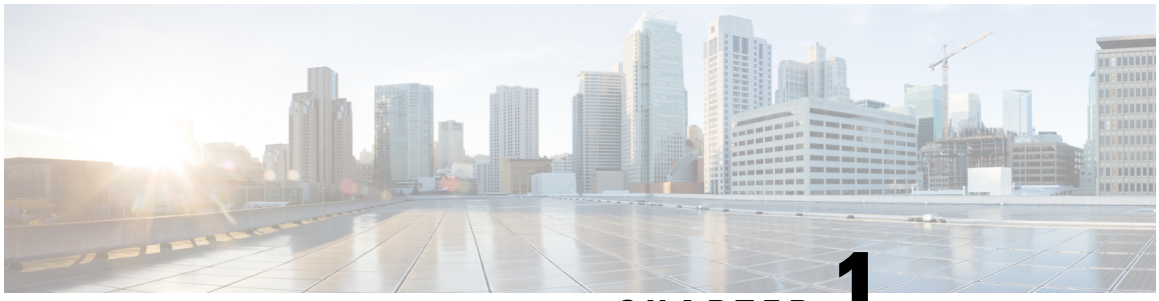
Security Configuration Guide, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

First Published: 2019-03-27

Last Modified: 2019-12-10

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

CONTENTS



CHAPTER 1

Security Features Overview

Security Features Overview

The security features are as follows:

- Web Authentication—Allows a supplicant (client) that does not support IEEE 802.1x functionality to be authenticated using a web browser.
- Local Web Authentication Banner—A custom banner or an image file displayed at a web authentication login screen.
- IEEE 802.1x Authentication with ACLs and the RADIUS Filter-Id Attribute
- Password-protected access (read-only and read-write access) to management interfaces (device manager, Network Assistant, and the CLI) for protection against unauthorized configuration changes
- Multilevel security for a choice of security level, notification, and resulting actions
- Static MAC addressing for ensuring security
- Protected port option for restricting the forwarding of traffic to designated ports on the same switch
- Port security option for limiting and identifying MAC addresses of the stations allowed to access the port
- VLAN aware port security option to shut down the VLAN on the port when a violation occurs, instead of shutting down the entire port.
- Port security aging to set the aging time for secure addresses on a port.
- Protocol storm protection to control the rate of incoming protocol traffic to a switch by dropping packets that exceed a specified ingress rate.
- BPDU guard for shutting down a Port Fast-configured port when an invalid configuration occurs.
- Standard and extended IP access control lists (ACLs) for defining inbound security policies on Layer 2 interfaces (port ACLs).
- Extended MAC access control lists for defining security policies in the inbound direction on Layer 2 interfaces.

- Source and destination MAC-based ACLs for filtering non-IP traffic.
- DHCP snooping to filter untrusted DHCP messages between untrusted hosts and DHCP servers.
- IP source guard to restrict traffic on nonrouted interfaces by filtering traffic based on the DHCP snooping database and IP source bindings.
- Dynamic ARP inspection to prevent malicious attacks on the switch by not relaying invalid ARP requests and responses to other ports in the same VLAN.
- IEEE 802.1x port-based authentication to prevent unauthorized devices (clients) from gaining access to the network. The following 802.1x features are supported:
  - Support for single-host, multi-host, multi-auth, and multi-domain-auth modes.

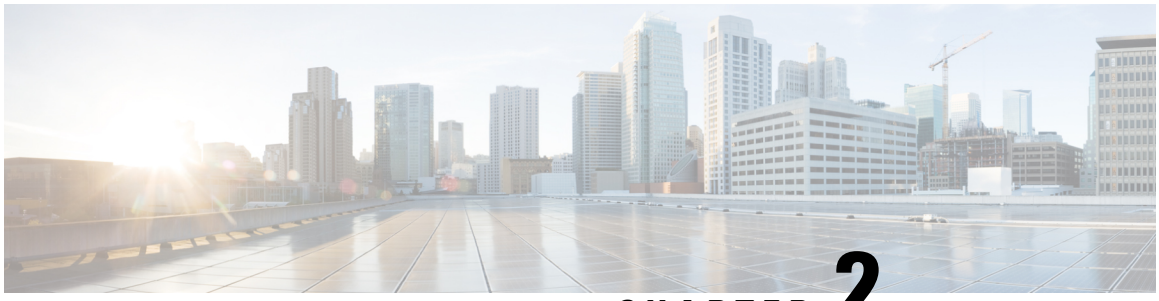
Mode	Description
Single-Host	Only one host can be authenticated. Security violation occurs if more than one client tries to authenticate.
Multi-Host	Only first host needs to authenticate. Remaining hosts get access without authentication.
Multi-Auth	Every client must get authenticated.
Multi-Domain-Auth	One VoIP client and one data client is allowed to authenticate. Security violation occurs if more than one client tries to authenticate.

  - Multidomain authentication (MDA) to allow both a data device and a voice device, such as an IP phone (Cisco or non-Cisco), to independently authenticate on the same IEEE 802.1x-enabled switch port.
  - Dynamic voice virtual LAN (VLAN) for MDA to allow a dynamic voice VLAN on an MDA-enabled port.
  - VLAN assignment for restricting 802.1x-authenticated users to a specified VLAN.
  - Support for VLAN assignment on a port configured for multi-auth mode. The RADIUS server assigns a VLAN to the first host to authenticate on the port, and subsequent hosts use the same VLAN. Voice VLAN assignment is supported for one IP phone.
  - Voice VLAN to permit a Cisco IP Phone to access the voice VLAN regardless of the authorized or unauthorized state of the port.
  - IP phone detection enhancement to detect and recognize a Cisco IP phone.
  - Guest VLAN to provide limited services to non-802.1x-compliant users.
  - Restricted VLAN to provide limited services to users who are 802.1x compliant, but do not have the credentials to authenticate via the standard 802.1x processes.
  - 802.1x accounting to track network usage.
  - 802.1x with wake-on-LAN to allow dormant PCs to be powered on based on the receipt of a specific Ethernet frame.

  - 802.1x readiness check to determine the readiness of connected end hosts before configuring IEEE 802.1x on the switch.
  - Voice aware 802.1x security to apply traffic violation actions only on the VLAN on which a security violation occurs.
  - MAC authentication bypass (MAB) to authorize clients based on the client MAC address.
  - Network Admission Control (NAC) Layer 2 802.1x validation of the antivirus condition or posture of endpoint systems or clients before granting the devices network access.
  - Network Edge Access Topology (NEAT) with 802.1X switch supplicant, host authorization with CISP, and auto enablement to authenticate a switch outside a wiring closet as a supplicant to another switch.
  - IEEE 802.1x with open access to allow a host to access the network before being authenticated.
  - IEEE 802.1x authentication with downloadable ACLs and redirect URLs to allow per-user ACL downloads from a RADIUS server or Cisco Identity Services Engine (ISE) to an authenticated switch.
  - Support for dynamic creation or attachment of an auth-default ACL on a port that has no configured static ACLs.
  - Flexible-authentication sequencing to configure the order of the authentication methods that a port tries when authenticating a new host.

- TACACS+, a proprietary feature for managing network security through a TACACS server for both IPv4 and IPv6.
- RADIUS for verifying the identity of, granting access to, and tracking the actions of remote users through authentication, authorization, and accounting (AAA) services.
- Enhancements to RADIUS, TACACS+, and SSH functionality.
- Secure Socket Layer (SSL) Version 3.0 support for the HTTP 1.1 server authentication, encryption, and message integrity and HTTP client authentication to allow secure HTTP communications (requires the cryptographic version of the software).
- IEEE 802.1x Authentication with ACLs and the RADIUS Filter-Id Attribute.
- RADIUS Change of Authorization (CoA) to change the attributes of a certain session after it is authenticated. When there is a change in policy for a user or user group in AAA, administrators can send the RADIUS CoA packets from the AAA server, such as Cisco Identity Services Engine, or Cisco Secure ACS to reinitialize authentication, and apply to the new policies.
- IEEE 802.1x User Distribution to allow deployments with multiple VLANs (for a group of users) to improve scalability of the network by load balancing users across different VLANs. Authorized users are assigned to the least populated VLAN in the group, assigned by RADIUS server.
- Support for critical VLAN multi-host/multi-auth enabled ports are placed in a critical VLAN in order to permit access to critical resources if AAA server becomes unreachable.
- Support for Network Edge Access Topology (NEAT) to change the port host mode and to apply a standard port configuration on the authenticator switch port.
- MAC address based authentication using MAC Authentication Bypass (MAB). Authenticated hosts are moved to a dynamic VLAN to prevent network access from unauthorized VLANs.

- MAC move to allow hosts (including the hosts connected behind an IP phone) to move across ports within the same switch without any restrictions to enable mobility. With MAC move, the switch treats the reappearance of the same MAC address on another port in the same way as a completely new MAC address.
- Support for 3DES and AES with version 3 of the Simple Network Management Protocol (SNMPv3). This release adds support for the 168-bit Triple Data Encryption Standard (3DES) and the 128-bit, 192-bit, and 256-bit Advanced Encryption Standard (AES) encryption algorithms to SNMPv3.
- Cisco TrustSec SXP protocol is not supported.



CHAPTER 2

Controlling Switch Access with Passwords and Privilege Levels

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** command in global configuration mode. This command produces the boot loader prompt (*switch:*) after the switch is power cycled.

Restrictions and Guidelines for Reversible Password Types

- Type 6 encrypted password is supported from Cisco IOS Release 15.2(7)E2 and later releases. Autoconversion to password type 6 is supported from Cisco IOS Release 15.2(7)E3 and later releases.
- If the startup configuration has a type 6 password and you downgrade to a version in which type 6 password is not supported, you can/may be locked out of the device.

Restrictions and Guidelines for Irreversible Password Types

- Username secret password type 5 and enable secret password type 5 must be migrated to the stronger password type 8 or 9. For more information, see Protecting Enable and Enable Secret Passwords with Encryption, on page 10.
- Plain text passwords are converted to nonreversible encrypted password type 9.

Note This is supported in Cisco IOS Release 15.2(7)E3 and later releases.

Information About Passwords and Privilege Levels

The following sections provide information on passwords and privilege levels.

Preventing Unauthorized Access

You can prevent unauthorized users from reconfiguring your device and viewing configuration information. Typically, you want network administrators to have access to your device 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 device, you should configure one or more of these security features:

- At a minimum, you should configure passwords and privileges at each device port. These passwords are locally stored on the device. When users attempt to access the device through a port or line, they must enter the password specified for the port or line before they can access the device.
- For an additional layer of security, you can also configure username and password pairs, which are locally stored on the device. These pairs are assigned to lines or ports and authenticate each user before that user can access the device. If you have defined privilege levels, you can also assign a specific privilege level (with associated rights and privileges) to each username 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.

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 1: 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.

Feature	Default Setting
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

Unmasked Secret Password

To provide an additional layer of security, particularly for passwords that cross the network or that are stored on a TFTP server, you can use either the **enable password** or **enable secret** commands in global configuration mode. 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 vty passwords.

Masked Secret Password

With **enable secret** command, password is encrypted but is visible on the terminal when you type the password. To mask the password on the terminal, use the **masked-secret** global configuration command. The encryption type for this password is type 9, by default.

You can use this command to configure masked secret password for common criteria policy.

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 Virtual Terminal Protocol (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** command in global configuration mode.

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** 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

The following sections provide various configuration examples on 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.

To set or change a static enable password, perform this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **enable password** *password*
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	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. *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.
Step 4	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	
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

To establish an encrypted password that users must enter to access privileged EXEC mode (the default) or any privilege level, perform this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Use one of the following:
   - **enable password** [**level** *level*] {*unencrypted-password* | *encryption-type encrypted-password*}
   - **enable secret** [**level** *level*] {*unencrypted-password* | *encryption-type encrypted-password*}
4. **service password-encryption**
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	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*] {*unencrypted-password* | *encryption-type encrypted-password*} • **enable secret** [**level** *level*] {*unencrypted-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.

	Command or Action	Purpose

	Example:
	`Device(config)# enable password level 12 example123`
	or
	`Device(config)# enable secret 9 $9$sMLBsTFXLnnHTk$0L82`
		• (Optional) For *level*, the range is from 0 to 15. Level 1 is normal user EXEC mode privileges. The default level is 15 (privileged EXEC mode privileges).
		• For *unencrypted-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.
		• For *encryption-type*, the available options for **enable password** are type 0 and 7, and type 0, 5, 8, and 9 for **enable secret**. If you specify an encryption type, you must provide an encrypted password—an encrypted password that you copy from another switch configuration. Secret encryption type 9 is more secure, so we recommend that you select type 9 to avoid any issues while upgrading or downgrading.
		Note
		• If you do not specify an encryption type for the secret password, the password is auto converted to type 9.
		• If you specify an encryption type and then enter a clear text password, it will result in an error.
		• You can also configure type 9 encryption for the secret password manually by using the **algorithm-type scrypt** command in global configuration mode. For example:
		`Device(config)# username user1 algorithm-type scrypt secret cisco`
		Or
		`Device(config)# enable algorithm-type scrypt secret cisco`
		Run the **write memory** command in privileged EXEC mode for the type 9 secret to be permanently written into the startup configuration.

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

Configuring Masked Secret Password

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Use one of the following:
   - **username** *name***masked-secret**
   - **username** *name***common-criteria-policy** *policy-name* **masked-secret**
4. **end**

DETAILED STEPS

	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: • **username** *name***masked-secret**	• Defines a masked secret password, which is saved using a nonreversible encryption method.

	Command or Action	Purpose

	• **username** *name***common-criteria-policy** *policy-name* **masked-secret**
	Example:
	`Device(config)# username cisco masked-secret`
	or
	`Device(config)# username common-criteria-policy test-policy masked-secret`
		• Defines a masked secret password for common criteria policy.
		• The masked secret password must be greater than 4 characters. The maximum length of masked-secret password is 256 characters. By default, no password is defined.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Disabling Password Recovery

To disable password recovery to protect the security of your switch, follow this procedure.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **system disable password recovery switch** *<1-9>*
4. **end**

DETAILED STEPS

	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** *<1-9>*	Disables password recovery.

	Command or Action	Purpose
	Example: `Device(config)# system disable password recovery switch all`	• *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. To remove **disable password recovery**, use the **no system disable password recovery switch all** command in global configuration mode.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Setting a Telnet Password for a Terminal Line

To set a Telnet password for the connected terminal line, perform this procedure.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line vty 0 15**
4. **password** *password*
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	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	**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. *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:** `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.

Configuring Username and Password Pairs

To configure username and password pairs, perform this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **username** *name* [**privilege** *level*] {**password** *encryption-type password*}
4. Use one of the following:
   - **line console 0**
   - **line vty 0 15**
5. **login local**
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

DETAILED STEPS

	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	**username** *name* [**privilege** *level*] {**password** *encryption-type password*}
	Example:
	`Device(config)# username adamsample privilege 1 password secret456`
	`Device(config)# username 111111111111 mac attribute`
		Sets the username, privilege level, and password for each user.
		• *name*: Specify the user ID as one word or the MAC address. Spaces and quotation marks are not allowed.
		• You can configure a maximum of 12000 clients each, for both username and MAC filter.
		• *level*: (Optional) Specify the privilege level the user has after gaining access. The range is 0 to 15. Level 15 gives privileged EXEC mode access. Level 1 gives user EXEC mode access.
		• *encryption-type*: Enter 0 to specify that an unencrypted password will follow. Enter 7 to specify that a hidden password will follow.
		• *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:**	Returns to privileged EXEC mode.

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

To set the privilege level for a command, follow this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **privilege** *mode* **level** *level command*
4. **enable password level** *level password*
5. **end**
6. **copy running-config startup-config**

DETAILED STEPS

	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	**privilege** *mode* **level** *level command* **Example:** `Device(config)# privilege exec level 14 configure`	Sets the privilege level for a command. • *mode*: Enter **configure** for global configuration mode, **exec** for EXEC mode, **interface** for interface configuration mode, or **line** for line configuration mode. • *level*: 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. • *command*: Specify the command to which you want to restrict access.

	Command or Action	Purpose
Step 4	**enable password level** *level password* **Example:** `Device(config)# enable password level 14 SecretPswd14`	Specifies the password to enable the privilege level. • *level*: Range is from 0 to 15. Level 1 is for normal user EXEC mode privileges. • *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:** `Device(config)# end`	Returns to privileged EXEC mode.
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

Users can override the privilege level you set using the **privilege level** command by logging in to the line and enabling a different privilege level. 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.

To change the default privilege level for the specified line, perform this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line vty** *line*
4. **privilege level** *level*
5. **end**
6. **copy running-config startup-config**

DETAILED STEPS

	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	**line vty** *line* **Example:** `Device(config)# line vty 10`	Selects the vty on which to restrict access.
Step 4	**privilege level** *level* **Example:** `Device(config)# privilege level 15`	Changes the default privilege level for the line. *level*: 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.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Logging into and Exiting a Privilege Level

Users can lower the privilege level by using the **disable** command.

To log into a specified privilege level and exit a specified privilege level, perform this procedure.

SUMMARY STEPS

1. **enable** *level*
2. **disable** *level*

DETAILED STEPS

	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. *level*: 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. *level*: Range is 0 to 15.

Configuration Examples for Controlling Switch Access with Passwords and Privilege Levels

The following section provides configuration examples for controlling switch access with passwords and privilege levels.

Example: Setting or Changing a Static Enable Password

The following 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> enable
Device# configure terminal
Device(config)# enable password l1u2c3k4y5
```

Example: Protecting Enable and Enable Secret Passwords with Encryption

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

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

Example: Configuring Masked Secret Password

The following example shows how to configure the masked secret password:

```
Device> enable
Device# configure terminal
Device(config)# username cisco masked-secret
Enter secret: ******
Confirm secret: ******
```

The following example shows how to configure the masked secret password forfor common criteria policy:

```
Device> enable
Device# configure terminal
Device(config)# username cisco common-criteria-policy test-policy masked-secret
Enter secret: ******
Confirm secret: ******
```

Example: Setting a Telnet Password for a Terminal Line

The following example shows how to set the Telnet password to *let45me67in89*:

```
Device> enable
Device# configure terminal
Device(config)# line vty 10
Device(config-line)# password let45me67in89
```

Example: Setting the Privilege Level for a Command

The following 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> enable
Device# configure terminal
Device(config)# line vty 10
Device(config)# privilege exec level 14 configure
Device(config)# enable password level 14 SecretPswd14
```

Monitoring Switch Access

Table 2: Commands for Displaying DHCP Information

Command	Purpose
show privilege	Displays the privilege level configuration.
show running | **secret username**	Verifies that the username is created and encypted to type9 by default.
show running | **secret enable**	Verifies that the secret password is encypted to type9 by default.

Feature History for Controlling Switch Access with Passwords and Privilege Levels

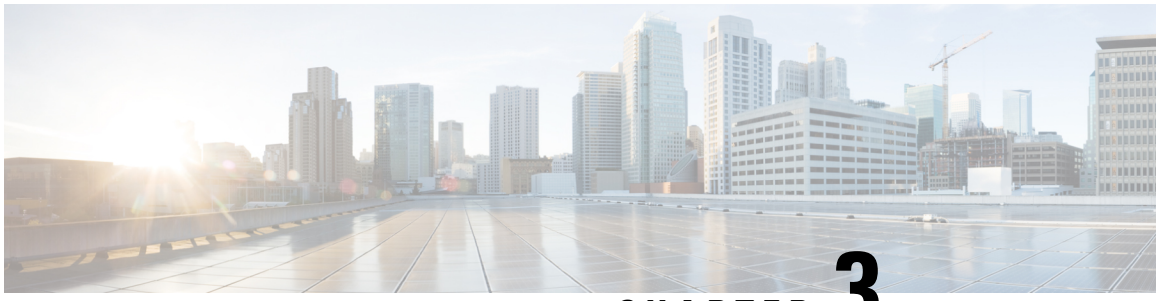
This table provides release and related information for features explained in this module.

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

Release	Feature	Feature Information
Cisco IOS Release 15.2(5)E	Controlling Switch Access with Passwords and 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.
Cisco IOS Release 15.2(7)E2	Type 6 Encryption	Type 6 encryption support for username and password has been introduced.

Release	Feature	Feature Information
Cisco IOS Release 15.2(7)E3	Autoconversion to Type 6	Autoconversion of type 0 and type 7 username and password to type 6 has been introduced.

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 3

Configuring TACACS+

TACACS+ is a security application that provides centralized validation of users attempting to gain access to a router or network access server. TACACS+ provides detailed accounting information and flexible administrative control over authentication and authorization processes. TACACS+ is facilitated through authentication, authorization and accounting (AAA) and can be enabled only through AAA commands.

Prerequisites for TACACS+

The following are the prerequisites for set up and configuration of device access with TACACS+ (must be performed in the order presented):

1. Configure the devices 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 device access with TACACS+:

- You must have access to a configured TACACS+ server to configure TACACS+ features on your device. 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 device.

- To use TACACS+, it must be enabled.
- Authorization must be enabled on the device 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+.

Restrictions for TACACS+

TACACS+ can be enabled only through AAA commands.

Information About 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.

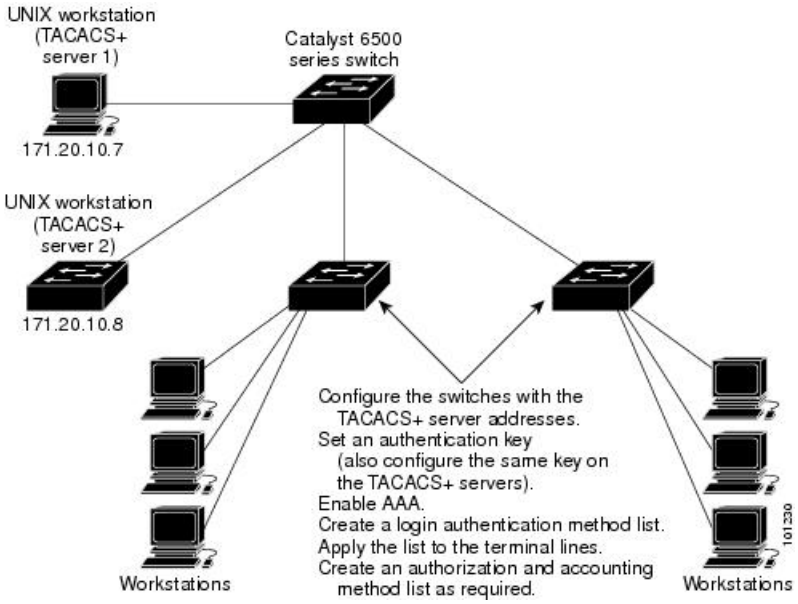
TACACS+ Overview

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 1: 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 device using TACACS+, this process occurs:

1. When the connection is established, the device contacts the TACACS+ daemon to obtain a username prompt to show to the user. The user enters a username, and the device then contacts the TACACS+

daemon to obtain a password prompt. The device 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 device eventually receives one of these responses from the TACACS+ daemon:
   - ACCEPT: The user is authenticated and service can begin. If the device 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 device. If an ERROR response is received, the device 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 device. 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 AV Pairs

The network access server implements TACACS+ authorization and accounting functions by transmitting and receiving TACACS+ attribute-value (AV) pairs for each user session.

