for multiple-hosts mode. In single-host mode, only one client can be connected to the 802.1x-enabled switch port. The switch detects the client by sending an EAPOL frame when the port link state changes to the up state. If a client leaves or is replaced with another client, the switch changes the port link state to down, and the port returns to the unauthorized state.

In multiple-hosts mode, you can attach multiple hosts to a single 802.1x-enabled port. In this mode, only one of the attached clients must be authorized for all clients to be granted network access. If the port becomes unauthorized (re-authentication fails or an EAPOL-logoff message is received), the switch denies network access to all of the attached clients.

The switch supports multidomain authentication (MDA), which allows both a data device and a voice device, such as an IP Phone (Cisco or non-Cisco), to connect to the same switch port.

802.1x Multiple Authentication Mode

Multiple-authentication (multiauth) mode allows multiple authenticated clients on the data VLAN and voice VLAN. Each host is individually authenticated. There is no limit to the number of data or voice device that can be authenticated on a multiauthport.

Note   When a port is in multiple-authentication mode, the authentication-failed VLAN features do not activate.

You can assign a RADIUS-server-supplied VLAN in multi-auth mode, under the following conditions:

- The host is the first host authorized on the port, and the RADIUS server supplies VLAN information

- Subsequent hosts are authorized with a VLAN that matches the operational VLAN.

- A host is authorized on the port with no VLAN assignment, and subsequent hosts either have no VLAN assignment, or their VLAN information matches the operational VLAN.

- The first host authorized on the port has a group VLAN assignment, and subsequent hosts either have no VLAN assignment, or their group VLAN matches the group VLAN on the port. Subsequent hosts must use the same VLAN from the VLAN group as the first host. If a VLAN list is used, all hosts are subject to the conditions specified in the VLAN list.

- After a VLAN is assigned to a host on the port, subsequent hosts must have matching VLAN information or be denied access to the port.

- You cannot configure a guest VLAN or an auth-fail VLAN in multi-auth mode.

- The behavior of the critical-auth VLAN is not changed for multi-auth mode. When a host tries to authenticate and the server is not reachable, all authorized hosts are reinitialized in the configured VLAN.

Multi-auth Per User VLAN assignment

The Multi-auth Per User VLAN assignment feature allows you to create multiple operational access VLANs based on VLANs assigned to the clients on the port that has a single configured access VLAN. The port configured as an access port where the traffic for all the VLANs associated with data domain is not dot1q tagged, and these VLANs are treated as native VLANs.

The number of hosts per multi-auth port is 8, however there can be more hosts.

The following scenarios are associated with the multi-auth Per User VLAN assignments:

Scenario one

When a hub is connected to an access port, and the port is configured with an access VLAN (V0).

The host (H1) is assigned to VLAN (V1) through the hub. The operational VLAN of the port is changed to V1. This behaviour is similar on a single-host or multi-domain-auth port.

When a second host (H2) is connected and gets assigned to VLAN (V2), the port will have two operational VLANs (V1 and V2). If H1 and H2 sends untagged ingress traffic, H1 traffic is mapped to VLAN (V1) and H2 traffic to VLAN (V2), all egress traffic going out of the port on VLAN (V1) and VLAN (V2) are untagged.

If both the hosts, H1 and H2 are logged out or the sessions are removed due to some reason then VLAN (V1) and VLAN (V2) are removed from the port, and the configured VLAN (V0) is restored on the port.

Scenario two

When a hub is connected to an access port, and the port is configured with an access VLAN (V0). The host (H1) is assigned to VLAN (V1) through the hub. The operational VLAN of the port is changed to V1.

When a second host (H2) is connected and gets authorized without explicit vlan policy, H2 is expected to use the configured VLAN (V0) that is restored on the port. A ll egress traffic going out of two operational VLANs, VLAN (V0) and VLAN (V1) are untagged.

If host (H2) is logged out or the session is removed due to some reason then the configured VLAN (V0) is removed from the port, and VLAN (V1) becomes the only operational VLAN on the port.

Scenario three

When a hub is connected to an access port in open mode, and the port is configured with an access VLAN (V0) .

The host (H1) is assigned to VLAN (V1) through the hub. The operational VLAN of the port is changed to V1. When a second host (H2) is connected and remains unauthorized, it still has access to operational VLAN (V1) due to open mode.

If host H1 is logged out or the session is removed due to some reason, VLAN (V1) is removed from the port and host (H2) gets assigned to VLAN (V0).

Note   The combination of Open mode and VLAN assignment has an adverse affect on host (H2) because it has an IP address in the subnet that corresponds to VLAN (V1).

Limitation in Multi-auth Per User VLAN assignment

In the Multi-auth Per User VLAN assignment feature, egress traffic from multiple vlans are untagged on a port where the hosts receive traffic that is not meant for them. This can be a problem with broadcast and multicast traffic.

- IPv4 ARPs: Hosts receive ARP packets from other subnets. This is a problem if two subnets in different Virtual Routing and Forwarding (VRF) tables with overlapping IP address range are active on the port. The host ARP cache may get invalid entries.

- IPv6 control packets: In IPv6 deployments, Router Advertisements (RA) are processed by hosts that are not supposed to receive them. When a host from one VLAN receives RA from a different VLAN, the host assign incorrect IPv6 address to itself. Such a host is unable to get access to the network.

The workaround is to enable the IPv6 first hop security so that the broadcast ICMPv6 packets are converted to unicast and sent out from multi-auth enabled ports.. The packet is replicated for each client in multi-auth port belonging to the VLAN and the destination MAC is set to an individual client. Ports having one VLAN, ICMPv6 packets broadcast normally.

- IP multicast: Multicast traffic destined to a multicast group gets replicated for different VLANs if the hosts on those VLANs join the multicast group. When two hosts in different VLANs join a multicast group (on the same mutli-auth port), two copies of each multicast packet are sent out from that port.

MAC Move

When a MAC address is authenticated on one switch port, that address is not allowed on another authentication manager-enabled port of the switch. If the switch detects that same MAC address on another authentication manager-enabled port, the address is not allowed.

There are situations where a MAC address might need to move from one port to another on the same switch. For example, when there is another device (for example a hub or an IP phone) between an authenticated host and a switch port, you might want to disconnect the host from the device and connect it directly to another port on the same switch.

You can globally enable MAC move so the device is reauthenticated on the new port. When a host moves to a second port, the session on the first port is deleted, and the host is reauthenticated on the new port. MAC move is supported on all host modes. (The authenticated host can move to any port on the switch, no matter which host mode is enabled on the that port.) When a MAC address moves from one port to another, the switch terminates the authenticated session on the original port and initiates a new authentication sequence on the new port. The MAC move feature applies to both voice and data hosts.

Note In open authentication mode, a MAC address is immediately moved from the original port to the new port, with no requirement for authorization on the new port.

MAC Replace

The MAC replace feature can be configured to address the violation that occurs when a host attempts to connect to a port where another host was previously authenticated.

Note This feature does not apply to ports in multi-auth mode, because violations are not triggered in that mode. It does not apply to ports in multiple host mode, because in that mode, only the first host requires authentication.

If you configure the authentication violation interface configuration command with the replace keyword, the authentication process on a port in multi-domain mode is:

- A new MAC address is received on a port with an existing authenticated MAC address.

- The authentication manager replaces the MAC address of the current data host on the port with the new MAC address.

- The authentication manager initiates the authentication process for the new MAC address.

- If the authentication manager determines that the new host is a voice host, the original voice host is removed.

If a port is in open authentication mode, any new MAC address is immediately added to the MAC address table.

802.1x Accounting

The 802.1x standard defines how users are authorized and authenticated for network access but does not keep track of network usage. 802.1x accounting is disabled by default. You can enable 802.1x accounting to monitor this activity on 802.1x-enabled ports:

- User successfully authenticates.

- User logs off.

- Link-down occurs.

- Re-authentication successfully occurs.

- Re-authentication fails.

The switch does not log 802.1x accounting information. Instead, it sends this information to the RADIUS server, which must be configured to log accounting messages.

802.1x Accounting Attribute-Value Pairs

The information sent to the RADIUS server is represented in the form of Attribute-Value (AV) pairs. These AV pairs provide data for different applications. (For example, a billing application might require information that is in the Acct-Input-Octets or the Acct-Output-Octets attributes of a RADIUS packet.)

AV pairs are automatically sent by a switch that is configured for 802.1x accounting. Three types of RADIUS accounting packets are sent by a switch:

- START–sent when a new user session starts

- INTERIM–sent during an existing session for updates

- STOP–sent when a session terminates

Note To view debug logs for RADIUS and AAA, use the show platform software trace message smd command. For more information, see the Tracing Commands section in Command Reference Guide, .

This table lists the AV pairs and when they are sent are sent by the switch.

Table 40: Accounting AV Pairs

Attribute Number	AV Pair Name	START	INTERIM	STOP
Attribute[1]	User-Name	Always	Always	Always
Attribute[4]	NAS-IP-Address	Always	Always	Always

Attribute Number	AV Pair Name	START	INTERIM	STOP
Attribute[5]	NAS-Port	Always	Always	Always
Attribute[8]	Framed-IP-Address	Never	Sometimes[11]	Sometimes
Attribute[30]	Called-Station-ID	Always	Always	Always
Attribute[31]	Calling-Station-ID	Always	Always	Always
Attribute[40]	Acct-Status-Type	Always	Always	Always
Attribute[41]	Acct-Delay-Time	Always	Always	Always
Attribute[42]	Acct-Input-Octets	Never	Always	Always
Attribute[43]	Acct-Output-Octets	Never	Always	Always
Attribute[47]	Acct-Input-Packets	Never	Always	Always
Attribute[48]	Acct-Output-Packets	Never	Always	Always
Attribute[44]	Acct-Session-ID	Always	Always	Always
Attribute[45]	Acct-Authentic	Always	Always	Always
Attribute[46]	Acct-Session-Time	Never	Always	Always
Attribute[49]	Acct-Terminate-Cause	Never	Never	Always
Attribute[61]	NAS-Port-Type	Always	Always	Always

[11] The Framed-IP-Address AV pair is sent when a valid static IP address is configured or w when a Dynamic Host Control Protocol (DHCP) binding exists for the host in the DHCP snooping bindings table.

802.1x Readiness Check

The 802.1x readiness check monitors 802.1x activity on all the switch ports and displays information about the devices connected to the ports that support 802.1x. You can use this feature to determine if the devices connected to the switch ports are 802.1x-capable. You use an alternate authentication such as MAC authentication bypass or web authentication for the devices that do not support 802.1x functionality.

This feature only works if the supplicant on the client supports a query with the NOTIFY EAP notification packet. The client must respond within the 802.1x timeout value.

Switch-to-RADIUS-Server Communication

RADIUS security servers are identified by their hostname or IP address, hostname and specific UDP port numbers, or IP address and specific UDP port numbers. The combination of the IP address and UDP port number creates a unique identifier, which enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, authentication—the second host entry configured acts as the fail-over backup to the first one. The RADIUS host entries are tried in the order that they were configured.

802.1x Authentication with VLAN Assignment

The switch supports 802.1x authentication with VLAN assignment. After successful 802.1x authentication of a port, the RADIUS server sends the VLAN assignment to configure the switch port. The RADIUS server database maintains the username-to-VLAN mappings, assigning the VLAN based on the username of the client connected to the switch port. You can use this feature to limit network access for certain users.

Voice device authentication is supported with multidomain host mode in Cisco IOS Release 12.2(37)SE. In Cisco IOS Release 12.2(40)SE and later, when a voice device is authorized and the RADIUS server returned an authorized VLAN, the voice VLAN on the port is configured to send and receive packets on the assigned voice VLAN. Voice VLAN assignment behaves the same as data VLAN assignment on multidomain authentication (MDA)-enabled ports.

When configured on the switch and the RADIUS server, 802.1x authentication with VLAN assignment has these characteristics:

- If no VLAN is supplied by the RADIUS server or if 802.1x authentication is disabled, the port is configured in its access VLAN after successful authentication. Recall that an access VLAN is a VLAN assigned to an access port. All packets sent from or received on this port belong to this VLAN.

- If 802.1x authentication is enabled but the VLAN information from the RADIUS server is not valid, authorization fails and configured VLAN remains in use. This prevents ports from appearing unexpectedly in an inappropriate VLAN because of a configuration error.

  Configuration errors could include specifying a VLAN for a routed port, a malformed VLAN ID, a nonexistent or internal (routed port) VLAN ID, an RSPAN VLAN, a shut down or suspended VLAN. In the case of a multidomain host port, configuration errors can also be due to an attempted assignment of a data VLAN that matches the configured or assigned voice VLAN ID (or the reverse).

- If 802.1x authentication is enabled and all information from the RADIUS server is valid, the authorized device is placed in the specified VLAN after authentication.

- If the multiple-hosts mode is enabled on an 802.1x port, all hosts are placed in the same VLAN (specified by the RADIUS server) as the first authenticated host.

- Enabling port security does not impact the RADIUS server-assigned VLAN behavior.

- If 802.1x authentication is disabled on the port, it is returned to the configured access VLAN and configured voice VLAN.

- If an 802.1x port is authenticated and put in the RADIUS server-assigned VLAN, any change to the port access VLAN configuration does not take effect. In the case of a multidomain host, the same applies to voice devices when the port is fully authorized with these exceptions:

  - If the VLAN configuration change of one device results in matching the other device configured or assigned VLAN, then authorization of all devices on the port is terminated and multidomain host mode is disabled until a valid configuration is restored where data and voice device configured VLANs no longer match.

  - If a voice device is authorized and is using a downloaded voice VLAN, the removal of the voice VLAN configuration, or modifying the configuration value to dot1p or untagged results in voice device un-authorization and the disablement of multi-domain host mode.

When the port is in the force authorized, force unauthorized, unauthorized, or shutdown state, it is put into the configured access VLAN.

To configure VLAN assignment you need to perform these tasks:

- Enable AAA authorization by using the network keyword to allow interface configuration from the RADIUS server.

- Enable 802.1x authentication. (The VLAN assignment feature is automatically enabled when you configure 802.1x authentication on an access port).

- Assign vendor-specific tunnel attributes in the RADIUS server. The RADIUS server must return these attributes to the switch:

    - [64] Tunnel-Type = VLAN

    - [65] Tunnel-Medium-Type = 802

    - [81] Tunnel-Private-Group-ID = VLAN name or VLAN ID

    - [83] Tunnel-Preference

Attribute [64] must contain the value VLAN (type 13). Attribute [65] must contain the value 802 (type 6). Attribute [81] specifies the VLAN name or VLAN ID assigned to the IEEE 802.1x-authenticated user.

802.1x Authentication with Per-User ACLs

You can enable per-user access control lists (ACLs) to provide different levels of network access and service to an 802.1x-authenticated user. When the RADIUS server authenticates a user connected to an 802.1x port, it retrieves the ACL attributes based on the user identity and sends them to the switch. The switch applies the attributes to the 802.1x port for the duration of the user session. The switch removes the per-user ACL configuration when the session is over, if authentication fails, or if a link-down condition occurs. The switch does not save RADIUS-specified ACLs in the running configuration. When the port is unauthorized, the switch removes the ACL from the port.

You can configure router ACLs and input port ACLs on the same switch. However, a port ACL takes precedence over a router ACL. If you apply input port ACL to an interface that belongs to a VLAN, the port ACL takes precedence over an input router ACL applied to the VLAN interface. Incoming packets received on the port, to which a port ACL is applied, are filtered by the port ACL. Incoming routed packets received on other ports are filtered by the router ACL. Outgoing routed packets are filtered by the router ACL. To avoid configuration conflicts, you should carefully plan the user profiles stored on the RADIUS server.

RADIUS supports per-user attributes, including vendor-specific attributes. These vendor-specific attributes (VSAs) are in octet-string format and are passed to the switch during the authentication process. The VSAs used for per-user ACLs are inacl#<n> for the ingress direction and outacl#<n> for the egress direction. MAC ACLs are supported only in the ingress direction. The switch supports VSAs only in the ingress direction. It does not support port ACLs in the egress direction on Layer 2 ports.

Use only the extended ACL syntax style to define the per-user configuration stored on the RADIUS server. When the definitions are passed from the RADIUS server, they are created by using the extended naming convention. However, if you use the Filter-Id attribute, it can point to a standard ACL.

You can use the Filter-Id attribute to specify an inbound or outbound ACL that is already configured on the switch. The attribute contains the ACL number followed by .in for ingress filtering or .out for egress filtering. If the RADIUS server does not allow the .in or .out syntax, the access list is applied to the outbound ACL by default. The user is marked unauthorized if the Filter-Id sent from the RADIUS server is not configured on

the device. Because of limited support of Cisco IOS access lists on the switch, the Filter-Id attribute is supported only for IP ACLs numbered in the range of 1 to 199 (IP standard ACLs) and 1300 to 2699 (IP extended ACLs).

The maximum size of the per-user ACL is 4000 ASCII characters but is limited by the maximum size of RADIUS-server per-user ACLs.

You must meet the following prerequisites to configure per-user ACLs:

- Enable AAA authentication.

- Enable AAA authorization by using the network keyword to allow interface configuration from the RADIUS server.

- Enable 802.1x authentication.

- Configure the user profile and VSAs on the RADIUS server.

- Configure the 802.1x port for single-host mode.

> **Note** Per-user ACLs are supported only in single-host mode.

802.1x Authentication with Downloadable ACLs and Redirect URLs

You can download ACLs and redirect URLs from a RADIUS server to the switch during 802.1x authentication or MAC authentication bypass of the host. You can also download ACLs during web authentication.

> **Note** A downloadable ACL is also referred to as a dACL.

You can apply the ACLs and redirect URLs to all the devices connected to the 802.1x-enabled port.

If no ACLs are downloaded during 802.1x authentication, the switch applies the static default ACL on the port to the host. On a voice VLAN port configured in multi-auth or MDA mode, the switch applies the ACL only to the phone as part of the authorization policies.

For a URL redirect ACL:

- Packets that match a permit access control entry (ACE) rule are sent to the CPU for forwarding to the AAA server.

- Packets that match a deny ACE rule are forwarded through the switch.

- Packets that match neither the permit ACE rule or deny ACE rule are processed by the next dACL, and if there is no dACL, the packets hit the implicit-deny ACL and are dropped.

Cisco Secure ACS and Attribute-Value Pairs for the Redirect URL

The switch uses these cisco-av-pair VSAs:

- url-redirect is the HTTP or HTTPS URL.

- url-redirect-acl is the switch ACL name or number.

The switch uses the CiscoSecure-defined-ACL attribute value pair to intercept an HTTP or HTTPS request from the end point. The switch then forwards the client web browser to the specified redirect address. The url-redirect AV pair on the Cisco Secure ACS contains the URL to which the web browser is redirected. The url-redirect-acl attribute value pair contains the name or number of an ACL that specifies the HTTP or HTTPS traffic to redirect.

Note
- Traffic that matches a permit ACE in the ACL is redirected.

- Define the URL redirect ACL and the default port ACL on the switch.

If a redirect URL is configured for a client on the authentication server, a default port ACL on the connected client switch port must also be configured.

Cisco Secure ACS and Attribute-Value Pairs for Downloadable ACLs

You can set the CiscoSecure-Defined-ACL Attribute-Value (AV) pair on the Cisco Secure ACS with the RADIUS cisco-av-pair vendor-specific attributes (VSAs). This pair specifies the names of the downloadable ACLs on the Cisco Secure ACS with the #ACL#-IP-name-number attribute.

- The name is the ACL name.

- The number is the version number (for example, 3f783768).

If a downloadable ACL is configured for a client on the authentication server, a default port ACL on the connected client switch port must also be configured.

If the default ACL is configured on the switch and the Cisco Secure ACS sends a host-access-policy to the switch, it applies the policy to traffic from the host connected to a switch port. If the policy does not apply, the switch applies the default ACL. If the Cisco Secure ACS sends the switch a downloadable ACL, this ACL takes precedence over the default ACL that is configured on the switch port. However, if the switch receives an host access policy from the Cisco Secure ACS but the default ACL is not configured, the authorization failure is declared.

VLAN ID-Based MAC Authentication

You can use VLAN ID-based MAC authentication if you wish to authenticate hosts based on a static VLAN ID instead of a downloadable VLAN. When you have a static VLAN policy configured on your switch, VLAN information is sent to an IAS (Microsoft) RADIUS server along with the MAC address of each host for authentication. The VLAN ID configured on the connected port is used for MAC authentication. By using VLAN ID-based MAC authentication with an IAS server, you can have a fixed number of VLANs in the network.

The feature also limits the number of VLANs monitored and handled by STP. The network can be managed as a fixed VLAN.

802.1x Authentication with Guest VLAN

You can configure a guest VLAN for each 802.1x port on the switch to provide limited services to clients, such as downloading the 802.1x client. These clients might be upgrading their system for 802.1x authentication, and some hosts, such as Windows 98 systems, might not be IEEE 802.1x-capable.

When you enable a guest VLAN on an 802.1x port, the switch assigns clients to a guest VLAN when the switch does not receive a response to its EAP request/identity frame or when EAPOL packets are not sent by the client.

The switch maintains the EAPOL packet history. If an EAPOL packet is detected on the interface during the lifetime of the link, the switch determines that the device connected to that interface is an IEEE 802.1x-capable supplicant, and the interface does not change to the guest VLAN state. EAPOL history is cleared if the interface link status goes down. If no EAPOL packet is detected on the interface, the interface changes to the guest VLAN state.

If the switch is trying to authorize an 802.1x-capable voice device and the AAA server is unavailable, the authorization attempt fails, but the detection of the EAPOL packet is saved in the EAPOL history. When the AAA server becomes available, the switch authorizes the voice device. However, the switch no longer allows other devices access to the guest VLAN. To prevent this situation, use one of these command sequences:

- Enter the authentication event no-response action authorize vlan vlan-id interface configuration command to allow access to the guest VLAN.

- Enter the shutdown interface configuration command followed by the no shutdown interface configuration command to restart the port.

If devices send EAPOL packets to the switch during the lifetime of the link, the switch no longer allows clients that fail authentication access to the guest VLAN.

Note   If an EAPOL packet is detected after the interface has changed to the guest VLAN, the interface reverts to an unauthorized state, and 802.1x authentication restarts.

Any number of 802.1x-incapable clients are allowed access when the switch port is moved to the guest VLAN. If an 802.1x-capable client joins the same port on which the guest VLAN is configured, the port is put into the unauthorized state in the user-configured access VLAN, and authentication is restarted.

Guest VLANs are supported on 802.1x ports in single host, multiple host, multi-auth and multi-domain modes.

You can configure any active VLAN except an RSPAN VLAN, a private VLAN, or a voice VLAN as an 802.1x guest VLAN. The guest VLAN feature is not supported on internal VLANs (routed ports) or trunk ports; it is supported only on access ports.

The switch supports MAC authentication bypass. When MAC authentication bypass is enabled on an 802.1x port, the switch can authorize clients based on the client MAC address when IEEE 802.1x authentication times out while waiting for an EAPOL message exchange. After detecting a client on an 802.1x port, the switch waits for an Ethernet packet from the client. The switch sends the authentication server a RADIUS-access/request frame with a username and password based on the MAC address. If authorization succeeds, the switch grants the client access to the network. If authorization fails, the switch assigns the port to the guest VLAN if one is specified.

802.1x Authentication with Restricted VLAN

You can configure a restricted VLAN (also referred to as an authentication failed VLAN) for each IEEE 802.1x port on a switch stack or a switch to provide limited services to clients that cannot access the guest VLAN. These clients are 802.1x-compliant and cannot access another VLAN because they fail the authentication process. A restricted VLAN allows users without valid credentials in an authentication server (typically,

visitors to an enterprise) to access a limited set of services. The administrator can control the services available to the restricted VLAN.

Note

You can configure a VLAN to be both the guest VLAN and the restricted VLAN if you want to provide the same services to both types of users.

Without this feature, the client attempts and fails authentication indefinitely, and the switch port remains in the spanning-tree blocking state. With this feature, you can configure the switch port to be in the restricted VLAN after a specified number of authentication attempts (the default value is 3 attempts).

The authenticator counts the failed authentication attempts for the client. When this count exceeds the configured maximum number of authentication attempts, the port moves to the restricted VLAN. The failed attempt count increments when the RADIUS server replies with either an EAP failure or an empty response without an EAP packet. When the port moves into the restricted VLAN, the failed attempt counter resets.

Users who fail authentication remain in the restricted VLAN until the next re-authentication attempt. A port in the restricted VLAN tries to re-authenticate at configured intervals (the default is 60 seconds). If re-authentication fails, the port remains in the restricted VLAN. If re-authentication is successful, the port moves either to the configured VLAN or to a VLAN sent by the RADIUS server. You can disable re-authentication. If you do this, the only way to restart the authentication process is for the port to receive a link down or EAP logoff event. We recommend that you keep re-authentication enabled if a client might connect through a hub. When a client disconnects from the hub, the port might not receive the link down or EAP logoff event.

After a port moves to the restricted VLAN, a simulated EAP success message is sent to the client. This prevents clients from indefinitely attempting authentication. Some clients (for example, devices running Windows XP) cannot implement DHCP without EAP success.

Restricted VLANs are supported on 802.1x ports in all host modes and on Layer 2 ports.

You can configure any active VLAN except an RSPAN VLAN, a primary private VLAN, or a voice VLAN as an 802.1x restricted VLAN. The restricted VLAN feature is not supported on internal VLANs (routed ports) or trunk ports; it is supported only on access ports.

Other security port features such as dynamic ARP Inspection, DHCP snooping, and IP source guard can be configured independently on a restricted VLAN.

802.1x Authentication with Inaccessible Authentication Bypass

Use the inaccessible authentication bypass feature, also referred to as critical authentication or the AAA fail policy, when the switch cannot reach the configured RADIUS servers and new hosts cannot be authenticated. You can configure the switch to connect those hosts to critical ports.