TACACS Authentication and Authorization AV Pairs

The following table lists and describes the supported TACACS+ authentication and authorization AV pairs and specifies the Cisco IOS release in which they are implemented.

Table 3: Supported TACACS+ Authentication and Authorization AV Pairs

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
acl=x	ASCII number representing a connection access list. Used only when service=shell.	yes	yes	yes	yes	yes	yes	yes
addr=x	A network address. Used with service=slip, service=ppp, and protocol=ip. Contains the IP address that the remote host should use when connecting via SLIP or PPP/IP. For example, addr=10.2.3.4.	yes	yes	yes	yes	yes	yes	yes
addr-pool=x	Specifies the name of a local pool from which to get the address of the remote host. Used with service=ppp and protocol=ip. Note that **addr-pool** works in conjunction with local pooling. It specifies the name of a local pool (which must be preconfigured on the network access server). Use the **ip-local pool** command to declare local pools. For example: ip address-pool local ip local pool boo 10.0.0.1 10.0.0.10 ip local pool moo 10.0.0.1 10.0.0.20 You can then use TACACS+ to return addr-pool=boo or addr-pool=moo to indicate the address pool from which you want to get this remote node's address.	yes	yes	yes	yes	yes	yes	yes
autocmd=x	Specifies an autocommand to be executed at EXEC startup (for example, autocmd=telnet example.com). Used only with service=shell.	yes	yes	yes	yes	yes	yes	yes

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
callback- dialstring	Sets the telephone number for a callback (for example: callback-dialstring= 408-555-1212). Value is NULL, or a dial-string. A NULL value indicates that the service might choose to get the dial string through other means. Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	no	yes	yes	yes	yes	yes	yes
callback-line	The number of a TTY line to use for callback (for example: callback-line=4). Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	no	yes	yes	yes	yes	yes	yes
callback-rotary	The number of a rotary group (between 0 and 100 inclusive) to use for callback (for example: callback-rotary=34). Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	no	yes	yes	yes	yes	yes	yes
cmd-arg=x	An argument to a shell (EXEC) command. This indicates an argument for the shell command that is to be run. Multiple cmd-arg attributes can be specified, and they are order dependent. **Note** This TACACS+ AV pair cannot be used with RADIUS attribute 26.	yes	yes	yes	yes	yes	yes	yes
cmd=x	A shell (EXEC) command. This indicates the command name for a shell command that is to be run. This attribute must be specified if service equals “shell.” A NULL value indicates that the shell itself is being referred to. **Note** This TACACS+ AV pair cannot be used with RADIUS attribute 26.	yes	yes	yes	yes	yes	yes	yes
data-service	Used with the service=outbound and protocol=ip.	no	no	no	no	no	yes	yes
dial-number	Defines the number to dial. Used with the service=outbound and protocol=ip.	no	no	no	no	no	yes	yes
dns-servers=	Identifies a DNS server (primary or secondary) that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation. To be used with service=ppp and protocol=ip. The IP address identifying each DNS server is entered in dotted decimal format.	no	no	no	yes	yes	yes	yes
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. To turn on this attribute, use the “true” value (force-56=true). Any other value is treated as false. Used with the service=outbound and protocol=ip.	no	no	no	no	no	yes	yes
gw-password	Specifies the password for the home gateway during the L2F tunnel authentication. Used with service=ppp and protocol=vpdn.	no	no	yes	yes	yes	yes	yes

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
idletime=x	Sets a value, in minutes, after which an idle session is terminated. A value of zero indicates no timeout.	no	yes	yes	yes	yes	yes	yes
inacl#<n>	ASCII access list identifier for an input access list to be installed and applied to an interface for the duration of the current connection. Used with service=ppp and protocol=ip, and service service=ppp and protocol =ipx. Per-user access lists do not currently work with ISDN interfaces.	no	no	no	yes	yes	yes	yes
inacl=x	ASCII identifier for an interface input access list. Used with service=ppp and protocol=ip. Per-user access lists do not currently work with ISDN interfaces.	yes	yes	yes	yes	yes	yes	yes
interface-config#<n>	Specifies user-specific AAA interface configuration information with Virtual Profiles. The information that follows the equal sign (=) can be any Cisco IOS interface configuration command. Multiple instances of the attributes are allowed, but each instance must have a unique number. Used with service=ppp and protocol=lcp. **Note** This attribute replaces the “interface-config=” attribute.	no	no	no	yes	yes	yes	yes
ip-addresses	Space-separated list of possible IP addresses that can be used for the end-point of a tunnel. Used with service=ppp and protocol=vpdn.	no	no	yes	yes	yes	yes	yes
l2tp-busy-disconnect	If a vpdn-group on an LNS uses a virtual-template that is configured to be pre-cloned, this attribute will control the disposition of a new L2TP session that finds no pre-cloned interface to which to connect. If the attribute is true (the default), the session will be disconnected by the LNS. Otherwise, a new interface will be cloned from the virtual-template. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
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. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
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. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
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. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
l2tp-hidden-avp	When enabled, sensitive AVPs in L2TP control messages are scrambled or hidden. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
l2tp-nosession-timeout	Specifies the number of seconds that a tunnel will stay active with no sessions before timing out and shutting down. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
l2tp-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. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
l2tp-tunnel- authen	If this attribute is set, it performs L2TP tunnel authentication. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
l2tp-tunnel-password	Shared secret used for L2TP tunnel authentication and AVP hiding. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
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. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
link- compression=	Defines whether to turn on or turn off “stac” compression over a PPP link. Used with service=ppp. Link compression is defined as a numeric value as follows: • 0: None • 1: Stac • 2: Stac-Draft-9 • 3: MS-Stac	no	no	no	yes	yes	yes	yes
load-threshold= <n>	Sets the load threshold for the caller at which additional links are either added to or deleted from the multilink bundle. If the load goes above the specified value, additional links are added. If the load goes below the specified value, links are deleted. Used with service=ppp and protocol=multilink. The range for <n> is from 1 to 255.	no	no	no	yes	yes	yes	yes
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. Used with the service=outbound and protocol=ip.	no	no	no	no	no	yes	yes
max-links=<n>	Restricts the number of links that a user can have in a multilink bundle. Used with service=ppp and protocol=multilink. The range for <n> is from 1 to 255.	no	no	no	yes	yes	yes	yes
min-links	Sets the minimum number of links for MLP. Used with service=ppp and protocol=multilink, protocol=vpdn.	no	no	no	no	no	yes	yes
nas-password	Specifies the password for the network access server during the L2F tunnel authentication. Used with service=ppp and protocol=vpdn.	no	no	yes	yes	yes	yes	yes
nocallback-verify	Indicates that no callback verification is required. The only valid value for this parameter is 1 (for example, nocallback-verify=1). Used with service=arap, service=slip, service=ppp, service=shell. There is no authentication on callback. Not valid for ISDN.	no	yes	yes	yes	yes	yes	yes

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
noescape=x	Prevents user from using an escape character. Used with service=shell. Can be either true or false (for example, noescape=true).	yes	yes	yes	yes	yes	yes	yes
nohangup=x	Used with service=shell. Specifies the nohangup option, which means that after an EXEC shell is terminated, the user is presented with another login (username) prompt. Can be either true or false (for example, nohangup=false).	yes	yes	yes	yes	yes	yes	yes
old-prompts	Allows providers to make the prompts in TACACS+ appear identical to those of earlier systems (TACACS and Extended TACACS). This allows administrators to upgrade from TACACS or Extended TACACS to TACACS+ transparently to users.	yes	yes	yes	yes	yes	yes	yes
outacl#<n>	ASCII access list identifier for an interface output access list to be installed and applied to an interface for the duration of the current condition. Used with service=ppp and protocol=ip, and service service=ppp and protocol=ipx. Per-user access lists do not currently work with ISDN interfaces.	no	no	no	yes	yes	yes	yes
outacl=x	ASCII identifier for an interface output access list. Used with service=ppp and protocol=ip, and service service=ppp and protocol=ipx. Contains an IP output access list for SLIP or PPP/IP (for example, outacl=4). The access list itself must be preconfigured on the router. Per-user access lists do not currently work with ISDN interfaces.	yes (PPP/IP only)	yes	yes	yes	yes	yes	yes
pool-def#<n>	Defines IP address pools on the network access server. Used with service=ppp and protocol=ip.	no	no	no	yes	yes	yes	yes
pool-timeout=	Defines (in conjunction with pool-def) IP address pools on the network access server. During IPCP address negotiation, if an IP pool name is specified for a user (see the addr-pool attribute), a check is made to see if the named pool is defined on the network access server. If it is, the pool is consulted for an IP address. Used with service=ppp and protocol=ip.	no	no	yes	yes	yes	yes	yes

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
port-type	Indicates the type of physical port the network access server is using to authenticate the user. Physical ports are indicated by a numeric value as follows: • 0: Asynchronous • 1: Synchronous • 2: ISDN-Synchronous • 3: ISDN-Asynchronous (V.120) • 4: ISDN- Asynchronous (V.110) • 5: Virtual Used with service=any and protocol=aaa.	no	no	no	no	no	yes	yes
ppp-vj-slot-compression	Instructs the Cisco router not to use slot compression when sending VJ-compressed packets over a PPP link.	no	no	no	yes	yes	yes	yes
priv-lvl=x	Privilege level to be assigned for the EXEC. Used with service=shell. Privilege levels range from 0 to 15, with 15 being the highest.	yes	yes	yes	yes	yes	yes	yes
protocol=x	A protocol that is a subset of a service. An example would be any PPP NCP. Currently known values are **lcp**, **ip**, **ipx**, **atalk**, **vines**, **lat**, **xremote**, **tn3270**, **telnet**, **rlogin**, **pad**, **vpdn**, **osicp**, **deccp**, **ccp**, **cdp**, **bridging**, **xns**, **nbf**, **bap**, **multilink**, and **unknown**.	yes	yes	yes	yes	yes	yes	yes
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. Used with the service=shell and protocol=exec.	no	no	no	no	no	yes	yes
route	Specifies a route to be applied to an interface. Used with service=slip, service=ppp, and protocol=ip. During network authorization, the route attribute can be used to specify a per-user static route, to be installed by TACACS+ as follows: route=”*dst_address mask* [*gateway*]” This indicates a temporary static route that is to be applied. The *dst_address*, *mask*, and *gateway* are expected to be in the usual dotted-decimal notation, with the same meanings as in the familiar **ip route** configuration command on a network access server. If *gateway* is omitted, the peer’s address is the gateway. The route is expunged when the connection terminates.	no	yes	yes	yes	yes	yes	yes

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
route#<n>	Like the route AV pair, this specifies a route to be applied to an interface, but these routes are numbered, allowing multiple routes to be applied. Used with service=ppp and protocol=ip, and service=ppp and protocol=ipx.	no	no	no	yes	yes	yes	yes
routing=x	Specifies whether routing information is to be propagated to and accepted from this interface. Used with service=slip, service=ppp, and protocol=ip. Equivalent in function to the /routing flag in SLIP and PPP commands. Can either be true or false (for example, routing=true).	yes	yes	yes	yes	yes	yes	yes
rte-fltr-in#<n>	Specifies an input access list definition to be installed and applied to routing updates on the current interface for the duration of the current connection. Used with service=ppp and protocol=ip, and with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes	yes
rte-fltr-out#<n>	Specifies an output access list definition to be installed and applied to routing updates on the current interface for the duration of the current connection. Used with service=ppp and protocol=ip, and with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes	yes
sap#<n>	Specifies static Service Advertising Protocol (SAP) entries to be installed for the duration of a connection. Used with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes	yes
sap-fltr-in#<n>	Specifies an input SAP filter access list definition to be installed and applied on the current interface for the duration of the current connection. Used with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes	yes
sap-fltr-out#<n>	Specifies an output SAP filter access list definition to be installed and applied on the current interface for the duration of the current connection. Used with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes	yes
send-auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication. Used with service=any and protocol=aaa.	no	no	no	no	no	yes	yes
send-secret	Specifies the password that the NAS needs to respond to a chap/pap request from the remote end of a connection on an outgoing call. Used with service=ppp and protocol=ip.	no	no	no	no	no	yes	yes
service=x	The primary service. Specifying a service attribute indicates that this is a request for authorization or accounting of that service. Current values are **slip**, **ppp**, **arap**, **shell**, **tty-daemon**, **connection**, and **system**. This attribute must always be included.	yes	yes	yes	yes	yes	yes	yes
source-ip=x	Used as the source IP address of all VPDN packets generated as part of a VPDN tunnel. This is equivalent to the Cisco **vpdn outgoing** global configuration command.	no	no	yes	yes	yes	yes	yes

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
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. Used with the service=mobileip and protocol=ip.	no	no	no	no	no	yes	yes
timeout=x	The number of minutes before an EXEC or ARA session disconnects (for example, timeout=60). A value of zero indicates no timeout. Used with service=arap.	yes	yes	yes	yes	yes	yes	yes
tunnel-id	Specifies the username that will be used to authenticate the tunnel over which the individual user MID will be projected. This is analogous to the *remote name* in the **vpdn outgoing** command. Used with service=ppp and protocol=vpdn.	no	no	yes	yes	yes	yes	yes
wins-servers=	Identifies a Windows NT server that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation. To be used with service=ppp and protocol=ip. The IP address identifying each Windows NT server is entered in dotted decimal format.	no	no	no	yes	yes	yes	yes
zonelist=x	A numeric zonelist value. Used with service=arap. Specifies an AppleTalk zonelist for ARA (for example, zonelist=5).	yes	yes	yes	yes	yes	yes	yes

See Configuring TACACS+. module for the documents used to configure TACACS+, and TACACS+ authentication and authorization.

TACACS Accounting AV Pairs

The following table lists and describes the supported TACACS+ accounting AV pairs and specifies the Cisco IOS release in which they are implemented.

Table 4: Supported TACACS+ Accounting AV Pairs

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
Abort-Cause	If the fax session gets cancelled, indicates the system component that signaled the cancellation. Examples of system components that could trigger a cancellation are FAP (Fax Application Process), TIFF (the TIFF reader or the TIFF writer), fax-mail client, fax-mail server, ESMTP client, or ESMTP server.	no	no	no	no	no	yes	yes
bytes_in	The number of input bytes transferred during this connection.	yes	yes	yes	yes	yes	yes	yes
bytes_out	The number of output bytes transferred during this connection.	yes	yes	yes	yes	yes	yes	yes
Call-Type	Describes the type of fax activity: fax receive or fax send.	no	no	no	no	no	yes	yes

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
cmd	The command the user executed.	yes	yes	yes	yes	yes	yes	yes
data-rate	This AV pair has been renamed. See nas-rx-speed.							
disc-cause	Specifies the reason a connection was taken off-line. The Disconnect-Cause attribute is sent in accounting-stop records. This attribute also causes stop records to be generated without first generating start records if disconnection occurs before authentication is performed. Refer to the following table (Disconnect Cause Extensions) for a list of Disconnect-Cause values and their meanings.	no	no	no	yes	yes	yes	yes
disc-cause-ext	Extends the disc-cause attribute to support vendor-specific reasons why a connection was taken off-line.	no	no	no	yes	yes	yes	yes
elapsed_time	The elapsed time in seconds for the action. Useful when the device does not keep real time.	yes	yes	yes	yes	yes	yes	yes
Email-Server-Address	Indicates the IP address of the e-mail server handling the on-ramp fax-mail message.	no	no	no	no	no	yes	yes
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.	no	no	no	no	no	yes	yes
event	Information included in the accounting packet that describes a state change in the router. Events described are accounting starting and accounting stopping.	yes	yes	yes	yes	yes	yes	yes
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** command.	no	no	no	no	no	yes	yes
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.	no	no	no	no	no	yes	yes
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.	no	no	no	no	no	yes	yes
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.	no	no	no	no	no	yes	yes
Fax-Dsn-Address	Indicates the address to which DSNs will be sent.	no	no	no	no	no	yes	yes
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.	no	no	no	no	no	yes	yes
Fax-Mdn-Address	Indicates the address to which MDNs will be sent.	no	no	no	no	no	yes	yes
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.	no	no	no	no	no	yes	yes

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
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.	no	no	no	no	no	yes	yes
Fax-Msg-Id=	Indicates a unique fax message identification number assigned by Store and Forward Fax.	no	no	no	no	no	yes	yes
Fax-Pages	Indicates the number of pages transmitted or received during this fax session. This page count includes cover pages.	no	no	no	no	no	yes	yes
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.	no	no	no	no	no	yes	yes
Fax-Recipient-Count	Indicates the number of recipients for this fax transmission. Until e-mail servers support Session mode, the number should be 1.	no	no	no	no	no	yes	yes
Gateway-Id	Indicates the name of the gateway that processed the fax session. The name appears in the following format: hostname.domain-name	no	no	no	no	no	yes	yes
mlp-links-max	Gives the count of links which are known to have been in a given multilink session at the time the accounting record is generated.	no	no	no	yes	yes	yes	yes
mlp-sess-id	Reports the identification number of the multilink bundle when the session closes. This attribute applies to sessions that are part of a multilink bundle. This attribute is sent in authentication-response packets.	no	no	no	yes	yes	yes	yes
nas-rx-speed	Specifies the average number of bits per second over the course of the connection’s lifetime. This attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes	yes
nas-tx-speed	Reports the transmit speed negotiated by the two modems.	no	no	no	yes	yes	yes	yes
paks_in	The number of input packets transferred during this connection.	yes	yes	yes	yes	yes	yes	yes
paks_out	The number of output packets transferred during this connection.	yes	yes	yes	yes	yes	yes	yes
port	The port the user was logged in to.	yes	yes	yes	yes	yes	yes	yes
Port-Used	Indicates the slot/port number of the Cisco AS5300 used to either transmit or receive this fax-mail.	no	no	no	no	no	yes	yes
pre-bytes-in	Records the number of input bytes before authentication. This attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes	yes
pre-bytes-out	Records the number of output bytes before authentication. This attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes	yes
pre-paks-in	Records the number of input packets before authentication. This attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes	yes
pre-paks-out	Records the number of output packets before authentication. The Pre-Output-Packets attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes	yes

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
pre-session-time	Specifies the length of time, in seconds, from when a call first connects to when it completes authentication.	no	no	no	yes	yes	yes	yes
priv_level	The privilege level associated with the action.	yes	yes	yes	yes	yes	yes	yes
protocol	The protocol associated with the action.	yes	yes	yes	yes	yes	yes	yes
reason	Information included in the accounting packet that describes the event that caused a system change. Events described are system reload, system shutdown, or when accounting is reconfigured (turned on or off).	yes	yes	yes	yes	yes	yes	yes
service	The service the user used.	yes	yes	yes	yes	yes	yes	yes
start_time	The time the action started (in seconds since the epoch, 12:00 a.m. Jan 1 1970). The clock must be configured to receive this information.	yes	yes	yes	yes	yes	yes	yes
stop_time	The time the action stopped (in seconds since the epoch.) The clock must be configured to receive this information.	yes	yes	yes	yes	yes	yes	yes
task_id	Start and stop records for the same event must have matching (unique) task_id numbers.	yes	yes	yes	yes	yes	yes	yes
timezone	The time zone abbreviation for all timestamps included in this packet.	yes	yes	yes	yes	yes	yes	yes
xmit-rate	This AV pair has been renamed. See nas-tx-speed.							

The following table lists the cause codes and descriptions for the Disconnect Cause Extended (disc-cause-ext) attribute.

Table 5: Disconnect Cause Extensions

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1000 - No Reason	No reason for the disconnect.	no	no	no	no	yes	yes	yes	yes
1001 - No Disconnect	The event was not a disconnect.	no	no	no	no	yes	yes	yes	yes
1002 - Unknown	The reason for the disconnect is unknown. This code can appear when the remote connection goes down.	no	no	no	no	yes	yes	yes	yes
1003 - Call Disconnect	The call has disconnected.	no	no	no	no	yes	yes	yes	yes
1004 - CLID Auth Fail	Calling line ID (CLID) authentication has failed.	no	no	no	no	yes	yes	yes	yes
1009 - No Modem Available	The modem is not available.	no	no	no	no	yes	yes	yes	yes
1010 - No Carrier	The modem never detected data carrier detect (DCD). This code can appear if a disconnect occurs during the initial modem connection.	no	no	no	no	yes	yes	yes	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1011 - Lost Carrier	The modem detected DCD but became inactive. This code can appear if a disconnect occurs during the initial modem connection.	no	no	no	no	yes	yes	yes	yes
1012 - No Modem Results	The result codes could not be parsed. This code can appear if a disconnect occurs during the initial modem connection.	no	no	no	no	yes	yes	yes	yes
1020 - TS User Exit	The user exited normally from the terminal server. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1021 - Idle Timeout	The user exited from the terminal server because the idle timer expired. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1022 - TS Exit Telnet	The user exited normally from a Telnet session. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1023 - TS No IP Addr	The user could not switch to Serial Line Internet Protocol (SLIP) or PPP because the remote host had no IP address or because the dynamic pool could not assign one. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1024 - TS TCP Raw Exit	The user exited normally from a raw TCP session. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1025 - TS Bad Password	The login process ended because the user failed to enter a correct password after three attempts. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1026 - TS No TCP Raw	The raw TCP option is not enabled. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1027 - TS CNTL-C	The login process ended because the user typed Ctrl-C. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1028 - TS Session End	The terminal server session has ended. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1029 - TS Close Vconn	The user closed the virtual connection. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1030 - TS End Vconn	The virtual connection has ended. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1031 - TS Rlogin Exit	The user exited normally from an Rlogin session. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1032 - TS Rlogin Opt Invalid	The user selected an invalid Rlogin option. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1033 - TS Insuff Resources	The access server has insufficient resources for the terminal server session. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1040 - PPP LCP Timeout	PPP link control protocol (LCP) negotiation timed out while waiting for a response from a peer. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1041 - PPP LCP Fail	There was a failure to converge on PPP LCP negotiations. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1042 - PPP Pap Fail	PPP Password Authentication Protocol (PAP) authentication failed. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1043 - PPP CHAP Fail	PPP Challenge Handshake Authentication Protocol (CHAP) authentication failed. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1044 - PPP Remote Fail	Authentication failed from the remote server. This code concerns PPP sessions.	no	no	no	no	yes	yes	yes	yes
1045 - PPP Receive Term	The peer sent a PPP termination request. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
PPP LCP Close (1046)	LCP got a close request from the upper layer while LCP was in an open state. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1047 - PPP No NCP	LCP closed because no NCPs were open. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1048 - PPP MP Error	LCP closed because it could not determine to which Multilink PPP bundle that it should add the user. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1049 - PPP Max Channels	LCP closed because the access server could not add any more channels to an MP session. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1050 - TS Tables Full	The raw TCP or Telnet internal session tables are full. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	no	no	no	no	yes	yes	yes	yes
1051 - TS Resource Full	Internal resources are full. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	no	no	no	no	yes	yes	yes	yes
1052 - TS Invalid IP Addr	The IP address for the Telnet host is invalid. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	no	no	no	no	yes	yes	yes	yes
1053 - TS Bad Hostname	The access server could not resolve the host name. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	no	no	no	no	yes	yes	yes	yes
1054 - TS Bad Port	The access server detected a bad or missing port number. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	no	no	no	no	yes	yes	yes	yes
1060 - TCP Reset	The host reset the TCP connection. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1061 - TCP Connection Refused	The host refused the TCP connection. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1062 - TCP Timeout	The TCP connection timed out. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1063 - TCP Foreign Host Close	A foreign host closed the TCP connection. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1064 - TCP Net Unreachable	The TCP network was unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1065 - TCP Host Unreachable	The TCP host was unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1066 - TCP Net Admin Unreachable	The TCP network was administratively unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1067 - TCP Host Admin Unreachable	The TCP host was administratively unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1068 - TCP Port Unreachable	The TCP port was unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1100 - Session Timeout	The session timed out because there was no activity on a PPP link. This code applies to all session types.	no	no	no	no	yes	yes	yes	yes
1101 - Security Fail	The session failed for security reasons. This code applies to all session types.	no	no	no	no	yes	yes	yes	yes
1102 - Callback	The session ended for callback. This code applies to all session types.	no	no	no	no	yes	yes	yes	yes
1120 - Unsupported	One end refused the call because the protocol was disabled or unsupported. This code applies to all session types.	no	no	no	no	yes	yes	yes	yes
1150 - Radius Disc	The RADIUS server requested the disconnect.	no	no	no	no	yes	yes	yes	yes
1151 - Local Admin Disc	The local administrator has disconnected.	no	no	no	no	yes	yes	yes	yes
1152 - SNMP Disc	Simple Network Management Protocol (SNMP) has disconnected.	no	no	no	no	yes	yes	yes	yes
1160 - V110 Retries	The allowed retries for V110 synchronization have been exceeded.	no	no	no	no	yes	yes	yes	yes
1170 - PPP Auth Timeout	Authentication timeout. This code applies to PPP sessions.	no	no	no	no	yes	yes	yes	yes
1180 - Local Hangup	The call disconnected as the result of a local hangup.	no	no	no	no	yes	yes	yes	yes
1185 - Remote Hangup	The call disconnected because the remote end hung up.	no	no	no	no	yes	yes	yes	yes
1190 - T1 Quiesced	The call disconnected because the T1 line that carried it was quiesced.	no	no	no	no	yes	yes	yes	yes
1195 - Call Duration	The call disconnected because the call duration exceeded the maximum amount of time allowed by the Max Call Mins or Max DS0 Mins parameter on the access server.	no	no	no	no	yes	yes	yes	yes
1600 - VPDN User Disconnect	The user disconnected. This value applies to virtual private dial-up network (VPDN) sessions.	no	no	no	no	no	no	yes	yes
1601 - VPDN Carrier Loss	Carrier loss has occurred. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1602 - VPDN No Resources	There are no resources. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1603 - VPDN Bad Control Packet	The control packet is invalid. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1604 - VPDN Admin Disconnect	The administrator disconnected. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1605 - VPDN Tunnel Down/Setup Fail	The tunnel is down or the setup failed. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1606 - VPDN Local PPP Disconnect	There was a local PPP disconnect. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1607 - VPDN Softshut/Session Limit	New sessions cannot be established on the VPN tunnel. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1608 - VPDN Call Redirected	The call was redirected. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1801 - Q850 Unassigned Number	The number has not been assigned. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1802 - Q850 No Route	The equipment that is sending this code has received a request to route the call through a particular transit network that it does not recognize. The equipment that is sending this code does not recognize the transit network because either the transit network does not exist or because that particular transit network, while it does exist, does not serve the equipment that is sending this code. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1803 - Q850 No Route To Destination	The called party cannot be reached because the network through which the call has been routed does not serve the destination that is desired. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1806 - Q850 Channel Unacceptable	The channel that has been most recently identified is not acceptable to the sending entity for use in this call. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1816 - Q850 Normal Clearing	The call is being cleared because one of the users who is involved in the call has requested that the call be cleared. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1817 - Q850 User Busy	The called party is unable to accept another call because the user-busy condition has been encountered. This code may be generated by the called user or by the network. In the case of the user, the user equipment is compatible with the call. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1818 - Q850 No User Responding	Used when a called party does not respond to a call-establishment message with either an alerting or connect indication within the prescribed period of time that was allocated. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1819 - Q850 No User Answer	The called party has been alerted but does not respond with a connect indication within a prescribed period of time. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1821 - Q850 Call Rejected	The equipment that is sending this code does not wish to accept this call although it could have accepted the call because the equipment that is sending this code is neither busy nor incompatible. This code may also be generated by the network, indicating that the call was cleared due to a supplementary service constraint. The diagnostic field may contain additional information about the supplementary service and reason for rejection. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1822 - Q850 Number Changed	The number that is indicated for the called party is no longer assigned. The new called party number may optionally be included in the diagnostic field. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1827 - Q850 Destination Out of Order	The destination that was indicated by the user cannot be reached because the interface to the destination is not functioning correctly. The term “not functioning correctly” indicates that a signaling message was unable to be delivered to the remote party. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1828 - Q850 Invalid Number Format	The called party cannot be reached because the called party number is not in a valid format or is not complete. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1829 - Q850 Facility Rejected	This code is returned when a supplementary service that was requested by the user cannot be provided by the network. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1830 - Q850 Responding to Status Enquiry	This code is included in the STATUS message when the reason for generating the STATUS message was the prior receipt of a STATUS ENQUIRY message. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1831 - Q850 Unspecified Cause	No other code applies. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1834 - Q850 No Circuit Available	No circuit or channel is available to handle the call. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1838 - Q850 Network Out of Order	The network is not functioning correctly and the condition is likely to last a relatively long period of time. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1841 - Q850 Temporary Failure	The network is not functioning correctly and the condition is not likely to last a long period of time. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1842 - Q850 Network Congestion	The network is congested. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1843 - Q850 Access Info Discarded	This code indicates that the network could not deliver access information to the remote user as requested. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1844 - Q850 Requested Channel Not Available	This code is returned when the circuit or channel that is indicated by the requesting entity cannot be provided by the other side of the interface. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1845 - Q850 Call Pre-empted	The call was preempted. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1847 - Q850 Resource Unavailable	This code is used to report a resource-unavailable event only when no other code in the resource-unavailable class applies. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1850 - Q850 Facility Not Subscribed	Not a subscribed facility. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1852 - Q850 Outgoing Call Barred	Although the calling party is a member of the closed user group for the outgoing closed user group call, outgoing calls are not allowed for this member. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
Q850 Incoming Call Barred (1854)	Although the called party is a member of the closed user group for the incoming closed user group call, incoming calls are not allowed to this member. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1858 - Q850 Bearer Capability Not Available	The user has requested a bearer capability that is implemented by the equipment that generated this code but that is not available at this time. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1863 - Q850 Service Not Available	The code is used to report a service- or option-not-available event only when no other code in the service- or option-not-available class applies. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1865 - Q850 Bearer Capability Not Implemented	The equipment that is sending this code does not support the bearer capability that was requested. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1866 - Q850 Channel Not Implemented	The equipment that is sending this code does not support the channel type that was requested. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1869 - Q850 Facility Not Implemented	The supplementary service requested by the user cannot be provided by the network. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1881 - Q850 Invalid Call Reference	The equipment that is sending this code has received a message having a call reference that is not currently in use on the user-network interface. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1882 - Q850 Channel Does Not Exist	The channel most recently identified is not acceptable to the sending entity for use in this call. This code applies to ISDN or modem calls that have come in over ISDN. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1888 - Q850 Incompatible Destination	The equipment that is sending this code has received a request to establish a call that has low-layer compatibility or other compatibility attributes that cannot be accommodated. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1896 - Q850 Mandatory Info Element Is Missing	The equipment that is sending this code has received a message that is missing an information element that must be present in the message before that message can be processed. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1897 - Q850 Non Existent Message Type	The equipment that is sending this code has received a message with a message type that it does not recognize either because this is a message that is not defined or that is defined but not implemented by the equipment that is sending this code. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1898 - Q850 Invalid Message	This code is used to report an invalid message when no other code in the invalid message class applies. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1899 - Q850 Bad Info Element	The information element not recognized. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1900 - Q850 Invalid Element Contents	The equipment that is sending this code has received an information element that it has implemented; however, one or more fields in the information element are coded in such a way that has not been implemented by the equipment that is sending this code. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1901 - Q850 Wrong Message for State	The message that was received is incompatible with the call state. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1902 - Q850 Recovery on Timer Expiration	A procedure has been initiated by the expiration of a timer in association with error-handling procedures. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1903 - Q850 Info Element Error	The equipment that is sending this code has received a message that includes information elements or parameters that are not recognized because the information element identifiers or paramenter names are not defined or are defined but not implemented by the equipment that is sending this code. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1911 - Q850 Protocol Error	This code is used to report a protocol error event only when no other code in the protocol error class applies. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1927 - Q850 Unspecified Internetworking Event	There has been an error when interworking with a network that does not provide codes for actions that it takes. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes

TACACS+ Configuration Options

You can configure the device 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 device 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+ Authentication

After you have identified the TACACS+ daemon and defined an associated TACACS+ encryption key, you must define method lists for TACACS+ authentication. Because TACACS+ authentication is operated via AAA, you need to issue the **aaa authentication** command, specifying TACACS+ as the authentication method.

TACACS+ Authorization

AAA authorization enables you to set parameters that restrict a user's access to the network. Authorization via TACACS+ may be applied to commands, network connections, and EXEC sessions. Because TACACS+ authorization is facilitated through AAA, you must issue the **aaa authorization** command, specifying TACACS+ as the authorization method.

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 device 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 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:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa group server tacacs+** *group-name*
5. **server** *ip-address*
6. **end**

DETAILED STEPS

	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 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. This command puts the device in a server group subconfiguration mode.
Step 5	**server** *ip-address* **Example:** `Device(config)# server 10.1.2.3`	(Optional) Associates a particular TACACS+ server with the defined server group. Repeat this step for each TACACS+ server in the AAA server group.
Step 6	**end** **Example:** `Device(config)# end`	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 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** {**default** | *list-name*} *method1* [*method2...*]
5. **line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*]
6. **login authentication** {**default** | *list-name*}
7. **end**

DETAILED STEPS

	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:**	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,

	Command or Action	Purpose
	`Device(config)# aaa authentication login default tacacs+ local`	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. • *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.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-line)# end`	Returns to privileged EXEC mode.

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:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization network** *authorization-list* **tacacs+**
4. **aaa authorization exec default tacacs+**
5. **end**

DETAILED STEPS

	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** *authorization-list* **tacacs+** **Example:** `Device(config)# aaa authorization network list1`	Configures the device for user TACACS+ authorization for all network-related service requests.

	Command or Action	Purpose
	`tacacs+`	
Step 4	**aaa authorization exec default tacacs+** **Example:** `Device(config)# aaa authorization exec default tacacs+`	Configures the device for user TACACS+ authorization if the user has privileged EXEC access. The **exec** keyword might return user profile information (such as **autocommand** information).
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Starting TACACS+ Accounting

Follow these steps to start TACACS+ Accounting:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting network** *authorization-list* **start-stop tacacs+**
4. **aaa accounting exec default start-stop tacacs+**
5. **end**

DETAILED STEPS

	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** *authorization-list* **start-stop tacacs+** **Example:** `Device(config)# aaa accounting network list1`	Enables TACACS+ accounting for all network-related service requests.

	Command or Action	Purpose
	`start-stop tacacs+`	
Step 4	**aaa accounting exec default start-stop tacacs+** **Example:** `Device(config)# aaa accounting exec default 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.

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

The **aaa accounting system guarantee-first** command 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.

Configuring Per VRF on a TACACS Server

The initial steps in this procedure are used to configure AAA and a server group, create a VRF routing table, and configure an interface. Steps 10 through 13 are used to configure the per VRF on a TACACS+ server feature:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **exit**
6. **interface** *interface-name*

7. **ip vrf forwarding** *vrf-name*
8. **ip address** *ip-address mask* [**secondary**]
9. **exit**
10. **aaa group server tacacs+** *group-name*
11. **server-private** {*ip-address* | *name*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*]
12. **ip vrf forwarding** *vrf-name*
13. **ip tacacs source-interface** *subinterface-name*
14. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `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 vrf** *vrf-name* **Example:** `Device(config)# ip vrf cisco`	Configures a VRF table and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF instance.
Step 5	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode.
Step 6	**interface** *interface-name* **Example:** `Device(config)# interface Loopback0`	Configures an interface and enters interface configuration mode.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding cisco`	Configures a VRF for the interface.

	Command or Action	Purpose
Step 8	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.2 255.0.0.0`	Sets a primary or secondary IP address for an interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 10	**aaa group server tacacs+** *group-name* **Example:** `Device(config)# aaa group server tacacs+ tacacs1`	Groups different TACACS+ server hosts into distinct lists and distinct methods and enters server-group configuration mode.
Step 11	**server-private** {*ip-address* | *name*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*] **Example:** `Device(config-sg-tacacs+)# server-private 10.1.1.1 port 19 key cisco`	Configures the IP address of the private TACACS+ server for the group server.
Step 12	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-sg-tacacs+)# ip vrf forwarding cisco`	Configures the VRF reference of a AAA TACACS+ server group.
Step 13	**ip tacacs source-interface** *subinterface-name* **Example:** `Device(config-sg-tacacs+)# ip tacacs source-interface Loopback0`	Uses the IP address of a specified interface for all outgoing TACACS+ packets.
Step 14	**exit** **Example:** `Device(config-sg-tacacs)# exit`	Exits server-group configuration mode, and enters global configuration mode.

Monitoring TACACS+

Table 6: Commands for Displaying TACACS+ Information

Command	Purpose
show tacacs	Displays TACACS+ server statistics.

Configuration Examples for TACACS+

Example: TACACS Authorization

The following example shows how to configure TACACS+ as the security protocol for PPP authentication using the default method list; it also shows how to configure network authorization via TACACS+:

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
aaa authorization network default group tacacs+
tacacs server server1
address IPv4 10.1.2.3
key goaway
exit
interface serial 0
 ppp authentication chap default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "default," to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **aaa authorization** command configures network authorization via TACACS+. Unlike authentication lists, this authorization list always applies to all incoming network connections made to the network access server.
- The **tacacs server** command identifies the TACACS+ daemon, and **address ipv4** command as having an IP address of 10.1.2.3. The **key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

Example: TACACS Accounting

The following example shows how to configure TACACS+ as the security protocol for PPP authentication using the default method list; it also shows how to configure accounting via TACACS+:

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
aaa accounting network default stop-only group tacacs+
tacacs server server1
address IPv4 10.1.2.3
key goaway
exit
interface serial 0
```

```
 ppp authentication chap default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "default," to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **aaa accounting** command configures network accounting via TACACS+. In this example, accounting records describing the session that just terminated will be sent to the TACACS+ daemon whenever a network connection terminates.
- The **tacacs server** command identifies the TACACS+ daemon, and **address ipv4** command as having an IP address of 10.1.2.3. The **key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

Example: TACACS Authentication

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

```
aaa new-model
aaa authentication ppp test group tacacs+ local
tacacs server server1
address IPv4 10.1.2.3
key goaway
exit
interface serial 0
 ppp authentication chap pap test
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "test," to be used on serial interfaces running PPP. The keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **tacacs server** command identifies the TACACS+ daemon, and **address ipv4** command as having an IP address of 10.1.2.3. The **key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the test method list to this line.

The following example shows how to configure TACACS+ as the security protocol for PPP authentication, but instead of the "test" method list, the "default" method list is used.

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
tacacs server server1
address IPv4 10.1.2.3
key goaway
exit
interface serial 0
 ppp authentication chap default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "default," to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **tacacs server** command identifies the TACACS+ daemon, and **address ipv4** command as having an IP address of 10.1.2.3. The **key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

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

```
aaa new-model
aaa authentication pap MIS-access if-needed group tacacs+ local
tacacs server server1
address IPv4 10.1.2.3
key goaway
exit
interface serial 0
 ppp authentication pap MIS-access
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "MIS-access," to be used on serial interfaces running PPP. The method list, "MIS-access," means that PPP authentication is applied to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **tacacs server** command identifies the TACACS+ daemon, and **address ipv4** command as having an IP address of 10.1.2.3. The **key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

The following example shows the configuration for a TACACS+ daemon with an IP address of 10.2.3.4 and an encryption key of “apple”:

```
aaa new-model
aaa authentication login default group tacacs+ local
tacacs server server1
address IPv4 10.2.3.4
key apple
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines the default method list. Incoming ASCII logins on all interfaces (by default) will use TACACS+ for authentication. If no TACACS+ server responds, then the network access server will use the information contained in the local username database for authentication.
- The **tacacs server** command identifies the TACACS+ daemon, and **address ipv4** command as having an IP address of 10.2.3.4. The **key** command defines the shared encryption key to be “apple.”

Example: Configuring Per VRF for TACACS Servers

The following output example shows that the group server **tacacs1** is configured for per VRF AAA services:

```
aaa group server tacacs+ tacacs1
    server-private 10.1.1.1 port 19 key cisco
    ip vrf forwarding cisco
    ip tacacs source-interface Loopback0
  ip vrf cisco
   rd 100:1
  interface Loopback0
   ip address 10.0.0.2 255.0.0.0
   ip vrf forwarding cisco
```

Additional References for TACACS+

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

MIBs

MB	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

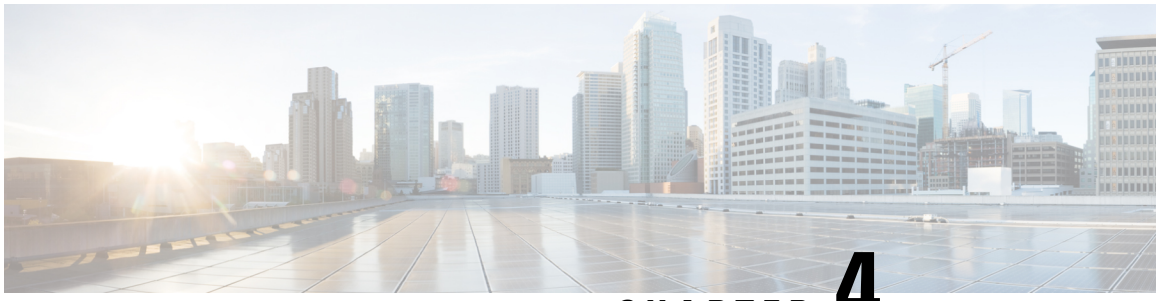
Feature History for TACACS+

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 Release 15.2(5)E	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.



CHAPTER 4

Configuring RADIUS

The RADIUS security system is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco devices and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

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

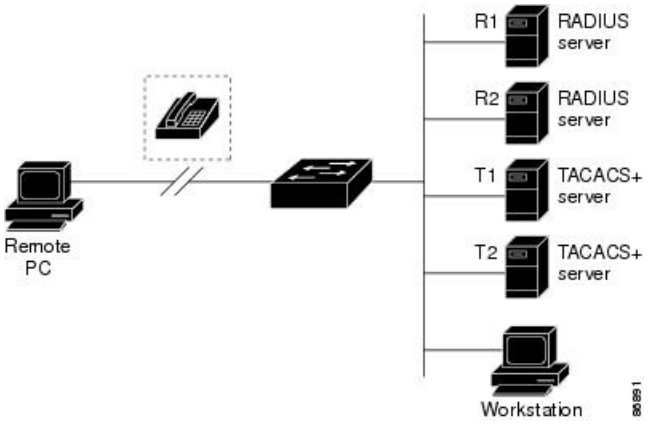
RADIUS is a distributed client/server system that secures networks against unauthorized access. RADIUS clients run on supported Cisco routers and switches. Clients send authentication requests to a central RADIUS server, which contains all user authentication and network service access information.

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 2: 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

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 device 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 device 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 device 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″
```

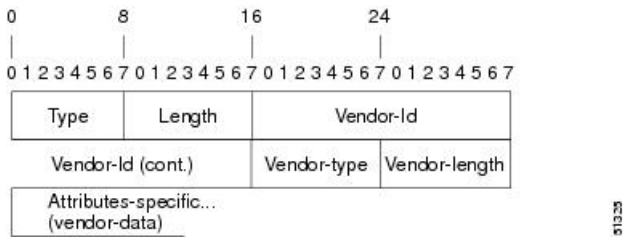
Other vendors have their own unique vendor-IDs, options, and associated VSAs. For more information about vendor-IDs and VSAs, see RFC 2138, "Remote Authentication Dial-In User Service (RADIUS)."