When a new host tries to connect to the critical port, that host is moved to a user-specified access VLAN, the critical VLAN. The administrator gives limited authentication to the hosts.

When the switch tries to authenticate a host connected to a critical port, the switch checks the status of the configured RADIUS server. If a server is available, the switch can authenticate the host. However, if all the RADIUS servers are unavailable, the switch grants network access to the host and puts the port in the critical-authentication state, which is a special case of the authentication state.

Note If critical authentication is configured on interface, then vlan used for critical authorization (critical vlan) should be active on the switch. If the critical vlan is inactive (or) down, critical authentication session will keep trying to enable inactive vlan and fail repeatedly. This can lead to large amount of memory holding.

Inaccessible Authentication Bypass Support on Multiple-Authentication Ports

When a port is configured on any host mode and the AAA server is unavailable, the port is then configured to multi-host mode and moved to the critical VLAN. To support this inaccessible bypass on multiple-authentication (multiauth) ports, use the authentication event server dead action reinitialize vlan vlan-id command. When a new host tries to connect to the critical port, that port is reinitialized and all the connected hosts are moved to the user-specified access VLAN.

This command is supported on all host modes.

Inaccessible Authentication Bypass Authentication Results

The behavior of the inaccessible authentication bypass feature depends on the authorization state of the port:

- If the port is unauthorized when a host connected to a critical port tries to authenticate and all servers are unavailable, the switch puts the port in the critical-authentication state in the RADIUS-configured or user-specified access VLAN.

- If the port is already authorized and reauthentication occurs, the switch puts the critical port in the critical-authentication state in the current VLAN, which might be the one previously assigned by the RADIUS server.

- If the RADIUS server becomes unavailable during an authentication exchange, the current exchange times out, and the switch puts the critical port in the critical-authentication state during the next authentication attempt.

You can configure the critical port to reinitialize hosts and move them out of the critical VLAN when the RADIUS server is again available. When this is configured, all critical ports in the critical-authentication state are automatically re-authenticated.

Inaccessible Authentication Bypass Feature Interactions

Inaccessible authentication bypass interacts with these features:

- Guest VLAN—Inaccessible authentication bypass is compatible with guest VLAN. When a guest VLAN is enabled on 8021.x port, the features interact as follows:

  - If at least one RADIUS server is available, the switch assigns a client to a guest VLAN when the switch does not receive a response to its EAP request/identity frame or when EAPOL packets are not sent by the client.

  - If all the RADIUS servers are not available and the client is connected to a critical port, the switch authenticates the client and puts the critical port in the critical-authentication state in the RADIUS-configured or user-specified access VLAN.

  - If all the RADIUS servers are not available and the client is not connected to a critical port, the switch might not assign clients to the guest VLAN if one is configured.

- If all the RADIUS servers are not available and if a client is connected to a critical port and was previously assigned to a guest VLAN, the switch keeps the port in the guest VLAN.

- Restricted VLAN—If the port is already authorized in a restricted VLAN and the RADIUS servers are unavailable, the switch puts the critical port in the critical-authentication state in the restricted VLAN.

- 802.1x accounting—Accounting is not affected if the RADIUS servers are unavailable.

- Private VLAN—You can configure inaccessible authentication bypass on a private VLAN host port. The access VLAN must be a secondary private VLAN.

- Voice VLAN—Inaccessible authentication bypass is compatible with voice VLAN, but the RADIUS-configured or user-specified access VLAN and the voice VLAN must be different.

- Remote Switched Port Analyzer (RSPAN)—Do not configure an RSPAN VLAN as the RADIUS-configured or user-specified access VLAN for inaccessible authentication bypass.

802.1x Critical Voice VLAN

When an IP phone connected to a port is authenticated by the Cisco Identity Services Engine (ISE), the phone is put into the voice domain. If the ISE is not reachable, the switch cannot determine if the device is a voice device. If the server is unavailable, the phone cannot access the voice network and therefore cannot operate.

For data traffic, you can configure inaccessible authentication bypass, or critical authentication, to allow traffic to pass through on the native VLAN when the server is not available. If the RADIUS authentication server is unavailable (down) and inaccessible authentication bypass is enabled, the switch grants the client access to the network and puts the port in the critical-authentication state in the RADIUS-configured or the user-specified access VLAN. When the switch cannot reach the configured RADIUS servers and new hosts cannot be authenticated, the switch connects those hosts to critical ports. A new host trying to connect to the critical port is moved to a user-specified access VLAN, the critical VLAN, and granted limited authentication.

Note Dynamic assignment of critical voice VLAN is not supported with nested service templates. It causes the device to switch between VLANs continuously in a loop.

You can enter the authentication event server dead action authorize voice interface configuration command to configure the critical voice VLAN feature. When the ISE does not respond, the port goes into critical authentication mode. When traffic coming from the host is tagged with the voice VLAN, the connected device (the phone) is put in the configured voice VLAN for the port. The IP phones learn the voice VLAN identification through Cisco Discovery Protocol (Cisco devices) or through LLDP or DHCP.

You can configure the voice VLAN for a port by entering the switchport voice vlan vlan-id interface configuration command.

This feature is supported in multidomain and multi-auth host modes. Although you can enter the command when the switch in single-host or multi-host mode, the command has no effect unless the device changes to multidomain or multi-auth host mode.

802.1x User Distribution

You can configure 802.1x user distribution to load-balance users with the same group name across multiple different VLANs.

The VLANs are either supplied by the RADIUS server or configured through the switch CLI under a VLAN group name.

- Configure the RADIUS server to send more than one VLAN name for a user. The multiple VLAN names can be sent as part of the response to the user. The 802.1x user distribution tracks all the users in a particular VLAN and achieves load balancing by moving the authorized user to the least populated VLAN.

- Configure the RADIUS server to send a VLAN group name for a user. The VLAN group name can be sent as part of the response to the user. You can search for the selected VLAN group name among the VLAN group names that you configured by using the switch CLI. If the VLAN group name is found, the corresponding VLANs under this VLAN group name are searched to find the least populated VLAN. Load balancing is achieved by moving the corresponding authorized user to that VLAN.

> **Note** The RADIUS server can send the VLAN information in any combination of VLAN-IDs, VLAN names, or VLAN groups.

802.1x User Distribution Configuration Guidelines

- Confirm that at least one VLAN is mapped to the VLAN group.

- You can map more than one VLAN to a VLAN group.

- You can modify the VLAN group by adding or deleting a VLAN.

- When you clear an existing VLAN from the VLAN group name, none of the authenticated ports in the VLAN are cleared, but the mappings are removed from the existing VLAN group.

- If you clear the last VLAN from the VLAN group name, the VLAN group is cleared.

- You can clear a VLAN group even when the active VLANs are mapped to the group. When you clear a VLAN group, none of the ports or users that are in the authenticated state in any VLAN within the group are cleared, but the VLAN mappings to the VLAN group are cleared.

IEEE 802.1x Authentication with Voice VLAN Ports

A voice VLAN port is a special access port associated with two VLAN identifiers:

- VVID to carry voice traffic to and from the IP phone. The VVID is used to configure the IP phone connected to the port.

- PVID to carry the data traffic to and from the workstation connected to the switch through the IP phone. The PVID is the native VLAN of the port.

The IP phone uses the VVID for its voice traffic, regardless of the authorization state of the port. This allows the phone to work independently of IEEE 802.1x authentication.

In single-host mode, only the IP phone is allowed on the voice VLAN. In multiple-hosts mode, additional clients can send traffic on the voice VLAN after a supplicant is authenticated on the PVID. When multiple-hosts mode is enabled, the supplicant authentication affects both the PVID and the VVID.

A voice VLAN port becomes active when there is a link, and the device MAC address appears after the first CDP message from the IP phone. Cisco IP phones do not relay CDP messages from other devices. As a result, if several IP phones are connected in series, the switch recognizes only the one directly connected to it. When IEEE 802.1x authentication is enabled on a voice VLAN port, the switch drops packets from unrecognized IP phones more than one hop away.

When IEEE 802.1x authentication is enabled on a switch port, you can configure an access port VLAN that is also a voice VLAN.

When IP phones are connected to an 802.1x-enabled switch port that is in single host mode, the switch grants the phones network access without authenticating them. We recommend that you use multidomain authentication (MDA) on the port to authenticate both a data device and a voice device, such as an IP phone

Note If you enable IEEE 802.1x authentication on an access port on which a voice VLAN is configured and to which a Cisco IP Phone is connected, the Cisco IP phone loses connectivity to the switch for up to 30 seconds.

IEEE 802.1x Authentication with Wake-on-LAN

The IEEE 802.1x authentication with wake-on-LAN (WoL) feature allows dormant PCs to be powered when the switch receives a specific Ethernet frame, known as the magic packet. You can use this feature in environments where administrators need to connect to systems that have been powered down.

When a host that uses WoL is attached through an IEEE 802.1x port and the host powers off, the IEEE 802.1x port becomes unauthorized. The port can only receive and send EAPOL packets, and WoL magic packets cannot reach the host. When the PC is powered off, it is not authorized, and the switch port is not opened.

When the switch uses IEEE 802.1x authentication with WoL, the switch forwards traffic to unauthorized IEEE 802.1x ports, including magic packets. While the port is unauthorized, the switch continues to block ingress traffic other than EAPOL packets. The host can receive packets but cannot send packets to other devices in the network.

Note If PortFast is not enabled on the port, the port is forced to the bidirectional state.

When you configure a port as unidirectional by using the authentication control-direction in interface configuration command, the port changes to the spanning-tree forwarding state. The port can send packets to the host but cannot receive packets from the host.

When you configure a port as bidirectional by using the authentication control-direction both interface configuration command, the port is access-controlled in both directions. The port does not receive packets from or send packets to the host.

IEEE 802.1x Authentication with MAC Authentication Bypass

You can configure the switch to authorize clients based on the client MAC address by using the MAC authentication bypass feature. For example, you can enable this feature on IEEE 802.1x ports connected to devices such as printers.

If IEEE 802.1x authentication times out while waiting for an EAPOL response from the client, the switch tries to authorize the client by using MAC authentication bypass.

When the MAC authentication bypass feature is enabled on an IEEE 802.1x port, the switch uses the MAC address as the client identity. The authentication server has a database of client MAC addresses that are allowed network access. After detecting a client on an IEEE 802.1x port, the switch waits for an Ethernet packet from the client. The switch sends the authentication server a RADIUS-access/request frame with a username and password based on the MAC address. If authorization succeeds, the switch grants the client access to the network. If authorization fails, the switch assigns the port to the guest VLAN if one is configured. This process works for most client devices; however, it does not work for clients that use an alternate MAC address format. You can configure how MAB authentication is performed for clients with MAC addresses that deviate from the standard format or where the RADIUS configuration requires the user name and password to differ.

If an EAPOL packet is detected on the interface during the lifetime of the link, the switch 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 switch already authorized a port by using MAC authentication bypass and detects an IEEE 802.1x supplicant, the switch does not unauthorize the client connected to the port. When re-authentication occurs, the switch uses the authentication or re-authentication methods configured on the port, if the previous session ended because the Termination-Action RADIUS attribute value is DEFAULT.

Clients that were authorized with MAC authentication bypass can be re-authenticated. The re-authentication process is the same as that for clients that were authenticated with IEEE 802.1x. During re-authentication, the port remains in the previously assigned VLAN. If re-authentication is successful, the switch keeps the port in the same VLAN. If re-authentication fails, the switch assigns the port to the guest VLAN, if one is configured.

If re-authentication 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 re-authentication. If MAC authentication bypass is enabled and the IEEE 802.1x authentication times out, the switch uses the MAC authentication bypass feature to initiate re-authorization. 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 features:

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

- Guest VLAN—If a client has an invalid MAC address identity, the switch assigns the client to a guest VLAN if one is configured.

- Restricted VLAN—This feature is not supported when the client connected to an IEEE 802.lx port is authenticated with MAC authentication bypass.

- Port security

- Voice VLAN

- Private VLAN—You can assign a client to a private VLAN.

- Network Edge Access Topology (NEAT)—MAB and NEAT are mutually exclusive. You cannot enable MAB when NEAT is enabled on an interface, and you should not enable NEAT when MAB is enabled on an interface.

Cisco IOS Release 12.2(55)SE and later supports filtering of verbose MAB system messages

Network Admission Control Layer 2 IEEE 802.1x Validation

The switch supports the Network Admission Control (NAC) Layer 2 IEEE 802.1x validation, which checks the antivirus condition or posture of endpoint systems or clients before granting the devices network access. With NAC Layer 2 IEEE 802.1x validation, you can do these tasks:

- Download the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute[29]) from the authentication server.

- Set the number of seconds between re-authentication attempts as the value of the Session-Timeout RADIUS attribute (Attribute[27]) and get an access policy against the client from the RADIUS server.

- Set the action to be taken when the switch tries to re-authenticate the client by using the Termination-Action RADIUS attribute (Attribute[29]). If the value is the DEFAULT or is not set, the session ends. If the value is RADIUS-Request, the re-authentication process starts.

- Set the list of VLAN number or name or VLAN group name as the value of the Tunnel Group Private ID (Attribute[81]) and the preference for the VLAN number or name or VLAN group name as the value of the Tunnel Preference (Attribute[83]). If you do not configure the Tunnel Preference, the first Tunnel Group Private ID (Attribute[81]) attribute is picked up from the list.

- View the NAC posture token, which shows the posture of the client, by using the show authentication privileged EXEC command.

- Configure secondary private VLANs as guest VLANs.

Configuring NAC Layer 2 IEEE 802.1x validation is similar to configuring IEEE 802.1x port-based authentication except that you must configure a posture token on the RADIUS server.

Flexible Authentication Ordering

You can use flexible authentication ordering to configure the order of methods that a port uses to authenticate a new host. The IEEE 802.1X Flexible Authentication feature supports three authentication methods:

- dot1X—IEEE 802.1X authentication is a Layer 2 authentication method.

- mab—MAC-Authentication Bypass is a Layer 2 authentication method.

- webauth—Web authentication is a Layer 3 authentication method.

Using this feature, you can control which ports use which authentication methods, and you can control the failover sequencing of methods on those ports. For example, MAC authentication bypass and 802.1x can be the primary or secondary authentication methods, and web authentication can be the fallback method if either or both of those authentication attempts fail.

The IEEE 802.1X Flexible Authentication feature supports the following host modes:

- multi-auth—Multiauthentication allows one authentication on a voice VLAN and multiple authentications on the data VLAN.

- multi-domain—Multidomain authentication allows two authentications: one on the voice VLAN and one on the data VLAN.

Open1x Authentication

Open1x authentication allows a device access to a port before that device is authenticated. When open authentication is configured, a new host can pass traffic according to the access control list (ACL) defined on the port. After the host is authenticated, the policies configured on the RADIUS server are applied to that host.

You can configure open authentication with these scenarios:

- Single-host mode with open authentication–Only one user is allowed network access before and after authentication.

- MDA mode with open authentication–Only one user in the voice domain and one user in the data domain are allowed.

- Multiple-hosts mode with open authentication–Any host can access the network.

- Multiple-authentication mode with open authentication–Similar to MDA, except multiple hosts can be authenticated.

Note If open authentication is configured, it takes precedence over other authentication controls. This means that if you use the authentication open interface configuration command, the port will grant access to the host irrespective of the authentication port-control interface configuration command.

Multidomain Authentication

The switch supports multidomain authentication (MDA), which allows both a data device and voice device, such as an IP phone (Cisco or non-Cisco), to authenticate on the same switch port. The port is divided into a data domain and a voice domain.

Note For all host modes, the line protocol stays up before authorization when port-based authentication is configured.

MDA does not enforce the order of device authentication. However, for best results, we recommend that a voice device is authenticated before a data device on an MDA-enabled port.

Follow these guidelines for configuring MDA:

- You must configure a switch port for MDA.

- You must configure the voice VLAN for the IP phone when the host mode is set to multidomain.

- Voice VLAN assignment on an MDA-enabled port is supported Cisco IOS Release 12.2(40)SE and later.

- To authorize a voice device, the AAA server must be configured to send a Cisco Attribute-Value (AV) pair attribute with a value of device-traffic-class=voice. Without this value, the switch treats the voice device as a data device.

Note	When traffic-class=voice is downloaded from AAA servers as a service-template, a session will be created in DATA domain instead of VOICE domain.

- The guest VLAN and restricted VLAN features only apply to the data devices on an MDA-enabled port. The switch treats a voice device that fails authorization as a data device.

- If more than one device attempts authorization on either the voice or the data domain of a port, it is error disabled.

- Until a device is authorized, the port drops its traffic. Non-Cisco IP phones or voice devices are allowed into both the data and voice VLANs. The data VLAN allows the voice device to contact a DHCP server to obtain an IP address and acquire the voice VLAN information. After the voice device starts sending on the voice VLAN, its access to the data VLAN is blocked.

- A voice device MAC address that is binding on the data VLAN is not counted towards the port security MAC address limit.

- MDA can use MAC authentication bypass as a fallback mechanism to allow the switch port to connect to devices that do not support IEEE 802.1x authentication.

- When a data or a voice device is detected on a port, its MAC address is blocked until authorization succeeds. If the authorization fails, the MAC address remains blocked for 5 minutes.

- If more than five devices are detected on the data VLAN or more than one voice device is detected on the voice VLAN while a port is unauthorized, the port is error disabled.

- When a port host mode is changed from single- or multihost to multidomain mode, an authorized data device remains authorized on the port. However, a Cisco IP phone that has been allowed on the port voice VLAN is automatically removed and must be reauthenticated on that port.

- Active fallback mechanisms such as guest VLAN and restricted VLAN remain configured after a port changes from single- or multihost mode to multidomain mode.

- Switching a port host mode from multidomain to single- or multihost mode removes all authorized devices from the port.

- If a data domain is authorized first and placed in the guest VLAN, non-IEEE 802.1x-capable voice devices need to tag their packets on the voice VLAN to trigger authentication.

- We do not recommend per-user ACLs with an MDA-enabled port. An authorized device with a per-user ACL policy might impact traffic on both the voice and data VLANs of the port. If used, only one device on the port should enforce per-user ACLs.

802.1x Supplicant and Authenticator Switches with Network Edge Access Topology (NEAT)

The Network Edge Access Topology (NEAT) feature extends identity to areas outside the wiring closet (such as conference rooms). This allows any type of device to authenticate on the port.

- 802.1x switch supplicant: You can configure a switch to act as a supplicant to another switch by using the 802.1x supplicant feature. This configuration is helpful in a scenario, where, for example, a switch

is outside a wiring closet and is connected to an upstream switch through a trunk port. A switch configured with the 802.1x switch supplicant feature authenticates with the upstream switch for secure connectivity. Once the supplicant switch authenticates successfully the port mode changes from access to trunk in an authenticator switch. In a supplicant switch you must manually configure trunk when enabling CISP.

• If the access VLAN is configured on the authenticator switch, it becomes the native VLAN for the trunk port after successful authentication.

In the default state, when you connect a supplicant switch to an authenticator switch that has BPDU guard enabled, the authenticator port could be error-disabled if it receives a Spanning Tree Protocol (STP) bridge protocol data unit (BPDU) packets before the supplicant switch has authenticated. Beginning with Cisco IOS Release 15.0(1)SE, you can control traffic exiting the supplicant port during the authentication period. Entering the dot1x supplicant controlled transient global configuration command temporarily blocks the supplicant port during authentication to ensure that the authenticator port does not shut down before authentication completes. If authentication fails, the supplicant port opens. Entering the no dot1x supplicant controlled transient global configuration command opens the supplicant port during the authentication period. This is the default behavior.

We strongly recommend using the dot1x supplicant controlled transientcommand on a supplicant switch when BPDU guard is enabled on the authenticator switch port with the spanning-tree bpduguard enable interface configuration command.

Note   If you globally enable BPDU guard on the authenticator switch by using the spanning-tree portfast bpduguard default global configuration command, entering the dot1x supplicant controlled transient command does not prevent the BPDU violation.

You can enable MDA or multiauth mode on the authenticator switch interface that connects to one more supplicant switches. Multihost mode is not supported on the authenticator switch interface.

When you reboot an authenticator switch with single-host mode enabled on the interface, the interface may move to err-disabled state before authentication. To recover from err-disabled state, flap the authenticator port to activate the interface again and initiate authentication.

Use the dot1x supplicant force-multicast global configuration command on the supplicant switch for Network Edge Access Topology (NEAT) to work in all host modes.

• Host Authorization: Ensures that only traffic from authorized hosts (connecting to the switch with supplicant) is allowed on the network. The switches use Client Information Signalling Protocol (CISP) to send the MAC addresses connecting to the supplicant switch to the authenticator switch.

• Auto enablement: Automatically enables trunk configuration on the authenticator switch, allowing user traffic from multiple VLANs coming from supplicant switches. Configure the cisco-av-pair as device-traffic-class=switch at the ISE. (You can configure this under the group or the user settings.)

Figure 30: Authenticator and Supplicant Switch using CISP
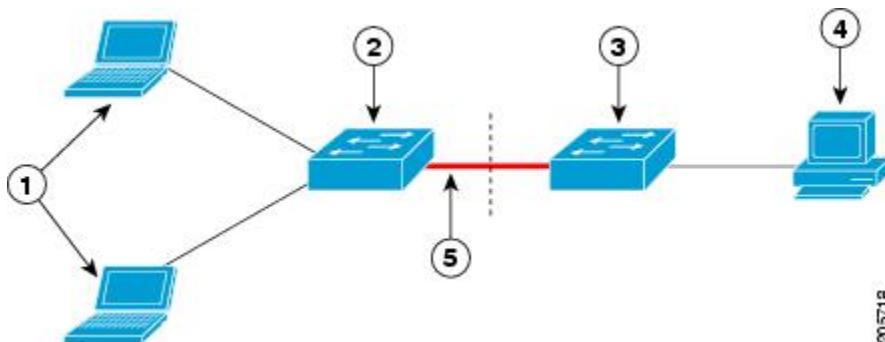
1	Workstations (clients)	2	Supplicant switch (outside wiring closet)
3	Authenticator switch	4	Cisco ISE
5	Trunk port		

Note The switchport nonegotiate command is not supported on supplicant and authenticator switches with NEAT. This command should not be configured at the supplicant side of the topology. If configured on the authenticator side, the internal macros will automatically remove this command from the port.

Voice Aware 802.1x Security

Note To use voice aware IEEE 802.1x authentication, the switch must be running the LAN base image.

You use the voice aware 802.1x security feature to configure the switch to disable only the VLAN on which a security violation occurs, whether it is a data or voice VLAN. In previous releases, when an attempt to authenticate the data client caused a security violation, the entire port shut down, resulting in a complete loss of connectivity.

You can use this feature in IP phone deployments where a PC is connected to the IP phone. A security violation found on the data VLAN results in the shutdown of only the data VLAN. The traffic on the voice VLAN flows through the switch without interruption.

Common Session ID

Authentication manager uses a single session ID (referred to as a common session ID) for a client no matter which authentication method is used. This ID is used for all reporting purposes, such as the show commands and MIBs. The session ID appears with all per-session syslog messages.

The session ID includes:

- The IP address of the Network Access Device (NAD)

- A monotonically increasing unique 32 bit integer

- The session start time stamp (a 32 bit integer)

This example shows how the session ID appears in the output of the show authentication command. The session ID in this example is 160000050000000B288508E5:

```
Device# show authentication sessions
Interface   MAC Address    Method   Domain   Status        Session ID
Fa4/0/4     0000.0000.0203 mab      DATA     Authz Success 160000050000000B288508E5
```

This is an example of how the session ID appears in the syslog output. The session ID in this example is also160000050000000B288508E5:

```
1w0d: %AUTHMGR-5-START: Starting 'mab' for client (0000.0000.0203) on Interface Fa4/0/4
AuditSessionID 160000050000000B288508E5
1w0d: %MAB-5-SUCCESS: Authentication successful for client (0000.0000.0203) on Interface
Fa4/0/4 AuditSessionID 160000050000000B288508E5
1w0d: %AUTHMGR-7-RESULT: Authentication result 'success' from 'mab' for client
(0000.0000.0203) on Interface Fa4/0/4 AuditSessionID 160000050000000B288508E5
```

The session ID is used by the NAD, the AAA server, and other report-analyzing applications to identify the client. The ID appears automatically. No configuration is required.

How to Configure 802.1x Port-Based Authentication

Default 802.1x Authentication Configuration

Table 41: Default 802.1x Authentication Configuration

Feature	Default Setting
Switch 802.1x enable state	Disabled.
Per-port 802.1x enable state	Disabled (force-authorized). The port sends and receives normal traffic without 802.1x-based authentication of the client.
AAA	Disabled.

RADIUS server
  • IP address
  • UDP authentication port
  • Default accounting port
  • Key
	• None specified.
	• 1645.
	• 1646.
	• None specified.

Host mode	Single-host mode.

Feature	Default Setting
Control direction	Bidirectional control.
Periodic re-authentication	Disabled.
Number of seconds between re-authentication attempts	3600 seconds.
Re-authentication number	2 times (number of times that the switch restarts the authentication process before the port changes to the unauthorized state).
Quiet period	60 seconds (number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client).
Retransmission time	30 seconds (number of seconds that the switch should wait for a response to an EAP request/identity frame from the client before resending the request).
Maximum retransmission number	2 times (number of times that the switch will send an EAP-request/identity frame before restarting the authentication process).
Client timeout period	30 seconds (when relaying a request from the authentication server to the client, the amount of time the switch waits for a response before resending the request to the client.)
Authentication server timeout period	30 seconds (when relaying a response from the client to the authentication server, the amount of time the switch waits for a reply before resending the response to the server.) You can change this timeout period by using the dot1x timeout server-timeout interface configuration command.
Inactivity timeout	Disabled.
Guest VLAN	None specified.
Inaccessible authentication bypass	Disabled.
Restricted VLAN	None specified.
Authenticator (switch) mode	None specified.
MAC authentication bypass	Disabled.
Voice-aware security	Disabled.