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 3: 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 7: 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 8: Vendor-Specific RADIUS IETF Attributes

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
MS-CHAP Attributes				
26	311	1	MSCHAP-Response	Contains the response value provided by a PPP MS-CHAP user in response to the challenge. It is only used in Access-Request packets. This attribute is identical to the PPP CHAP Identifier. (RFC 2548
26	311	11	MSCHAP-Challenge	Contains the challenge sent by a network access server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets. (RFC 2548)
VPDN Attributes				
26	9	1	l2tp-cm-local-window-size	Specifies the maximum receive window size for L2TP control messages. This value is advertised to the peer during tunnel establishment.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	l2tp-drop-out-of-order	Respects sequence numbers on data packets by dropping those that are received out of order. This does not ensure that sequence numbers will be sent on data packets, just how to handle them if they are received.
26	9	1	l2tp-hello-interval	Specifies the number of seconds for the hello keepalive interval. Hello packets are sent when no data has been sent on a tunnel for the number of seconds configured here.
26	9	1	l2tp-hidden-avp	When enabled, sensitive AVPs in L2TP control messages are scrambled or hidden.
26	9	1	l2tp-nosession-timeout	Specifies the number of seconds that a tunnel will stay active with no sessions before timing out and shutting down.
26	9	1	tunnel-tos-reflect	Copies the IP ToS field from the IP header of each payload packet to the IP header of the tunnel packet for packets entering the tunnel at the LNS.
26	9	1	l2tp-tunnel-authen	If this attribute is set, it performs L2TP tunnel authentication.
26	9	1	l2tp-tunnel-password	Shared secret used for L2TP tunnel authentication and AVP hiding.
26	9	1	l2tp-udp-checksum	This is an authorization attribute and defines whether L2TP should perform UDP checksums for data packets. Valid values are “yes” and “no.” The default is no.
Store and Forward Fax Attributes				
26	9	3	Fax-Account-Id-Origin	Indicates the account ID origin as defined by system administrator for the **mmoip aaa receive-id** or the **mmoip aaa send-id** commands.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	4	Fax-Msg-Id=	Indicates a unique fax message identification number assigned by Store and Forward Fax.
26	9	5	Fax-Pages	Indicates the number of pages transmitted or received during this fax session. This page count includes cover pages.
26	9	6	Fax-Coverpage-Flag	Indicates whether or not a cover page was generated by the off-ramp gateway for this fax session. True indicates that a cover page was generated; false means that a cover page was not generated.
26	9	7	Fax-Modem-Time	Indicates the amount of time in seconds the modem sent fax data (x) and the amount of time in seconds of the total fax session (y), which includes both fax-mail and PSTN time, in the form x/y. For example, 10/15 means that the transfer time took 10 seconds, and the total fax session took 15 seconds.
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.
26	9	20	Port-Used	Indicates the slot/port number of the Cisco AS5300 used to either transmit or receive this fax-mail.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	21	Abort-Cause	If the fax session gets cancelled, indicates the system component that signaled the cancellation. Examples of system components that could trigger a cancellation 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.
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.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
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.
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.

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
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				
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.

RADIUS Disconnect-Cause Attribute Values

Disconnect-cause attribute values specify the reason a connection was taken offline. The attribute values are sent in Accounting request packets. These values are sent at the end of a session, even if the session fails to be authenticated. If the session is not authenticated, the attribute can cause stop records to be generated without first generating start records.

The table below lists the cause codes, values, and descriptions for the Disconnect-Cause (195) attribute.

Note The Disconnect-Cause is incremented by 1000 when it is used in RADIUS AVPairs; for example, disc-cause 4 becomes 1004.

Table 9: Disconnect-Cause Attribute Values

Cause Code	Value	Description
0	No-Reason	No reason is given for the disconnect.
1	No-Disconnect	The event was not disconnected.
2	Unknown	Reason unknown.
3	Call-Disconnect	The call has been disconnected.
4	CLID-Authentication-Failure	Failure to authenticate number of the calling-party.
9	No-Modem-Available	A modem in not available to connect the call.

Cause Code	Value	Description
10	No-Carrier	No carrier detected. **Note** Codes 10, 11, and 12 can be sent if there is a disconnection during initial modem connection.
11	Lost-Carrier	Loss of carrier.
12	No-Detected-Result-Codes	Failure to detect modem result codes.
20	User-Ends-Session	User terminates a session. **Note** Codes 20, 22, 23, 24, 25, 26, 27, and 28 apply to EXEC sessions.
21	Idle-Timeout	Timeout waiting for user input. Codes 21, 100, 101, 102, and 120 apply to all session types.
22	Exit-Telnet-Session	Disconnect due to exiting Telnet session.
23	No-Remote-IP-Addr	Could not switch to SLIP/PPP; the remote end has no IP address.
24	Exit-Raw-TCP	Disconnect due to exiting raw TCP.
25	Password-Fail	Bad passwords.
26	Raw-TCP-Disabled	Raw TCP disabled.
27	Control-C-Detected	Control-C detected.
28	EXEC-Process-Destroyed	EXEC process destroyed.
29	Close-Virtual-Connection	User closes a virtual connection.
30	End-Virtual-Connection	Virtual connected has ended.
31	Exit-Rlogin	User exists Rlogin.
32	Invalid-Rlogin-Option	Invalid Rlogin option selected.
33	Insufficient-Resources	Insufficient resources.
40	Timeout-PPP-LCP	PPP LCP negotiation timed out. **Note** Codes 40 through 49 apply to PPP sessions.
41	Failed-PPP-LCP-Negotiation	PPP LCP negotiation failed.
42	Failed-PPP-PAP-Auth-Fail	PPP PAP authentication failed.
43	Failed-PPP-CHAP-Auth	PPP CHAP authentication failed.
44	Failed-PPP-Remote-Auth	PPP remote authentication failed.
45	PPP-Remote-Terminate	PPP received a Terminate Request from remote end.

Cause Code	Value	Description
46	PPP-Closed-Event	Upper layer requested that the session be closed.
47	NCP-Closed-PPP	PPP session closed because there were no NCPs open.
48	MP-Error-PPP	PPP session closed because of an MP error.
49	PPP-Maximum-Channels	PPP session closed because maximum channels were reached.
50	Tables-Full	Disconnect due to full terminal server tables.
51	Resources-Full	Disconnect due to full internal resources.
52	Invalid-IP-Address	IP address is not valid for Telnet host.
53	Bad-Hostname	Hostname cannot be validated.
54	Bad-Port	Port number is invalid or missing.
60	Reset-TCP	TCP connection has been reset. **Note** Codes 60 through 67 apply to Telnet or raw TCP sessions.
61	TCP-Connection-Refused	TCP connection has been refused by the host.
62	Timeout-TCP	TCP connection has timed out.
63	Foreign-Host-Close-TCP	TCP connection has been closed.
64	TCP-Network-Unreachable	TCP network is unreachable.
65	TCP-Host-Unreachable	TCP host is unreachable.
66	TCP-Network-Admin Unreachable	TCP network is unreachable for administrative reasons.
67	TCP-Port-Unreachable	TCP port in unreachable.
100	Session-Timeout	Session timed out.
101	Session-Failed-Security	Session failed for security reasons.
102	Session-End-Callback	Session terminated due to callback.
120	Invalid-Protocol	Call refused because the detected protocol is disabled.
150	RADIUS-Disconnect	Disconnected by RADIUS request.
151	Local-Admin-Disconnect	Administrative disconnect.
152	SNMP-Disconnect	Disconnected by SNMP request.
160	V110-Retries	Allowed V.110 retries have been exceeded.
170	PPP-Authentication-Timeout	PPP authentication timed out.
180	Local-Hangup	Disconnected by local hangup.

Cause Code	Value	Description
185	Remote-Hangup	Disconnected by remote end hangup.
190	T1-Quiesced	Disconnected because T1 line was quiesced.
195	Call-Duration	Disconnected because the maximum duration of the call was exceeded.
600	VPN-User-Disconnect	Call disconnected by client (through PPP). Code is sent if the LNS receives a PPP terminate request from the client.
601	VPN-Carrier-Loss	Loss of carrier. This can be the result of a physical line going dead. Code is sent when a client is unable to dial out using a dialer.
602	VPN-No-Resources	No resources available to handle the call. Code is sent when the client is unable to allocate memory (running low on memory).
603	VPN-Bad-Control-Packet	Bad L2TP or L2F control packets. This code is sent when an invalid control packet, such as missing mandatory Attribute-Value pairs (AVP), from the peer is received. When using L2TP, the code will be sent after six retransmits; when using L2F, the number of retransmits is user configurable. **Note** VPN-Tunnel-Shut will be sent if there are active sessions in the tunnel.
604	VPN-Admin-Disconnect	Administrative disconnect. This can be the result of a VPN soft shutdown, which is when a client reaches maximum session limit or exceeds maximum hopcount. Code is sent when a tunnel is brought down by issuing the **clear vpdn tunnel** command.
605	VPN-Tunnel-Shut	Tunnel teardown or tunnel setup has failed. Code is sent when there are active sessions in a tunnel and the tunnel goes down. **Note** This code is not sent when tunnel authentication fails.
606	VPN-Local-Disconnect	Call is disconnected by LNS PPP module. Code is sent when the LNS sends a PPP terminate request to the client. It indicates a normal PPP disconnection initiated by the LNS.
607	VPN-Session-Limit	VPN soft shutdown is enabled. Code is sent when a call has been refused due to any of the soft shutdown restrictions previously mentioned.
608	VPN-Call-Redirect	VPN call redirect is enabled.

RADIUS Progress Codes

The RADIUS Progress Codes feature adds additional progress codes to RADIUS attribute 196 (Ascend-Connect-Progress), which indicates a connection state before a call is disconnected through progress codes.

Attribute 196 is sent in network, exec, and resource accounting *start* and *stop* records. This attribute can facilitate call failure debugging because each progress code identifies accounting information relevant to the connection state of a call. The attribute is activated by default; when an accounting *start* or *stop* accounting record is requested, authentication, authorization, and accounting (AAA) adds attribute 196 into the record as part of the standard attribute list. Attribute 196 is valuable because the progress codes, which are sent in accounting *start* and *stop* records, facilitate the debugging of call failures.

Note In accounting *start* records, attribute 196 does not have a value.

Table 10: Newly Supported Progress Codes for Attribute 196

Code	Description
10	Modem allocation and negotiation is complete; the call is up.
30	The modem is up.
33	The modem is waiting for result codes.
41	The max TNT is establishing the TCP connection by setting up a TCP clear call.
60	Link control protocol (LCP) is the open state with PPP and IP Control Protocol (IPCP) negotiation; the LAN session is up.
65	PPP negotiation occurs and, initially, the LCP negotiation occurs; LCP is in the open state.
67	After PPP negotiation with LCP in the open state occurs, IPCP negotiation begins.

Note Progress codes 33, 30, and 67 are generated and seen through debugs on the NAS; all other codes are generated and seen through debugs and the accounting record on the RADIUS server.

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.

Enhanced Test Command

The Enhanced Test Command feature allows a named user profile to be created with calling line ID (CLID) or dialed number identification service (DNIS) attribute values. The CLID or DNIS attribute values can be associated with the RADIUS record that is sent with the user profile so that the RADIUS server can access CLID or DNIS attribute information for all incoming calls.

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 key**, and **radius-server retransmit**.

You can configure the device to use AAA server groups to group existing server hosts for authentication. For more information, see Related Topics below.

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. For more information, see the RADIUS server documentation.

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. For information on configuring these settings on all RADIUS servers, see Related Topics below.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius server** *server name*
4. **address** {**ipv4** | **ipv6**} *ip address* { **auth-port** *port number* | **acct-port** *port number*}
5. **key** *string*
6. **end**

DETAILED STEPS

	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	**radius server** *server name* **Example:** `Device(config)# radius server server1`	Specifies the name for the RADIUS server configuration, 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 172.2.2.12 auth-port 1612`	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 key1`	Specifies the RADIUS server parameters. **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.
Step 6	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring Settings for All RADIUS Servers

Beginning in privileged EXEC mode, follow these steps to configure settings for all RADIUS servers:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server key** *string*
4. **radius-server retransmit** *retries*
5. **radius-server timeout** *seconds*

6. **radius-server deadtime** *minutes*
7. **end**

DETAILED STEPS

	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 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 4	**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 5	**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 6	**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 7	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** {**default** | *list-name*} *method1* [*method2*...]
5. **line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*]
6. **login authentication** {**default** | *list-name*}
7. **end**

DETAILED STEPS

	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*...]	Creates a login authentication method list.

	Command or Action	Purpose
	Example: `Device(config)# aaa authentication login default local`	• 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 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:**	Applies the authentication list to a line or set of lines.

	Command or Action	Purpose
	`Device(config)# login authentication default`	• 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.

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:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius server** *name*
4. **address** {**ipv4** | **ipv6**} {*ip-address* | *hostname*} **auth-port** *port-number* **acct-port** *port-number*
5. **key** [**0** | **6** | **7**] *string*
6. **end**

DETAILED STEPS

	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	**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** [**0** | **6** | **7**] *string* **Example:** `Device(config-radius-server)# key 0 cisco123`	Specifies the authentication and encryption key for all RADIUS communications between the device and the RADIUS server. **Note** • Key 0, 6, and 7 indicates cleartext password, type 6 encryption, and type 7 encryption respectively. If the key is configured as type 7 then a valid type 7 encrypted string should also be configured. Similarly, a valid type 6 encrypted string should follow key type 6. • The text string 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.
Step 6	**end** **Example:** `Device(config-radius-server)# end`	Exits RADIUS server configuration mode and returns to privileged EXEC mode.

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:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization network radius**
4. **aaa authorization exec radius**
5. **end**

DETAILED STEPS

	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).
Step 5	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

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:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting network start-stop radius**
4. **aaa accounting exec start-stop radius**
5. **end**

DETAILED STEPS

	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 radius** **Example:** `Device(config)# aaa accounting network start-stop radius`	Enables RADIUS accounting for all network-related service requests.

	Command or Action	Purpose
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.

Verifying Attribute 196

No configuration is required to configure RADIUS Progress Codes. To verify attribute 196 in accounting *start* and *stop* records, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug aaa accounting**
3. **show radius statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug aaa accounting** **Example:** `Device# debug aaa accounting`	Displays information on accountable events as they occur.
Step 3	**show radius statistics** **Example:** `Device# debug aaa authorization`	Displays the RADIUS statistics for accounting and authentication packets.

Configuring the Device to Use Vendor-Specific RADIUS Attributes

Follow these steps to configure the device to use vendor-specific RADIUS attributes:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server vsa send** [**accounting** | **authentication**]
4. **end**

DETAILED STEPS

	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.

Configuring the Device for Vendor-Proprietary RADIUS Server Communication

Follow these steps to configure the device to use vendor-proprietary RADIUS server communication:

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **radius server** *server name*
4. **address** {**ipv4** | **ipv6**} *ip address*
5. **non-standard**
6. **key** [**0** | **6** | **7**] *string*
7. **end**

DETAILED STEPS

	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 server1`	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 172.2.2.12`	(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** [**0**	**6**	**7**] *string* **Example:** `Device(config-radius-server)# key 0 cisco123`	Specifies the shared secret type and string used between the device and the vendor-proprietary RADIUS server. The device and the RADIUS server use this to encrypt passwords and exchange responses.

	Command or Action	Purpose
		Note • Key 0, 6, and 7 indicates cleartext password, type 6 encryption, and type 7 encryption respectively. If the key is configured as type 7 then a valid type 7 encrypted string should also be configured. Similarly, a valid type 6 encrypted string should follow key type 6. • The text string 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.
Step 7	**end** **Example:** `Device(config-radius-server)# end`	Returns to privileged EXEC mode.

Configuring a User Profile and Associating it with the RADIUS Record

This section describes how to create a named user profile with CLID or DNIS attribute values and associate it with the RADIUS record.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa user profile** *profile-name*
4. **aaa attribute** {**dnis** | **clid**}
5. **exit**
6. **test aaa group** {*group-name* | **radius**} *username password* **new-code** [**profile** *profile-name*]

DETAILED STEPS

	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	**aaa user profile** *profile-name* **Example:** `Device(config)# aaa user profile profilename1`	Creates a user profile.
Step 4	**aaa attribute** {**dnis** | **clid**} **Example:** `Device(config)# aaa attribute dnis`	Adds DNIS or CLID attribute values to the user profile and enters AAA-user configuration mode.
Step 5	**exit** **Example:** `Device(config)# exit`	Exit Global Configuration mode.
Step 6	**test aaa group** {*group-name* | **radius**} *username password* **new-code** [**profile** *profile-name*] **Example:** `Device# test aaa group radius secret new-code profile profilename1`	Associates a DNIS or CLID named user profile with the record sent to the RADIUS server. **Note** The *profile-name* must match the profile-name specified in the **aaa user profile** command.

Verifying the Enhanced Test Command Configuration

To verify the Enhanced Test Command configuration, use the following commands in privileged EXEC mode:

Command	Purpose
`Device# debug radius`	Displays information associated with RADIUS.
`Device# more system:running-config`	Displays the contents of the current running configuration file. (Note that the **more system:running-config** command has replaced the **show running-config** command.)

Configuration Examples for RADIUS

Example: Identifying the RADIUS Server Host

This example shows how to configure one RADIUS server to be used for authentication and another to be used for accounting:

```
Device# configure terminal
Device(config)# radius server server1
Device(config-radius-server)# address ipv4 172.2.2.12 auth-port 1612
Device(config-radius-server)# key key1
Device(config-radius-server)# exit
Device(config)# radius server server2
Device(config-radius-server)# address ipv4 172.2.2.20 auth-port 1618
Device(config-radius-server)# key key2
Device(config-radius-server)# exit
```

Example: AAA Server Groups

The following example shows how to create server group radgroup1 with three different RADIUS server members, each using the default authentication port (1645) and accounting port (1646):

```
aaa group server radius radgroup1
 server 172.16.1.11
 server 172.17.1.21
 server 172.18.1.31
```

The following example shows how to create server group radgroup2 with three RADIUS server members, each with the same IP address but with unique authentication and accounting ports:

```
aaa group server radius radgroup2
 server 172.16.1.1 auth-port 1000 acct-port 1001
 server 172.16.1.1 auth-port 2000 acct-port 2001
 server 172.16.1.1 auth-port 3000 acct-port 3001
```

Troubleshooting Tips for RADIUS Progress Codes

The following example is a sample debug output from the **debug ppp negotiation** command. This debug output is used to verify that accounting *stop* records have been generated and that attribute 196 (Ascend-Connect-Progress) has a value of 65.

```
Tue Aug 7 06:21:03 2001
        NAS-IP-Address = 10.0.58.62
        NAS-Port = 20018
        Vendor-Specific = ""
        NAS-Port-Type = ISDN
        User-Name = "peer_16a"
        Called-Station-Id = "5213124"
        Calling-Station-Id = "5212175"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
```

```
Service-Type = Framed-User
Acct-Session-Id = "00000014"
Framed-Protocol = PPP
Framed-IP-Address = 172.16.0.2
Acct-Input-Octets = 3180
Acct-Output-Octets = 3186
Acct-Input-Packets = 40
Acct-Output-Packets = 40
Ascend-Connect-Pr = 65
Acct-Session-Time = 49
Acct-Delay-Time = 0
Timestamp = 997190463
Request-Authenticator = Unverified
```

Example: Configuring the Device to Use Vendor-Specific RADIUS Attributes

For example, this AV pair activates Cisco *multiple named ip address pools* feature during IP authorization (during PPP IPCP address assignment):

```
cisco-avpair= "ip:addr-pool=first"
```

This example shows how to provide a user logging in from a device with immediate access to privileged EXEC commands:

```
cisco-avpair= "shell:priv-lvl=15"
```

This example shows how to specify an authorized VLAN in the RADIUS server database:

```
cisco-avpair= "tunnel-type(#64)=VLAN(13)"
cisco-avpair= "tunnel-medium-type(#65)=802 media(6)"
cisco-avpair= "tunnel-private-group-id(#81)=vlanid"
```

This example shows how to apply an input ACL in ASCII format to an interface for the duration of this connection:

```
cisco-avpair= "ip:inacl#1=deny ip 10.10.10.10 0.0.255.255 20.20.20.20 255.255.0.0"
cisco-avpair= "ip:inacl#2=deny ip 10.10.10.10 0.0.255.255 any"
cisco-avpair= "mac:inacl#3=deny any any decnet-iv"
```

This example shows how to apply an output ACL in ASCII format to an interface for the duration of this connection:

```
cisco-avpair= "ip:outacl#2=deny ip 10.10.10.10 0.0.255.255 any"
```

Example: Configuring the Device for Vendor-Proprietary RADIUS Server Communication

This example shows how to specify a vendor-proprietary RADIUS host and to use a secret key of *rad124* between the device and the server:

```
Device# configure terminal
Device(config)# radius server server1
Device(config-radius-server)# address ipv4 172.2.2.12
Device(config-radius-server)# nonstandard
Device(config-radius-server)# key rad124
Device(config-radius-server)# exit
```

Example: User Profile Associated With the test aaa group Command

The following example shows how to configure the dnis = dnisvalue user profile *prfl1* and associate it with a **test aaa group** command. In this example, the **debug radius** command has been enabled and the output follows the configuration.

```
aaa user profile prfl1
  aaa attribute dnis
  aaa attribute dnis dnisvalue
  no aaa attribute clid
! Attribute not found.
  aaa attribute clid clidvalue
  no aaa attribute clid
  exit
!
! Associate the dnis user profile with the test aaa group command.
test aaa group radius user1 pass new-code profile prfl1
!
!
!
! debug radius output, which shows that the dnis value has been passed to the radius !
server.
*Dec 31 16:35:48: RADIUS: Sending packet for Unique id = 0
 *Dec 31 16:35:48: RADIUS: Initial Transmit unknown id 8 172.22.71.21:1645, Access-Request,
 len 68
 *Dec 31 16:35:48: RADIUS: code=Access-Request id=08 len=0068
         authenticator=1E CA 13 F2 E2 81 57 4C - 02 EA AF 9D 30 D9 97 90
         T=User-Password[2]                 L=12 V=*
         T=User-Name[1]                     L=07 V="test"
         T=Called-Station-Id[30]            L=0B V="dnisvalue"
         T=Service-Type[6]                  L=06 V=Login                    [1]
         T=NAS-IP-Address[4]                L=06 V=10.0.1.81

 *Dec 31 16:35:48: RADIUS: Received from id 8 172.22.71.21:1645, Access-Accept, len 38
 *Dec 31 16:35:48: RADIUS: code=Access-Accept id=08 len=0038
```

Additional References for RADIUS

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

Standards and RFCs

Standard/RFC	Title
RFC 5176	RADIUS Change of Authorization (CoA) extensions

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

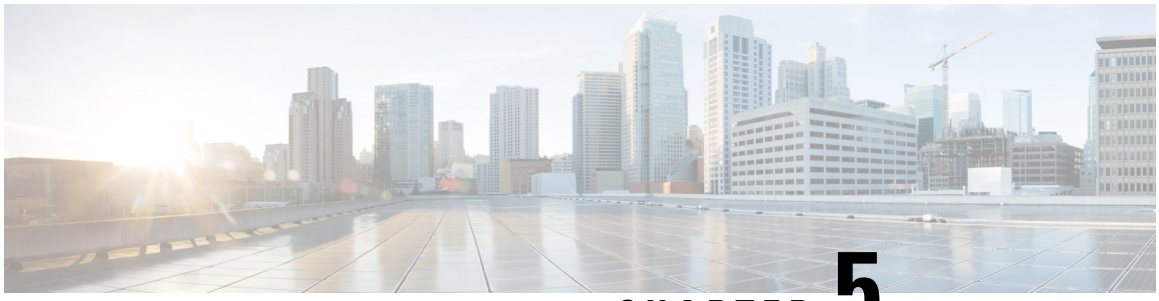
Feature History for RADIUS

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 Release 15.2(5)E	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.