802.1x Authentication Configuration Guidelines

802.1x Authentication

These are the 802.1x authentication configuration guidelines:

- You must enable SISF-Based device tracking to use 802.1x authentication. By default, SISF-Based device tracking is disabled on a switch.

- When 802.1x authentication is enabled, ports are authenticated before any other Layer 2 or Layer 3 features are enabled.

- If the VLAN to which an 802.1x-enabled port is assigned changes, this change is transparent and does not affect the switch. For example, this change occurs if a port is assigned to a RADIUS server-assigned VLAN and is then assigned to a different VLAN after re-authentication.

  If the VLAN to which an 802.1x port is assigned to shut down, disabled, or removed, the port becomes unauthorized. For example, the port is unauthorized after the access VLAN to which a port is assigned shuts down or is removed.

- The 802.1x protocol is supported on Layer 2 static-access ports, voice VLAN ports, and Layer 3 routed ports, but it is not supported on these port types:

  - Dynamic ports—A port in dynamic mode can negotiate with its neighbor to become a trunk port. If you try to enable 802.1x authentication on a dynamic port, an error message appears, and 802.1x authentication is not enabled. If you try to change the mode of an 802.1x-enabled port to dynamic, an error message appears, and the port mode is not changed.

  - EtherChannel port—Do not configure a port that is an active or a not-yet-active member of an EtherChannel as an 802.1x port. If you try to enable 802.1x authentication on an EtherChannel port, an error message appears, and 802.1x authentication is not enabled.

  - Switched Port Analyzer (SPAN) and Remote SPAN (RSPAN) destination ports—You can enable 802.1x authentication on a port that is a SPAN or RSPAN destination port. However, 802.1x authentication is disabled until the port is removed as a SPAN or RSPAN destination port. You can enable 802.1x authentication on a SPAN or RSPAN source port.

- Before globally enabling 802.1x authentication on a switch by entering the dot1x system-auth-control global configuration command, remove the EtherChannel configuration from the interfaces on which 802.1x authentication and EtherChannel are configured.

- Cisco IOS Release 12.2(55)SE and later supports filtering of system messages related to 802.1x authentication.

VLAN Assignment, Guest VLAN, Restricted VLAN, and Inaccessible Authentication Bypass

These are the configuration guidelines for VLAN assignment, guest VLAN, restricted VLAN, and inaccessible authentication bypass:

- When 802.1x authentication is enabled on a port, you cannot configure a port VLAN that is equal to a voice VLAN.

- You can configure any VLAN except an RSPAN VLAN or a voice VLAN as an 802.1x guest VLAN. The guest VLAN feature is not supported on internal VLANs (routed ports) or trunk ports; it is supported only on access ports.

- After you configure a guest VLAN for an 802.1x port to which a DHCP client is connected, you might need to get a host IP address from a DHCP server. You can change the settings for restarting the 802.1x authentication process on the switch before the DHCP process on the client times out and tries to get a host IP address from the DHCP server. Decrease the settings for the 802.1x authentication process (authentication timer inactivity and authentication timer reauthentication interface configuration commands). The amount to decrease the settings depends on the connected 802.1x client type.

- When configuring the inaccessible authentication bypass feature, follow these guidelines:

  - The feature is supported on 802.1x port in single-host mode and multihosts mode.

  - If the client is running Windows XP and the port to which the client is connected is in the critical-authentication state, Windows XP might report that the interface is not authenticated.

  - If the Windows XP client is configured for DHCP and has an IP address from the DHCP server, receiving an EAP-Success message on a critical port might not re-initiate the DHCP configuration process.

  - You can configure the inaccessible authentication bypass feature and the restricted VLAN on an 802.1x port. If the switch tries to re-authenticate a critical port in a restricted VLAN and all the RADIUS servers are unavailable, switch changes the port state to the critical authentication state and remains in the restricted VLAN.

  - If the CTS links are in Critical Authentication mode and the active switch reloads, the policy where SGT was configured on a device will not be available on the new active switch. This is because the internal bindings will not be synced to the standby switch in a 3750-X switch stack.

- You can configure any VLAN except an RSPAN VLAN or a voice VLAN as an 802.1x restricted VLAN. The restricted VLAN feature is not supported on internal VLANs (routed ports) or trunk ports; it is supported only on access ports.

MAC Authentication Bypass

These are the MAC authentication bypass configuration guidelines:

- Unless otherwise stated, the MAC authentication bypass guidelines are the same as the 802.1x authentication guidelines.

- If you disable MAC authentication bypass from a port after the port has been authorized with its MAC address, the port state is not affected.

- If the port is in the unauthorized state and the client MAC address is not the authentication-server database, the port remains in the unauthorized state. However, if the client MAC address is added to the database, the switch can use MAC authentication bypass to re-authorize the port.

- If the port is in the authorized state, the port remains in this state until re-authorization occurs.

- You can configure a timeout period for hosts that are connected by MAC authentication bypass but are inactive. The range is 1to 65535 seconds.

Maximum Number of Allowed Devices Per Port

This is the maximum number of devices allowed on an 802.1x-enabled port:

- In single-host mode, only one device is allowed on the access VLAN. If the port is also configured with a voice VLAN, an unlimited number of Cisco IP phones can send and receive traffic through the voice VLAN.

- In multidomain authentication (MDA) mode, one device is allowed for the access VLAN, and one IP phone is allowed for the voice VLAN.

- In multihost mode, only one 802.1x supplicant is allowed on the port, but an unlimited number of non-802.1x hosts are allowed on the access VLAN. An unlimited number of devices are allowed on the voice VLAN.

Configuring 802.1x Readiness Check

The 802.1x readiness check monitors 802.1x activity on all the switch ports and displays information about the devices connected to the ports that support 802.1x. You can use this feature to determine if the devices connected to the switch ports are 802.1x-capable.

The 802.1x readiness check is allowed on all ports that can be configured for 802.1x. The readiness check is not available on a port that is configured as dot1x force-unauthorized.

Follow these steps to enable the 802.1x readiness check on the switch:

Before you begin

Follow these guidelines to enable the readiness check on the switch:

- The readiness check is typically used before 802.1x is enabled on the switch.
- If you use the dot1x test eapol-capable privileged EXEC command without specifying an interface, all the ports on the switch stack are tested.
- When you configure the dot1x test eapol-capable command on an 802.1x-enabled port, and the link comes up, the port queries the connected client about its 802.1x capability. When the client responds with a notification packet, it is 802.1x-capable. A syslog message is generated if the client responds within the timeout period. If the client does not respond to the query, the client is not 802.1x-capable. No syslog message is generated
- When you configure the dot1x test eapol-capable command on an 802.1x-enabled port, and the link comes up, the port queries the connected client about its 802.1x capability. When the client responds with a notification packet, it is 802.1x-capable. A syslog message is generated if the client responds within the timeout period. If the client does not respond to the query, the client is not 802.1x-capable. No syslog message is generated
- The readiness check can be sent on a port that handles multiple hosts (for example, a PC that is connected to an IP phone). A syslog message is generated for each of the clients that respond to the readiness check within the timer period.

Procedure

	Command or Action	**Purpose**
Step 1	enable	Enables privileged EXEC mode.
	Example:	• Enter your password if prompted.
	Device> **enable**	

	Command or Action	**Purpose**
Step 2	configure terminal	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	dot1x test eapol-capable [interface interface-id]	Enables the 802.1x readiness check on the switch.
	Example:	(Optional) For interface-id specify the port on which to check for IEEE 802.1x readiness.
	Device# **dot1x test eapol-capable interface gigabitethernet1/0/13** DOT1X_PORT_EAPOL_CAPABLE:DOT1X: MAC 00-01-02-4b-f1-a3 on gigabitethernet1/0/13 is EAPOL capable	**Note** If you omit the optional interface keyword, all interfaces on the switch are tested.
Step 4	dot1x test timeout timeout	(Optional) Configures the timeout used to wait for EAPOL response. The range is from 1 to 65535 seconds. The default is 10 seconds.
Step 5	end	Returns to privileged EXEC mode.
	Example:	
	Device(config)# **end**	
Step 6	show running-config	Verifies your entries.
	Example:	
	Device# **show running-config**	
Step 7	copy running-config startup-config	(Optional) Saves your entries in the configuration file.
	Example:	
	Device# **copy running-config startup-config**	

Configuring Voice Aware 802.1x Security

Note   To use voice aware IEEE 802.1x authentication, the switch must be running the LAN base image.

You use the voice aware 802.1x security feature on the switch to disable only the VLAN on which a security violation occurs, whether it is a data or voice VLAN. You can use this feature in IP phone deployments where a PC is connected to the IP phone. A security violation found on the data VLAN results in the shutdown of only the data VLAN. The traffic on the voice VLAN flows through the switch without interruption.

Follow these guidelines to configure voice aware 802.1x voice security on the switch:

- You enable voice aware 802.1x security by entering the errdisable detect cause security-violation shutdown vlan global configuration command. You disable voice aware 802.1x security by entering the no version of this command. This command applies to all 802.1x-configured ports in the switch.

> **Note** If you do not include the shutdown vlan keywords, the entire port is shut down when it enters the error-disabled state.

- If you use the errdisable recovery cause security-violation global configuration command to configure error-disabled recovery, the port is automatically re-enabled. If error-disabled recovery is not configured for the port, you re-enable it by using the shutdown and no shutdown interface configuration commands.

- You can re-enable individual VLANs by using the clear errdisable interface interface-id vlan [vlan-list] privileged EXEC command. If you do not specify a range, all VLANs on the port are enabled.

Beginning in privileged EXEC mode, follow these steps to enable voice aware 802.1x security:

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal	Enter global configuration mode.
Step 2	errdisable detect cause security-violation shutdown vlan	Shut down any VLAN on which a security violation error occurs. **Note** If the shutdown vlan keywords are not included, the entire port enters the error-disabled state and shuts down.
Step 3	errdisable recovery cause security-violation	Enter global configuration mode.
Step 4	clear errdisable interfaceinterface-id vlan [vlan-list]	(Optional) Reenable individual VLANs that have been error disabled. • For interface-id specify the port on which to reenable individual VLANs. • (Optional) For vlan-list specify a list of VLANs to be re-enabled. If vlan-list is not specified, all VLANs are re-enabled.
Step 5	Enter the following: • shutdown • no shutdown	(Optional) Re-enable an error-disabled VLAN, and clear all error-disable indications.
Step 6	end	Return to privileged EXEC mode.
Step 7	show errdisable detect	Verify your entries.

Example

This example shows how to configure the switch to shut down any VLAN on which a security violation error occurs:

```
Switch(config)# errdisable detect cause security-violation shutdown vlan
```

This example shows how to re-enable all VLANs that were error disabled on port Gigabit Ethernet 40/2.

```
Switch# clear errdisable interface gigabitethernet40/2
vlan
```

You can verify your settings by entering the show errdisable detect privileged EXEC command.

Configuring 802.1x Violation Modes

You can configure an 802.1x port so that it shuts down, generates a syslog error, or discards packets from a new device when:

- a device connects to an 802.1x-enabled port
- the maximum number of allowed about devices have been authenticated on the port

Beginning in privileged EXEC mode, follow these steps to configure the security violation actions on the switch:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	aaa new-model **Example:** Device(config)# **aaa new-model**	Enables AAA.
Step 3	aaa authentication dot1x {default} method1 **Example:** Device(config)# **aaa authentication dot1x** **default group radius**	Creates an 802.1x authentication method list. To create a default list that is used when a named list is not specified in the authentication command, use the default keyword followed by the method that is to be used in default situations. The default method list is automatically applied to all ports.

	Command or Action	Purpose
		For method1, enter the group radius keywords to use the list of all RADIUS servers for authentication.
Step 4	interface interface-id **Example:** Device(config)# **interface gigabitethernet1/0/4**	Specifies the port connected to the client that is to be enabled for IEEE 802.1x authentication, and enter interface configuration mode.
Step 5	switchport mode access **Example:** Device(config-if)# **switchport mode access**	Sets the port to access mode.
Step 6	authentication violation {shutdown | restrict | protect | replace} **Example:** Device(config-if)# **authentication violation restrict**	Configures the violation mode. The keywords have these meanings: • shutdown–Error disable the port. • restrict–Generate a syslog error. • protect–Drop packets from any new device that sends traffic to the port. • replace–Removes the current session and authenticates with the new host.
Step 7	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

Configuring 802.1x Authentication

To allow per-user ACLs or VLAN assignment, you must enable AAA authorization to configure the switch for all network-related service requests.

This is the 802.1x AAA process:

Before you begin

To configure 802.1x port-based authentication, you must enable authentication, authorization, and accounting (AAA) and specify the authentication method list. A method list describes the sequence and authentication method to be queried to authenticate a user.

Procedure

	Command or Action	Purpose
Step 1	A user connects to a port on the switch.	
Step 2	Authentication is performed.	
Step 3	VLAN assignment is enabled, as appropriate, based on the RADIUS server configuration.	
Step 4	The switch sends a start message to an accounting server.	
Step 5	Re-authentication is performed, as necessary.	
Step 6	The switch sends an interim accounting update to the accounting server that is based on the result of re-authentication.	
Step 7	The user disconnects from the port.	
Step 8	The switch sends a stop message to the accounting server.	

Configuring 802.1x Port-Based Authentication

Perform these steps to configure 802.1x port-based authentication:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	aaa new-model **Example:** Device(config)# **aaa new-model**	Enables AAA.
Step 4	aaa authentication dot1x {default} method1 **Example:**	Creates an 802.1x authentication method list. To create a default list that is used when a named list is not specified in the authentication

	Command or Action	Purpose
	`Device(config)# aaa authentication dot1x default group radius`	command, use the default keyword followed by the method that is to be used in default situations. The default method list is automatically applied to all ports.
		For method1, enter the group radius keywords to use the list of all RADIUS servers for authentication.
		Note Though other keywords are visible in the command-line help string, only the group radius keywords are supported.
Step 5	dot1x system-auth-control **Example:** `Device(config)# dot1x system-auth-control`	Enables 802.1x authentication globally on the switch.
Step 6	aaa authorization network {default} group radius **Example:** `Device(config)# aaa authorization network default group radius`	(Optional) Configures the switch to use user-RADIUS authorization for all network-related service requests, such as per-user ACLs or VLAN assignment.
Step 7	radius server server name **Example:** `Device(config)# radius server rsim address ipv4 124.2.2.12`	(Optional) Specifies the IP address of the RADIUS server.
Step 8	address {ipv4	ipv6} ip address **Example:** `Device(config-radius-server)# address ipv4 10.0.1.12`	Configures the IP address for the RADIUS server.
Step 9	key string **Example:** `Device(config-radius-server)# key rad123`	(Optional) Specifies the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.

	Command or Action	**Purpose**
Step 10	exit **Example:** Device(config-radius-server)# **exit**	Exits the RADIUS server mode and enters the global configuration mode.
Step 11	interface interface-id **Example:** Device(config)# **interface gigabitethernet 1/0/2**	Specifies the port connected to the client that is to be enabled for IEEE 802.1x authentication, and enter interface configuration mode.
Step 12	switchport mode access **Example:** Device(config-if)# **switchport mode access**	(Optional) Sets the port to access mode only if you configured the RADIUS server in Step 6 and Step 7.
Step 13	authentication port-control auto **Example:** Device(config-if)# **authentication port-control auto**	Enables 802.1x authentication on the port.
Step 14	dot1x pae authenticator **Example:** Device(config-if)# **dot1x pae authenticator**	Sets the interface Port Access Entity to act only as an authenticator and ignore messages meant for a supplicant.
Step 15	end **Example:** Device(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Switch-to-RADIUS-Server Communication

Follow these steps to configure the RADIUS server parameters:

Procedure

	Command or Action	**Purpose**
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	**Purpose**
	Example: Device> **enable**	• Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip radius source-interface vlan vlan interface number **Example:** Device(config)# **ip radius source-interface vlan 80**	Specifies that the RADIUS packets have the IP address of the indicated interface.
Step 4	radius server server name **Example:** Device(config)# **radius server rsim address ipv4 172.16.0.1**	(Optional) Specifies the IP address of the RADIUS server.
Step 5	address {ipv4 | ipv6} ip address **Example:** Device(config-radius-server)# **address ipv4 10.0.1.2 auth-port 1550 acct-port 1560**	Configures the IP address for the RADIUS server.
Step 6	key string **Example:** Device(config-radius-server)# **key rad123**	(Optional) Specifies the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.
Step 7	exit **Example:** Device(config-radius-server)# **exit**	Exits the RADIUS server mode and enters the global configuration mode.
Step 8	radius-server dead-criteria tries num-tries **Example:**	Specifies the number of unanswered sent messages to a RADIUS server before considering the server to be inactive. The range of num-tries is 1 to 100.

	Command or Action	Purpose
	Device(config)# **radius-server dead-criteria tries 30**	
Step 9	end **Example:** Device(config)# **end**	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the Host Mode

Beginning in privileged EXEC mode, follow these steps to allow multiple hosts (clients) on an IEEE 802.1x-authorized port that has the authentication port-control interface configuration command set to auto. Use the multi-domain keyword to configure and enable multidomain authentication (MDA), which allows both a host and a voice device, such as an IP phone (Cisco or non-Cisco), on the same switch port. This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet2/0/1**	Specifies the port to which multiple hosts are indirectly attached, and enter interface configuration mode.
Step 3	authentication host-mode [multi-auth | multi-domain | multi-host | single-host] **Example:** Device(config-if)# **authentication host-mode multi-host**	Allows multiple hosts (clients) on an 802.1x-authorized port. The keywords have these meanings: • multi-auth–Allow multiple authenticated clients on both the voice VLAN and data VLAN. **Note** The multi-auth keyword is only available with the authentication host-mode command.

	Command or Action	**Purpose**
		• multi-host–Allow multiple hosts on an 802.1x-authorized port after a single host has been authenticated.
		• multi-domain–Allow both a host and a voice device, such as an IP phone (Cisco or non-Cisco), to be authenticated on an IEEE 802.1x-authorized port.
		Note You must configure the voice VLAN for the IP phone when the host mode is set to multi-domain.
		Make sure that the authentication port-control interface configuration command is set to auto for the specified interface.
Step 4	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

Configuring Periodic Re-Authentication

You can enable periodic 802.1x client re-authentication and specify how often it occurs. If you do not specify a time period before enabling re-authentication, the number of seconds between attempts is 3600.

Beginning in privileged EXEC mode, follow these steps to enable periodic re-authentication of the client and to configure the number of seconds between re-authentication attempts. This procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet2/0/1**	Specifies the port to be configured, and enter interface configuration mode.

	Command or Action	Purpose
Step 3	authentication periodic **Example:** Device(config-if)# **authentication periodic**	Enables periodic re-authentication of the client, which is disabled by default. **Note** The default value is 3600 seconds. To change the value of the reauthentication timer or to have the switch use a RADIUS-provided session timeout, enter the authentication timer reauthenticate command.
Step 4	authentication timer {{[inactivity | reauthenticate | restart | unauthorized]} {value}} **Example:** Device(config-if)# **authentication timer reauthenticate 180**	Sets the number of seconds between re-authentication attempts. The authentication timer keywords have these meanings: • inactivity—Interval in seconds after which if there is no activity from the client then it is unauthorized • reauthenticate—Time in seconds after which an automatic re-authentication attempt is initiated • restart value—Interval in seconds after which an attempt is made to authenticate an unauthorized port • unauthorized value—Interval in seconds after which an unauthorized session will get deleted This command affects the behavior of the switch only if periodic re-authentication is enabled.
Step 5	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

Changing the Quiet Period

When the switch cannot authenticate the client, the switch remains idle for a set period of time and then tries again. The authentication timer restart interface configuration command controls the idle period. A failed authentication of the client might occur because the client provided an invalid password. You can provide a faster response time to the user by entering a number smaller than the default.

Beginning in privileged EXEC mode, follow these steps to change the quiet period. This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet2/0/1**	Specifies the port to be configured, and enter interface configuration mode.
Step 3	authentication timer restart seconds **Example:** Device(config-if)# **authentication timer restart 30**	Sets the number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client. The range is 1 to 65535 seconds; the default is 60.
Step 4	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.
Step 5	show authentication sessions interface interface-id **Example:** Device# **show authentication sessions interface gigabitethernet2/0/1**	Verifies your entries.
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Changing the Switch-to-Client Retransmission Time

The client responds to the EAP-request/identity frame from the switch with an EAP-response/identity frame. If the switch does not receive this response, it waits a set period of time (known as the retransmission time) and then resends the frame.

> **Note** You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

Beginning in privileged EXEC mode, follow these steps to change the amount of time that the switch waits for client notification. This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet2/0/1**	Specifies the port to be configured, and enter interface configuration mode.
Step 3	authentication timer reauthenticate seconds **Example:** Device(config-if)# **authentication timer reauthenticate 60**	Sets the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before resending the request. The range is 1 to 65535 seconds; the default is 5.
Step 4	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.
Step 5	show authentication sessions interface interface-id **Example:** Device# **show authentication sessions interface gigabitethernet2/0/1**	Verifies your entries.
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Setting the Switch-to-Client Frame-Retransmission Number

In addition to changing the switch-to-client retransmission time, you can change the number of times that the switch sends an EAP-request/identity frame (assuming no response is received) to the client before restarting the authentication process.

✎

Note   You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

Beginning in privileged EXEC mode, follow these steps to set the switch-to-client frame-retransmission number. This procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet2/0/1**	Specifies the port to be configured, and enter interface configuration mode.
Step 3	dot1x max-reauth-req count **Example:** Device(config-if)# **dot1x max-reauth-req 5**	Sets the number of times that the switch sends an EAP-request/identity frame to the client before restarting the authentication process. The range is 1 to 10; the default is 2.
Step 4	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

Setting the Re-Authentication Number

You can also change the number of times that the switch restarts the authentication process before the port changes to the unauthorized state.

![Note icon]

Note You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

Beginning in privileged EXEC mode, follow these steps to set the re-authentication number. This procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device# **interface gigabitethernet2/0/1**	Specifies the port to be configured, and enter interface configuration mode.
Step 3	switchport mode access **Example:** Device(config-if)# **switchport mode access**	Sets the port to access mode only if you previously configured the RADIUS server.
Step 4	dot1x max-req count **Example:** Device(config-if)# **dot1x max-req 4**	Sets the number of times that the switch restarts the authentication process before the port changes to the unauthorized state. The range is 0 to 10; the default is 2.
Step 5	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

Enabling MAC Move

MAC move allows an authenticated host to move from one port on the switch to another.

Beginning in privileged EXEC mode, follow these steps to globally enable MAC move on the switch. This procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	authentication mac-move permit **Example:** Device(config)# **authentication mac-move permit**	Enables MAC move on the switch. Default is deny. In Session Aware Networking mode, the default CLI is access-session mac-move deny. To enable Mac Move in Session Aware Networking, use the no access-session mac-move global configuration command. In legacy mode (IBNS 1.0), default value for mac-move is deny and in C3PL mode (IBNS 2.0) default value is permit.
Step 3	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 4	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 5	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Enabling MAC Replace

MAC replace allows a host to replace an authenticated host on a port.

Beginning in privileged EXEC mode, follow these steps to enable MAC replace on an interface. This procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet2/0/2**	Specifies the port to be configured, and enter interface configuration mode.
Step 3	authentication violation {protect | replace | restrict | shutdown} **Example:** Device(config-if)# **authentication violation replace**	Use the replace keyword to enable MAC replace on the interface. The port removes the current session and initiates authentication with the new host. The other keywords have these effects: • protect: the port drops packets with unexpected MAC addresses without generating a system message. • restrict: violating packets are dropped by the CPU and a system message is generated. • shutdown: the port is error disabled when it receives an unexpected MAC address.
Step 4	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.
Step 5	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring 802.1x Accounting

Enabling AAA system accounting with 802.1x accounting allows system reload events to be sent to the accounting RADIUS server for logging. The server can then infer that all active 802.1x sessions are closed.

Note

In Cisco IOS XE Denali 16.3.x and Cisco IOS XE Everest 16.6.x, periodic AAA accounting updates are not supported. The switch does not send periodic interim accounting records to the accounting server. Periodic AAA accounting updates are available in Cisco IOS XE Fuji 16.9.x and later releases.

Because RADIUS uses the unreliable UDP transport protocol, accounting messages might be lost due to poor network conditions. If the switch does not receive the accounting response message from the RADIUS server after a configurable number of retransmissions of an accounting request, this system message appears:

```
Accounting message %s for session %s failed to receive Accounting Response.
```

When the stop message is not sent successfully, this message appears:

```
00:09:55: %RADIUS-4-RADIUS_DEAD: RADIUS server 172.20.246.201:1645,1646 is not responding.
```

Note

You must configure the RADIUS server to perform accounting tasks, such as logging start, stop, and interim-update messages and time stamps. To turn on these functions, enable logging of "Update/Watchdog packets from this AAA client" in your RADIUS server Network Configuration tab. Next, enable "CVS RADIUS Accounting" in your RADIUS server System Configuration tab.

Beginning in privileged EXEC mode, follow these steps to configure 802.1x accounting after AAA is enabled on your switch. This procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet1/0/3**	Specifies the port to be configured, and enter interface configuration mode.

		Command or Action	Purpose
Step 3		aaa accounting dot1x default start-stop group radius	Enables 802.1x accounting using the list of all RADIUS servers.
		Example:	
		Device(config-if)# **aaa accounting dot1x default start-stop group radius**	
Step 4		aaa accounting system default start-stop group radius	(Optional) Enables system accounting (using the list of all RADIUS servers) and generates system accounting reload event messages when the switch reloads.
		Example:	
		Device(config-if)# **aaa accounting system default start-stop group radius**	
Step 5		end	Returns to privileged EXEc mode.
		Example:	
		Device(config-if)# **end**	
Step 6		show running-config	Verifies your entries.
		Example:	
		Device# **show running-config**	
Step 7		copy running-config startup-config	(Optional) Saves your entries in the configuration file.
		Example:	
		Device# **copy running-config startup-config**	

Configuring a Guest VLAN

When you configure a guest VLAN, clients that are not 802.1x-capable are put into the guest VLAN when the server does not receive a response to its EAP request/identity frame. Clients that are 802.1x-capable but that fail authentication are not granted network access. The switch supports guest VLANs in single-host or multiple-hosts mode.

Beginning in privileged EXEC mode, follow these steps to configure a guest VLAN. This procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet 2/0/2**	Specifies the port to be configured, and enter interface configuration mode.

Step 3	Use one of the following:
	
	    • switchport mode access
	    • switchport mode private-vlan host
	
	Example:
	
	Device(config-if)# **switchport mode private-vlan host**
		• Sets the port to access mode.
		
		• Configures the Layer 2 port as a private-VLAN host port.

Step 4	authentication event no-response action authorize vlan vlan-id **Example:** Device(config-if)# **authentication event no-response action authorize vlan 2**	Specifies an active VLAN as an 802.1x guest VLAN. The range is 1 to 4094. You can configure any active VLAN except an internal VLAN (routed port), an RSPAN VLAN or a voice VLAN as an 802.1x guest VLAN.
Step 5	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

Configuring a Restricted VLAN

When you configure a restricted VLAN on a switch stack or a switch, clients that are IEEE 802.1x-compliant are moved into the restricted VLAN when the authentication server does not receive a valid username and password. The switch supports restricted VLANs only in single-host mode.

Beginning in privileged EXEC mode, follow these steps to configure a restricted VLAN. This procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet 2/0/2**	Specifies the port to be configured, and enter interface configuration mode.
Step 3	Use one of the following: • switchport mode access • switchport mode private-vlan host **Example:** Device(config-if)# **switchport mode access**	• Sets the port to access mode. • Configures the Layer 2 port as a private-VLAN host port.
Step 4	authentication port-control auto **Example:** Device(config-if)# **authentication port-control auto**	Enables 802.1x authentication on the port.
Step 5	authentication event fail action authorize vlan vlan-id **Example:** Device(config-if)# **authentication event fail action authorize vlan 2**	Specifies an active VLAN as an 802.1x restricted VLAN. The range is 1 to 4094. You can configure any active VLAN except an internal VLAN (routed port), an RSPAN VLAN or a voice VLAN as an 802.1x restricted VLAN.
Step 6	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

Configuring Number of Authentication Attempts on a Restricted VLAN

You can configure the maximum number of authentication attempts allowed before a user is assigned to the restricted VLAN by using the authentication event retry retry count interface configuration command. The range of allowable authentication attempts is 1 to 3. The default is 3 attempts.

Beginning in privileged EXEC mode, follow these steps to configure the maximum number of allowed authentication attempts. This procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet 2/0/3**	Specifies the port to be configured, and enter interface configuration mode.
Step 3	Use one of the following: • switchport mode access • switchport mode private-vlan host **Example:** or Device(config-if)# **switchport mode access**	• Sets the port to access mode. • Configures the Layer 2 port as a private-VLAN host port.
Step 4	authentication port-control auto **Example:** Device(config-if)# **authentication port-control auto**	Enables 802.1x authentication on the port.
Step 5	authentication event fail action authorize vlan vlan-id **Example:** Device(config-if)# **authentication event fail action authorize vlan 8**	Specifies an active VLAN as an 802.1x restricted VLAN. The range is 1 to 4094. You can configure any active VLAN except an internal VLAN (routed port), an RSPAN VLAN or a voice VLAN as an 802.1x restricted VLAN.

	Command or Action	**Purpose**
Step 6	authentication event retry retry count	Specifies a number of authentication attempts to allow before a port moves to the restricted VLAN. The range is 1 to 3, and the default is 3.
	Example:	
	Device(config-if)# **authentication event retry 2**	
Step 7	end	Returns to privileged EXEC mode.
	Example:	
	Device(config-if)# **end**	

Configuring 802.1x Inaccessible Authentication Bypass with Critical Voice VLAN

Beginning in privileged EXEC mode, follow these steps to configure critical voice VLAN on a port and enable the inaccessible authentication bypass feature.

Procedure

	Command or Action	**Purpose**
Step 1	enable	Enables privileged EXEC mode.
	Example:	• Enter your password if prompted.
	Device> **enable**	
Step 2	configure terminal	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	aaa new-model	Enables AAA.
	Example:	
	Device(config)# **aaa new-model**	
Step 4	radius-server dead-criteria{time seconds } [tries number]	Sets the conditions that determine when a RADIUS server is considered un-available or down (dead).
	Example:	• time— 1 to 120 seconds. The switch dynamically determines a default seconds value between 10 and 60.
	Device(config)# **radius-server dead-criteria time 20 tries 10**	

	Command or Action	**Purpose**
		• number—1 to 100 tries. The switch dynamically determines a default triesnumber between 10 and 100.
Step 5	radius-serverdeadtimeminutes **Example:** Device(config)# **radius-server deadtime 60**	(Optional) Sets the number of minutes during which a RADIUS server is not sent requests. The range is from 0 to 1440 minutes (24 hours). The default is 0 minutes.
Step 6	radius server server name **Example:** Device(config)# **radius server rsim address ipv4 124.2.2.12**	(Optional) Specifies the IP address of the RADIUS server.
Step 7	address {ipv4 | ipv6} ip address auth-port port_number acct-port port_number **Example:** Device(config-radius-server)# **address ipv4 10.0.1.2 auth-port 1550 acct-port 1560**	Configures the IP address for the RADIUS server.
Step 8	key string **Example:** Device(config-radius-server)# **key rad123**	(Optional) Specifies the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.
Step 9	exit **Example:** Device(config-radius-server)# **exit**	Exits the RADIUS server mode and enters the global configuration mode.
Step 10	dot1x critical {eapol | recovery delay milliseconds} **Example:** Device(config)# **dot1x critical eapol** Device(config)# **dot1x critical recovery delay 2000**	(Optional) Configure the parameters for inaccessible authentication bypass: • eapol—Specify that the switch sends an EAPOL-Success message when the switch successfully authenticates the critical port. • recovery delaymilliseconds—Set the recovery delay period during which the switch waits to re-initialize a critical port

	Command or Action	**Purpose**
		when a RADIUS server that was unavailable becomes available. The range is from 1 to 10000 milliseconds. The default is 1000 milliseconds (a port can be re-initialized every second).
Step 11	interface interface-id **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specify the port to be configured, and enters interface configuration mode.
Step 12	authentication event server dead action {authorize | reinitialize} vlan vlan-id] **Example:** Device(config-if)# **authentication event server dead action reinitialicze vlan 20**	Use these keywords to move hosts on the port if the RADIUS server is unreachable: • authorize—Move any new hosts trying to authenticate to the user-specified critical VLAN. • reinitialize—Move all authorized hosts on the port to the user-specified critical VLAN.
Step 13	switchport voice vlan vlan-id **Example:** Device(config-if)# **switchport voice vlan**	Specifies the voice VLAN for the port. The voice VLAN cannot be the same as the critical data VLAN configured in Step 6.
Step 14	authentication event server dead action authorize voice **Example:** Device(config-if)# **authentication event server dead action authorize voice**	Configures critical voice VLAN to move data traffic on the port to the voice VLAN if the RADIUS server is unreachable.
Step 15	end **Example:** Device(config-if)# **end**	Exits interface configuration mode and returns to privileged EXEC mode.
Step 16	show authentication interface interface-id **Example:** Device(config-if)# **show authentication**	(Optional) Verify your entries.

Command or Action	Purpose
`interface gigabitethernet 1/0/1`	

Example

To return to the RADIUS server default settings, use the no radius-server dead-criteria, the no radius-server deadtime, and the no radius server global configuration commands. To disable inaccessible authentication bypass, use the no authentication event server dead action interface configuration command. To disable critical voice VLAN, use the no authentication event server dead action authorize voice interface configuration command.

Example: Configuring Inaccessible Authentication Bypass

This example shows how to configure the inaccessible authentication bypass feature:

```
Device> enable
Device# configure terminal
Device(config)# radius-server dead-criteria time 30 tries 20
Device(config)# radius-server deadtime 60
Device(config)# radius server server1
Device(config-radius-server)# address ipv4 172.29.36.49 acct-port 1618 auth-port 1612
Device(config-radius-server)# key abc1234
Device(config-radius-server)# exit
Device(config)# dot1x critical eapol
Device(config)# dot1x critical recovery delay 2000
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# dot1x critical
Device(config-if)# dot1x critical recovery action reinitialize
Device(config-if)# dot1x critical vlan 20
Device(config-if)# end
```

Configuring 802.1x Authentication with WoL

Beginning in privileged EXEC mode, follow these steps to enable 802.1x authentication with WoL. This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:**	Specifies the port to be configured, and enter interface configuration mode.

	Command or Action	Purpose
	Device(config)# **interface gigabitethernet2/0/3**	
Step 3	authentication control-direction {both | in} **Example:** Device(config-if)# **authentication control-direction both**	Enables 802.1x authentication with WoL on the port, and use these keywords to configure the port as bidirectional or unidirectional. • both—Sets the port as bidirectional. The port cannot receive packets from or send packets to the host. By default, the port is bidirectional. • in—Sets the port as unidirectional. The port can send packets to the host but cannot receive packets from the host.
Step 4	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.
Step 5	show authentication sessions interface interface-id **Example:** Device# **show authentication sessions interface gigabitethernet2/0/3**	Verifies your entries.
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring MAC Authentication Bypass

Beginning in privileged EXEC mode, follow these steps to enable MAC authentication bypass. This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	Device# **configure terminal**	
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet 2/0/1**	Specifies the port to be configured, and enter interface configuration mode.
Step 3	authentication port-control auto **Example:** Device(config-if)# **authentication port-control auto**	Enables 802.1x authentication on the port.
Step 4	mab [eap] **Example:** Device(config-if)# **mab**	Enables MAC authentication bypass. (Optional) Use the eap keyword to configure the switch to use EAP for authorization.
Step 5	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

Configuring 802.1x User Distribution

Beginning in privileged EXEC mode, follow these steps to configure a VLAN group and to map a VLAN to it:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	vlan group vlan-group-name vlan-list vlan-list **Example:** Device(config)# **vlan group eng-dept**	Configures a VLAN group, and maps a single VLAN or a range of VLANs to it.

	Command or Action	**Purpose**
	`vlan-list 10`	
Step 3	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 4	no vlan group vlan-group-name vlan-list vlan-list **Example:** Device(config)# **no vlan group eng-dept vlan-list 10**	Clears the VLAN group configuration or elements of the VLAN group configuration.

Example of Configuring VLAN Groups

This example shows how to configure the VLAN groups, to map the VLANs to the groups, to and verify the VLAN group configurations and mapping to the specified VLANs:

```
Device(config)# vlan group eng-dept vlan-list 10

Device(config)# show vlan group group-name eng-dept
Group Name                  Vlans Mapped
-------------               --------------
eng-dept                    10

Device(config)# show dot1x vlan-group all
Group Name                  Vlans Mapped
-------------               --------------
eng-dept                    10
hr-dept                     20
```

This example shows how to add a VLAN to an existing VLAN group and to verify that the VLAN was added:

```
Device(config)# vlan group eng-dept vlan-list 30
Device(config)# show vlan group eng-dept
Group Name                  Vlans Mapped
-------------               --------------
eng-dept                    10,30
```

This example shows how to remove a VLAN from a VLAN group:

```
Device# no vlan group eng-dept vlan-list 10
```

This example shows that when all the VLANs are cleared from a VLAN group, the VLAN group is cleared:

```
Device(config)# no vlan group eng-dept vlan-list 30
Vlan 30 is successfully cleared from vlan group eng-dept.
```

```
Device(config)# show vlan group group-name eng-dept
```

This example shows how to clear all the VLAN groups:

```
Device(config)# no vlan group end-dept vlan-list all
Device(config)# show vlan-group all
```

For more information about these commands, see the Cisco IOS Security Command Reference.

Configuring NAC Layer 2 802.1x Validation

You can configure NAC Layer 2 802.1x validation, which is also referred to as 802.1x authentication with a RADIUS server.

Beginning in privileged EXEC mode, follow these steps to configure NAC Layer 2 802.1x validation. The procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet2/0/3**	Specifies the port to be configured, and enter interface configuration mode.
Step 3	switchport mode access **Example:** Device(config-if)# **switchport mode access**	Sets the port to access mode only if you configured the RADIUS server.
Step 4	authentication event no-response action authorize vlan vlan-id **Example:** Device(config-if)# **authentication event no-response action authorize vlan 8**	Specifies an active VLAN as an 802.1x guest VLAN. The range is 1 to 4094. You can configure any active VLAN except an internal VLAN (routed port), an RSPAN VLAN, or a voice VLAN as an 802.1x guest VLAN.
Step 5	authentication periodic **Example:**	Enables periodic re-authentication of the client, which is disabled by default.

	Command or Action	**Purpose**
	Device(config-if)# **authentication periodic**	
Step 6	authentication timer reauthenticate **Example:** Device(config-if)# **authentication timer reauthenticate**	Sets re-authentication attempt for the client (set to one hour). This command affects the behavior of the switch only if periodic re-authentication is enabled.
Step 7	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.
Step 8	show authentication sessions interface interface-id **Example:** Device# **show authentication sessions interface gigabitethernet2/0/3**	Verifies your entries.
Step 9	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring an Authenticator Switch with NEAT

Configuring this feature requires that one switch outside a wiring closet is configured as a supplicant and is connected to an authenticator switch.

Note
- The authenticator switch interface configuration must be restored to access mode by explicitly flapping it if a line card is removed and inserted in the chassis when CISP or NEAT session is active.

- The cisco-av-pairs must be configured as device-traffic-class=switch on the ISE, which sets the interface as a trunk after the supplicant is successfully authenticated.

Beginning in privileged EXEC mode, follow these steps to configure a switch as an authenticator:

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	cisp enable **Example:** Device(config)# **cisp enable**	Enables CISP.
Step 3	interface interface-id **Example:** Device(config)# **interface gigabitethernet 2/0/1**	Specifies the port to be configured, and enter interface configuration mode.
Step 4	switchport mode access **Example:** Device(config-if)# **switchport mode access**	Sets the port mode to access.
Step 5	authentication port-control auto **Example:** Device(config-if)# **authentication port-control auto**	Sets the port-authentication mode to auto.
Step 6	dot1x pae authenticator **Example:** Device(config-if)# **dot1x pae authenticator**	Configures the interface as a port access entity (PAE) authenticator.
Step 7	spanning-tree portfast **Example:** Device(config-if)# **spanning-tree portfast trunk**	Enables Port Fast on an access port connected to a single workstation or server..

	Command or Action	Purpose
Step 8	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.
Step 9	show running-config interface interface-id **Example:** Device# **show running-config interface gigabitethernet 2/0/1**	Verifies your configuration.
Step 10	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file. **Note** Saving changes to the configuration file will mean that the authenticator interface will continue to be in trunk mode after reload. If you want the authenticator interface to remain as an access port, do not save your changes to the configuration file.

Configuring a Supplicant Switch with NEAT

Beginning in privileged EXEC mode, follow these steps to configure a switch as a supplicant:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	cisp enable **Example:** Device(config)# **cisp enable**	Enables CISP.

	Command or Action	**Purpose**
Step 3	dot1x credentials profile **Example:** Device(config)# **dot1x credentials test**	Creates 802.1x credentials profile. This must be attached to the port that is configured as supplicant.
Step 4	username suppswitch **Example:** Device(config)# **username suppswitch**	Creates a username.
Step 5	password password **Example:** Device(config)# **password myswitch**	Creates a password for the new username.
Step 6	dot1x supplicant force-multicast **Example:** Device(config)# **dot1x supplicant force-multicast**	Forces the switch to send only multicast EAPOL packets when it receives either unicast or multicast packets. This also allows NEAT to work on the supplicant switch in all host modes.
Step 7	interface interface-id **Example:** Device(config)# **interface gigabitethernet1/0/1**	Specifies the port to be configured, and enter interface configuration mode.
Step 8	switchport trunk encapsulation dot1q **Example:** Device(config-if)# **switchport trunk encapsulation dot1q**	Sets the port to trunk mode.
Step 9	switchport mode trunk **Example:** Device(config-if)# **switchport mode trunk**	Configures the interface as a VLAN trunk port.
Step 10	dot1x pae supplicant **Example:**	Configures the interface as a port access entity (PAE) supplicant.

	Command or Action	**Purpose**
	Device(config-if)# **dot1x pae supplicant**	
Step 11	dot1x credentials profile-name **Example:** Device(config-if)# **dot1x credentials test**	Attaches the 802.1x credentials profile to the interface.
Step 12	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.
Step 13	show running-config interface interface-id **Example:** Device# **show running-config interface gigabitethernet1/0/1**	Verifies your configuration.
Step 14	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.
Step 15	Configuring NEAT with Auto Smartports Macros	You can also use an Auto Smartports user-defined macro instead of the switch VSA to configure the authenticator switch. For more information, see the Auto Smartports Configuration Guide for this release.

Configuring 802.1x Authentication with Downloadable ACLs and Redirect URLs

Note You must configure a downloadable ACL on the ACS before downloading it to the switch.

After authentication on the port, you can use the show ip access-list privileged EXEC command to display the downloaded ACLs on the port.

![note icon]

Note   The output of the show ip access-lists interface command does not display dACL or ACL filter IDs. This is because the ACLs are attached to the virtual ports created by multidomain authentication for each authentication session; instead of the physical interface. To display dACL or ACL filter IDs, use the show ip access-lists access-list-name command.

Configuring Downloadable ACLs

The policies take effect after client authentication and the client IP address addition to the IP device tracking table. The switch then applies the downloadable ACL to the port.

Beginning in privileged EXEC mode:

Before you begin

SISF-Based device tracking is a prerequisite to configuring 802.1x authentication. Ensure that you have enabled device tracking programmatically or manually. For more information, see the Confguring SISF-Based Tracking chapter.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	aaa new-model **Example:** Device(config)# **aaa new-model**	Enables AAA.
Step 3	aaa authorization network default local group radius **Example:** Device(config)# **aaa authorization network default local group radius**	Sets the authorization method to local. To remove the authorization method, use the no aaa authorization network default local group radius command.
Step 4	radius-server vsa send authentication **Example:** Device(config)# **radius-server vsa send authentication**	Configures the radius vsa send authentication.

	Command or Action	**Purpose**
Step 5	interface interface-id **Example:** Device(config)# **interface gigabitethernet2/0/4**	Specifies the port to be configured, and enter interface configuration mode.
Step 6	ip access-group acl-id in **Example:** Device(config-if)# **ip access-group default_acl in**	Configures the default ACL on the port in the input direction. **Note** The acl-id is an access list name or number.
Step 7	show running-config interface interface-id **Example:** Device(config-if)# **show running-config interface gigabitethernet2/0/4**	Verifies your configuration.
Step 8	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring a Downloadable Policy

Beginning in privileged EXEC mode:

Before you begin

SISF-Based device tracking is a prerequisite to configuring 802.1x authentication. Ensure that you have enabled device tracking programmatically or manually.

For more information, see Confguring SISF-Based Tracking.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.

	Command or Action	**Purpose**
Step 2	access-list access-list-number { deny | permit } { hostname | any | host } log **Example:** Device(config)# **access-list 1 deny any log**	Defines the default port ACL. The access-list-number is a decimal number from 1 to 99 or 1300 to 1999. Enter deny or permit to specify whether to deny or permit access if conditions are matched. The source is the source address of the network or host that sends a packet, such as this: - hostname: The 32-bit quantity in dotted-decimal format. - any: The keyword any as an abbreviation for source and source-wildcard value of 0.0.0.0 255.255.255.255. You do not need to enter a source-wildcard value. - host: The keyword host as an abbreviation for source and source-wildcard of source 0.0.0.0. (Optional) Applies the source-wildcard wildcard bits to the source. (Optional) Enters log to cause an informational logging message about the packet that matches the entry to be sent to the console.
Step 3	interface interface-id **Example:** Device(config)# **interface gigabitethernet2/0/2**	Enters interface configuration mode.
Step 4	ip access-group acl-id in **Example:** Device(config-if)# **ip access-group default_acl in**	Configures the default ACL on the port in the input direction. **Note** The acl-id is an access list name or number.
Step 5	exit **Example:** Device(config-if)# **exit**	Returns to global configuration mode.
Step 6	aaa new-model **Example:**	Enables AAA.

	Command or Action	Purpose
	Device(config)# **aaa new-model**	
Step 7	aaa authorization network default group radius **Example:** Device(config)# **aaa authorization network default group radius**	Sets the authorization method to local. To remove the authorization method, use the no aaa authorization network default group radius command.
Step 8	radius-server vsa send authentication **Example:** Device(config)# **radius-server vsa send authentication**	Configures the network access server to recognize and use vendor-specific attributes. **Note** The downloadable ACL must be operational.
Step 9	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

Configuring VLAN ID-based MAC Authentication

Beginning in privileged EXEC mode, follow these steps:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	mab request format attribute 32 vlan access-vlan **Example:** Device(config)# **mab request format attribute 32 vlan access-vlan**	Enables VLAN ID-based MAC authentication.
Step 3	copy running-config startup-config **Example:** Device# **copy running-config**	(Optional) Saves your entries in the configuration file.

Command or Action	Purpose
`startup-config`	

Configuring Flexible Authentication Ordering

The examples used in the instructions below changes the order of Flexible Authentication Ordering so that MAB is attempted before IEEE 802.1X authentication (dot1x). MAB is configured as the first authentication method, so MAB will have priority over all other authentication methods.

Beginning in privileged EXEC mode, follow these steps:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# `configure terminal`	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# `interface gigabitethernet 1/0/1`	Specifies the port to be configured, and enter interface configuration mode.
Step 3	switchport mode access **Example:** Device(config-if)# `switchport mode access`	Sets the port to access mode only if you previously configured the RADIUS server.
Step 4	authentication order [dot1x | mab] | {webauth} **Example:** Device(config-if)# `authentication order mab dot1x`	(Optional) Sets the order of authentication methods used on a port.
Step 5	authentication priority [dot1x | mab] | {webauth} **Example:** Device(config-if)# `authentication priority mab dot1x`	(Optional) Adds an authentication method to the port-priority list.

	Command or Action	Purpose
Step 6	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

Configuring Open1x

Beginning in privileged EXEC mode, follow these steps to enable manual control of the port authorization state:

Procedure

	Command or Action	Purpose
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specifies the port to be configured, and enter interface configuration mode.
Step 3	switchport mode access **Example:** Device(config-if)# **switchport mode access**	Sets the port to access mode only if you configured the RADIUS server.
Step 4	authentication control-direction {both | in} **Example:** Device(config-if)# **authentication control-direction both**	(Optional) Configures the port control as unidirectional or bidirectional.
Step 5	authentication fallback name **Example:** Device(config-if)# **authentication fallback profile1**	(Optional) Configures a port to use web authentication as a fallback method for clients that do not support 802.1x authentication.

	Command or Action	Purpose
Step 6	authentication host-mode [multi-auth | multi-domain | multi-host | single-host] **Example:** Device(config-if)# **authentication host-mode multi-auth**	(Optional) Sets the authorization manager mode on a port.
Step 7	authentication open **Example:** Device(config-if)# **authentication open**	(Optional) Enables or disable open access on a port.
Step 8	authentication order [dot1x | mab] | {webauth} **Example:** Device(config-if)# **authentication order dot1x webauth**	(Optional) Sets the order of authentication methods used on a port.
Step 9	authentication periodic **Example:** Device(config-if)# **authentication periodic**	(Optional) Enables or disable reauthentication on a port.
Step 10	authentication port-control {auto | force-authorized | force-un authorized} **Example:** Device(config-if)# **authentication port-control auto**	(Optional) Enables manual control of the port authorization state.
Step 11	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

Disabling 802.1x Authentication on the Port

You can disable 802.1x authentication on the port by using the no dot1x pae interface configuration command.

Beginning in privileged EXEC mode, follow these steps to disable 802.1x authentication on the port. This procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet 2/0/1**	Specifies the port to be configured, and enter interface configuration mode.
Step 3	switchport mode access **Example:** Device(config-if)# **switchport mode access**	(Optional) Sets the port to access mode only if you configured the RADIUS server.
Step 4	no dot1x pae authenticator **Example:** Device(config-if)# **no dot1x pae authenticator**	Disables 802.1x authentication on the port.
Step 5	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

Resetting the 802.1x Authentication Configuration to the Default Values

Beginning in privileged EXEC mode, follow these steps to reset the 802.1x authentication configuration to the default values. This procedure is optional.

Procedure

	Command or Action	**Purpose**
Step 1	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	interface interface-id **Example:** Device(config)# **interface gigabitethernet 1/0/2**	Enters interface configuration mode, and specify the port to be configured.
Step 3	dot1x default **Example:** Device(config-if)# **dot1x default**	Resets the 802.1x parameters to the default values.
Step 4	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.