CHAPTER 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-model**command 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.

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.

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

Accounting method lists must be applied to specific lines or interfaces before any of the defined methods are performed. The only exception is the default method list (which is named "default"). If the **aaa accounting** command for a particular accounting type is issued without 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 LDAP, 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, a subset of the configured server hosts can be specified and use them for a particular service. For example, server groups allows R1 and R2 to be defined as separate server groups, and T1 and T2 as separate server groups. This allows either R1 and T1 to be specified in the method list or R2 and T2 in the method list, which provides more flexibility in the way that RADIUS and TACACS+ resources are assigned.

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

AAA Accounting Methods

The Cisco IOS software supports the following two methods for accounting:

- TACACS+: The network access server reports user activity to the TACACS+ security server in the form of accounting records. Each accounting record contains accounting 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.

AAA Accounting Types

This section describes the different 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     updatetask_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 IOS 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:

VRRS Accounting

Virtual Router Redundancy Service (VRRS) provides a multiclient information abstraction and management service between a First Hop Redundancy Protocol (FHRP) and a registered client. The VRRS multiclient service provides a consistent interface with FHRP protocols by abstracting over several FHRPs and providing an idealized view of their state. VRRS manages data updates, allowing interested clients to register in one place and receive updates for named FHRP groups or all registered FHRP groups.

VRRS Accounting Plug-in

The VRRS Accounting plug-in provides a configurable AAA method list mechanism that provides updates to a RADIUS server when a VRRS group transitions its state. The VRRS accounting plug-in is an extension of existing AAA system accounting messages. The VRRS Accounting plug-in provides accounting-on and accounting-off messages and an additional Vendor-Specific Attribute (VSA) that sends the configured VRRS name in RADIUS accounting messages. The VRRS name is configured using the **vrrp name** command in interface configuration mode.

The VRRS Accounting plug-in provides a configurable AAA method list mechanism that provides updates to a RADIUS server when a VRRS group transitions its state.

The VRRS accounting plug-in is an extension of existing AAA system accounting messages. The VRRS Accounting plug-in provides accounting-on and accounting-off messages and an additional Vendor-Specific Attribute (VSA) that sends the configured VRRS name in RADIUS accounting messages. The VRRS name is configured using the **vrrp name** command in interface configuration mode. The VRRS Accounting plug-in sends an accounting-on message to RADIUS when a VRRS group transitions to the primary state, and it sends an accounting-off message when a VRRS group transitions from the primary state.

The following RADIUS attributes are included in VRRS accounting messages by default:

- Attribute 4, NAS-IP-Address
- Attribute 26, Cisco VSA Type 1, VRRS Name
- Attribute 40, Acct-Status-Type
- Attribute 41, Acct-Delay-Time
- Attribute 44, Acct-Session-Id

Accounting messages for a VRRS transitioning out of primary state are sent after all PPPoE accounting stop messages for sessions that are part of that VRRS.

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 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.

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 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [*method1* [*method2*...]]
4. Do one of the following:
   - **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
   - **interface** *interface-type interface-number*
5. Do one of the following:
   - **accounting** {**arap** | **commands** *level* | **connection** | **exec**} {**default** | *list-name*}
   - **ppp accounting**{**default** | *list-name*}
6. **end**

DETAILED STEPS

	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** {**system**	**network**	**exec**	**connection**	**commands** *level*} {**default**	*list-name*} {**start-stop**	**stop-only**	**none**} [*method1* [*method2*...]] **Example:**	Creates an accounting method list and enables accounting. The argument *list-name* is a character string used to name the created list.

	Command or Action	Purpose
	`Device(config)# aaa accounting system default start-stop`	
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.

Configuring RADIUS System Accounting

Perform this task to configure RADIUS system accounting on the global RADIUS server:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **radius-server accounting system host-config**
5. **aaa group server radius** *server-name*
6. **server-private** {*host-name* | *ip-address*} **key** {[**0** *server-key* | **7** *server-key*] *server-key*
7. **accounting system host-config**
8. **end**

DETAILED STEPS

	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	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA network security services.
Step 4	**radius-server accounting system host-config** **Example:** `Device(config)# radius-server accounting system host-config`	Enables the device to send a system accounting record for the addition and deletion of a RADIUS server.
Step 5	**aaa group server radius** *server-name* **Example:** `Device(config)# aaa group server radius radgroup1`	Adds the RADIUS server and enters server-group configuration mode. • The *server-name* argument specifies the RADIUS server group name.
Step 6	**server-private** {*host-name* | *ip-address*} **key** {[**0** *server-key* | **7** *server-key*] *server-key* **Example:** `Device(config-sg-radius)# server-private 172.16.1.11 key cisco`	Enters the hostname or IP address of the RADIUS server and hidden server key. • (Optional) **0** with the *server-key*argument specifies that an unencrypted (cleartext) hidden server key follows. • (Optional) **7** with the *server-key* argument specifies that an encrypted hidden server key follows. • The *server-key* argument specifies the hidden server key. If the *server-key*argument is configured without the **0** or **7** preceding it, it is unencrypted. **Note** Once the **server-private** command is configured, RADIUS system accounting is enabled.
Step 7	**accounting system host-config** **Example:** `Device(config-sg-radius)# accounting system host-config`	Enables the generation of system accounting records for private server hosts when they are added or deleted.
Step 8	**end** **Example:**	Exits server-group configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-sg-radius)# end`	

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

When used with the keyword **periodic**, interim accounting records are sent periodically as defined by the *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.

Generating Accounting Records for Failed Login or Session

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

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

Command	Purpose
`Device(config)# aaa accounting send stop-record authentication failure`	Generates "stop" records for users who fail to authenticate at login or during session negotiation using PPP.
`Device(config)# aaa accounting send stop-record always`	Sends AAA stop records regardless of whether a start record was sent earlier.

Specifying Accounting NETWORK-Stop Records Before EXEC-Stop Records

For PPP users who start EXEC terminal sessions, you can specify the NETWORK records to be generated before EXEC-stop records. In cases such as billing customers for specific services, it can be desirable to keep network start and stop records together, essentially "nesting" them within the framework of the EXEC start and stop messages. For example, a user dialing in using PPP can create the following records: EXEC-start, NETWORK-start, EXEC-stop, NETWORK-stop. By nesting the accounting records, NETWORK-stop records follow NETWORK-start messages: EXEC-start, NETWORK-start, NETWORK-stop, EXEC-stop.

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

Command	Purpose
`Device(config)# aaa accounting nested`	Nests network accounting records.

Configuring AAA Resource Failure Stop Accounting

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

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.

Configuring AAA Broadcast Accounting

To configure AAA broadcast accounting, use the **aaa accounting** command in global configuration mode:

Command	
`Device(config)# aaa accounting {system | network | exec | connection | commands level} {default | list-name} {start-stop | stop-only | none} [broadcast] method1 [method2...]`	

Configuring Per-DNIS AAA Broadcast Accounting

To configure AAA broadcast accounting per DNIS, use the **aaa dnis map accounting network**command 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.

Configuring AAA Session MIB

The following tasks must be performed before configuring the AAA session MIB feature:

- Configure SNMP.
- Configure AAA.
- Define the RADIUS or TACACS+ server characteristics.

Note Overusing SNMP can affect the overall system performance; therefore, normal network management performance must be considered when this feature is used.

To configure AAA session MIB, use the following command in global configuration mode

SUMMARY STEPS

1. Device (config)# **aaa session-mib disconnect**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Device (config)# **aaa session-mib disconnect**	Monitors and terminates authenticated client connections using SNMP. To terminate the call, the **disconnect** keyword must be used.

Configuring VRRS Accounting

Perform the following task to configure Virtual Router Redundancy Service (VRRS) to send AAA Accounting messages to the AAA server:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting vrrs** {**default** | *list-name*} **start-stop** *method1* [*method2...*]
4. **aaa attribute list** *list-name*

5. **attribute type** *name value* [**service** *service*] [**protocol** *protocol*][**mandatory**][**tag** *tag-value*]
6. **exit**
7. **vrrs** *vrrs-group-name*
8. **accounting delay** *seconds*
9. **accounting method** {**default** | *accounting-method-list*}
10. **end**

DETAILED STEPS

	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 vrrs** {**default** | *list-name*} **start-stop** *method1* [*method2...*] **Example:** `Device(config)# aaa accounting vrrs default start-stop`	Enables AAA accounting for VRRS.
Step 4	**aaa attribute list** *list-name* **Example:** `Device(config)# aaa attribute list list1`	Defines a AAA attribute list locally on a device, and enters attribute list configuration mode.
Step 5	**attribute type** *name value* [**service** *service*] [**protocol** *protocol*][**mandatory**][**tag** *tag-value*] **Example:** `Device(config-attr-list)# attribute type example 1`	Defines an attribute type that is to be added to an attribute list locally on a device.
Step 6	**exit** **Example:** `Device(config-attr-list)# exit`	Exits attribute list configuration mode and returns to global configuration mode.
Step 7	**vrrs** *vrrs-group-name* **Example:** `Device(config)# vrrs vrrs1`	(Optional) Defines a VRRP group and configures parameters for the VRRS group, and enters VRRS configuration mode.

	Command or Action	Purpose
Step 8	**accounting delay** *seconds* **Example:** `Device(config-vrrs)# accounting delay 10`	(Optional) Specifies the delay time for sending accounting-off messages to the VRRS.
Step 9	**accounting method** {**default** | *accounting-method-list*} **Example:** `Device(config-vrrs)# accounting method default`	(Optional) Enables VRRS accounting for a VRRP group.
Step 10	**end** **Example:** `Device(config-vrrs)# end`	Exits VRRS configuration mode and returns to privileged EXEC mode.

Establishing a Session with a Device if the AAA Server is Unreachable

To establish a console or telnet session with a device if the AAA server is unreachable, use the following command in global configuration mode:

Command	Purpose
`Device(config)# no aaa accounting system guarantee-first`	Guarantees system accounting as the first record, which is the default condition. In some situations, users may be prevented from starting a session on the console or terminal connection until after the system reloads, which can take more than three minutes. To resolve this problem, the **no aaa accounting system guarantee-first** command can be used.

Note Entering the **no aaa accounting system guarantee-first** command is not the only condition by which the console or telnet session can be started. For example, if the privileged EXEC session is being authenticated by TACACS and the TACACS server is not reachable, then the session cannot start.

Monitoring Accounting

No specific **show** command exists for either RADIUS or TACACS+ accounting. To obtain accounting records displaying information about users currently logged in, use the following command in privileged EXEC mode:

Command	Purpose
`Device# show accounting`	Allows display of the active accountable events on the network and helps collect information in the event of a data loss on the accounting server.

Troubleshooting Accounting

To troubleshoot accounting information, use the following command in privileged EXEC mode:

Command	Purpose
`Device# debug aaa accounting`	Displays information on accountable events as they occur.

Configuration Examples for Accounting

Example: Configuring Named Method List

The following example shows how to configure a 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> enable
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 blue1 group radius local
Device(config)# aaa accounting network red1 start-stop group radius group tacacs+
Device(config)# username root password ALongPassword
Device(config)# tacacs server server1
Device(config-tacacs-server)# address IPv4 172.31.255.0
Device(config-tacacs-server)# key goaway
Device(config-tacacs-server)# exit
Device(config)# radius server server2
Device(config-radius-server)# address IPv4 172.16.2.7
Device(config-radius-server)# key myRaDiUSpassWoRd
Device(config-radius-server)# 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 blue1
Device(config-if)# ppp accounting red1
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 blue1 group radius local** command defines the network authorization method list named “blue1”, which specifies that RADIUS authorization is used on serial lines using PPP. If the RADIUS server fails to respond, then local network authorization is performed.
- The **aaa accounting network red1 start-stop group radius group tacacs+**command defines the network accounting method list named red1, which specifies that RADIUS accounting services (in this case, start and stop records for specific events) are used on serial lines using PPP. If the RADIUS server fails to respond, accounting services are handled by a TACACS+ server.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **tacacs server** command defines the name of the TACACS+ server host, and the **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, and 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.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication chap dialins**command selects Challenge Handshake Authentication Protocol (CHAP) as the method of PPP authentication and applies the “dialins” method list to the specified interfaces.
- The **ppp authorization blue1**command applies the blue1 network authorization method list to the specified interfaces.
- The **ppp accounting red1**command applies the red1 network accounting method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the admins method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

The **show accounting**command yields the following output for the preceding configuration:

```
Device# show accounting

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.
aaa new-model
!Enable authentication at login and list the AOL string name to use for login authentication.
aaa authentication login AOL group radius local
!Enable authentication for ppp and list the default method to use for PPP authentication.
aaa authentication ppp default group radius local
!Enable authorization for all exec sessions and list the AOL string name to use for
authorization.
aaa authorization exec AOL group radius if-authenticated
!Enable authorization for all network-related service requests and list the default method
 to use for all network-related authorizations.
aaa authorization network default group radius if-authenticated
!Enable accounting for all exec sessions and list the default method to use for all start-stop
 accounting services.
aaa accounting exec default start-stop group radius
!Enable accounting for all network-related service requests and list the default method to
 use for all start-stop accounting services.
aaa accounting network default start-stop group radius
!Enable failure stop accounting.
aaa accounting resource default stop-failure group radius
!Enable resource accounting for start-stop records.
aaa accounting resource default start-stop group radius
```

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)# radius server server
Device(config-radius-server)# address IPv4 10.0.0.1
Device(config-radius-server)# key key_1
Device(config-radius-server)# exit
Device(config)# radius server server
Device(config-radius-server)# address IPv4 10.0.0.2
Device(config-radius-server)# key key_1
Device(config-radius-server)# exit
Device(config)# tacacs server server1
Device(config-tacacs-server)# address IPv4 172.0.0.1
Device(config-tacacs-server)# key key2
Device(config-tacacs-server)# 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 network**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-radius)# server 172.0.0.1
Device(config-sg-radius)# 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)# radius server server
Device(config-radius-server)# address IPv4 10.0.0.1
Device(config-radius-server)# key key_1
Device(config-radius-server)# exit
Device(config)# radius server server
Device(config-radius-server)# address IPv4 10.0.0.2
Device(config-radius-server)# key key_1
Device(config-radius-server)# exit
Device(config)# tacacs server server
Device(config-tacacs-server)# address IPv4 172.0.0.1
Device(config-tacacs-server)# key key_2
Device(config-tacacs-server)# end
s
```

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
```

Example Configuring VRRS Accounting

The following example shows how to configure VRRS to send AAA Accounting messages to the AAA server:

```
Device> enable
Device# configure terminal
Device(config)# aaa accounting vrrs vrrp-mlist-1 start-stop group radius
Device(config)# aaa attribute list vrrp-1-attr
Device(config-attr-list)# attribute type account-delay 10
Device(config-attr-list)# exit
Device(config)# vrrs vrrp-group-1
Device(config-vrrs)# accounting delay 10
Device(config-vrrs)# accounting method vrrp-mlist-1
Device(config-vrrs)# end
```

Additional References for Configuring Accounting

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

RFCs

RFC	Title
RFC 2903	*Generic AAA Architecture*
RFC 2904	*AAA Authorization Framework*
RFC 2906	*AAA Authorization Requirements*
RFC 2989	*Criteria for Evaluating AAA Protocols for Network Access*

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

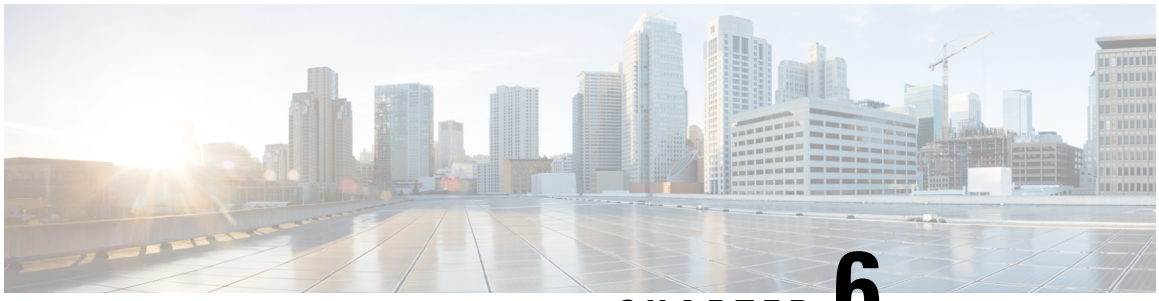
Feature History for Configuring Accounting

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 Release 15.2(5)E	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.

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 6

Configuring Local Authentication and Authorization

How to Configure the Switch for Local Authentication and Authorization

You can configure authentication, authorization, and accounting (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** command in global configuration mode. Configuring AAA authentication does not secure the switch for HTTP access by using AAA methods.

To configure AAA to operate without a server by setting the switch to implement AAA in local mode, perform this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login default local**
5. **aaa authorization exec default local**
6. **aaa authorization network default local**
7. **username** *name* [**privilege** *level*] {**password** *encryption-type password*}
8. **end**
9. **show running-config**
10. **copy running-config startup-config**

DETAILED STEPS

	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 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. • *name*: Specify the user ID as one word. Spaces and quotation marks are not allowed. • *level*: (Optional) 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. • *encryption-type*: Enter 0 to specify an unencrypted password. Enter 7 to specify a hidden password . • *password*: Specify the password the user must enter to gain access to the 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.

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config-line)# 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

Table 14: Commands for Displaying Local Authentication and Authorization

Command	Purpose
show running-config	Displays the local authentication and authorization configuration.

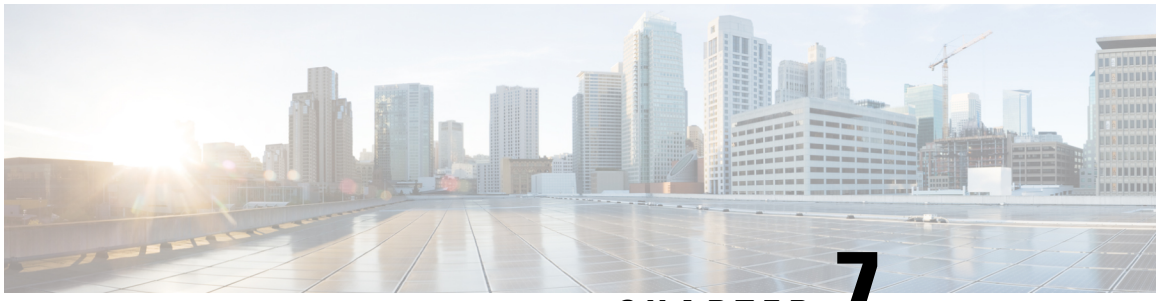
Feature History for Local Authentication and Authorization

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 Release 15.2(5)E	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

MAC Authentication Bypass

The MAC Authentication Bypass feature is a MAC-address-based authentication mechanism that allows clients in a network to integrate with the Cisco Identity Based Networking Services (IBNS) and Network Admission Control (NAC) strategy using the client MAC address. The MAC Authentication Bypass feature is applicable to the following network environments:

- Network environments in which a supplicant code is not available for a given client platform.
- Network environments in which the end client configuration is not under administrative control, that is, the IEEE 802.1X requests are not supported on these networks.

Prerequisites for Configuring MAC Authentication Bypass

IEEE 802.1x—Port-Based Network Access Control

You should understand the concepts of port-based network access control and have an understanding of how to configure port-based network access control on your Cisco platform.

RADIUS and ACLs

You should understand the concepts of the RADIUS protocol and have an understanding of how to create and apply access control lists (ACLs). For more information, see the documentation for your Cisco platform and the *Securing User Services Configuration Guide Library*.

The device must have a RADIUS configuration and be connected to the Cisco secure access control server (ACS). For more information, see the *User Guide for Secure ACS Appliance 3.2*.

Information About MAC Authentication Bypass

Overview of the Cisco IOS Auth Manager

The capabilities of devices connecting to a given network can be different, thus requiring that the network support different authentication methods and authorization policies. The Cisco IOS Auth Manager handles network authentication requests and enforces authorization policies regardless of authentication method. The Auth Manager maintains operational data for all port-based network connection attempts, authentications, authorizations, and disconnections and, as such, serves as a session manager.

The possible states for Auth Manager sessions are as follows:

- Idle—In the idle state, the authentication session has been initialized, but no methods have yet been run. This is an intermediate state.
- Running—A method is currently running. This is an intermediate state.
- Authc Success—The authentication method has run successfully. This is an intermediate state.
- Authc Failed—The authentication method has failed. This is an intermediate state.
- Authz Success—All features have been successfully applied for this session. This is a terminal state.
- Authz Failed—At least one feature has failed to be applied for this session. This is a terminal state.
- No methods—There were no results for this session. This is a terminal state.

Overview of the Configurable MAB Username and Password

A MAC Authentication Bypass (MAB) operation involves authentication using RADIUS Access-Request packets with both the username and password attributes. By default, the username and the password values are the same and contain the MAC address. The Configurable MAB Username and Password feature enables you to configure both the username and the password attributes in the following scenarios:

- To enable MAB for an existing large database that uses formatted username attributes, the username format in the client MAC needs to be configured. Use the **mab request format attribute 1** command to configure the username format.
- Some databases do not accept authentication if the username and password values are the same. In such instances, the password needs to be configured to ensure that the password is different from the username. Use the **mab request format attribute 2** command to configure the password.

The Configurable MAB Username and Password feature allows interoperability between the Cisco IOS Authentication Manager and the existing MAC databases and RADIUS servers. The password is a global password and hence is the same for all MAB authentications and interfaces. This password is also synchronized across all supervisor devices to achieve high availability.

If the password is not provided or configured, the password uses the same value as the username. The table below describes the formatting of the username and the password:

MAC Address	Username Format (Group Size, Separator)	Username	Password Configured	Password Created
08002b8619de	(1, :) (1, -) (1, .)	0:8:0:0:2:b:8:6:1:9:d:e 0-8-0-0-2-b-8-6-1-9-d-e 0.8.0.0.2.b.8.6.1.9.d.e	None	0:8:0:0:2:b:8:6:1:9:d:e 0-8-0-0-2-b-8-6-1-9-d-e 0.8.0.0.2.b.8.6.1.9.d.e
08002b8619de	(1, :) (1, -) (1, .)	0:8:0:0:2:b:8:6:1:9:d:e 0-8-0-0-2-b-8-6-1-9-d-e 0.8.0.0.2.b.8.6.1.9.d.e	Password	Password
08002b8619de	(2, :) (2, -) (2, .)	08:00:2b:86:19:de 08-00-2b-86-19-de 08.00.2b.86.19.de	None	08:00:2b:86:19:de 08-00-2b-86-19-de 08.00.2b.86.19.de
08002b8619de	(2, :) (2, -) (2, .)	08:00:2b:86:19:de 08-00-2b-86-19-de 08.00.2b.86.19.de	Password	Password
08002b8619de	(4, :) (4, -) (4, .)	0800:2b86:19de 0800-2b86-19de 0800.2b86.19de	None	0800:2b86:19de 0800-2b86-19de 0800.2b86.19de
08002b8619de	(4, :) (4, -) (4, .)	0800:2b86:19de 0800-2b86-19de 0800.2b86.19de	Password	Password
08002b8619de	(12, <not applicable>)	08002b8619de	None	08002b8619de
08002b8619de	(12, <not applicable>)	08002b8619de	Password	Password

How to Configure MAC Authentication Bypass

Enabling MAC Authentication Bypass

Perform this task to enable the MAC Authentication Bypass feature on an 802.1X port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type slot* / *port*
4. **mab**
5. **end**
6. **show authentication sessions interface** *type slot* / *port* **details**

DETAILED STEPS

	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** *type slot* / *port* **Example:** `Device(config)# interface gigabitethernet 2/1`	Enters interface configuration mode.
Step 4	**mab** **Example:** `Device(config-if)# mab`	Enables MAB.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show authentication sessions interface** *type slot* / *port* **details** **Example:** `Device# show authentication sessions interface gigabitethernet 2/1`	Displays the interface configuration and the authenticator instances on the interface.

Enabling Reauthentication on a Port

By default, ports are not automatically reauthenticated. You can enable automatic reauthentication and specify how often reauthentication attempts are made.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface** *type slot* / *port*
4. **switchport**
5. **switchport mode access**
6. **authentication port-control auto**
7. **mab** [**eap**]
8. **authentication periodic**
9. **authentication timer reauthenticate** {*seconds* | **server**}
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `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** *type slot* / *port* **Example:** `Device(config)# interface gigabitethernet 2/1`	Enters interface configuration mode.
Step 4	**switchport** **Example:** `Device(config-if)# switchport`	Places interface in Layer 2 switched mode.
Step 5	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Sets the interface type as a nontrunking, nontagged single VLAN Layer 2 interface.
Step 6	**authentication port-control auto** **Example:** `Device(config-if)# authentication port-control auto`	Configures the authorization state of the port.
Step 7	**mab** [**eap**] **Example:** `Device(config-if)# mab`	Enables MAB.

	Command or Action	Purpose
Step 8	**authentication periodic** **Example:** `Device(config-if)# authentication periodic`	Enables reauthentication.
Step 9	**authentication timer reauthenticate** {*seconds* | **server**} **Example:** `Device(config-if)# authentication timer reauthenticate 900`	Configures the time, in seconds, between reauthentication attempts.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Specifying the Security Violation Mode

When there is a security violation on a port, the port can be shut down or traffic can be restricted. By default, the port is shut down. You can configure the period of time for which the port is shut down.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **switchport**
5. **switchport mode access**
6. **authentication port-control auto**
7. **mab** [**eap**]
8. **authentication violation** {**protect** | **replace** | **restrict** | **shutdown**}
9. **authentication timer restart** *seconds*
10. **end**

DETAILED STEPS

	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	**interface** *type slot* / *port* **Example:** `Device(config)# interface gigabitethernet 2/1`	Enters interface configuration mode.
Step 4	**switchport** **Example:** `Device(config-if)# switchport`	Places interface in Layer 2 switched mode.
Step 5	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Sets the interface type as a nontrunking, nontagged single VLAN Layer 2 interface.
Step 6	**authentication port-control auto** **Example:** `Device(config-if)# authentication port-control auto`	Configures the authorization state of the port.
Step 7	**mab** [**eap**] **Example:** `Device(config-if)# mab`	Enables MAB.
Step 8	**authentication violation** {**protect** | **replace** | **restrict** | **shutdown**} **Example:** `Device(config-if)# authentication violation shutdown`	Configures the action to be taken when a security violation occurs on the port.
Step 9	**authentication timer restart** *seconds* **Example:** `Device(config-if)# authentication timer restart 30`	Configures the period of time, in seconds, after which an attempt is made to authenticate an unauthorized port.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Enabling Configurable MAB Username and Password

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mab request format attribute 1 groupsize** {**1** | **2** | **4** | **12**} **separator** {**-** | **:** | **.**} [**lowercase** | **uppercase**]
4. **mab request format attribute 2** [**0** | **7**] *password*
5. **end**

DETAILED STEPS

	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	**mab request format attribute 1 groupsize** {**1** | **2** | **4** | **12**} **separator** {**-** | **:** | **.**} [**lowercase** | **uppercase**] **Example:** `Device(config)# mab request format attribute 1 groupsize 2 separator :`	Configures the username format for MAB requests.
Step 4	**mab request format attribute 2** [**0** | **7**] *password* **Example:** `Device(config)# mab request format attribute 2 password1`	Configures a global password for all MAB requests.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuration Examples for MAC Authentication Bypass

Example: MAC Authentication Bypass Configuration

In the following example, the **mab** command has been configured to enable the MAC Authorization Bypass (MAB) feature on the specified interface. The optional **show authentication sessions** command has been enabled to display the interface configuration and the authentication instances on the interface.

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 2/1
Device(config-if)# mab
Device(config-if)# end
Device# show authentication sessions interface gigabitethernet 2/1 details
```

Example: Enabling Configurable MAB Username and Password

The following example shows how to configure the username format and password for MAC Authentication Bypass (MAB). In this example, the username format is configured as a group of 12 hexadecimal digits with no separator and the global password as **password1**.

```
Device> enable
Device# configure terminal
Device(config)# mab request format attribute 1 groupsize 2 separator :
Device(config)# mab request format attribute 2 password1
Device(config)# end
```

Additional References for MAC Authentication Bypass

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

MIBs

MIB	MIBs Link
• CISCO-AUTH-FRAMEWORK-MIB • CISCO-MAC-AUTH-BYPASS-MIB • CISCO-PAE-MIB • IEEE8021-PAE-MIB	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3580	*IEEE 802.1x Remote Authentication Dial In User Service (RADIUS)*

Technical Assistance

Description	Link
The Cisco Support 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

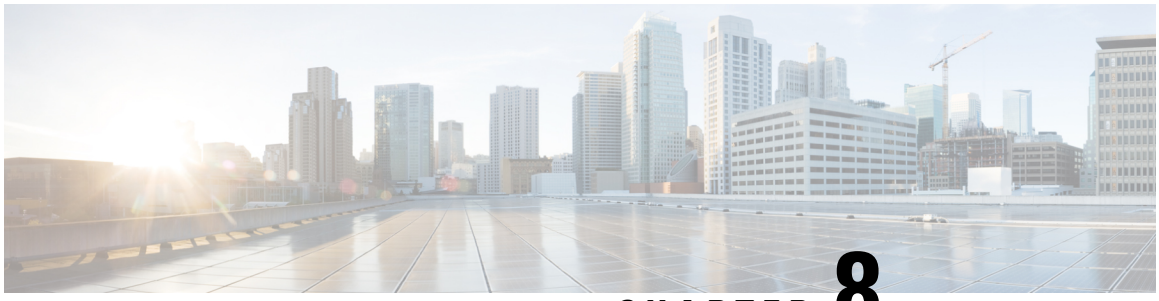
Feature History for MAC Authentication Bypass

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 Release 15.2(5)E	MAC Authentication Bypass	The MAC Authentication Bypass feature is a MAC-address-based authentication mechanism that allows clients in a network to integrate with the IBNS and NAC strategy using the client MAC address.

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 8

Password Strength and Management for Common Criteria

The Password Strength and Management for Common Criteria feature is used to specify password policies and security mechanisms for storing, retrieving, and providing rules to specify user passwords.

For local users, the user profile and the password information with the key parameters are stored on the Cisco device, and this profile is used for local authentication of users. The user can be an administrator (terminal access) or a network user (for example, PPP users being authenticated for network access).

For remote users, where the user profile information is stored in a remote server, a third-party authentication, authorization, and accounting (AAA) server may be used for providing AAA services, both for administrative and network access.

Restrictions for Password Strength and Management for Common Criteria

Only four concurrent users can log on to the system by using vty at any moment.

Information About Password Strength and Management for Common Criteria

The following sections provide information on password strength and management.

Password Composition Policy

The password composition policy allows you to create passwords of any combination of upper and lowercase characters, numbers, and special characters that include “!”, “@”, “#”, “$”, “%”,“^”, “&”, “*”, “(“, and “)”.

Password Length Policy

The administrator has the flexibility to set the password's minimum and maximum length. The recommended minimum password length is 8 characters. The administrator can specify both the minimum (1) and the maximum (64) length for the password.

Password Lifetime Policy

The security administrator can provide a configurable option for a password to have a maximum lifetime. If the lifetime parameter is not configured, the configured password will never expire. The maximum lifetime can be configured by providing the configurable value in years, months, days, hours, minutes, and seconds. The lifetime configuration will survive across reloads as it is a part of the configuration, but every time the system reboots, the password creation time will be updated to the new time. For example, if a password is configured with a lifetime of one month and on the 29th day, the system reboots, then the password will be valid for one month after the system reboots.

Password Expiry Policy

If the user attempts to log on and if the user's password credentials have expired, then the following happens:

1. The user is prompted to set the new password after successfully entering the expired password.
2. When the user enters the new password, the password is validated against the password security policy.
3. If the new password matches the password security policy, then the authentication, authorization, and accounting (AAA) database is updated, and the user is authenticated with the new password.
4. If the new password does not match the password security policy, then the user is prompted again for the password. From AAA perspective, there is no restriction on the number of retries. The number of retries for password prompt in case of unsuccessful authentication is controlled by the respective terminal access interactive module. For example, for telnet, after three unsuccessful attempts, the session will be terminated.

If the password's lifetime is not configured for a user and the user has already logged on and if the security administrator configures the lifetime for that user, then the lifetime will be set in the database. When the same user is authenticated the next time, the system will check for password expiry. The password expiry is checked only during the authentication phase.

If the user has been already authenticated and logged on to the system and if the password expires, then no action will be taken. The user will be prompted to change the password only during the next authentication for the same user.

Password Change Policy

The new password must contain a minimum of 4 character changes from the previous password. A password change can be triggered by the following scenarios:

- The security administrator wants to change the password.
- The user is trying to get authenticated using a profile, and the password for that profile has expired.

When the security administrator changes the password security policy and the existing profile does not meet the password security policy rules, no action will be taken if the user has already logged on to the system. The user will be prompted to change the password only when the user tries to get authenticated using the profile that does not meet the password security restriction.

When the user changes the password, the lifetime parameters set by the security administrator for the old profile will be the lifetime parameters for the new password.

For noninteractive clients such as dot1x, when the password expires, appropriate error messages will be sent to the clients, and the clients must contact the security administrator to renew the password.

User Reauthentication Policy

Users are reauthenticated when they change their passwords.

When users change their passwords on expiry, they will be authenticated against the new password. In such cases, the actual authentication happens based on the previous credentials, and the new password is updated in the database.

Note Users can change their passwords only when they are logging on and after the expiry of the old password; however, a security administrator can change the user's password at any time.

Support for Framed (Noninteractive) Session

When a client such as dot1x uses the local database for authentication, the Password Strength and Management for Common Criteria feature will be applicable; however, upon password expiry, clients will not be able to change the password. An appropriate failure message will be sent to such clients, and the user must request the security administrator to change the password.

How to Configure Password Strength and Management for Common Criteria

The following sections provide information on configuring password strength and management.

Configuring the Password Security Policy

To create a password security policy and to apply the policy to a specific user profile, perform this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **aaa new-model**
4. **aaa common-criteria policy** *policy-name*
5. **char-changes** *number*
6. **max-length** *number*
7. **min-length** *number*
8. **numeric-count** *number*
9. **special-case** *number*
10. **exit**
11. **username** *username* **common-criteria-policy** *policy-name* **password** *password*
12. **end**

DETAILED STEPS

	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 common-criteria policy** *policy-name* **Example:** `Device(config)# aaa common-criteria policy policy1`	Creates the AAA security password policy and enters common criteria configuration policy mode.
Step 5	**char-changes** *number* **Example:** `Device(config-cc-policy)# char-changes 4`	(Optional) Specifies the number of changed characters between old and new passwords.
Step 6	**max-length** *number* **Example:** `Device(config-cc-policy)# max-length 25`	(Optional) Specifies the maximum length of the password.
Step 7	**min-length** *number* **Example:** `Device(config-cc-policy)# min-length 8`	(Optional) Specifies the minimum length of the password.
Step 8	**numeric-count** *number* **Example:** `Device(config-cc-policy)# numeric-count 4`	(Optional) Specifies the number of numeric characters in the password.

	Command or Action	Purpose
Step 9	**special-case** *number* **Example:** `Device(config-cc-policy)# special-case 3`	(Optional) Specifies the number of special characters in the password.
Step 10	**exit** **Example:** `Device(config-cc-policy)# exit`	(Optional) Exits common criteria configuration policy mode and returns to global configuration mode.

Step 11	**username** *username* **common-criteria-policy** *policy-name* **password** *password*
	Example:
	`Device(config)# username user1 common-criteria-policy policy1 password password1`
		(Optional) Applies a specific policy and password to a user profile.
		Note A single numerical character is not accepted as password. The following console message is displayed if you try to configure a password with a single numerical character.
		`username user2 common-criteria-policy Hay_passwd_policy_2 password 3`
		`% Incomplete command.`

Step 12	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Verifying the Common Criteria Policy

To verify all the common criteria security policies, perform this procedure.

SUMMARY STEPS

1. **enable**
2. **show aaa common-criteria policy name** *policy-name*
3. **show aaa common-criteria policy all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.

Step 2	**show aaa common-criteria policy name** *policy-name*
	Example:
	`Device# show aaa common-criteria policy name policy1`
	`Policy name: policy1`
		Displays the password security policy information for a specific policy.

	Command or Action	Purpose
	Minimum length: 1 Maximum length: 64 Upper Count: 20 Lower Count: 20 Numeric Count: 5 Special Count: 2 Number of character changes 4 Valid forever. User tied to this policy will not expire.	
Step 3	**show aaa common-criteria policy all** **Example:** Device# **show aaa common-criteria policy all** Policy name: policy1 Minimum length: 1 Maximum length: 64 Upper Count: 20 Lower Count: 20 Numeric Count: 5 Special Count: 2 Number of character changes 4 Valid forever. User tied to this policy will not expire. Policy name: policy2 Minimum length: 1 Maximum length: 34 Upper Count: 10 Lower Count: 5 Numeric Count: 4 Special Count: 2 Number of character changes 2 Valid forever. User tied to this policy will not expire.	Displays password security policy information for all the configured policies.

Configuration Example for Password Strength and Management for Common Criteria

The following section provides a configuration example for password strength and management for common criteria.

Example: Password Strength and Management for Common Criteria

The following example shows how to create a common criteria security policy and apply the specific policy to a user profile:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa common-criteria policy policy1
Device(config-cc-policy)# char-changes 4
```

```
Device(config-cc-policy)# max-length 20
Device(config-cc-policy)# min-length 6
Device(config-cc-policy)# numeric-count 2
Device(config-cc-policy)# special-case 2
Device(config-cc-policy)# exit
Device(config)# username user1 common-criteria-policy policy1 password password1
Device(config)# end
```

Additional References for Password Strength and Management for Common Criteria

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)E (Catalyst 2960-L Switches)

RFCs

RFC	Title
RFC 2865	*Remote Authentication Dial-in User Service*
RFC 3576	*Dynamic Authorization Extensions to RADIUS*

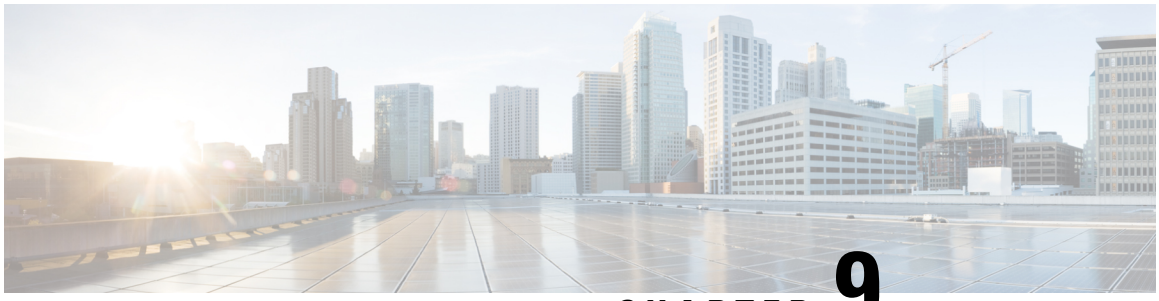
Feature History for Password Strength and Management for Common Criteria

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 Release 15.2(5)E	Password Strength and Management for Common Criteria	The Password Strength and Management for Common Criteria feature is used to specify password policies and security mechanisms for storing, retrieving, and providing rules to specify user passwords.

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 9

AAA-SERVER-MIB Set Operation

The AAA-SERVER-MIB Set Operation feature allows the authentication, authorization, and accounting (AAA) server configuration to be extended or expanded by using the CISCO-AAA-SERVER-MIB to create and add new AAA servers, modify the “KEY” under the CISCO-AAA-SERVER-MIB, and delete the AAA server configuration.

Prerequisites for AAA-SERVER-MIB Set Operation

AAA must have been enabled on the router, that is, the **aaa new-model** command must have been configured. If this configuration has not been accomplished, the set operation fails.

Restrictions for AAA-SERVER-MIB Set Operation

Currently, the CISCO SNMP set operation is supported only for the RADIUS protocol. Therefore, only RADIUS servers in global configuration mode can be added, modified, or deleted.

Information About AAA-SERVER-MIB Set Operation

The AAA-SERVER-MIB Set Operation feature allows the authentication, authorization, and accounting (AAA) server configuration to be extended or expanded by using the CISCO-AAA-SERVER-MIB to create and add new AAA servers, modify the “KEY” under the CISCO-AAA-SERVER-MIB, and delete the AAA server configuration.

CISCO-AAA-SERVER-MIB

The CISCO-AAA-SERVER-MIB provides that statistics reflect both the state of the AAA server operation with the server itself and of AAA communications with external servers. The CISCO-AAA-SERVER-MIB provides the following information:

- Statistics for each AAA operation
- Status of servers that are providing AAA functions
- Identities of external AAA servers

CISCO-AAA-SERVER-MIB Set Operation

With the SET operation, you can do the following:

- Create or add a new AAA server.
- Modify the KEY under the CISCO-AAA-SERVER-MIB. This “secret key” is used for secure connectivity to the AAA server, which is present with the network access server (NAS) and the AAA server.
- Delete the AAA server configuration.

How to Configure AAA-SERVER-MIB Set Operation

The following sections provide information about how to configure AAA-SERVER-MIB set operation:

Configuring AAA-SERVER-MIB Set Operations

No special configuration is required for this feature. The Simple Network Management Protocol (SNMP) framework can be used to manage MIBs. See the Additional References section for a reference to configuring SNMP.

Verifying SNMP Values

SNMP values can be verified by performing the following steps.

SUMMARY STEPS

1. **enable**
2. **show running-config aaa**
3. **show aaa servers**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enters privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**show running-config aaa** **Example:** `Device# show running-config aaa`	Displays all the authentication, authorization, and accounting (AAA) servers that are configured in the global configuration mode.
Step 3	**show aaa servers** **Example:** `Device# show aaa servers`	Displays information about the number of requests sent to and received from authentication, authorization, and accounting (AAA) servers.

Configuration Examples for AAA-SERVER-MIB Set Operation

This section provides information about configuration examples for AAA-SERVER-MIB set operation:

RADIUS Server Configuration and Server Statistics Example

The following sample output shows the RADIUS server configuration and server statistics before and after the set operation.