Monitoring 802.1x Statistics and Status

Table 42: Privileged EXEC show Commands

Command	Purpose
show dot1x all statistics	Displays 802.1x statistics for all ports
show dot1x interface interface-id statistics	Displays 802.1x statistics for a specific port
show dot1x all [count | details | statistics | summary]	Displays the 802.1x administrative and operational status for a switch
show dot1x interface interface-id	Displays the 802.1x administrative and operational status for a specific port

Table 43: Global Configuration Commands

Command	Purpose
no dot1x logging verbose	Filters verbose 802.1x authentication messages (beginning with Cisco IOS Release 12.2(55)SE)

For detailed information about the fields in these displays, see the command reference for this release.

Feature History for IEEE 802.1x Port-Based Authentication

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	IEEE 802.1x Port-Based Authentication	IEEE 802.1x authentication prevents unauthorized devices (clients) from gaining access to the network.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.
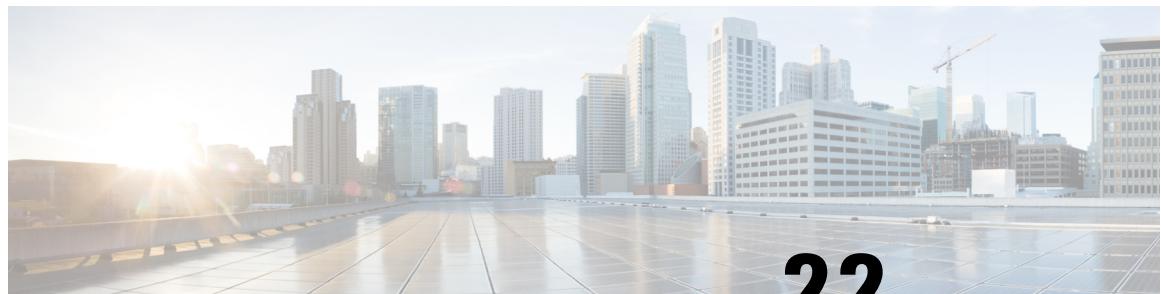
CHAPTER 22

Web-Based Authentication

This chapter describes how to configure web-based authentication on the device. It contains these sections:

Web-Based Authentication Overview

Use the web-based authentication feature, known as web authentication proxy, to authenticate end users on host systems that do not run the IEEE 802.1x supplicant.

When you initiate an HTTP session, web-based authentication intercepts ingress HTTP packets from the host and sends an HTML login page to the users. The users enter their credentials, which the web-based authentication feature sends to the authentication, authorization, and accounting (AAA) server for authentication.

If authentication succeeds, web-based authentication sends a Login-Successful HTML page to the host and applies the access policies returned by the AAA server.

If authentication fails, web-based authentication forwards a Login-Fail HTML page to the user, prompting the user to retry the login. If the user exceeds the maximum number of attempts, web-based authentication forwards a Login-Expired HTML page to the host, and the user is placed on a watch list for a waiting period.

Note HTTPS traffic interception for central web authentication redirect is not supported.

Note You should use global parameter-map (for method-type, custom, and redirect) only for using the same web authentication methods like consent, web consent, and webauth, for all the clients and SSIDs. This ensures that all the clients have the same web-authentication method.

If the requirement is to use Consent for one SSID and Web-authentication for another SSID, then you should use two named parameter-maps. You should configure Consent in first parameter-map and configure webauth in second parameter-map.

Note
The traceback that you receive when webauth client tries to do authentication does not have any performance or behavioral impact. It happens rarely when the context for which FFM replied back to EPM for ACL application is already dequeued (possibly due to timer expiry) and the session becomes 'unauthorized'.

Based on where the web pages are hosted, the local web authention can be categorozied as follows:

- Internal—The internal default HTML pages (Login, Success, Fail, and Expire) in the controller are used during the local web authentication.

- Customized—The customized web pages (Login, Success, Fail, and Expire) are downloaded onto the controller and used during the local web authentication.

- External—The customized web pages are hosted on the external web server instead of using the in-built or custom web pages.

Based on the various web authentication pages, the types of web authentication are as follows:

- Webauth—This is a basic web authentication. Herein, the controller presents a policy page with the user name and password. You need to enter the correct credentials to access the network.

- Consent or web-passthrough—Herein, the controller presents a policy page with the Accept or Deny buttons. You need to click the Accept button to access the network.

- Webconsent—This is a combination of webauth and consent web authentication types. Herein, the controller presents a policy page with Accept or Deny buttons along with user name or password. You need to enter the correct credentials and click the Accept button to access the network.

Device Roles

With web-based authentication, the devices in the network have these specific roles:

- Client—The device (workstation) that requests access to the LAN and the services and responds to requests from the switch. The workstation must be running an HTML browser with Java Script enabled.

- Authentication server—Authenticates the client. The authentication server validates the identity of the client and notifies the switch that the client is authorized to access the LAN and the switch services or that the client is denied.

- Switch—Controls the physical access to the network based on the authentication status of the client. The switch acts as an intermediary (proxy) between the client and the authentication server, requesting identity information from the client, verifying that information with the authentication server, and relaying a response to the client.

Figure 31: Web-Based Authentication Device Roles

This figure shows the roles of these devices in a
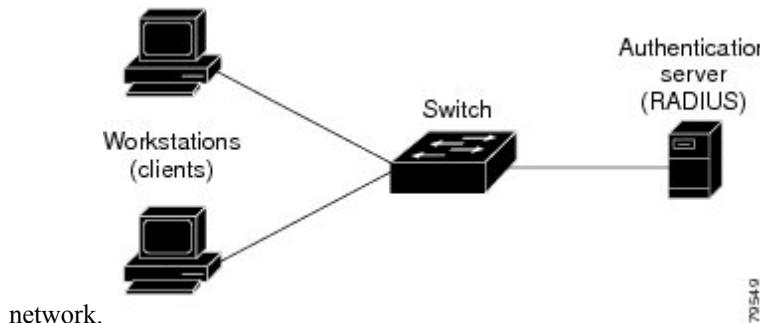
network.

Host Detection

The switch maintains an IP device tracking table to store information about detected hosts.

For Layer 2 interfaces, web-based authentication detects IP hosts by using these mechanisms:

- ARP based trigger—ARP redirect ACL allows web-based authentication to detect hosts with a static IP address or a dynamic IP address.

- Dynamic ARP inspection

- DHCP snooping—Web-based authentication is notified when the switch creates a DHCP-binding entry for the host.

Session Creation

When web-based authentication detects a new host, it creates a session as follows:

- Reviews the exception list.

  If the host IP is included in the exception list, the policy from the exception list entry is applied, and the session is established.

- Reviews for authorization bypass

  If the host IP is not on the exception list, web-based authentication sends a nonresponsive-host (NRH) request to the server.

  If the server response is access accepted, authorization is bypassed for this host. The session is established.

- Sets up the HTTP intercept ACL

  If the server response to the NRH request is access rejected, the HTTP intercept ACL is activated, and the session waits for HTTP traffic from the host.

Authentication Process

When you enable web-based authentication, these events occur:

- The user initiates an HTTP session.

- The HTTP traffic is intercepted, and authorization is initiated. The switch sends the login page to the user. The user enters a username and password, and the switch sends the entries to the authentication server.

- If the authentication succeeds, the switch downloads and activates the user's access policy from the authentication server. The login success page is sent to the user.

- If the authentication fails, the switch sends the login fail page. The user retries the login. If the maximum number of attempts fails, the switch sends the login expired page, and the host is placed in a watch list. After the watch list times out, the user can retry the authentication process.

- If the authentication server does not respond to the switch, and if an AAA fail policy is configured, the switch applies the failure access policy to the host. The login success page is sent to the user.

- The switch reauthenticates a client when the host does not respond to an ARP probe on a Layer 2 interface, or when the host does not send any traffic within the idle timeout on a Layer 3 interface.

- The switch reauthenticates a client when the host does not respond to an ARP probe on a Layer 2 interface.

- The feature applies the downloaded timeout or the locally configured session timeout.

- If the terminate action is RADIUS, the feature sends a nonresponsive host (NRH) request to the server. The terminate action is included in the response from the server.

- If the terminate action is default, the session is dismantled, and the applied policy is removed.

Local Web Authentication Banner

With Web Authentication, you can create a default and customized web-browser banners that appears when you log in to a switch.

The banner appears on both the login page and the authentication-result pop-up pages. The default banner messages are as follows:

- Authentication Successful

- Authentication Failed

- Authentication Expired

The Local Web Authentication Banner can be configured in legacy CLIs as follows:

- Legacy mode—Use the ip admission auth-proxy-banner http global configuration command.

- New-style mode—Use the parameter-map type webauth global banner global configuration command.

The default banner Cisco Systems and Switch host-name Authentication appear on the Login Page. Cisco Systems appears on the authentication result pop-up page.

Figure 32: Authentication Successful Banner
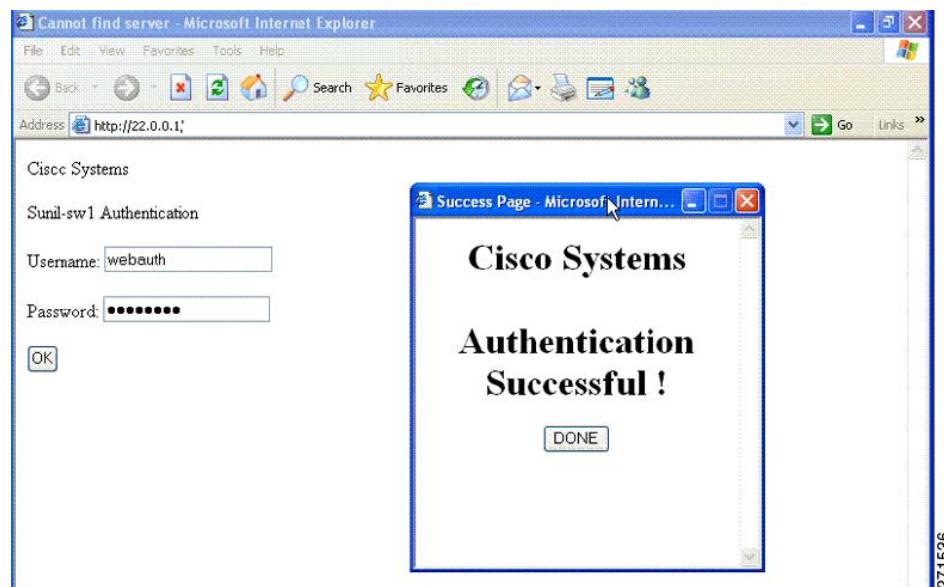
The banner can be customized as follows:

- Add a message, such as switch, router, or company name to the banner:

  - Legacy mode—Use the ip admission auth-proxy-banner http banner-textglobal configuration command.

  - New-style mode—Use the parameter-map type webauth global banner global configuration command.

- Add a logo or text file to the banner:

  - Legacy mode—Use the ip admission auth-proxy-banner http file-path global configuration command.

  - New-style mode—Use the parameter-map type webauth global banner global configuration command.

Figure 33: Customized Web Banner
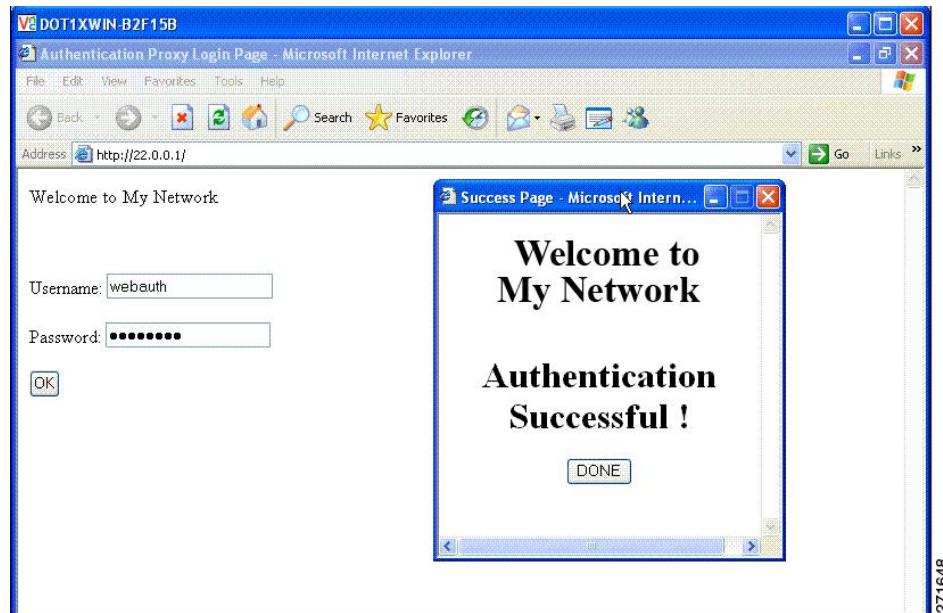
If you do not enable a banner, only the username and password dialog boxes appear in the web authentication login screen, and no banner appears when you log into the switch.

Figure 34: Login Screen With No Banner
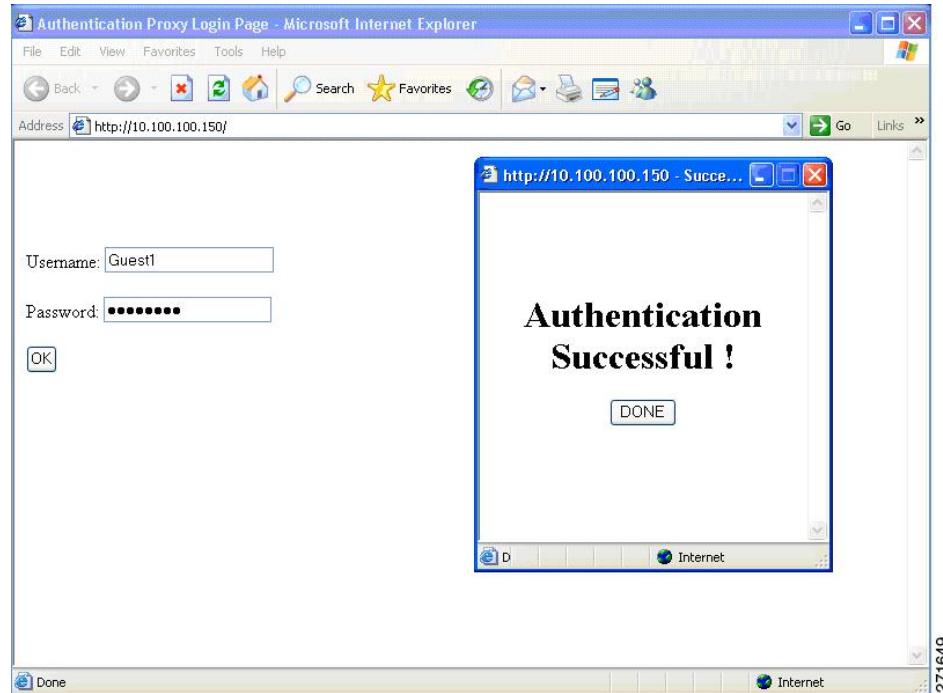
Web Authentication Customizable Web Pages

During the web-based authentication process, the switch internal HTTP server hosts four HTML pages to deliver to an authenticating client. The server uses these pages to notify you of these four-authentication process states:

- Login—Your credentials are requested.

- Success—The login was successful.

- Fail—The login failed.

- Expire—The login session has expired because of excessive login failures.

Guidelines

- You can substitute your own HTML pages for the default internal HTML pages.

- You can use a logo or specify text in the login, success, failure, and expire web pages.

- On the banner page, you can specify text in the login page.

- The pages are in HTML.

- You must include an HTML redirect command in the success page to access a specific URL.

- The URL string must be a valid URL (for example, http://www.cisco.com). An incomplete URL might cause page not found or similar errors on a web browser.

- If you configure web pages for HTTP authentication, they must include the appropriate HTML commands (for example, to set the page time out, to set a hidden password, or to confirm that the same page is not submitted twice).

- The CLI command to redirect users to a specific URL is not available when the configured login form is enabled. The administrator should ensure that the redirection is configured in the web page.

- If the CLI command redirecting users to specific URL after authentication occurs is entered and then the command configuring web pages is entered, the CLI command redirecting users to a specific URL does not take effect.

- Configured web pages can be copied to the switch boot flash or flash.

- The login page can be on one flash, and the success and failure pages can be another flash (for example, the flash on the active switch or a member switch).

- You must configure all four pages.

- The banner page has no effect if it is configured with the web page.

- All of the logo files (image, flash, audio, video, and so on) that are stored in the system directory (for example, flash, disk0, or disk) and that must be displayed on the login page must use web_auth_<filename> as the file name.

- The configured authentication proxy feature supports both HTTP and SSL.

You can substitute your HTML pages for the default internal HTML pages. You can also specify a URL to which users are redirected after authentication occurs, which replaces the internal Success page.

Figure 35: Customizable Authentication Page
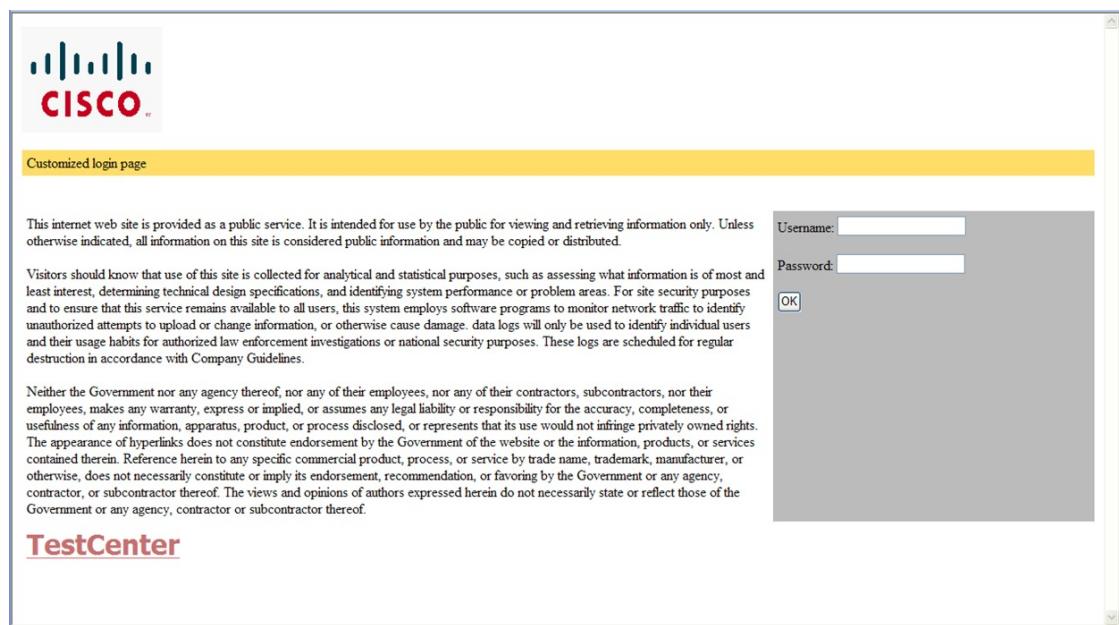
Authentication Proxy Web Page Guidelines

When configuring customized authentication proxy web pages, follow these guidelines:

- To enable the custom web pages feature, specify all four custom HTML files. If you specify fewer than four files, the internal default HTML pages are used.

- The four custom HTML files must be present on the flash memory of the switch. The maximum size of each HTML file is 8 KB.

- Any images on the custom pages must be on an accessible HTTP server. Configure an intercept ACL within the admission rule.

- Any external link from a custom page requires configuration of an intercept ACL within the admission rule.

- To access a valid DNS server, any name resolution required for external links or images requires configuration of an intercept ACL within the admission rule.

- If the custom web pages feature is enabled, a configured auth-proxy-banner is not used.

- If the custom web pages feature is enabled, the redirection URL for successful login feature is not available.

- To remove the specification of a custom file, use the no form of the command.

Because the custom login page is a public web form, consider these guidelines for the page:

- The login form must accept user entries for the username and password and must show them as uname and pwd.

- The custom login page should follow best practices for a web form, such as page timeout, hidden password, and prevention of redundant submissions.

Redirection URL for Successful Login Guidelines

When configuring a redirection URL for successful login, consider these guidelines:

- If the custom authentication proxy web pages feature is enabled, the redirection URL feature is disabled and is not available in the CLI. You can perform redirection in the custom-login success page.

- If the redirection URL feature is enabled, a configured auth-proxy-banner is not used

- To remove the specification of a redirection URL, use the no form of the command.

- If the redirection URL is required after the web-based authentication client is successfully authenticated, then the URL string must start with a valid URL (for example, http://) followed by the URL information. If only the URL is given without http://, then the redirection URL on successful authentication might cause page not found or similar errors on a web browser.

Web-based Authentication Interactions with Other Features

Port Security

You can configure web-based authentication and port security on the same port. Web-based authentication authenticates the port, and port security manages network access for all MAC addresses, including that of the client. You can then limit the number or group of clients that can access the network through the port.

LAN Port IP

You can configure LAN port IP (LPIP) and Layer 2 web-based authentication on the same port. The host is authenticated by using web-based authentication first, followed by LPIP posture validation. The LPIP host policy overrides the web-based authentication host policy.

If the web-based authentication idle timer expires, the NAC policy is removed. The host is authenticated, and posture is validated again.

Gateway IP

You cannot configure Gateway IP (GWIP) on a Layer 3 VLAN interface if web-based authentication is configured on any of the switch ports in the VLAN.

You can configure web-based authentication on the same Layer 3 interface as Gateway IP. The host policies for both features are applied in software. The GWIP policy overrides the web-based authentication host policy.

ACLs

If you configure a VLAN ACL or a Cisco IOS ACL on an interface, the ACL is applied to the host traffic only after the web-based authentication host policy is applied.

For Layer 2 web-based authentication, it is more secure, though not required, to configure a port ACL (PACL) as the default access policy for ingress traffic from hosts connected to the port. After authentication, the web-based authentication host policy overrides the PACL. The Policy ACL is applied to the session even if there is no ACL configured on the port.

You cannot configure a MAC ACL and web-based authentication on the same interface.

You cannot configure web-based authentication on a port whose access VLAN is configured for VACL capture.

Context-Based Access Control

Web-based authentication cannot be configured on a Layer 2 port if context-based access control (CBAC) is configured on the Layer 3 VLAN interface of the port VLAN.

EtherChannel

You can configure web-based authentication on a Layer 2 EtherChannel interface. The web-based authentication configuration applies to all member channels.

How to Configure Web-Based Authentication

Default Web-Based Authentication Configuration

The following table shows the default web-based authentication configuration.

Table 44: Default Web-based Authentication Configuration

Feature	Default Setting
AAA	Disabled
RADIUS server • IP address • UDP authentication port • Key	• None specified • None specified
Default value of inactivity timeout	3600 seconds
Inactivity timeout	Enabled

Web-Based Authentication Configuration Guidelines and Restrictions

- Web-based authentication is an ingress-only feature.

- You can configure web-based authentication only on access ports. Web-based authentication is not supported on trunk ports, EtherChannel member ports, or dynamic trunk ports.

- External web authentication, where the switch redirects a client to a particular host or web server for displaying login message, is not supported.

- You cannot authenticate hosts on Layer 2 interfaces with static ARP cache assignment. These hosts are not detected by the web-based authentication feature because they do not send ARP messages.

- By default, the IP device tracking feature is disabled on a switch. You must enable the IP device tracking feature to use web-based authentication.

- You must enable SISF-Based device tracking to use web-based authentication. By default, SISF-Based device tracking is disabled on a switch.

- You must configure at least one IP address to run the switch HTTP server. You must also configure routes to reach each host IP address. The HTTP server sends the HTTP login page to the host.

- Hosts that are more than one hop away might experience traffic disruption if an STP topology change results in the host traffic arriving on a different port. This occurs because the ARP and DHCP updates might not be sent after a Layer 2 (STP) topology change.

- Web-based authentication does not support VLAN assignment as a downloadable-host policy.

- Web-based authentication supports IPv6 in Session-aware policy mode. IPv6 Web-authentication requires at least one IPv6 address configured on the switch and IPv6 Snooping configured on the switchport.

- Web-based authentication and Network Edge Access Topology (NEAT) are mutually exclusive. You cannot use web-based authentication when NEAT is enabled on an interface, and you cannot use NEAT when web-based authentication is running on an interface.

- Identify the following RADIUS security server settings that will be used while configuring switch-to-RADIUS-server communication:

  - Host name

  - Host IP address

  - Host name and specific UDP port numbers

  - IP address and specific UDP port numbers

  The combination of the IP address and UDP port number creates a unique identifier, that enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service (for example, authentication) the second host entry that is configured functions as the failover backup to the first one. The RADIUS host entries are chosen in the order that they were configured.

- When you configure the RADIUS server parameters:

  - Specify the key string on a separate command line.

  - For key string, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server.

  - When you specify the key string, use spaces within and at the end of the key. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon.

  - You can globally configure the timeout, retransmission, and encryption key values for all RADIUS servers by using with the radius-server host global configuration command. If you want to configure these options on a per-server basis, use the radius-server timeout, radius-server transmit, and the radius-server key global configuration commands.

    Note You need to configure some settings on the RADIUS server, including: the switch IP address, the key string to be shared by both the server and the switch, and the downloadable ACL (DACL). For more information, see the RADIUS server documentation.

- For a URL redirect ACL:

  - Packets that match a permit access control entry (ACE) rule are sent to the CPU for forwarding to the AAA server.

  - Packets that match a deny ACE rule are forwarded through the switch.

  - Packets that match neither the permit ACE rule or deny ACE rule are processed by the next dACL, and if there is no dACL, the packets hit the implicit-deny ACL and are dropped.

Configuring the Authentication Rule and Interfaces

Follow these steps to configure the authentication rule and interfaces:

Before you begin

SISF-Based device tracking is a prerequisite to Web Authentication. Ensure that you have enabled device tracking programmatically or manually.