Before the Set Operation

```
Device# show aaa servers

RADIUS: id 2, priority 1, host 172.19.192.238, auth-port 2095, acct-port 2096
State: current UP, duration 25s, previous duration 0s
     Dead: total time 0s, count 7
Authen: request 8, timeouts 8
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 2
Author: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Account: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Elapsed time since counters last cleared: 5m
RADIUS: id 3, priority 2, host 172.19.192.238, auth-port 1645, acct-port 1646
State: current UP, duration 5s, previous duration 0s
     Dead: total time 0s, count 2
Authen: request 8, timeouts 8
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 4
Author: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Account: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Elapsed time since counters last cleared: 3m
```

SNMP Get Operation to Check the Configuration and Statistics of the RADIUS Servers

```
aaa-server5:/users/smetri> getmany 10.0.1.42 casConfigTable
casAddress.2.2 = 172.19.192.238
casAddress.2.3 = 172.19.192.238
casAuthenPort.2.2 = 2095
casAuthenPort.2.3 = 1645
casAcctPort.2.2 = 2096
casAcctPort.2.3 = 1646
casKey.2.2 =
casKey.2.3 =
! The following line shows priority for server 1.
casPriority.2.2 = 1
! The following line shows priority for server 2.
casPriority.2.3 = 2
casConfigRowStatus.2.2 = active(1)
casConfigRowStatus.2.3 = active(1)
aaa-server5:/users/smetri>
```

SNMP Set Operation

The key of the existing RADIUS server is being changed. The index “1” is being used. That index acts as a wildcard for addition, deletion, or modification of any entries.

```
Change the key for server 1:=>
aaa-server5:/users/smetri>  setany -v2c 10.0.1.42 public casAddress.2.1 -a 172.19.192.238
casAuthenPort.2.1 -i 2095 casAcctPort.2.1 -i 2096 casKey.2.1 -o king
casAddress.2.1 = 172.19.192.238
casAuthenPort.2.1 = 2095
casAcctPort.2.1 = 2096
casKey.2.1 = king
aaa-server5:/users/smetri>
```

After the Set Operation

After the above SNMP set operation, the configurations on the device change. The following output shows the output after the set operation.

```
Device# show aaa servers

RADIUS: id 3, priority 1, host 172.19.192.238, auth-port 1645, acct-port 1646
State: current UP, duration 189s, previous duration 0s
     Dead: total time 0s, count 2
Authen: request 8, timeouts 8
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 4
Author: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Account: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Elapsed time since counters last cleared: 6m

! The following line shows a new server with new statistics.
RADIUS: id 4, priority 2, host 172.19.192.238, auth-port 2095, acct-port 2096
State: current UP, duration 209s, previous duration 0s
     Dead: total time 0s, count 7
Authen: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
```

```
     Transaction: success 0, failure 0
Author: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Account: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
```

Additional References for AAA-SERVER-MIB Set Operation

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

MIBs

MIB	MIBs Link
AAA-SERVER-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

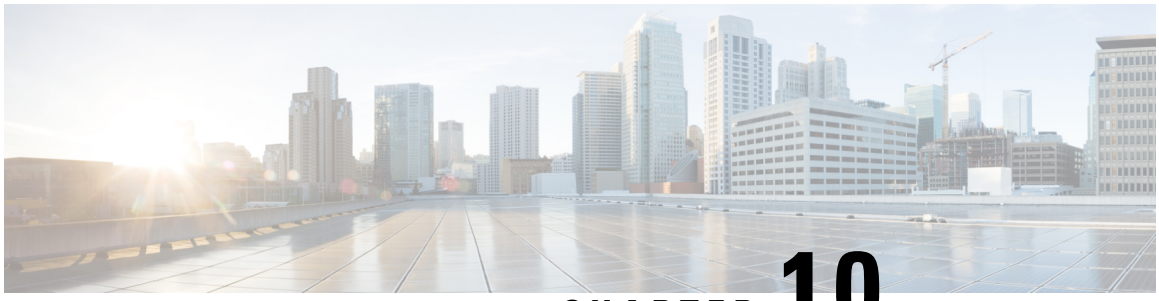
Feature History for AAA-SERVER-MIB Set Operation

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 Release 15.2(5)E	AAA-SERVER-MIB Set Operation	The AAA-SERVER-MIB Set Operation feature allows the authentication, authorization, and accounting (AAA) server configuration to be extended or expanded by using the CISCO-AAA-SERVER-MIB to create and add new AAA servers, modify the “KEY” under the CISCO-AAA-SERVER-MIB, and delete the AAA server configuration.

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 10

Configuring Secure Shell

The Secure Shell (SSH) feature is an application and a protocol that provides a secure replacement to the Berkeley r-tools. The protocol secures sessions using standard cryptographic mechanisms, and the application can be used similarly to the Berkeley rexec and rsh tools. Two versions of SSH are available: SSH Version 1 and SSH Version 2.

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 1 (SSHv1) and 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 1 (SSHv1) and 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.

The switch supports an SSHv1 or an SSHv2 server.

The switch supports an SSHv1 client.

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

RSA Authentication Support

Rivest, Shamir, and Adleman (RSA) authentication available in Secure Shell (SSH) clients is not supported on the SSH server for Cisco software by default.

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.

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 Secure Copy Works

The behavior of Secure Copy (SCP) is similar to that of remote copy (RCP), which comes from the Berkeley r-tools suite (Berkeley university's own set of networking applications), except that SCP relies on Secure Shell (SSH) for security. In addition, SCP requires that authentication, authorization, and accounting (AAA) authorization be configured so that the device can determine whether the user has the correct privilege level.

SCP allows a user only with a privilege level of 15 to copy any file that exists in the Cisco IOS File System (IFS) to and from a device by using the **copy** command. An authorized administrator may also perform this action from a workstation.

Note Enable the SCP option while using the pscp.exe file with the Cisco software.

Reverse Telnet

Reverse telnet allows you to telnet to a certain port range and connect to terminal or auxiliary lines. Reverse telnet has often been used to connect a Cisco device that has many terminal lines to the consoles of other Cisco devices. Telnet makes it easy to reach the device console from anywhere simply by telnet to the terminal server on a specific line. This telnet approach can be used to configure a device even if all network connectivity to that device is disconnected. Reverse telnet also allows modems that are attached to Cisco devices to be used for dial-out (usually with a rotary device).

Reverse SSH

Reverse telnet can be accomplished using SSH. Unlike reverse telnet, SSH provides for secure connections. The Reverse SSH Enhancements feature provides you with a simplified method of configuring SSH. Using this feature, you no longer have to configure a separate line for every terminal or auxiliary line on which you want to enable SSH. The previous method of configuring reverse SSH limited the number of ports that can be accessed to 100. The Reverse SSH Enhancements feature removes the port number limitation.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *hostname*
4. **ip domain-name** *domain_name*
5. **crypto key generate rsa**
6. **end**

DETAILED STEPS

	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.
Step 4	**ip domain-name** *domain_name* **Example:** `Device(config)# ip domain-name your_domain`	Configures a host domain for your device.

	Command or Action	Purpose
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.

Configuring the SSH Server

Follow the procedure given below to configure the SSH server:

Note This procedure is only required if you are configuring the device as an SSH server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh version** [**1** | **2**]
4. **ip ssh** {**timeout** *seconds* | **authentication-retries** *number*}
5. Use one or both of the following:
   - `line vty` *line_number*[*ending_line_number*]
   - **transport input ssh**
6. **end**

DETAILED STEPS

	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	**ip ssh version** [**1** | **2**] **Example:** `Device(config)# ip ssh version 1`	(Optional) Configures the device to run SSH Version 1 or SSH Version 2. • **1**: Configure the device to run SSH Version 1. • **2**: Configure 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. For example, if the SSH client supports SSHv1 and SSHv2, the SSH server selects SSHv2.
Step 4	**ip ssh** {**timeout** *seconds* | **authentication-retries** *number*} **Example:** `Device(config)# ip ssh timeout 90 authentication-retries 2`	Configures the SSH control parameters: • 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. • 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.

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 *line_number* and *ending_line_number*, specify a pair of lines. The range is 0 to 15.
		• Specifies that the device prevent non-SSH Telnet connections. This limits the router to only SSH connections.

Step 6	**end** **Example:**	Returns to privileged EXEC mode.

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

Troubleshooting Tips

- If your Secure Shell (SSH) configuration commands are rejected as illegal commands, you have not successfully generated an Rivest, Shamir, and Adleman (RSA) key pair for your device. Make sure that you have specified a hostname and domain. Then use the **crypto key generate rsa** command to generate an RSA key pair and enable the SSH server.
- When configuring the RSA key pair, you might encounter the following error messages:
  - No hostname specified.

    You must configure a hostname for the device using the **hostname** global configuration command.
  - No domain specified.

    You must configure a host domain for the device using the **ip domain-name** global configuration command.
- The number of allowable SSH connections is limited to the maximum number of vtys configured for the device. Each SSH connection uses a vty resource.
- SSH uses either local security or the security protocol that is configured through AAA on your device for user authentication. When configuring Authentication, Authorization, and Accounting (AAA), you must ensure that AAA is disabled on the console for user authentication. 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.

Configuring Reverse SSH for Console Access

To configure reverse SSH console access on the SSH server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number ending-line-number*
4. **no exec**
5. **login authentication** *listname*
6. **transport input ssh**
7. **exit**
8. **exit**
9. **ssh -l** *userid* **:** {*number*} {*ip-address*}

DETAILED STEPS

	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** *line-number ending-line-number* **Example:** `Device# line 1 3`	Identifies a line for configuration and enters line configuration mode.
Step 4	**no exec** **Example:** `Device(config-line)# no exec`	Disables EXEC processing on a line.
Step 5	**login authentication** *listname* **Example:** `Device(config-line)# login authentication default`	Defines a login authentication mechanism for the lines. **Note** The authentication method must use a username and password.
Step 6	**transport input ssh** **Example:** `Device(config-line)# transport input ssh`	Defines which protocols to use to connect to a specific line of the device. • The **ssh** keyword must be used for the Reverse SSH Enhancements feature.
Step 7	**exit** **Example:** `Device(config-line)# exit`	Exits line configuration mode.
Step 8	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 9	**ssh -l** *userid* **:** {*number*} {*ip-address*} **Example:** `Device# ssh -l lab:1 router.example.com`	Specifies the user ID to use when logging in on the remote networking device that is running the SSH server. • *userid* : User ID.

	Command or Action	Purpose
		• **:** : Signifies that a port number and terminal IP address will follow the userid argument. • *number*: Terminal or auxiliary line number. • *ip-address* : Terminal server IP address. **Note** The *userid* argument and **:rotary**{*number*}{*ip-address*} delimiter and arguments are mandatory when configuring the alternative method of Reverse SSH for modem access.

Configuring Reverse SSH for Modem Access

In this configuration, reverse SSH is being configured on a modem used for dial-out lines. To get any of the dial-out modems, you can use any SSH client and start a SSH session to get to the next available modem from the rotary device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number ending-line-number*
4. **no exec**
5. **login authentication** *listname*
6. **rotary** *group*
7. **transport input ssh**
8. **exit**
9. **exit**
10. **ssh -l** *userid* **:rotary** {*number*} {*ip-address*}

DETAILED STEPS

	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	**line** *line-number ending-line-number* **Example:** `Device# line 1 200`	Identifies a line for configuration and enters line configuration mode.
Step 4	**no exec** **Example:** `Device(config-line)# no exec`	Disables EXEC processing on a line.
Step 5	**login authentication** *listname* **Example:** `Device(config-line)# login authentication default`	Defines a login authentication mechanism for the lines. **Note** The authentication method must use a username and password.
Step 6	**rotary** *group* **Example:** `Device(config-line)# rotary 1`	Defines a group of lines consisting of one or more virtual terminal lines or one auxiliary port line.
Step 7	**transport input ssh** **Example:** `Device(config-line)# transport input ssh`	Defines which protocols to use to connect to a specific line of the device. • The **ssh** keyword must be used for the Reverse SSH Enhancements feature.
Step 8	**exit** **Example:** `Device(config-line)# exit`	Exits line configuration mode.
Step 9	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 10	**ssh -l** *userid* **:rotary** {*number*} {*ip-address*} **Example:** `Device# ssh -l lab:rotary1 router.example.com`	Specifies the user ID to use when logging in on the remote networking device that is running the SSH server. • *userid* : User ID. • **:** : Signifies that a port number and terminal IP address will follow the *userid* argument. • *number* : Terminal or auxiliary line number. • *ip-address* : Terminal server IP address.

	Command or Action	Purpose
		Note The *userid* argument and **:rotary**{*number*}{*ip-address*} delimiter and arguments are mandatory when configuring the alternative method of Reverse SSH for modem access.

Troubleshooting Reverse SSH on the Client

To troubleshoot the reverse SSH configuration on the client (remote device), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug ip ssh client**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip ssh client** **Example:** `Device# debug ip ssh client`	Displays debugging messages for the SSH client.

Troubleshooting Reverse SSH on the Server

To troubleshoot the reverse SSH configuration on the terminal server, perform the following steps. The steps may be configured in any order or independent of one another.

SUMMARY STEPS

1. **enable**
2. **debug ip ssh**
3. **show ssh**
4. **show line**

DETAILED STEPS

	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	**debug ip ssh** **Example:** `Device# debug ip ssh`	Displays debugging messages for the SSH server.
Step 3	**show ssh** **Example:** `Device# show ssh`	Displays the status of the SSH server connections.
Step 4	**show line** **Example:** `Device# show line`	Displays parameters of a terminal line.

Monitoring the SSH Configuration and Status

This table displays the SSH server configuration and status.

Table 15: 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.

Configuring Secure Copy

To configure a Cisco device for Secure Copy (SCP) server-side functionality, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** {**default** | *list-name*} *method1* [*method2...*]
5. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [*method1* [*method2...*]]
6. **username** *name* [**privilege** *level*] **password** *encryption-type encrypted-password*
7. **ip scp server enable**
8. **exit**

9. **debug ip scp**

DETAILED STEPS

	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`	Sets AAA authentication at login.
Step 4	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Device(config)# aaa authentication login default group tacacs+`	Enables the AAA access control system.
Step 5	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [*method1* [*method2...*]] **Example:** `Device(config)# aaa authorization exec default group tacacs+`	Sets parameters that restrict user access to a network. **Note** The **exec** keyword runs authorization to determine if the user is allowed to run an EXEC shell; therefore, you must use the **exec** keyword when you configure SCP.
Step 6	**username** *name* [**privilege** *level*] **password** *encryption-type encrypted-password* **Example:** `Device(config)# username superuser privilege 2 password 0 superpassword`	Establishes a username-based authentication system. **Note** You may omit this step if a network-based authentication mechanism, such as TACACS+ or RADIUS, has been configured.
Step 7	**ip scp server enable** **Example:** `Device(config)# ip scp server enable`	Enables SCP server-side functionality.
Step 8	**exit** **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# exit`	
Step 9	**debug ip scp** **Example:** `Device# debug ip scp`	(Optional) Troubleshoots SCP authentication problems.

Configuration Examples for Secure Shell

Example: Secure Copy Configuration Using Local Authentication

The following example shows how to configure the server-side functionality of Secure Copy (SCP). This example uses a locally defined username and password.

```
! AAA authentication and authorization must be configured properly in order for SCP to work.
aaa new-model
aaa authentication login default local
aaa authorization exec default local
username user1 privilege 15 password 0 lab
! SSH must be configured and functioning properly.
ip scp server enable
```

Example: SCP Server-Side Configuration Using Network-Based Authentication

The following example shows how to configure the server-side functionality of SCP using a network-based authentication mechanism:

```
! AAA authentication and authorization must be configured properly for SCP to work.
aaa new-model
aaa authentication login default group tacacs+
aaa authorization exec default group tacacs+
! SSH must be configured and functioning properly.
ip ssh time-out 120
ip ssh authentication-retries 3
ip scp server enable
```

Example Reverse SSH Console Access

Terminal Server Configuration

The following configuration example shows that reverse SSH has been configured for console access for terminal lines 1 through 3:

```
line 1 3
   no exec
   login authentication default
   transport input ssh
```

Client Configuration

The following commands configured on the SSH client will form the reverse SSH session with lines 1, 2, and 3, respectively:

```
ssh -l lab:1 router.example.com
ssh -l lab:2 router.example.com
ssh -l lab:3 router.example.com
```

Example Reverse SSH Modem Access

The following configuration example shows that dial-out lines 1 through 200 have been grouped under rotary group 1 for modem access:

```
line 1 200
   no exec
   login authentication default
   rotary 1
   transport input ssh
   exit
```

The following command shows that reverse SSH will connect to the first free line in the rotary group:

```
ssh -l lab:rotary1 router.example.com
```

Example: Monitoring the SSH Configuration and Status

To verify that the Secure Shell (SSH) server is enabled and to display the version and configuration data for your SSH connection, use the **show ip ssh** command. The following example shows that SSH is enabled:

```
Device# show ip ssh

SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
```

The following example shows that SSH is disabled:

```
Device# show ip ssh

%SSH has not been enabled
```

To verify the status of your SSH server connections, use the **show ssh** command. The following example shows the SSH server connections on the device when SSH is enabled:

```
Device# show ssh

Connection      Version     Encryption State Username
 0 1.5 3DES Session Started  guest
```

The following example shows that SSH is disabled:

```
Device# show ssh

%No SSH server connections running.
```

Additional References for Secure Shell

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

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

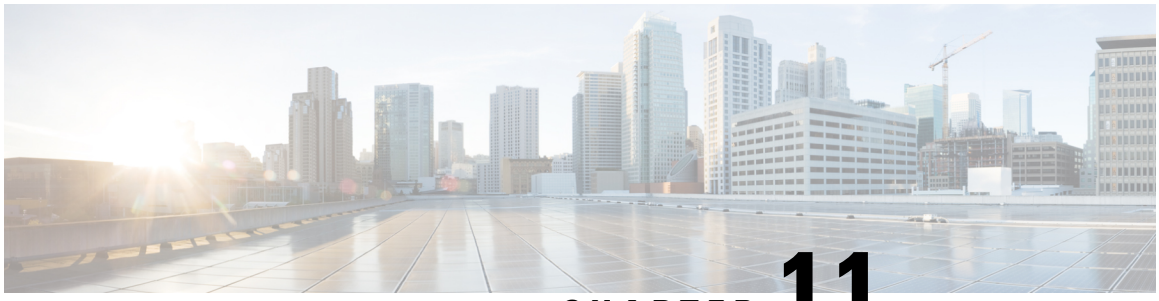
Feature History for Configuring Secure Shell

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 Release 15.2(5)E	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.

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 11

Secure Shell Version 2 Support

The Secure Shell Version 2 Support feature allows you to configure Secure Shell (SSH) Version 2. (SSH Version 1 support was implemented in an earlier Cisco software release.) SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. The only reliable transport that is defined for SSH is TCP. SSH provides a means to securely access and securely execute commands on another computer over a network. The Secure Copy Protocol (SCP) feature that is provided with SSH allows for the secure transfer of files.

Information About Secure Shell Version 2 Support

Secure Shell Version 2

The Secure Shell Version 2 Support feature allows you to configure SSH Version 2.

The configuration for the SSH Version 2 server is similar to the configuration for SSH Version 1. The **ip ssh version** command defines the SSH version to be configured. If you do not configure this command, SSH by default runs in compatibility mode; that is, both SSH Version 1 and SSH Version 2 connections are honored.

Note SSH Version 1 is a protocol that has never been defined in a standard. If you do not want your device to fall back to the undefined protocol (Version 1), you should use the **ip ssh version** command and specify Version 2.

The **ip ssh rsa keypair-name** command enables an SSH connection using the Rivest, Shamir, and Adleman (RSA) keys that you have configured. Previously, SSH was linked to the first RSA keys that were generated (that is, SSH was enabled when the first RSA key pair was generated). This behavior still exists, but by using the **ip ssh rsa keypair-name** command, you can overcome this behavior. If you configure the **ip ssh rsa keypair-name** command with a key pair name, SSH is enabled if the key pair exists or SSH will be enabled if the key pair is generated later. If you use this command to enable SSH, you are not forced to configure a hostname and a domain name, which was required in SSH Version 1 of the Cisco software.

Note The login banner is supported in SSH Version 2, but it is not supported in Secure Shell Version 1.

Secure Shell Version 2 Enhancements for RSA Keys

Cisco SSH Version 2 supports keyboard-interactive and password-based authentication methods. The SSH Version 2 Enhancements for RSA Keys feature also supports RSA-based public key authentication for the client and the server.

User authentication—RSA-based user authentication uses a private/public key pair associated with each user for authentication. The user must generate a private/public key pair on the client and configure a public key on the Cisco SSH server to complete the authentication.

An SSH user trying to establish credentials provides an encrypted signature using the private key. The signature and the user's public key are sent to the SSH server for authentication. The SSH server computes a hash over the public key provided by the user. The hash is used to determine if the server has a matching entry. If a match is found, an RSA-based message verification is performed using the public key. Hence, the user is authenticated or denied access based on the encrypted signature.

Server authentication—While establishing an SSH session, the Cisco SSH client authenticates the SSH server by using the server host keys available during the key exchange phase. SSH server keys are used to identify the SSH server. These keys are created at the time of enabling SSH and must be configured on the client.

For server authentication, the Cisco SSH client must assign a host key for each server. When the client tries to establish an SSH session with a server, the client receives the signature of the server as part of the key exchange message. If the strict host key checking flag is enabled on the client, the client checks if it has the host key entry corresponding to the server. If a match is found, the client tries to validate the signature by using the server host key. If the server is successfully authenticated, the session establishment continues; otherwise, it is terminated and displays a "Server Authentication Failed" message.

Note Storing public keys on a server uses memory; therefore, the number of public keys configurable on an SSH server is restricted to ten users, with a maximum of two public keys per user.

Note RSA-based user authentication is supported by the Cisco server, but Cisco clients cannot propose public key as an authentication method. If the Cisco server receives a request from an open SSH client for RSA-based authentication, the server accepts the authentication request.

Note For server authentication, configure the RSA public key of the server manually and configure the **ip ssh stricthostkeycheck** command on the Cisco SSH client.

SNMP Trap Generation

Depending on your release, Simple Network Management Protocol (SNMP) traps are generated automatically when an SSH session terminates if the traps have been enabled and SNMP debugging has been enabled. For information about enabling SNMP traps, see the "Configuring SNMP Support" module in the *SNMP Configuration Guide*.

Note When you configure the **snmp-server host** command, the IP address must be the address of the PC that has the SSH (telnet) client and that has IP connectivity to the SSH server.

You must also enable SNMP debugging using the **debug snmp packet** command to display the traps. The trap information includes information such as the number of bytes sent and the protocol that was used for the SSH session.