For more information, see Confguring SISF-Based Tracking.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip admission name name proxy http **Example:** Device(config)# **ip admission name webauth1 proxy http**	Configures an authentication rule for web-based authorization.
Step 4	interface type slot/port **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Enters interface configuration mode and specifies the ingress Layer 2 or Layer 3 interface to be enabled for web-based authentication. type can be fastethernet, gigabit ethernet, or tengigabitethernet.

	Command or Action	**Purpose**
Step 5	ip access-group name **Example:** Device(config-if)# **ip access-group webauthag**	Applies the default ACL.
Step 6	ip admission name **Example:** Device(config)# **ip admission name**	Configures an authentication rule for web-based authorization for the interface.
Step 7	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 8	show ip admission **Example:** Device# **show ip admission**	Displays the configuration.
Step 9	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring AAA Authentication

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	Device# **configure terminal**	
Step 3	aaa new-model	Enables AAA functionality.
	Example:	
	Device(config)# **aaa new-model**	
Step 4	aaa authentication login default group {tacacs+ | radius}	Defines the list of authentication methods at login.
	Example:	named_authentication_list refers to any name that is not greater than 31 characters.
	Device(config)# **aaa authentication login default group tacacs+**	AAA_group_name refers to the server group name. You need to define the server-group server_name at the beginning itself.
Step 5	aaa authorization auth-proxy default group {tacacs+ | radius}	Creates an authorization method list for web-based authorization.
	Example:	
	Device(config)# **aaa authorization auth-proxy default group tacacs+**	
Step 6	tacacs server server-name	Specifies an AAA server.
	Example:	
	Device(config)# **tacacs server yourserver**	
Step 7	address {ipv4 | ipv6} ip address	Configures the IP address for the TACACS server.
	Example:	
	Device(config-server-tacacs)# **address ipv4 10.0.1.12**	
Step 8	key string	Configures the authorization and encryption key used between the switch and the TACACS server.
	Example:	
	Device(config-server-tacacs)# **key cisco123**	
Step 9	exit	Exits the TACACS server mode and enters the global configuration mode.
	Example:	

	Command or Action	Purpose
	Device(config-server-tacacs)# **exit**	
Step 10	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 11	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 12	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring Switch-to-RADIUS-Server Communication

Follow these steps to configure the RADIUS server parameters:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip radius source-interface vlan vlan interface number **Example:** Device(config)# **ip radius**	Specifies that the RADIUS packets have the IP address of the indicated interface.

	Command or Action	Purpose
	`source-interface vlan 80`	
Step 4	radius server server name **Example:** Device(config)# **radius server rsim address ipv4 172.16.0.1**	(Optional) Specifies the IP address of the RADIUS server.
Step 5	address {ipv4 | ipv6} ip address **Example:** Device(config-radius-server)# **address ipv4 10.0.1.2 auth-port 1550 acct-port 1560**	Configures the IP address for the RADIUS server.
Step 6	key string **Example:** Device(config-radius-server)# **key rad123**	(Optional) Specifies the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.
Step 7	exit **Example:** Device(config-radius-server)# **exit**	Exits the RADIUS server mode and enters the global configuration mode.
Step 8	radius-server dead-criteria tries num-tries **Example:** Device(config)# **radius-server dead-criteria tries 30**	Specifies the number of unanswered sent messages to a RADIUS server before considering the server to be inactive. The range of num-tries is 1 to 100.
Step 9	end **Example:** Device(config)# **end**	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the HTTP Server

To use web-based authentication, you must enable the HTTP server within the Device. You can enable the server for either HTTP or HTTPS.

![Note icon]

Note   The Apple psuedo-browser will not open if you configure only the ip http secure-server command. You should also configure the ip http server command.

Follow the procedure given below to enable the server for either HTTP or HTTPS:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip http server **Example:** Device(config)# **ip http server**	Enables the HTTP server. The web-based authentication feature uses the HTTP server to communicate with the hosts for user authentication.
Step 4	ip http secure-server **Example:** Device(config)# **ip http secure-server**	Enables HTTPS. You can configure custom authentication proxy web pages or specify a redirection URL for successful login. **Note** To ensure secure authentication when you enter the ip http secure-server command, the login page is always in HTTPS (secure HTTP) even if the user sends an HTTP request.
Step 5	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

Customizing the Authentication Proxy Web Pages

You can configure web authentication to display four substitute HTML pages to the user in place of the Device default HTML pages during web-based authentication.

Follow these steps to specify the use of your custom authentication proxy web pages:

Before you begin

Store your custom HTML files on the Device flash memory.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip admission proxy http login page file device:login-filename **Example:** Device(config)# **ip admission proxy http login page file disk1:login.htm**	Specifies the location in the Device memory file system of the custom HTML file to use in place of the default login page. The device: is flash memory.
Step 4	ip admission proxy http success page file device:success-filename **Example:** Device(config)# **ip admission proxy http success page file disk1:success.htm**	Specifies the location of the custom HTML file to use in place of the default login success page.
Step 5	ip admission proxy http failure page file device:fail-filename **Example:** Device(config)# **ip admission proxy http fail page file disk1:fail.htm**	Specifies the location of the custom HTML file to use in place of the default login failure page.

	Command or Action	Purpose
Step 6	ip admission proxy http login expired page file device:expired-filename **Example:** Device(config)# **ip admission proxy http login expired page file disk1:expired.htm**	Specifies the location of the custom HTML file to use in place of the default login expired page.
Step 7	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

Specifying a Redirection URL for Successful Login

Follow these steps to specify a URL to which the user is redirected after authentication, effectively replacing the internal Success HTML page:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip admission proxy http success redirect url-string **Example:** Device(config)# **ip admission proxy http success redirect www.example.com**	Specifies a URL for redirection of the user in place of the default login success page.
Step 4	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

Configuring Web-Based Authentication Parameters

Follow these steps to configure the maximum number of failed login attempts before the client is placed in a watch list for a waiting period:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	ip admission max-login-attempts number **Example:** Device(config)# **ip admission max-login-attempts 10**	Sets the maximum number of failed login attempts. The range is 1 to 2147483647 attempts. The default is 5.
Step 4	exit **Example:** Device# **exit**	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Web-Based Authentication Local Banner

Follow these steps to configure a local banner on a switch that has web authentication configured.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	Device# **configure terminal**	
Step 3	ip admission auth-proxy-banner http [banner-text | file-path] **Example:** Device(config)# **ip admission auth-proxy-banner http C My Switch C**	Enables the local banner. (Optional) Create a custom banner by entering C banner-text C (where C is a delimiting character), or file-path that indicates a file (for example, a logo or text file) that appears in the banner.
Step 4	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 5	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 6	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Removing Web-Based Authentication Cache Entries

Follow these steps to remove web-based authentication cache entries:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	clear ip auth-proxy cache {* | host ip address} **Example:** Device# **clear ip auth-proxy cache**	Delete authentication proxy entries. Use an asterisk to delete all cache entries. Enter a specific IP address to delete the entry for a single host.

	Command or Action	Purpose
	`192.168.4.5`	

Step 3	clear ip admission cache {*	host ip address}
		
		Example:
		
		`Device# clear ip admission cache`
		`192.168.4.5`
			Delete authentication proxy entries. Use an asterisk to delete all cache entries. Enter a specific IP address to delete the entry for a single host.

Verifying Web-Based Authentication Status

Use the commands in this topic to display the web-based authentication settings for all interfaces or for specific ports.

Table 45: Privileged EXEC show Commands

Command	Purpose
show authentication sessions method webauth	Displays the web-based authentication settings for all interfaces for fastethernet, gigabitethernet, or tengigabitethernet
show authentication sessions interface type slot/port[details]	Displays the web-based authentication settings for the specified interface for fastethernet, gigabitethernet, or tengigabitethernet. In Session Aware Networking mode, use the show access-session interface command.

Feature Information for Web-Based Authentication

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 46: Feature Information for Web-Based Authentication

Feature Name	Releases	Feature Information
Web-Based Authentication	Cisco IOS XE Everest 16.6.1	Use the web-based authentication feature, known as web authentication proxy, to authenticate end users on host systems that do not run the IEEE 802.1x supplicant. This feature was implemented on the following platforms: • Cisco Catalyst 9400 Series Switches
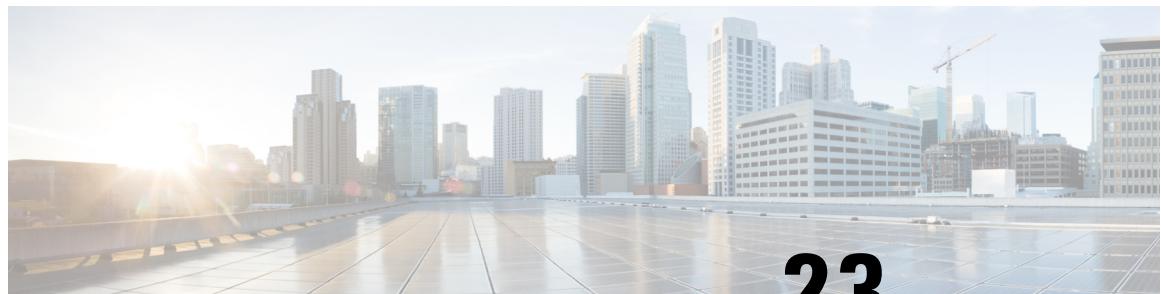
CHAPTER **23**

Configuring Port-Based Traffic Control

- [Overview of Port-Based Traffic Control , on page 467](navigation)

Overview of Port-Based Traffic Control

Port-based traffic control is a set of Layer 2 features on the Cisco Catalyst switches used to filter or block packets at the port level in response to specific traffic conditions. The following port-based traffic control features are supported:

- Storm Control
- Protected Ports
- Port Blocking

Information About Storm Control

Storm Control

Storm control prevents traffic on a LAN from being disrupted by a broadcast, multicast, or unicast storm on one of the physical interfaces. A LAN storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. Errors in the protocol-stack implementation, mistakes in network configurations, or users issuing a denial-of-service attack can cause a storm.

Storm control (or traffic suppression) monitors packets passing from an interface to the switching bus and determines if the packet is unicast, multicast, or broadcast. The switch counts the number of packets of a specified type received within the 1-second time interval and compares the measurement with a predefined suppression-level threshold.

How Traffic Activity is Measured

Storm control uses one of these methods to measure traffic activity:

- Bandwidth as a percentage of the total available bandwidth of the port that can be used by the broadcast, multicast, or unicast traffic
- Traffic rate in packets per second at which broadcast, multicast, or unicast packets are received
- Traffic rate in bits per second at which broadcast, multicast, or unicast packets are received

With each method, the port blocks traffic when the rising threshold is reached. The port remains blocked until the traffic rate drops below the falling threshold (if one is specified) and then resumes normal forwarding. If the falling suppression level is not specified, the switch blocks all traffic until the traffic rate drops below the rising suppression level. In general, the higher the level, the less effective the protection against broadcast storms.

Note   When the storm control threshold for multicast traffic is reached, all multicast traffic except control traffic, such as bridge protocol data unit (BDPU) and Cisco Discovery Protocol frames, are blocked. However, the switch does not differentiate between routing updates, such as OSPF, and regular multicast data traffic, so both types of traffic are blocked.

Traffic Patterns

Figure 36: Broadcast Storm Control Example

This example shows broadcast traffic patterns on an interface over a given period of time.
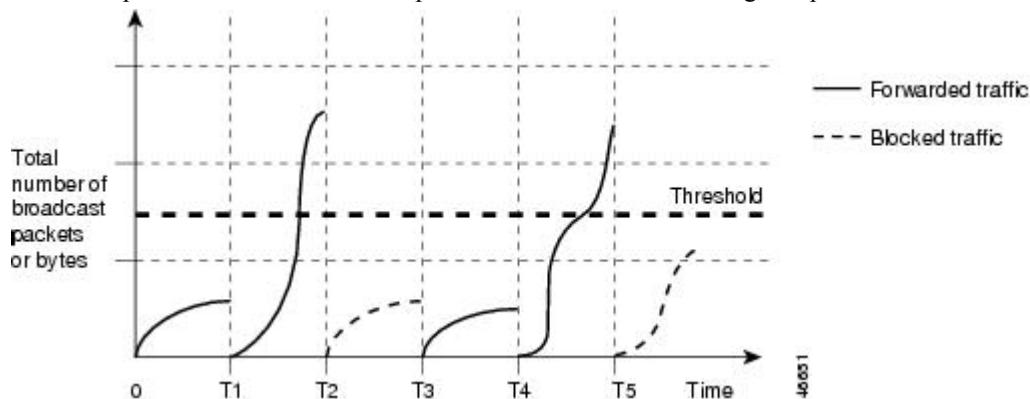
Broadcast traffic being forwarded exceeded the configured threshold between time intervals T1 and T2 and between T4 and T5. When the amount of specified traffic exceeds the threshold, all traffic of that kind is dropped for the next time period. Therefore, broadcast traffic is blocked during the intervals following T2 and T5. At the next time interval (for example, T3), if broadcast traffic does not exceed the threshold, it is again forwarded.

The combination of the storm-control suppression level and the 1-second time interval controls the way the storm control algorithm works. A higher threshold allows more packets to pass through. A threshold value of 100 percent means that no limit is placed on the traffic. A value of 0.0 means that all broadcast, multicast, or unicast traffic on that port is blocked.

Note   Because packets do not arrive at uniform intervals, the 1-second time interval during which traffic activity is measured can affect the behavior of storm control.

You use the storm-control interface configuration commands to set the threshold value for each traffic type.

How to Configure Storm Control

Configuring Storm Control and Threshold Levels

You configure storm control on a port and enter the threshold level that you want to be used for a particular type of traffic.

However, because of hardware limitations and the way in which packets of different sizes are counted, threshold percentages are approximations. Depending on the sizes of the packets making up the incoming traffic, the actual enforced threshold might differ from the configured level by several percentage points.

Note   Storm control is supported on physical interfaces. You can also configure storm control on an EtherChannel. When storm control is configured on an EtherChannel, the storm control settings propagate to the EtherChannel physical interfaces.

Follow these steps to storm control and threshold levels:

Before you begin

Storm control is supported on physical interfaces. You can also configure storm control on an EtherChannel. When storm control is configured on an EtherChannel, the storm control settings propagate to the EtherChannel physical interfaces.

Procedure

	Command or Action	**Purpose**
Step 1	enable	Enables privileged EXEC mode.
	Example:	• Enter your password if prompted.
	Device> **enable**	
Step 2	configure terminal	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	interface interface-id	Specifies the interface to be configured, and enter interface configuration mode.
	Example:	
	Device(config)# **interface gigabitethernet1/0/1**	
Step 4	storm-control {broadcast	multicast	unicast} level {level [level-low]	bps bps [bps-low]	pps pps [pps-low]}	Configures broadcast, multicast, or unicast storm control. By default, storm control is disabled.
	Example:	The keywords have these meanings:

	Command or Action	**Purpose**
	`Device(config-if)# ` **`storm-control unicast level 87 65`**	• For level, specifies the rising threshold level for broadcast, multicast, or unicast traffic as a percentage (up to two decimal places) of the bandwidth. The port blocks traffic when the rising threshold is reached. The range is 0.00 to 100.00.
		• (Optional) For level-low, specifies the falling threshold level as a percentage (up to two decimal places) of the bandwidth. This value must be less than or equal to the rising suppression value. The port forwards traffic when traffic drops below this level. If you do not configure a falling suppression level, it is set to the rising suppression level. The range is 0.00 to 100.00.
		If you set the threshold to the maximum value (100 percent), no limit is placed on the traffic. If you set the threshold to 0.0, all broadcast, multicast, and unicast traffic on that port is blocked.
		• For bps bps, specifies the rising threshold level for broadcast, multicast, or unicast traffic in bits per second (up to one decimal place). The port blocks traffic when the rising threshold is reached. The range is 0.0 to 10000000000.0.
		• (Optional) For bps-low, specifies the falling threshold level in bits per second (up to one decimal place). It can be less than or equal to the rising threshold level. The port forwards traffic when traffic drops below this level. The range is 0.0 to 10000000000.0.
		• For pps pps, specifies the rising threshold level for broadcast, multicast, or unicast traffic in packets per second (up to one decimal place). The port blocks traffic when the rising threshold is reached. The range is 0.0 to 10000000000.0.
		• (Optional) For pps-low, specifies the falling threshold level in packets per second (up to one decimal place). It can be less than or equal to the rising threshold level. The port forwards traffic when

	Command or Action	**Purpose**
		traffic drops below this level. The range is 0.0 to 10000000000.0.
		For BPS and PPS settings, you can use metric suffixes such as k, m, and g for large number thresholds.
Step 5	storm-control action {shutdown	trap} **Example:** Device(config-if)# **storm-control action trap**	Specifies the action to be taken when a storm is detected. The default is to filter out the traffic and not to send traps. • Select the shutdown keyword to error-disable the port during a storm. • Select the trap keyword to generate an SNMP trap when a storm is detected.
Step 6	end **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.
Step 7	show storm-control [interface-id] [broadcast	multicast	unicast] **Example:** Device# **show storm-control gigabitethernet1/0/1 unicast**	Verifies the storm control suppression levels set on the interface for the specified traffic type. If you do not enter a traffic type, details for all traffic types (broadcast, multicast and unicast) are displayed.
Step 8	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring Small-Frame Arrival Rate

Incoming VLAN-tagged packets smaller than 67 bytes are considered small frames. They are forwarded by the switch, but they do not cause the switch storm-control counters to increment.

You globally enable the small-frame arrival feature on the switch and then configure the small-frame threshold for packets on each interface. Packets smaller than the minimum size and arriving at a specified rate (the threshold) are dropped since the port is error disabled.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	Device> **enable**	
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	errdisable detect cause small-frame **Example:** Device(config)# **errdisable detect cause small-frame**	Enables the small-frame rate-arrival feature on the switch.
Step 4	errdisable recovery interval interval **Example:** Device(config)# **errdisable recovery interval 60**	(Optional) Specifies the time to recover from the specified error-disabled state.
Step 5	errdisable recovery cause small-frame **Example:** Device(config)# **errdisable recovery cause small-frame**	(Optional) Configures the recovery time for error-disabled ports to be automatically re-enabled after they are error disabled by the arrival of small frames Storm control is supported on physical interfaces. You can also configure storm control on an EtherChannel. When storm control is configured on an EtherChannel, the storm control settings propagate to the EtherChannel physical interfaces.
Step 6	interface interface-id **Example:** Device(config)# **interface gigabitethernet1/0/2**	Enters interface configuration mode, and specify the interface to be configured.
Step 7	small-frame violation-rate pps **Example:** Device(config-if)# **small-frame violation rate 10000**	Configures the threshold rate for the interface to drop incoming packets and error disable the port. The range is 1 to 10,000 packets per second (pps)

	Command or Action	**Purpose**
Step 8	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 9	show interfaces interface-id **Example:** Device# **show interfaces gigabitethernet1/0/2**	Verifies the configuration.
Step 10	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 11	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Information About Protected Ports

Protected Ports

Some applications require that no traffic be forwarded at Layer 2 between ports on the same switch so that one neighbor does not see the traffic generated by another neighbor. In such an environment, the use of protected ports ensures that there is no exchange of unicast, broadcast, or multicast traffic between these ports on the switch.

Protected ports have these features:

- A protected port does not forward any traffic (unicast, multicast, or broadcast) to any other port that is also a protected port. Data traffic cannot be forwarded between protected ports at Layer 2; only control traffic, such as PIM packets, is forwarded because these packets are processed by the CPU and forwarded in software. All data traffic passing between protected ports must be forwarded through a Layer 3 device.

- Forwarding behavior between a protected port and a nonprotected port proceeds as usual.

Default Protected Port Configuration

The default is to have no protected ports defined.

Protected Ports Guidelines

You can configure protected ports on a physical interface (for example, Gigabit Ethernet port 1) or an EtherChannel group (for example, port-channel 5). When you enable protected ports for a port channel, it is enabled for all ports in the port-channel group.

How to Configure Protected Ports

Configuring a Protected Port

Before you begin

Protected ports are not pre-defined. This is the task to configure one.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	interface interface-id **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specifies the interface to be configured, and enter interface configuration mode.
Step 4	switchport protected **Example:** Device(config-if)# **switchport protected**	Configures the interface to be a protected port.
Step 5	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

	Command or Action	**Purpose**
Step 6	show interfaces interface-id switchport **Example:** Device# **show interfaces gigabitethernet 1/0/1 switchport**	Verifies your entries.
Step 7	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 8	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Monitoring Protected Ports

Table 47: Commands for Displaying Protected Port Settings

Command	**Purpose**
show interfaces [interface-id] switchport	Displays the administrative and operational status of all switching (nonrouting) ports or the specified port, including port blocking and port protection settings.

Information About Port Blocking

Port Blocking

By default, the switch floods packets with unknown destination MAC addresses out of all ports. If unknown unicast and multicast traffic is forwarded to a protected port, there could be security issues. To prevent unknown unicast or multicast traffic from being forwarded from one port to another, you can block a port (protected or nonprotected) from flooding unknown unicast or multicast packets to other ports.

Note With multicast traffic, the port blocking feature blocks only pure Layer 2 packets. Multicast packets that contain IPv4 or IPv6 information in the header are not blocked.

How to Configure Port Blocking

Blocking Flooded Traffic on an Interface

Before you begin

The interface can be a physical interface or an EtherChannel group. When you block multicast or unicast traffic for a port channel, it is blocked on all ports in the port-channel group.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	interface interface-id **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specifies the interface to be configured, and enter interface configuration mode.
Step 4	switchport block multicast **Example:** Device(config-if)# **switchport block multicast**	Blocks unknown multicast forwarding out of the port.
Step 5	switchport block unicast **Example:** Device(config-if)# **switchport block unicast**	Blocks unknown unicast forwarding out of the port.
Step 6	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 7	show interfaces interface-id switchport **Example:** Device# **show interfaces gigabitethernet 1/0/1 switchport**	Verifies your entries.
Step 8	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 9	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Monitoring Port Blocking

Table 48: Commands for Displaying Port Blocking Settings

Command	Purpose
show interfaces [interface-id] switchport	Displays the administrative and operational status of all switching (nonrouting) ports or the specified port, including port blocking and port protection settings.

Prerequisites for Port Security

Note If you try to set the maximum value to a number less than the number of secure addresses already configured on an interface, the command is rejected.

Restrictions for Port Security

- The maximum number of secure MAC addresses that you can configure on a switch is set by the maximum number of available MAC addresses allowed in the system. This number is the total of available MAC addresses, including those used for other Layer 2 functions and any other secure MAC addresses configured on interfaces.

- Port Security is not supported on EtherChannel interfaces.

Information About Port Security

Port Security

You can use the port security feature to restrict input to an interface by limiting and identifying MAC addresses of the stations allowed to access the port. When you assign secure MAC addresses to a secure port, the port does not forward packets with source addresses outside the group of defined addresses. If you limit the number of secure MAC addresses to one and assign a single secure MAC address, the workstation attached to that port is assured the full bandwidth of the port.

If a port is configured as a secure port and the maximum number of secure MAC addresses is reached, when the MAC address of a station attempting to access the port is different from any of the identified secure MAC addresses, a security violation occurs. Also, if a station with a secure MAC address configured or learned on one secure port attempts to access another secure port, a violation is flagged.

Types of Secure MAC Addresses

The switch supports these types of secure MAC addresses:

- Static secure MAC addresses—These are manually configured by using the switchport port-security mac-address mac-address interface configuration command, stored in the address table, and added to the switch running configuration.

- Dynamic secure MAC addresses—These are dynamically configured, stored only in the address table, and removed when the switch restarts.

- Sticky secure MAC addresses—These can be dynamically learned or manually configured, stored in the address table, and added to the running configuration. If these addresses are saved in the configuration file, when the switch restarts, the interface does not need to dynamically reconfigure them.

Sticky Secure MAC Addresses

You can configure an interface to convert the dynamic MAC addresses to sticky secure MAC addresses and to add them to the running configuration by enabling sticky learning. The interface converts all the dynamic secure MAC addresses, including those that were dynamically learned before sticky learning was enabled, to sticky secure MAC addresses. All sticky secure MAC addresses are added to the running configuration.

The sticky secure MAC addresses do not automatically become part of the configuration file, which is the startup configuration used each time the switch restarts. If you save the sticky secure MAC addresses in the configuration file, when the switch restarts, the interface does not need to relearn these addresses. If you do not save the sticky secure addresses, they are lost.

If sticky learning is disabled, the sticky secure MAC addresses are converted to dynamic secure addresses and are removed from the running configuration.

Security Violations

It is a security violation when one of these situations occurs:

- The maximum number of secure MAC addresses have been added to the address table, and a station whose MAC address is not in the address table attempts to access the interface.

- An address learned or configured on one secure interface is seen on another secure interface in the same VLAN.

- Running diagnostic tests with port security enabled.

You can configure the interface for one of three violation modes, based on the action to be taken if a violation occurs:

- protect—when the number of secure MAC addresses reaches the maximum limit allowed on the port, packets with unknown source addresses are dropped until you remove a sufficient number of secure MAC addresses to drop below the maximum value or increase the number of maximum allowable addresses. You are not notified that a security violation has occurred.

> **Note** We do not recommend configuring the protect violation mode on a trunk port. The protect mode disables learning when any VLAN reaches its maximum limit, even if the port has not reached its maximum limit.

- restrict—when the number of secure MAC addresses reaches the maximum limit allowed on the port, packets with unknown source addresses are dropped until you remove a sufficient number of secure MAC addresses to drop below the maximum value or increase the number of maximum allowable addresses. In this mode, you are notified that a security violation has occurred. An SNMP trap is sent, a syslog message is logged, and the violation counter increments.

- shutdown—a port security violation causes the interface to become error-disabled and to shut down immediately, and the port LED turns off. When a secure port is in the error-disabled state, you can bring it out of this state by entering the errdisable recovery cause psecure-violation global configuration command, or you can manually re-enable it by entering the shutdown and no shut down interface configuration commands. This is the default mode.

- shutdown vlan—Use to set the security violation mode per-VLAN. In this mode, the VLAN is error disabled instead of the entire port when a violation occurs

This table shows the violation mode and the actions taken when you configure an interface for port security.

Table 49: Security Violation Mode Actions

Violation Mode	Traffic is forwarded [12]	Sends SNMP trap	Sends syslog message	Displays error message [13]	Violation counter increments	Shuts down port
protect	No	No	No	No	No	No
restrict	No	Yes	Yes	No	Yes	No
shutdown	No	No	No	No	Yes	Yes
shutdown vlan	No	No	Yes	No	Yes	No [14]

[12] Packets with unknown source addresses are dropped until you remove a sufficient number of secure MAC addresses.

[13] The switch returns an error message if you manually configure an address that would cause a security violation.

[14] Shuts down only the VLAN on which the violation occurred.

Port Security Aging

You can use port security aging to set the aging time for all secure addresses on a port. Two types of aging are supported per port:

• Absolute—The secure addresses on the port are deleted after the specified aging time.

• Inactivity—The secure addresses on the port are deleted only if the secure addresses are inactive for the specified aging time.

Port Security and Switch Stacks

When a switch joins a stack, the new switch will get the configured secure addresses. All dynamic secure addresses are downloaded by the new stack member from the other stack members.

When a switch (either the active switch or a stack member) leaves the stack, the remaining stack members are notified, and the secure MAC addresses configured or learned by that switch are deleted from the secure MAC address table.

Default Port Security Configuration

Table 50: Default Port Security Configuration

Feature	Default Setting
Port security	Disabled on a port.
Sticky address learning	Disabled.
Maximum number of secure MAC addresses per port	1.
Violation mode	Shutdown. The port shuts down when the maximum number of secure MAC addresses is exceeded.
Port security aging	Disabled. Aging time is 0. Static aging is disabled. Type is absolute.

Port Security Configuration Guidelines

• Port security can only be configured on static access ports or trunk ports. A secure port cannot be a dynamic access port.

• A secure port cannot be a destination port for Switched Port Analyzer (SPAN).

• Voice VLAN is only supported on access ports and not on trunk ports, even though the configuration is allowed.

• When you enable port security on an interface that is also configured with a voice VLAN, set the maximum allowed secure addresses on the port to two. When the port is connected to a Cisco IP phone, the IP phone requires one MAC address. The Cisco IP phone address is learned on the voice VLAN, but is not

learned on the access VLAN. If you connect a single PC to the Cisco IP phone, no additional MAC addresses are required. If you connect more than one PC to the Cisco IP phone, you must configure enough secure addresses to allow one for each PC and one for the phone.

- When a trunk port configured with port security and assigned to an access VLAN for data traffic and to a voice VLAN for voice traffic, entering the switchport voice and switchport priority extend interface configuration commands has no effect.

  When a connected device uses the same MAC address to request an IP address for the access VLAN and then an IP address for the voice VLAN, only the access VLAN is assigned an IP address.

- When you enter a maximum secure address value for an interface, and the new value is greater than the previous value, the new value overwrites the previously configured value. If the new value is less than the previous value and the number of configured secure addresses on the interface exceeds the new value, the command is rejected.

- The switch does not support port security aging of sticky secure MAC addresses.

This table summarizes port security compatibility with other port-based features.

Table 51: Port Security Compatibility with Other Switch Features

Type of Port or Feature on Port	Compatible with Port Security
DTP [15] port [16]	No
Trunk port	Yes
Dynamic-access port [17]	No
Routed port	No
SPAN source port	Yes
SPAN destination port	No
EtherChannel	No
Tunneling port	Yes
Protected port	Yes
IEEE 802.1x port	Yes
Voice VLAN port [18]	Yes
IP source guard	Yes
Dynamic Address Resolution Protocol (ARP) inspection	Yes
Flex Links	Yes

[15] DTP=Dynamic Trunking Protocol

[16] A port configured with the switchport mode dynamic interface configuration command.

17 A VLAN Query Protocol (VQP) port configured with the switchport access vlan dynamic interface configuration command.

18 You must set the maximum allowed secure addresses on the port to two plus the maximum number of secure addresses allowed on the access VLAN.

Overview of Port-Based Traffic Control

Port-based traffic control is a set of Layer 2 features on the Cisco Catalyst switches used to filter or block packets at the port level in response to specific traffic conditions. The following port-based traffic control features are supported:

- Storm Control

- Protected Ports

- Port Blocking

How to Configure Port Security

Enabling and Configuring Port Security

Before you begin

This task restricts input to an interface by limiting and identifying MAC addresses of the stations allowed to access the port:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	interface interface-id **Example:** Device(config)# **interface gigabitethernet1/0/1**	Specifies the interface to be configured, and enter interface configuration mode.
Step 4	switchport mode {access | trunk} **Example:**	Sets the interface switchport mode as access or trunk; an interface in the default mode (dynamic auto) cannot be configured as a secure port.

	Command or Action	**Purpose**
	Device(config-if)# **switchport mode access**	
Step 5	switchport voice vlan vlan-id **Example:** Device(config-if)# **switchport voice vlan 22**	Enables voice VLAN on a port. vlan-id—Specifies the VLAN to be used for voice traffic.
Step 6	switchport port-security **Example:** Device(config-if)# **switchport port-security**	Enable port security on the interface. **Note** Under certain conditions, when port security is enabled on the member ports in a switch stack, the DHCP and ARP packets would be dropped. To resolve this, configure a shut and no shut on the interface.
Step 7	switchport port-security [maximum value [vlan {vlan-list | {access | voice}}]] **Example:** Device(config-if)# **switchport port-security maximum 20**	(Optional) Sets the maximum number of secure MAC addresses for the interface. The maximum number of secure MAC addresses that you can configure on a switch or switch stack is set by the maximum number of available MAC addresses allowed in the system. This number is the total of available MAC addresses, including those used for other Layer 2 functions and any other secure MAC addresses configured on interfaces. (Optional) vlan—sets a per-VLAN maximum value Enter one of these options after you enter the vlan keyword: • vlan-list—On a trunk port, you can set a per-VLAN maximum value on a range of VLANs separated by a hyphen or a series of VLANs separated by commas. For nonspecified VLANs, the per-VLAN maximum value is used. • access—On an access port, specifies the VLAN as an access VLAN. • voice—On an access port, specifies the VLAN as a voice VLAN.

	Command or Action	**Purpose**
		Note The voice keyword is available only if a voice VLAN is configured on a port and if that port is not the access VLAN. If an interface is configured for voice VLAN, configure a maximum of two secure MAC addresses.
Step 8	switchport port-security violation {protect | restrict | shutdown | shutdown vlan} **Example:** Device(config-if)# **switchport port-security violation restrict**	(Optional) Sets the violation mode, the action to be taken when a security violation is detected, as one of these: • protect—When the number of port secure MAC addresses reaches the maximum limit allowed on the port, packets with unknown source addresses are dropped until you remove a sufficient number of secure MAC addresses to drop below the maximum value or increase the number of maximum allowable addresses. You are not notified that a security violation has occurred. **Note** We do not recommend configuring the protect mode on a trunk port. The protect mode disables learning when any VLAN reaches its maximum limit, even if the port has not reached its maximum limit. • restrict—When the number of secure MAC addresses reaches the limit allowed on the port, packets with unknown source addresses are dropped until you remove a sufficient number of secure MAC addresses or increase the number of maximum allowable addresses. An SNMP trap is sent, a syslog message is logged, and the violation counter increments. • shutdown—The interface is error-disabled when a violation occurs, and the port LED turns off. An SNMP trap is sent, a syslog message is logged, and the violation counter increments. • shutdown vlan—Use to set the security violation mode per VLAN. In this mode,

	Command or Action	Purpose
		the VLAN is error disabled instead of the entire port when a violation occurs.
		Note When a secure port is in the error-disabled state, you can bring it out of this state by entering the errdisable recovery cause psecure-violation global configuration command. You can manually re-enable it by entering the shutdown and no shutdown interface configuration commands or by using the clear errdisable interface vlan privileged EXEC command.
Step 9	switchport port-security [mac-address mac-address [vlan {vlan-id | {access | voice}}] **Example:** Device(config-if)# **switchport port-security mac-address 00:A0:C7:12:C9:25 vlan 3 voice**	(Optional) Enters a secure MAC address for the interface. You can use this command to enter the maximum number of secure MAC addresses. If you configure fewer secure MAC addresses than the maximum, the remaining MAC addresses are dynamically learned. **Note** If you enable sticky learning after you enter this command, the secure addresses that were dynamically learned are converted to sticky secure MAC addresses and are added to the running configuration. (Optional) vlan—sets a per-VLAN maximum value. Enter one of these options after you enter the vlan keyword: • vlan-id—On a trunk port, you can specify the VLAN ID and the MAC address. If you do not specify a VLAN ID, the native VLAN is used. • access—On an access port, specifies the VLAN as an access VLAN. • voice—On an access port, specifies the VLAN as a voice VLAN.

	Command or Action	**Purpose**
		Note The voice keyword is available only if a voice VLAN is configured on a port and if that port is not the access VLAN. If an interface is configured for voice VLAN, configure a maximum of two secure MAC addresses.
Step 10	switchport port-security mac-address sticky **Example:** Device(config-if)# **switchport port-security mac-address sticky**	(Optional) Enables sticky learning on the interface.
Step 11	switchport port-security mac-address sticky [mac-address | vlan {vlan-id | {access | voice}}] **Example:** Device(config-if)# **switchport port-security mac-address sticky 00:A0:C7:12:C9:25 vlan voice**	(Optional) Enters a sticky secure MAC address, repeating the command as many times as necessary. If you configure fewer secure MAC addresses than the maximum, the remaining MAC addresses are dynamically learned, are converted to sticky secure MAC addresses, and are added to the running configuration. **Note** If you do not enable sticky learning before this command is entered, an error message appears, and you cannot enter a sticky secure MAC address. (Optional) vlan—sets a per-VLAN maximum value. Enter one of these options after you enter the vlan keyword: • vlan-id—On a trunk port, you can specify the VLAN ID and the MAC address. If you do not specify a VLAN ID, the native VLAN is used. • access—On an access port, specifies the VLAN as an access VLAN. • voice—On an access port, specifies the VLAN as a voice VLAN. **Note** The voice keyword is available only if a voice VLAN is configured on a port and if that port is not the access VLAN.

	Command or Action	**Purpose**
Step 12	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 13	show port-security **Example:** Device# **show port-security**	Verifies your entries.
Step 14	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 15	copy running-config startup-config **Example:** Device# **copy running-config** **startup-config**	(Optional) Saves your entries in the configuration file.

Enabling and Configuring Port Security Aging

Use this feature to remove and add devices on a secure port without manually deleting the existing secure MAC addresses and to still limit the number of secure addresses on a port. You can enable or disable the aging of secure addresses on a per-port basis.

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	interface interface-id **Example:**	Specifies the interface to be configured, and enter interface configuration mode.

	Command or Action	Purpose
	`Device(config)# `**`interface gigabitethernet`** **`1/0/1`**	
Step 4	switchport port-security aging {static | time time | type {absolute | inactivity}} **Example:** `Device(config-if)# `**`switchport`** **`port-security aging time 120`**	Enables or disable static aging for the secure port, or set the aging time or type. **Note** The switch does not support port security aging of sticky secure addresses. Enter static to enable aging for statically configured secure addresses on this port. For time, specifies the aging time for this port. The valid range is from 0 to 1440 minutes. For type, select one of these keywords: • absolute—Sets the aging type as absolute aging. All the secure addresses on this port age out exactly after the time (minutes) specified lapses and are removed from the secure address list. • inactivity—Sets the aging type as inactivity aging. The secure addresses on this port age out only if there is no data traffic from the secure source addresses for the specified time period.
Step 5	end **Example:** `Device(config)# `**`end`**	Returns to privileged EXEC mode.
Step 6	show port-security [interface interface-id] [address] **Example:** `Device# `**`show port-security interface`** **`gigabitethernet 1/0/1`**	Verifies your entries.
Step 7	show running-config **Example:** `Device# `**`show running-config`**	Verifies your entries.

	Command or Action	**Purpose**
Step 8	copy running-config startup-config **Example:** Device# `copy running-config` `startup-config`	(Optional) Saves your entries in the configuration file.

Configuration Examples for Port Security

This example shows how to enable port security on a port and to set the maximum number of secure addresses to 50. The violation mode is the default, no static secure MAC addresses are configured, and sticky learning is enabled.

```
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# switchport mode access
Device(config-if)# switchport port-security
Device(config-if)# switchport port-security maximum 50
Device(config-if)# switchport port-security mac-address sticky
```

This example shows how to configure a static secure MAC address on VLAN 3 on a port:

```
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# switchport mode trunk
Device(config-if)# switchport port-security
Device(config-if)# switchport port-security mac-address 0000.0200.0004 vlan 3
```

This example shows how to enable sticky port security on a port, to manually configure MAC addresses for data VLAN and voice VLAN, and to set the total maximum number of secure addresses to 20 (10 for data VLAN and 10 for voice VLAN).

```
Device(config)# interface tengigabitethernet 1/0/1
Device(config-if)# switchport access vlan 21
Device(config-if)# switchport mode access
Device(config-if)# switchport voice vlan 22
Device(config-if)# switchport port-security
Device(config-if)# switchport port-security maximum 20
Device(config-if)# switchport port-security violation restrict
Device(config-if)# switchport port-security mac-address sticky
Device(config-if)# switchport port-security mac-address sticky 0000.0000.0002
Device(config-if)# switchport port-security mac-address 0000.0000.0003
Device(config-if)# switchport port-security mac-address sticky 0000.0000.0001 vlan voice
Device(config-if)# switchport port-security mac-address 0000.0000.0004 vlan voice
Device(config-if)# switchport port-security maximum 10 vlan access
Device(config-if)# switchport port-security maximum 10 vlan voice
```

Information About Protocol Storm Protection

Protocol Storm Protection

When a switch is flooded with Address Resolution Protocol (ARP) or control packets, high CPU utilization can cause the CPU to overload. These issues can occur:

- Routing protocol can flap because the protocol control packets are not received, and neighboring adjacencies are dropped.

- Spanning Tree Protocol (STP) reconverges because the STP bridge protocol data unit (BPDU) cannot be sent or received.

- CLI is slow or unresponsive.

Using protocol storm protection, you can control the rate at which control packets are sent to the switch by specifying the upper threshold for the packet flow rate. The supported protocols are ARP, ARP snooping, Dynamic Host Configuration Protocol (DHCP) v4, DHCP snooping, Internet Group Management Protocol (IGMP), and IGMP snooping.

When the packet rate exceeds the defined threshold, the switch drops all traffic arriving on the specified virtual port for 30 seconds. The packet rate is measured again, and protocol storm protection is again applied if necessary.

For further protection, you can manually error disable the virtual port, blocking all incoming traffic on the virtual port. You can manually enable the virtual port or set a time interval for automatic re-enabling of the virtual port.

Note
Excess packets are dropped on no more than two virtual ports.

Virtual port error disabling is not supported for EtherChannel and Flexlink interfaces

Default Protocol Storm Protection Configuration

Protocol storm protection is disabled by default. When it is enabled, auto-recovery of the virtual port is disabled by default.

How to Configure Protocol Storm Protection

Enabling Protocol Storm Protection

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. - Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	psp {arp	dhcp	igmp} pps value **Example:** Device(config)# **psp dhcp pps 35**	Configures protocol storm protection for ARP, IGMP, or DHCP. For value, specifies the threshold value for the number of packets per second. If the traffic exceeds this value, protocol storm protection is enforced. The range is from 5 to 50 packets per second.
Step 4	errdisable detect cause psp **Example:** Device(config)# **errdisable detect cause psp**	(Optional) Enables error-disable detection for protocol storm protection. If this feature is enabled, the virtual port is error disabled. If this feature is disabled, the port drops excess packets without error disabling the port.
Step 5	errdisable recovery interval time **Example:** Device	(Optional) Configures an auto-recovery time (in seconds) for error-disabled virtual ports. When a virtual port is error-disabled, the switch auto-recovers after this time. The range is from 30 to 86400 seconds.
Step 6	end **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 7	show psp config {arp	dhcp	igmp} **Example:** Device# **show psp config dhcp**	Verifies your entries.

Monitoring Protocol Storm Protection

Command	Purpose
show psp config {arp	dhcp	igmp}	Verify your entries.

Additional References for Port-Based Traffic Control

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History for Port-Based Traffic Control

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	Port-Based Traffic Control	Port-based traffic control is a set of Layer 2 features on the Cisco Catalyst switches used to filter or block packets at the port level in response to specific traffic conditions.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

Cisco TrustSec SGT Caching

The Cisco TrustSec SGT Caching feature enhances the ability of Cisco TrustSec to make Security Group Tag (SGT) transportability flexible. This feature identifies IP-SGT bindings, and caches the corresponding SGTs so that network packets are forwarded through all the network services for normal deep-packet inspection processing, and at the service egress point the packets are re-tagged with the appropriate SGT.

Only IPv4 SGT caching is supported. High availability is supported for SGT caching.

Information about Cisco TrustSec

Cisco TrustSec provides security improvements to Cisco network devices based on the capability to strongly identify users, hosts, and network devices within a network. TrustSec provides topology-independent and scalable access controls by uniquely classifying data traffic for a particular role. TrustSec ensures data confidentiality and integrity by establishing trust among authenticated peers and encrypting links with those peers.

The key component of Cisco TrustSec is the Cisco Identity Services Engine (ISE). Cisco ISE can provision switches with TrustSec Identities and Security Group ACLs (SGACLs), though these may be configured manually on the switch.

Feature Information for Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 52: Feature Information for Cisco TrustSec

Feature Name	Releases	Feature Information
Cisco TrustSec	Cisco IOS XE Everest 16.6.1	Cisco TrustSec provides security improvements to Cisco network devices based on the capability to strongly identify users, hosts, and network devices within a network. CiscoTrustSec provides topology-independent and scalable access controls by uniquely classifying data traffic for a particular role. Cisco TrustSec ensures data confidentiality and integrity by establishing trust among authenticated peers and encrypting links with those peers.
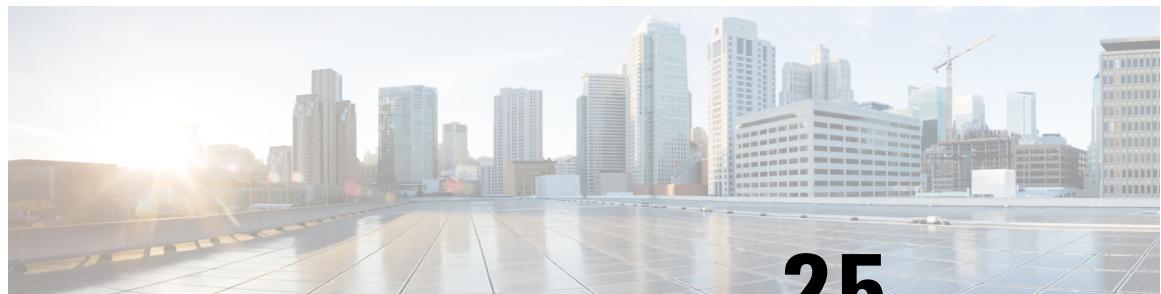
C H A P T E R 25

Configuring Control Plane Policing

Restrictions for CoPP

Restrictions for control plane policing (CoPP) include the following:

- Only ingress CoPP is supported. The system-cpp-policy policy-map is available on the control plane interface, and only in the ingress direction.

- Only the system-cpp-policy policy-map can be installed on the control plane interface.

- The system-cpp-policy policy-map and the 17 system-defined classes cannot be modified or deleted.

- Only the police action is allowed under the system-cpp-policy policy-map. The police rate for system-defined classes must be configured only in packets per second (pps); for user-defined class maps this must be configured only in bits per second (bps).

- We recommend not disabling the policer for a system-defined class map, that is, do not configure the no police rate   rate   pps command. Doing so affects the overall system health in case of high traffic towards the CPU. Further, even if you disable the policer rate for a system-defined class map, the system automatically reverts to the default policer rate after system bootup in order to protect the system bring-up process.

- When setting the policer rate, note that a clock frequency limitation causes differences in the `default rate` and the `set rate` values displayed for some classes (even if you set the default rate for all classes). See the User-Configurable Aspects of CoPP and Example: Setting the Default Policer Rates for All CPU Queues topics in this chapter for more information.

- Removing the policer rate configuration, disables CoPP on all affected queues.

- The show run command does not display information about classes configured under `system-cpp policy`, when they are left at default values. Use the show policy-map system-cpp-policy or the show policy-map control-plane commands instead.

You can continue use the show run command to display information about custom policies.

- A protocol with a huge number of CPU-bound packets may impact other protocols in the same class, as some of these protocols share the same policer. For example, Address Resolution Protocol (ARP) shares 4000 hardware policers with an array of host protocols like Telnet, Internet Control Message Protocol (ICMP), SSH, FTP, and SNMP in the system-cpp-police-forus class. If there is an ARP poisoning or an ICMP attack, hardware policers start throttling any incoming traffic that exceeds 4000 packets per second to protect the CPU and the overall integrity of the system. As a result, ARP and ICMP host protocols are dropped, along with any other host protocols that share the same class.

Information About CoPP

This chapter describes how control plane policing (CoPP) works on your device and how to configure it.

CoPP Overview

The CoPP feature improves security on your device protecting the CPU from unnecessary traffic and DoS attacks. It can also protect control and management traffic from traffic drops caused by high volumes of other, lower priority traffic.

Your device is typically segmented into three planes of operation, each with its own objective:

- The data plane, to forward data packets.

- The control plane, to route data correctly.

- The management plane, to manage network elements.

You can use CoPP to protect most of the CPU-bound traffic and ensure routing stability, reachability, and packet delivery. Most importantly, you can use CoPP to protect the CPU from a DoS attack.

CoPP uses the modular QoS command-line interface (MQC) and CPU queues to achieve these objectives. Different types of control plane traffic are grouped together based on certain criteria, and assigned to a CPU queue. You can manage these CPU queues by configuring dedicated policers in hardware. For example, you can modify the policer rate for certain CPU queues (traffic-type), or you can disable the policer for a certain type of traffic.

Although the policers are configured in hardware, CoPP does not affect CPU performance or the performance of the data plane. But since it limits the number of packets going to CPU, the CPU load is controlled. This means that services waiting for packets from hardware may see a more controlled rate of incoming packets (the rate being user-configurable).

System-Defined Aspects of CoPP

When you power-up the device for the first time, the system automatically performs the following tasks:

- Looks for policy-map system-cpp-policy. If not found, the system creates and installs it on the control-plane.

- Creates seventeen class-maps under system-cpp-policy.

  The next time you power-up the device, the system detects the policy and class maps that have already been created.

- Enables all CPU queues by default, with their respective default rate. The default rates are indicated in the table System-Defined Values for CoPP.

The following table lists the class-maps that the system creates when you load the device. It lists the policer that corresponds to each class-map and one or more CPU queues that are grouped under each class-map. There is a one-to-one mapping of a class-map to a policer; and one-to-many mapping of a class-map to CPU queues.

Table 53: System-Defined Values for CoPP

Class Maps Names	Policer Index (Policer No.)	CPU queues (Queue No.)
system-cpp- police-data	WK_CPP_POLICE_DATA(0)	WK_CPU_Q_ICMP_GEN(3) WK_CPU_Q_BROADCAST(12) WK_CPU_Q_ICMP_REDIRECT(6)
system-cpp-police-l2- control	WK_CPP_POLICE_L2_ CONTROL(1)	WK_CPU_Q_L2_CONTROL(1)
system-cpp-police-routing-control	WK_CPP_POLICE_ROUTING_CONTROL(2)	WK_CPU_Q_ROUTING_CONTROL(4) WK_CPU_Q_LOW_LATENCY (27)
system-cpp-police-control-low-priority	WK_CPP_POLICE_CO NTROL_LOW_PRI(3)	WK_CPU_Q_GENERAL_PUNT(25)
system-cpp-police-punt-webauth	WK_CPP_POLICE_PU NT_WEBAUTH(7)	WK_CPU_Q_PUNT_WEBAUTH(22)
system-cpp-police- topology-control	WK_CPP_POLICE_TOPOLOGY_CONTROL(8)	WK_CPU_Q_TOPOLOGY_CONTROL(15)
system-cpp-police- multicast	WK_CPP_POLICE_MULTICAST(9)	WK_CPU_Q_TRANSIT_TRAFFIC(18) WK_CPU_Q_MCAST_DATA(30)
system-cpp-police-sys- data	WK_CPP_POLICE_SYS _DATA(10)	WK_CPU_Q_LEARNING_CACHE_OVFL(13) WK_CPU_Q_CRYPTO_CONTROL(23) WK_CPU_Q_EXCEPTION(24) WK_CPU_Q_EGR_EXCEPTION(28) WK_CPU_Q_NFL_SAMPLED_DATA(26) WK_CPU_Q_GOLD_PKT(31) WK_CPU_Q_RPF_FAILED(19)
system-cpp-police-dot1x-auth	WK_CPP_POLICE_DOT1X(11)	WK_CPU_Q_DOT1X_AUTH(0)
system-cpp-police- protocol-snooping	WK_CPP_POLICE_PR(12)	WK_CPU_Q_PROTO_SNOOPING(16)
system-cpp-police-sw-forward	WK_CPP_POLICE_SW_FWD (13)	WK_CPU_Q_SW_FORWARDING_Q(14) WK_CPU_Q_LOGGING(21) WK_CPU_Q_L2_LVX_DATA_PACK(11)

Class Maps Names	Policer Index (Policer No.)	CPU queues (Queue No.)
system-cpp-police-forus	WK_CPP_POLICE_FORUS(14)	WK_CPU_Q_FORUS_ADDR_RESOLUTION(5) WK_CPU_Q_FORUS_TRAFFIC(2)
system-cpp-police-multicast-end-station	WK_CPP_POLICE_MULTICAST_SNOOPING(5)	WK_CPU_Q_MCAST_END_STATION_SERVICE(20)
system-cpp-default	WK_CPP_POLICE_DEFAULT_POLICER(16)	WK_CPU_Q_DHCP_SNOOPING(17) WK_CPU_Q_UNUSED(7) WK_CPU_Q_EWLC_CONTROL(9) WK_CPU_Q_EWLC_DATA(10)
system-cpp-police-stackwise-virt-control	WK_CPP_STACKWISE_VIRTUAL_CONTROL(6)	WK_CPU_Q_STACKWISE_VIRTUAL_CONTROL(29)
system-cpp-police-l2lvx-control	WK_CPP_L2_LVX_CONT_PACK(4)	WK_CPU_Q_L2_LVX_CONT_PACK(8)

The following table lists the CPU queues and the feature(s) associated with each CPU queue.

Table 54: CPU Queues and Associated Feature(s)

CPU queues (Queue No.)	Feature(s)
WK_CPU_Q_DOT1X_AUTH(0)	IEEE 802.1x Port-Based Authentication
WK_CPU_Q_L2_CONTROL(1)	Dynamic Trunking Protocol (DTP) VLAN Trunking Protocol (VTP) Port Aggregation Protocol (PAgP) Client Information Signaling Protocol (CISP) Message session relay protocol Multiple VLAN Registration Protocol (MVRP) Metropolitan Mobile Network (MMN) Link Level Discovery Protocol (LLDP) UniDirectional Link Detection (UDLD) Link Aggregation Control Protocol (LACP) Cisco Discovery Protocol (CDP) Spanning Tree Protocol (STP)
WK_CPU_Q_FORUS_TRAFFIC(2)	Host such as Telnet,Pingv4 and Pingv6, and SNMP Keepalive / loopback detection Initiate-Internet Key Exchange (IKE) protocol (IPSec)

CPU queues (Queue No.)	Feature(s)
WK_CPU_Q_ICMP_GEN(3)	ICMP - destination unreachable ICMP-TTL expired
WK_CPU_Q_ROUTING_CONTROL(4)	Routing Information Protocol version 1 (RIPv1) RIPv2 Interior Gateway Routing Protocol (IGRP) Border Gateway Protocol (BGP) PIM-UDP Virtual Router Redundancy Protocol (VRRP) Hot Standby Router Protocol version 1 (HSRPv1) HSRPv2 Gateway Load Balancing Protocol (GLBP) Label Distribution Protocol (LDP) Web Cache Communication Protocol (WCCP) Routing Information Protocol next generation (RIPng) Open Shortest Path First (OSPF) Open Shortest Path First version 3(OSPFv3) Enhanced Interior Gateway Routing Protocol (EIGRP) Enhanced Interior Gateway Routing Protocol version 6 (EIGRPv6) DHCPv6 Protocol Independent Multicast (PIM) Protocol Independent Multicast version 6 (PIMv6) Hot Standby Router Protocol next generation (HSRPng) IPv6 control Generic Routing Encapsulation (GRE) keepalive Network Address Translation (NAT) punt Intermediate System-to-Intermediate System (IS-IS)
WK_CPU_Q_FORUS_ADDR_RESOLUTION(5)	Address Resolution Protocol (ARP) IPv6 neighbor advertisement and neighbor solicitation
WK_CPU_Q_ICMP_REDIRECT(6)	Internet Control Message Protocol (ICMP) redirect
WK_CPU_Q_UNUSED (7)	Unused
WK_CPU_Q_L2_LVX_CONT_PACK(8)	Exchange ID (XID) packet

CPU queues (Queue No.)	Feature(s)
WK_CPU_Q_EWLC_CONTROL(9)	Embedded Wirelss Controller (eWLC) [Control and Provisioning of Wireless Access Points (CAPWAP) (UDP 5246)]
WK_CPU_Q_EWLC_DATA(10)	eWLC data packet (CAPWAP DATA, UDP 5247)
WK_CPU_Q_L2_LVX_DATA_PACK(11)	Unknown unicast packet punted for map request.
WK_CPU_Q_BROADCAST(12)	All types of broadcast
WK_CPU_Q_LEARNING_CACHE_OVFL(13)	Learning cache overflow (Layer 2 + Layer 3)
WK_CPU_Q_SW_FORWARDING_Q(14)	Data - access control list (ACL) Full Data - IPv4 options Data - IPv6 hop-by-hop Data - out-of-resources / catch all Data - Reverse Path Forwarding (RPF) incomplete Glean packet
WK_CPU_Q_TOPOLOGY_CONTROL(15)	Spanning Tree Protocol (STP) Resilient Ethernet Protocol (REP) Shared Spanning Tree Protocol (SSTP)
WK_CPU_Q_PROTO_SNOOPING(16)	Address Resolution Protocol (ARP) snooping for Dynamic ARP Inspection (DAI)
WK_CPU_Q_DHCP_SNOOPING(17)	DHCP snooping
WK_CPU_Q_TRANSIT_TRAFFIC(18)	This is used for packets punted by NAT, which need to be handled in the software path.
WK_CPU_Q_RPF_FAILED(19)	Data – mRPF (multicast RPF) failed
WK_CPU_Q_MCAST_END_STATION_SERVICE(20)	Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD) control
WK_CPU_Q_LOGGING(21)	Access control list (ACL) logging
WK_CPU_Q_PUNT_WEBAUTH(22)	Web Authentication
WK_CPU_Q_CRYPTO_CONTROL(23)	Crypto Controls

CPU queues (Queue No.)	Feature(s)
WK_CPU_Q_EXCEPTION(24)	IKE indication IP learning violation IP port security violation IP Static address violation IPv6 scope check Remote Copy Protocol (RCP) exception Unicast RPF fail
WK_CPU_Q_GENERAL_PUNT(25)	General punt
WK_CPU_Q_NFL_SAMPLED_DATA(26)	Netflow sampled data and Media Services Proxy (MSP)
WK_CPU_Q_LOW_LATENCY(27)	Bidirectional Forwarding Detection (BFD), Precision Time Protocol (PTP)
WK_CPU_Q_EGR_EXCEPTION(28)	Egress resolution exception
WK_CPU_Q_STACKWISE_VIRTUAL _CONTROL(29)	Front side stacking protocols, namely SVL
WK_CPU_Q_MCAST_DATA(30)	Data - (S,G) creation Data - local joins Data - PIM Registration Data - SPT switchover Data - Multicast
WK_CPU_Q_GOLD_PKT(31)	Gold

User-Configurable Aspects of CoPP

You can perform these tasks to manage control plane traffic:

✎

Note All `system-cpp-policy` configurations must be saved so they are retained after reboot.

Enable or Disable a Policer for CPU Queues

Enable a policer for a CPU queue, by configuring a policer action (in packets per second) under the corresponding class-map, within the `system-cpp-policy` policy-map.

Disable a policer for CPU queue, by removing the policer action under the corresponding class-map, within the `system-cpp-policy` policy-map.

Note　If a default policer is already present, carefully consider and control its removal; otherwise the system may see a CPU hog or other anomalies, such as control packet drops.

Change the Policer Rate

You can do this by configuring a policer rate action (in packets per second), under the corresponding class-map, within the `system-cpp-policy` policy-map.

When setting a policer rate, note that the rate you set is automatically converted to the nearest multiple of 200. For instance, if you set the policer rate of a CPU queue 100 pps, the system changes it to 200; or if set the policer rate to 650, the system changes it to 600. See Example: Setting the Default Policer Rates for All CPU Queues in this chapter, for sample output that displays this behavior.

Set Policer Rates to Default

Set the policer for CPU queues to their default values, by entering the cpp system-default command in global configuration mode.

Create User-Defined Class Maps

If a given traffic class does not have a designated class map, and you want to protect this traffic, you can create specific class maps (with filters) for such traffic packets and add these user-defined class maps to `system-cpp-policy`.

While `system-cpp-policy` is applied in the ingress direction, the forwarding engine driver (FED) changes policers on user-defined class maps to the egress. The filters and the policers in all user-defined classes must therefore be applied as egress classifications and actions, respectively. The policy map itself is unaffected by this change in the direction.

When you add a user-defined class map to `system-cpp-policy`, the system automatically installs it on all 32 CPU queues (in addition to the control plane), resulting in 33 instances of the policy. You can see this by entering the command in privileged EXEC mode.

The police rate on these class maps is controlled by the Active Queue Management (AQM) policer. AQM provides buffering control of traffic flows prior to queuing a packet into the transmit queue of a port, ensuring that certain flows do not hog the switch packet memory. If the AQM policer feature is enabled, any user-defined police rates exceeding the AQM policer limits are disregarded.

User defined class maps have normal QoS or ACL classification filters.

Upgrading or Downgrading the Software Version

Software Version Upgrades and CoPP

When you upgrade the software version on your device, the system checks and make the necessary updates as required for CoPP (For instance, it checks for the `system-cpp-policy` policy map and creates it if missing). You may also have to complete certain tasks before or after the upgrade activity. This is to ensure that any configuration updates are reflected correctly and CoPP continues to work as expected. Depending on the method you use to upgrade the software, upgrade-related tasks may be optional or recommended in some scenarios, and mandatory in others.

The system actions and user actions for an upgrade, are described here. Also included, are any release-specfic caveats.

System Actions for an Upgrade

When you upgrade the software version on your device, the system performs these actions. This applies to all upgrade methods:

- If the device did not have a `system-cpp-policy` policy map before upgrade, then on upgrade, the system creates a default policy map.

- If the device had a `system-cpp-policy` policy map before upgrade, then on upgrade, the system does not re-generate the policy.

User Actions for an Upgrade

User actions for an upgrade – depending on upgrade method:

Upgrade Method	Condition	Action Time and Action	Purpose
Regular[19]	None	After upgrade (required) Enter the cpp system-default command in global configuration mode	To get the latest, default policer rates.

[19] Refers to a software upgrade method that involves a reload of the switch. Can be install or bundle mode.

Software Version Downgrades and CoPP

The system actions and user actions for a downgrade, are described here.

System Actions for a Downgrade

When you downgrade the software version on your device, the system performs these actions. This applies to all downgrade methods:

- The system retains the `system-cpp-policy` policy map on the device, and installs it on the control plane.

User Actions for a Downgrade

User actions for a downgrade:

Upgrade Method	Condition	Action Time and Action	Purpose
Regular[20]	None	No action required	Not applicable

[20] Refers to a software upgrade method that involves a reload of the switch. Can be install or bundle mode.

If you downgrade the software version and then again upgrade, the system action and user actions that apply are the same as those mentioned for upgrades.

How to Configure CoPP

Enabling a CPU Queue or Changing the Policer Rate

The procedure to enable a CPU queue and change the policer rate of a CPU queue is the same. Follow these steps:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	policy-map policy-map-name **Example:** Device(config)# **policy-map system-cpp-policy** Device(config-pmap)#	Enters the policy map configuration mode.
Step 4	class class-name **Example:** Device(config-pmap)# **class system-cpp-police-protocol-snooping** Device(config-pmap-c)#	Enters the class action configuration mode. Enter the name of the class that corresponds to the CPU queue you want to enable. See table System-Defined Values for CoPP.
Step 5	police rate rate pps **Example:** Device(config-pmap-c)# **police rate 100 pps** Device(config-pmap-c-police)#	Specifies an upper limit on the number of incoming packets processed per second, for the specified traffic class. **Note** The rate you specify is applied to all CPU queues that belong to the class-map you have specified.
Step 6	exit **Example:** Device(config-pmap-c-police)# **exit**	Returns to the global configuration mode.

	Command or Action	Purpose

	`Device(config-pmap-c)# exit`
	`Device(config-pmap)# exit`
	`Device(config)#`	

Step 7	control-plane
	Example:
	`Device(config)# control-plane`
	`Device(config-cp)#`
		Enters the control plane (config-cp) configuration mode

Step 8	service-policy input policy-name
	Example:
	`Device(config)# control-plane`
	`Device(config-cp)#service-policy input`
	` system-cpp-policy`
	`Device(config-cp)#`
		Installs system-cpp-policy in FED. This command is required for you to see the FED policy. Not configuring this command will lead to an error.

Step 9	end **Example:** `Device(config-cp)# end`	Returns to the privileged EXEC mode.
Step 10	show policy-map control-plane **Example:** `Device# show policy-map control-plane`	Displays all the classes configured under `system-cpp policy`, the rates configured for the various traffic types, and statistics

Disabling a CPU Queue

Follow these steps to disable a CPU queue:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	**Purpose**
Step 3	policy-map policy-map-name **Example:** Device(config)# **policy-map system-cpp-policy** Device(config-pmap)#	Enters the policy map configuration mode.
Step 4	class class-name **Example:** Device(config-pmap)# **class system-cpp-police-protocol-snooping** Device(config-pmap-c)#	Enters the class action configuration mode. Enter the name of the class that corresponds to the CPU queue you want to disable. See the table, System-Defined Values for CoPP.
Step 5	no police rate rate pps **Example:** Device(config-pmap-c)# **no police rate 100 pps**	Disables incoming packet processing for the specified traffic class. **Note** This disables all CPU queues that belong to the class-map you have specified.
Step 6	end **Example:** Device(config-pmap-c)# **end**	Returns to the privileged EXEC mode.
Step 7	show policy-map control-plane **Example:** Device# **show policy-map control-plane**	Displays all the classes configured under system-cpp policy and the rates configured for the various traffic types and statistics.

Setting the Default Policer Rates for All CPU Queues

Follow these steps to set the policer rates for all CPU queues to their default rates:

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:**	Enters global configuration mode.

	Command or Action	**Purpose**
	Device# **configure terminal**	
Step 3	cpp system-default **Example:** Device(config)# **cpp system-default** Defaulting CPP : Policer rate for all classes will be set to their defaults	Sets the policer rates for all the classes to the default rate.
Step 4	end **Example:** Device(config)# **end**	Returns to the privileged EXEC mode.
Step 5	show platform hardware fed{active | standby}qos queue stats internal cpu policer **Example:** Device# **show platform hardware fed active qos queue stats internal cpu policer**	Displays device-specific internal queue information.

Creating A User-Defined Class Map

Follow these steps to create user-defined class maps in system-cpp-policy and set the policer rates in bps

Procedure

	Command or Action	**Purpose**
Step 1	enable **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	class-map class-map-name **Example:**	Specify the class map you want to create. Enters the class map configuration mode.

	Command or Action	Purpose
	Device(config)# **class-map example_class** Device(config-cmap)#	
Step 4	exit **Example:** Device(config-cmap)# **exit** Device(config)#	Exits the class map configuration mode.
Step 5	policy-map policy-map-name **Example:** Device(config)# **policy-map** **system-cpp-policy** Device(config-pmap)#	Enter the policy map name. Enters the policy map configuration mode.
Step 6	class-map class-map-name **Example:** Device(config-pmap)# **class example_class** Device(config-pmap-c)#	Enters the class action configuration mode. Enter the name of the class.
Step 7	[no] police rate target_bit_rate **Example:** Device(config-pmap-c)# **police 90000**	Specifies the bit rate per second, enter a value between 8000 and 10000000000. **Note** The police rate for user-defined class-maps must not exceed 10000 pps worth of traffic.
Step 8	end **Example:** Device(config-pmap-c-police)# **end** Device#	Returns to the privileged EXEC mode.
Step 9	show policy-map control-plane **Example:** Device# **show policy-map control-plane**	Displays all the classes configured under system-cpp policy, including the user-defined class maps, and the rates configured.

Configuration Examples for CoPP

Example: Enabling a CPU Queue or Changing the Policer Rate of a CPU Queue

This example shows how to enable a CPU queue or to change the policer rate of a CPU queue. Here the **class system-cpp-police-protocol-snooping** CPU queue is enabled with the policer rate of **2000 pps** .

```
Device> enable
Device# configure terminal
Device(config)# policy-map system-cpp-policy
Device(config-pmap)# class system-cpp-police-protocol-snooping
Device(config-pmap-c)# police rate 2000 pps
Device(config-pmap-c-police)# end


Device# show policy-map control-plane
Control Plane

  Service-policy input: system-cpp-policy

    <output truncated>


    Class-map: system-cpp-police-dot1x-auth (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: none
      police:
          rate 1000 pps, burst 244 packets
        conformed 0 bytes; actions:
          transmit
        exceeded 0 bytes; actions:
          drop

    Class-map: system-cpp-police-protocol-snooping (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: none
      police:
          rate 2000 pps, burst 488 packets
        conformed 0 bytes; actions:
          transmit
        exceeded 0 bytes; actions:
          drop

    <output truncated>

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

Example: Disabling a CPU Queue

This example shows how to disable a CPU queue. Here the **class system-cpp-police-protocol-snooping** CPU queue is disabled.

Example: Setting the Default Policer Rates for All CPU Queues

This example shows how to set the policer rates for all CPU queues to their default and then verify the setting.

Note For some CPU queues, the `default rate` and the `set rate` values will not be the same, even if you set the default rate for all classes. This because the set rate is rounded off to the nearest multiple of 200. This behavior is controlled by the clock speed of your device. In the sample output below, the default and set rate values for `DHCP Snooping` and `NFL SAMPLED DATA` display this difference.

```
Device> enable
Device# configure terminal
Device(config)# cpp system-default
Defaulting CPP : Policer rate for all classes will be set to their defaults
Device(config)# end

Device# show platform hardware fed active qos queue stats internal cpu policer

CPU Queue Statistics
================================================================================================

                                           (default) (set)    Queue        Queue
QId PlcIdx  Queue Name           Enabled   Rate      Rate     Drop(Bytes)  Drop(Frames)
------------------------------------------------------------------------------------------------
0   11      DOT1X Auth           Yes       1000      1000     0            0

1   1       L2 Control           Yes       2000      2000     0            0

2   14      Forus traffic        Yes       1000      1000     0            0

3   0       ICMP GEN             Yes       600       600      0            0

4   2       Routing Control      Yes       5400      5400     0            0

5   14      Forus Address resolution  Yes  1000      1000     0            0

6   0       ICMP Redirect        Yes       600       600      0            0

7   16      Inter FED Traffic    Yes       2000      2000     0            0

8   4       L2 LVX Cont Pack     Yes       1000      1000     0            0

9   16      EWLC Control         Yes       2000      2000     0            0

10  16      EWLC Data            Yes       2000      2000     0            0

11  13      L2 LVX Data Pack     Yes       1000      1000     0            0

12  0       BROADCAST            Yes       600       600      0            0

13  10      Openflow             Yes       100       200      0            0

14  13      Sw forwarding        Yes       1000      1000     0            0

15  8       Topology Control     Yes       13000     13000    0            0

16  12      Proto Snooping       Yes       2000      2000     0            0

17  6       DHCP Snooping        Yes       500       400      0            0

18  9       Transit Traffic      Yes       500       400      0            0

19  10      RPF Failed           Yes       100       200      0            0
```

20	15	MCAST END STATION	Yes	2000	2000	0	0
21	13	LOGGING	Yes	1000	1000	0	0
22	7	Punt Webauth	Yes	1000	1000	0	0
23	18	High Rate App	Yes	13000	13000	0	0
24	10	Exception	Yes	100	200	0	0
25	3	System Critical	Yes	1000	1000	0	0
26	10	NFL SAMPLED DATA	Yes	100	200	0	0
27	2	Low Latency	Yes	5400	5400	0	0
28	10	EGR Exception	Yes	100	200	0	0
29	5	Stackwise Virtual OOB	Yes	8000	8000	0	0
30	9	MCAST Data	Yes	500	400	0	0
31	10	Gold Pkt	Yes	100	200	0	0

```
* NOTE: CPU queue policer rates are configured to the closest hardware supported value

                     CPU Queue Policer Statistics
========================================================================
Policer   Policer Accept   Policer Accept  Policer Drop  Policer Drop
  Index       Bytes           Frames         Bytes          Frames
-----------------------------------------------------------------
0         0                0               0             0
1         0                0               0             0
2         0                0               0             0
3         0                0               0             0
4         0                0               0             0
5         0                0               0             0
6         0                0               0             0
7         0                0               0             0
8         0                0               0             0
9         0                0               0             0
10        0                0               0             0
11        0                0               0             0
12        0                0               0             0
13        0                0               0             0
14        0                0               0             0
15        0                0               0             0
16        0                0               0             0
17        0                0               0             0
18        0                0               0             0

                     CPP Classes to queue map
===============================================================================
PlcIdx CPP Class                                   :  Queues
-------------------------------------------------------------------------------
0      system-cpp-police-data                      :  ICMP GEN/BROADCAST/ICMP Redirect/
10     system-cpp-police-sys-data                  :  Openflow/Exception/EGR Exception/NFL
SAMPLED DATA/Gold Pkt/RPF Failed/
13     system-cpp-police-sw-forward                :  Sw forwarding/LOGGING/L2 LVX Data Pack/
9      system-cpp-police-multicast                 :  Transit Traffic/MCAST Data/
15     system-cpp-police-multicast-end-station     :  MCAST END STATION /
7      system-cpp-police-punt-webauth              :  Punt Webauth/
```

```
1     system-cpp-police-l2-control              :  L2 Control/
2     system-cpp-police-routing-control         :  Routing Control/Low Latency/
3     system-cpp-police-system-critical         :  System Critical/
4     system-cpp-police-l2lvx-control           :  L2 LVX Cont Pack/
8     system-cpp-police-topology-control        :  Topology Control/
11    system-cpp-police-dot1x-auth              :  DOT1X Auth/
12    system-cpp-police-protocol-snooping       :  Proto Snooping/
6     system-cpp-police-dhcp-snooping           :  DHCP Snooping/
14    system-cpp-police-forus                   :  Forus Address resolution/Forus traffic/
5     system-cpp-police-stackwise-virt-control  :  Stackwise Virtual OOB/
16    system-cpp-default                        :  Inter FED Traffic/EWLC Control/EWLC Data/
18    system-cpp-police-high-rate-app           :  High Rate App/
```

Example: Creating a User-Defined Class Map

Device

This example shows how to create a user-defined class map, apply it to `system-cpp-policy` and display information about where the policy is applied.

A user-defined class map is applied to `system-cpp-policy`, which means that any control traffic matching the user-defined class map `class-cpp-user` is subject to the aggregate policer, under the user-defined class map. Statistics for the user defined traffic class are reported in Bytes.

```
Device> enable
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# class-map match-any class-cpp-user
Device(config-cmap)# match dscp cs1
Device(config-cmap)# exit
Device(config)# policy-map system-cpp-policy
Device(config-pmap)# class class-cpp-user
Device(config-pmap-c)# police rate 2m bps
Device(config-pmap-c-police)# end

Device# show policy-map control-plane
<output truncated>
Class-map: class-cpp-user (match-any)
     0 packets, 0 bytes
     5 minute offered rate 0000 bps, drop rate 0000 bps
     Match:  dscp cs1 (8)
     police:
         rate 2000000 bps, burst 62500 bytes
       conformed 0 bytes; actions:
         transmit
       exceeded 0 bytes; actions:
         drop
       conformed 0000 bps, exceeded 0000 bps
<output truncated>
```

When you add a user-defined class map to `system-cpp-policy`, the system automatically installs it on all 32 CPU queues, in addition to the control plane (resulting in 33 instances of the policy).

Note how the direction is display as egress (OUT), even though system-cpp-policy is applied in the ingress

```
Device# show platform software fed switch active qos policy target status

TCG status summary:

Loc Interface           IIF-ID          Dir State:(cfg,opr) Policy
```

```
--- -------------------- --------------- --- -------------- --------------------
?:255 Control Plane       0x00000001000001 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-0          0x0000000100000d OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-1          0x0000000100000e OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-2          0x0000000100000f OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-3          0x00000001000010 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-4          0x00000001000011 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-5          0x00000001000012 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-6          0x00000001000013 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-7          0x00000001000014 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-8          0x00000001000015 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-9          0x00000001000016 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-10         0x00000001000017 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-11         0x00000001000018 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-12         0x00000001000019 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-13         0x0000000100001a OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-14         0x0000000100001b OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-15         0x0000000100001c OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-16         0x0000000100001d OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-17         0x0000000100001e OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-18         0x0000000100001f OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-19         0x00000001000020 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-20         0x00000001000021 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-21         0x00000001000022 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-22         0x00000001000023 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-23         0x00000001000024 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-24         0x00000001000025 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-25         0x00000001000026 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-26         0x00000001000027 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-27         0x00000001000028 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-28         0x00000001000029 OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-29         0x0000000100002a OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-30         0x0000000100002b OUT VALID,SET_INHW  system-cpp-policy
?:0 CoPP-Queue-31         0x0000000100002c OUT VALID,SET_INHW  system-cpp-policy
```

Monitoring CoPP

Use these commands to display policer settings, such as, traffic types and policer rates (user-configured and default rates) for CPU queues:

Command	Purpose
show policy-map control-plane	Displays the rates configured for the various traffic types
show policy-map system-cpp-policy	Displays all the classes configured under system-cpp policy, and policer rates

Feature History and Information for CoPP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Feature	Release	Feature Information
Control Plane Policing (CoPP) or CPP	Cisco IOS XE Everest 16.6.1	This feature was introduced.
Change in the system behavior for policer rates that are set.	Cisco IOS XE Everest 16.6.4	For some CPU queues, the default rate and the set rate values will not be the same, even if you set the default rate for all classes. This because the set rate is rounded off to the nearest multiple of 200.