The following example shows that an SNMP trap is set. The trap notification is generated automatically when the SSH session terminates. In the example, a.b.c.d is the IP address of the SSH client.

```
snmp-server
snmp-server host a.b.c.d public tty
```

The following is sample output from the **debug snmp packet** command. The output provides SNMP trap information for an SSH session.

```
Switch# debug snmp packet

SNMP packet debugging is on
Device1# ssh -l lab 10.0.0.2
Password:

Switch# exit

[Connection to 10.0.0.2 closed by foreign host]
Device1#
*Jul 18 10:18:42.619: SNMP: Queuing packet to 10.0.0.2
*Jul 18 10:18:42.619: SNMP: V1 Trap, ent cisco, addr 10.0.0.1, gentrap 6, spectrap 1
local.9.3.1.1.2.1 = 6
tcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 4
ltcpConnEntry.5.10.0.0.1.22.10.0.0.2.55246 = 1015
ltcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 1056
ltcpConnEntry.2.10.0.0.1.22.10.0.0.2.55246 = 1392
local.9.2.1.18.2 = lab
*Jul 18 10:18:42.879: SNMP: Packet sent via UDP to 10.0.0.2

Switch#
```

SSH Keyboard Interactive Authentication

The SSH Keyboard Interactive Authentication feature, also known as Generic Message Authentication for SSH, is a method that can be used to implement different types of authentication mechanisms. Basically, any currently supported authentication method that requires only user input can be performed with this feature. The feature is automatically enabled.

The following methods are supported:

- Password

- SecurID and hardware tokens printing a number or a string in response to a challenge sent by the server
- Pluggable Authentication Module (PAM)
- S/KEY (and other One-Time-Pads)

Example: Enabling Client-Side Debugs

The following example shows that the client-side debugs are turned on, and the maximum number of prompts is six (three for the SSH keyboard interactive authentication method and three for the password authentication method).

```
Password:
Password:
Password:
Password:
Password:
Password: cisco123
Last login: Tue Dec 6 13:15:21 2005 from 10.76.248.213
user1@courier:~> exit
logout
[Connection to 10.76.248.200 closed by foreign host]
Device1# debug ip ssh client

SSH Client debugging is on

Device1# ssh -l lab 10.1.1.3

Password:
*Nov 17 12:50:53.199: SSH0: sent protocol version id SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.199: SSH CLIENT0: protocol version id is - SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.199: SSH CLIENT0: sent protocol version id SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.199: SSH CLIENT0: protocol version exchange successful
*Nov 17 12:50:53.203: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.335: SSH CLIENT0: key exchange successful and encryption on
*Nov 17 12:50:53.335: SSH2 CLIENT 0: using method keyboard-interactive
Password:
Password:
Password:
*Nov 17 12:51:01.887: SSH2 CLIENT 0: using method password authentication
Password:
Password: lab
Device2>

*Nov 17 12:51:11.407: SSH2 CLIENT 0: SSH2_MSG_USERAUTH_SUCCESS message received
*Nov 17 12:51:11.407: SSH CLIENT0: user authenticated
*Nov 17 12:51:11.407: SSH2 CLIENT 0: pty-req request sent
*Nov 17 12:51:11.411: SSH2 CLIENT 0: shell request sent
*Nov 17 12:51:11.411: SSH CLIENT0: session open
```

Example: Enabling ChPass with a Blank Password Change

In the following example, the ChPass feature is enabled, and a blank password change is accomplished using the SSH Keyboard Interactive Authentication method. A TACACS+ access control server (ACS) is used as the back-end AAA server.

```
Device1# ssh -l cisco 10.1.1.3

Password:
```

```
Old Password: cisco
New Password: cisco123
Re-enter New password: cisco123

Device2> exit

[Connection to 10.1.1.3 closed by foreign host]
```

Example: Enabling ChPass and Changing the Password on First Login

In the following example, the ChPass feature is enabled and TACACS+ ACS is used as the back-end server. The password is changed on the first login using the SSH keyboard interactive authentication method.

```
Device1# ssh -l cisco 10.1.1.3

Password: cisco
Your password has expired.
Enter a new one now.
New Password: cisco123
Re-enter New password: cisco123

Device2> exit

[Connection to 10.1.1.3 closed by foreign host]

Device1# ssh -l cisco 10.1.1.3

Password:cisco1
Your password has expired.
Enter a new one now.
New Password: cisco
Re-enter New password: cisco12
The New and Re-entered passwords have to be the same.
Try again.
New Password: cisco
Re-enter New password: cisco

Device2>
```

Example: Enabling ChPass and Expiring the Password After Three Logins

In the following example, the ChPass feature is enabled and TACACS+ ACS is used as the back-end AAA server. The password expires after three logins using the SSH keyboard interactive authentication method.

```
Device# ssh -l cisco. 10.1.1.3

Password: cisco

Device2> exit

[Connection to 10.1.1.3 closed by foreign host]

Device1# ssh -l cisco 10.1.1.3

Password: cisco

Device2> exit

Device1# ssh -l cisco 10.1.1.3
```

```
Password: cisco

Device2> exit

[Connection to 10.1.1.3 closed by foreign host]

Device1# ssh -l cisco 10.1.1.3

Password: cisco
Your password has expired.
Enter a new one now.
New Password: cisco123
Re-enter New password: cisco123

Device2>
```

How to Configure Secure Shell Version 2 Support

Configuring a Device for SSH Version 2 Using a Hostname and Domain Name

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **ip domain-name** *name*
5. **crypto key generate rsa**
6. **ip ssh** [**time-out** *seconds* | **authentication-retries** *integer*]
7. **ip ssh version** [**1** | **2**]
8. **exit**

DETAILED STEPS

	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** *name* **Example:** `Device(config)# hostname cisco7200`	Configures a hostname for your device.

	Command or Action	Purpose
Step 4	**ip domain-name** *name* **Example:** `cisco7200(config)# ip domain-name example.com`	Configures a domain name for your device.
Step 5	**crypto key generate rsa** **Example:** `cisco7200(config)# crypto key generate rsa`	Enables the SSH server for local and remote authentication.
Step 6	**ip ssh** [**time-out** *seconds* | **authentication-retries** *integer*] **Example:** `cisco7200(config)# ip ssh time-out 120`	(Optional) Configures SSH control variables on your device.
Step 7	**ip ssh version** [**1** | **2**] **Example:** `cisco7200(config)# ip ssh version 1`	(Optional) Specifies the version of SSH to be run on your device.
Step 8	**exit** **Example:** `cisco7200(config)# exit`	Exits global configuration mode and enters privileged EXEC mode. • Use **no hostname** command to return to the default host.

Configuring a Device for SSH Version 2 Using RSA Key Pairs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh rsa keypair-name** *keypair-name*
4. **crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size*
5. **ip ssh** [**time-out** *seconds* \| **authentication-retries** *integer*]
6. **ip ssh version 2**
7. **exit**

DETAILED STEPS

	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	**ip ssh rsa keypair-name** *keypair-name* **Example:** `Device(config)# ip ssh rsa keypair-name sshkeys`	Specifies the RSA key pair to be used for SSH. **Note** A Cisco device can have many RSA key pairs.
Step 4	**crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size* **Example:** `Device(config)# crypto key generate rsa usage-keys label sshkeys modulus 768`	Enables the SSH server for local and remote authentication on the device. • For SSH Version 2, the modulus size must be at least 768 bits. **Note** To delete the RSA key pair, use the **crypto key zeroize rsa** command. When you delete the RSA key pair, you automatically disable the SSH server.
Step 5	**ip ssh** [**time-out** *seconds* | **authentication-retries** *integer*] **Example:** `Device(config)# ip ssh time-out 12`	Configures SSH control variables on your device.
Step 6	**ip ssh version 2** **Example:** `Device(config)# ip ssh version 2`	Specifies the version of SSH to be run on the device.
Step 7	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

Configuring the Cisco SSH Server to Perform RSA-Based User Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **ip domain-name** *name*
5. **crypto key generate rsa**
6. **ip ssh pubkey-chain**

7. **username** *username*
8. **key-string**
9. **key-hash** *key-type key-name*
10. **end**

DETAILED STEPS

	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** *name* **Example:** `Device(config)# hostname host1`	Specifies the hostname.
Step 4	**ip domain-name** *name* **Example:** `host1(config)# ip domain-name name1`	Defines a default domain name that the Cisco software uses to complete unqualified hostnames.
Step 5	**crypto key generate rsa** **Example:** `host1(config)# crypto key generate rsa`	Generates RSA key pairs.
Step 6	**ip ssh pubkey-chain** **Example:** `host1(config)# ip ssh pubkey-chain`	Configures SSH-RSA keys for user and server authentication on the SSH server and enters public-key configuration mode. • The user authentication is successful if the RSA public key stored on the server is verified with the public or the private key pair stored on the client.
Step 7	**username** *username* **Example:** `host1(conf-ssh-pubkey)# username user1`	Configures the SSH username and enters public-key user configuration mode.
Step 8	**key-string** **Example:**	Specifies the RSA public key of the remote peer and enters public-key data configuration mode.

	Command or Action	Purpose
	`host1(conf-ssh-pubkey-user)# key-string`	**Note** You can obtain the public key value from an open SSH client; that is, from the .ssh/id_rsa.pub file.
Step 9	**key-hash** *key-type key-name* **Example:** `host1(conf-ssh-pubkey-data)# key-hash ssh-rsa key1`	(Optional) Specifies the SSH key type and version. • The key type must be ssh-rsa for the configuration of private public key pairs. • This step is optional only if the **key-string** command is configured. • You must configure either the **key-string** command or the **key-hash** command. **Note** You can use a hashing software to compute the hash of the public key string, or you can also copy the hash value from another Cisco device. Entering the public key data using the **key-string** command is the preferred way to enter the public key data for the first time.
Step 10	**end** **Example:** `host1(conf-ssh-pubkey-data)# end`	Exits public-key data configuration mode and returns to privileged EXEC mode. • Use **no hostname** command to return to the default host.

Configuring the Cisco IOS SSH Client to Perform RSA-Based Server Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **ip domain-name** *name*
5. **crypto key generate rsa**
6. **ip ssh pubkey-chain**
7. **server** *server-name*
8. **key-string**
9. **exit**
10. **key-hash** *key-type key-name*
11. **end**
12. **configure terminal**
13. **ip ssh stricthostkeycheck**

DETAILED STEPS

	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** *name* **Example:** `Device(config)# hostname host1`	Specifies the hostname.
Step 4	**ip domain-name** *name* **Example:** `host1(config)# ip domain-name name1`	Defines a default domain name that the Cisco software uses to complete unqualified hostnames.
Step 5	**crypto key generate rsa** **Example:** `host1(config)# crypto key generate rsa`	Generates RSA key pairs.
Step 6	**ip ssh pubkey-chain** **Example:** `host1(config)# ip ssh pubkey-chain`	Configures SSH-RSA keys for user and server authentication on the SSH server and enters public-key configuration mode.
Step 7	**server** *server-name* **Example:** `host1(conf-ssh-pubkey)# server server1`	Enables the SSH server for public-key authentication on the device and enters public-key server configuration mode.
Step 8	**key-string** **Example:** `host1(conf-ssh-pubkey-server)# key-string`	Specifies the RSA public-key of the remote peer and enters public key data configuration mode. **Note** You can obtain the public key value from an open SSH client; that is, from the .ssh/id_rsa.pub file.
Step 9	**exit** **Example:** `host1(conf-ssh-pubkey-data)# exit`	Exits public-key data configuration mode and enters public-key server configuration mode.

	Command or Action	Purpose
Step 10	**key-hash** *key-type key-name* **Example:** `host1(conf-ssh-pubkey-server)# key-hash ssh-rsa key1`	(Optional) Specifies the SSH key type and version. • The key type must be ssh-rsa for the configuration of private/public key pairs. • This step is optional only if the **key-string** command is configured. • You must configure either the **key-string** command or the **key-hash** command. **Note** You can use a hashing software to compute the hash of the public key string, or you can copy the hash value from another Cisco device. Entering the public key data using the **key-string** command is the preferred way to enter the public key data for the first time.
Step 11	**end** **Example:** `host1(conf-ssh-pubkey-server)# end`	Exits public-key server configuration mode and returns to privileged EXEC mode.
Step 12	**configure terminal** **Example:** `host1# configure terminal`	Enters global configuration mode.
Step 13	**ip ssh stricthostkeycheck** **Example:** `host1(config)# ip ssh stricthostkeycheck`	Ensures that server authentication takes place. • The connection is terminated in case of a failure. • Use **no hostname** command to return to the default host.

Starting an Encrypted Session with a Remote Device

Note The device with which you want to connect must support a Secure Shell (SSH) server that has an encryption algorithm that is supported in Cisco software. Also, you need not enable your device. SSH can be run in disabled mode.

SUMMARY STEPS

1. **ssh** [**-v** {**1** | **2**} | **-c** {**aes128-ctr** | **aes192-ctr** | **aes256-ctr** | **aes128-cbc** | **3des** | **aes192-cbc** | **aes256-cbc**} | **-l** *user-id* | **-l** *user-id***:***vrf-name number ip-address ip-address* | **-l** *user-id***:***rotary number ip-address*

| **-m** {**hmac-md5-128** | **hmac-md5-96** | **hmac-sha1-160** | **hmac-sha1-96**} | **-o numberofpasswordprompts** *n* | **-p** *port-num*] {*ip-addr* | *hostname*} [**command** | **-vrf**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ssh** [**-v** {**1**	**2**}	**-c** {**aes128-ctr**	**aes192-ctr**	**aes256-ctr**	**aes128-cbc**	**3des**	**aes192-cbc**	**aes256-cbc**}	**-l** *user-id*	**-l** *user-id***:***vrf-name number ip-address ip-address*	**-l** *user-id***:***rotary number ip-address*	**-m** {**hmac-md5-128**	**hmac-md5-96**	**hmac-sha1-160**	**hmac-sha1-96**}	**-o numberofpasswordprompts** *n*	**-p** *port-num*] {*ip-addr*	*hostname*} [**command**	**-vrf**] **Example:** `Device# ssh -v 2 -c aes256-ctr -m hmac-sha1-96 -l user2 10.76.82.24`	Starts an encrypted session with a remote networking device.

Enabling Secure Copy Protocol on the SSH Server

Note The following task configures the server-side functionality for SCP. This task shows a typical configuration that allows the device to securely copy files from a remote workstation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login default local**
5. **aaa authorization exec defaultlocal**
6. **username***name* **privilege** *privilege-level* **password** *password*
7. **ip ssh time-out***seconds*
8. **ip ssh authentication-retries** *integer*
9. **ip scpserverenable**
10. **exit**
11. **debug ip scp**

DETAILED STEPS

	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	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa authentication login default local** **Example:** `Device(config)# aaa authentication login default local`	Sets AAA authentication at login to use the local username database for authentication.
Step 5	**aaa authorization exec defaultlocal** **Example:** `Device(config)# aaa authorization exec default local`	Sets the parameters that restrict user access to a network, runs the authorization to determine if the user ID is allowed to run an EXEC shell, and specifies that the system must use the local database for authorization.
Step 6	**username***name* **privilege** *privilege-level* **password** *password* **Example:** `Device(config)# username samplename privilege 15 password password1`	Establishes a username-based authentication system, and specifies the username, privilege level, and an unencrypted password. **Note** The minimum value for the *privilege-level* argument is 15. A privilege level of less than 15 results in the connection closing.
Step 7	**ip ssh time-out***seconds* **Example:** `Device(config)# ip ssh time-out 120`	Sets the time interval (in seconds) that the device waits for the SSH client to respond.
Step 8	**ip ssh authentication-retries** *integer* **Example:** `Device(config)# ip ssh authentication-retries 3`	Sets the number of authentication attempts after which the interface is reset.
Step 9	**ip scpserverenable** **Example:** `Device(config)# ip scp server enable`	Enables the device to securely copy files from a remote workstation.
Step 10	**exit** **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# exit`	
Step 11	**debug ip scp** **Example:** `Device# debug ip scp`	(Optional) Provides diagnostic information about SCP authentication problems.

Verifying the Status of the Secure Shell Connection

SUMMARY STEPS

1. **enable**
2. **show ssh**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ssh** **Example:** `Device# show ssh`	Displays the status of SSH server connections.
Step 3	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode and returns to user EXEC mode.

The following sample output from the **show ssh** command displays status of various SSH Version 1 and Version 2 connections for Version 1 and Version 2 connections:

```
------------------------------------------------------------------------
Device# show ssh

Connection      Version Encryption      State                   Username
 0              1.5     3DES            Session started         lab
Connection Version Mode Encryption  Hmac                State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
------------------------------------------------------------------------
```

The following sample output from the **show ssh** command displays status of various SSH Version 1 and Version 2 connections for a Version 2 connection with no Version 1 connection:

```
------------------------------------------------------------------------
Device# show ssh

Connection Version Mode Encryption  Hmac                State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
%No SSHv1 server connections running.
------------------------------------------------------------------------
```

The following sample output from the **show ssh** command displays status of various SSH Version 1 and Version 2 connections for a Version 1 connection with no Version 2 connection:

```
------------------------------------------------------------------------
Device# show ssh

Connection      Version Encryption      State                   Username
 0              1.5     3DES            Session started         lab
%No SSHv2 server connections running.
------------------------------------------------------------------------
```

Verifying the Secure Shell Status

SUMMARY STEPS

1. **enable**
2. **show ip ssh**
3. **exit**

DETAILED STEPS

	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`	Displays the version and configuration data for SSH.
Step 3	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode and returns to user EXEC mode.

Examples

The following sample output from the **show ip ssh** command displays the version of SSH that is enabled, the authentication timeout values, and the number of authentication retries for Version 1 and Version 2 connections:

```
------------------------------------------------------------------------
Device# show ip ssh

SSH Enabled - version 1.99
Authentication timeout: 120 secs; Authentication retries: 3
------------------------------------------------------------------------
```

The following sample output from the **show ip ssh** command displays the version of SSH that is enabled, the authentication timeout values, and the number of authentication retries for a Version 2 connection with no Version 1 connection:

```
------------------------------------------------------------------------
Device# show ip ssh

SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
------------------------------------------------------------------------
```

The following sample output from the **show ip ssh** command displays the version of SSH that is enabled, the authentication timeout values, and the number of authentication retries for a Version 1 connection with no Version 2 connection:

```
------------------------------------------------------------------------
Device# show ip ssh

3d06h: %SYS-5-CONFIG_I: Configured from console by console
SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
------------------------------------------------------------------------
```

Monitoring and Maintaining Secure Shell Version 2

SUMMARY STEPS

1. **enable**
2. **debug ip ssh**
3. **debug snmp packet**

DETAILED STEPS

	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	**debug ip ssh** **Example:** `Device# debug ip ssh`	Enables debugging of SSH.
Step 3	**debug snmp packet** **Example:** `Device# debug snmp packet`	Enables debugging of every SNMP packet sent or received by the device.

Example

The following sample output from the **debug ip ssh** command shows the connection is an SSH Version 2 connection:

```
Device# debug ip ssh

00:33:55: SSH1: starting SSH control process
00:33:55: SSH1: sent protocol version id SSH-1.99-Cisco-1.25
00:33:55: SSH1: protocol version id is - SSH-2.0-OpenSSH_2.5.2p2
00:33:55: SSH2 1: send: len 280 (includes padlen 4)
00:33:55: SSH2 1: SSH2_MSG_KEXINIT sent
00:33:55: SSH2 1: ssh_receive: 536 bytes received
00:33:55: SSH2 1: input: packet len 632
00:33:55: SSH2 1: partial packet 8, need 624, maclen 0
00:33:55: SSH2 1: ssh_receive: 96 bytes received
00:33:55: SSH2 1: partial packet 8, need 624, maclen 0
00:33:55: SSH2 1: input: padlen 11
00:33:55: SSH2 1: received packet type 20
00:33:55: SSH2 1: SSH2_MSG_KEXINIT received
00:33:55: SSH2: kex: client->server aes128-cbc hmac-md5 none
00:33:55: SSH2: kex: server->client aes128-cbc hmac-md5 none
00:33:55: SSH2 1: expecting SSH2_MSG_KEXDH_INIT
00:33:55: SSH2 1: ssh_receive: 144 bytes received
00:33:55: SSH2 1: input: packet len 144
00:33:55: SSH2 1: partial packet 8, need 136, maclen 0
00:33:55: SSH2 1: input: padlen 5
00:33:55: SSH2 1: received packet type 30
00:33:55: SSH2 1: SSH2_MSG_KEXDH_INIT received
00:33:55: SSH2 1: signature length 111
00:33:55: SSH2 1: send: len 384 (includes padlen 7)
00:33:55: SSH2: kex_derive_keys complete
00:33:55: SSH2 1: send: len 16 (includes padlen 10)
00:33:55: SSH2 1: newkeys: mode 1
00:33:55: SSH2 1: SSH2_MSG_NEWKEYS sent
00:33:55: SSH2 1: waiting for SSH2_MSG_NEWKEYS
00:33:55: SSH2 1: ssh_receive: 16 bytes received
00:33:55: SSH2 1: input: packet len 16
00:33:55: SSH2 1: partial packet 8, need 8, maclen 0
00:33:55: SSH2 1: input: padlen 10
00:33:55: SSH2 1: newkeys: mode 0
00:33:55: SSH2 1: received packet type 2100:33:55: SSH2 1: SSH2_MSG_NEWKEYS received
00:33:56: SSH2 1: ssh_receive: 48 bytes received
00:33:56: SSH2 1: input: packet len 32
00:33:56: SSH2 1: partial packet 16, need 16, maclen 16
```

```
00:33:56: SSH2 1: MAC #3 ok
00:33:56: SSH2 1: input: padlen 10
00:33:56: SSH2 1: received packet type 5
00:33:56: SSH2 1: send: len 32 (includes padlen 10)
00:33:56: SSH2 1: done calc MAC out #3
00:33:56: SSH2 1: ssh_receive: 64 bytes received
00:33:56: SSH2 1: input: packet len 48
00:33:56: SSH2 1: partial packet 16, need 32, maclen 16
00:33:56: SSH2 1: MAC #4 ok
00:33:56: SSH2 1: input: padlen 9
00:33:56: SSH2 1: received packet type 50
00:33:56: SSH2 1: send: len 32 (includes padlen 13)
00:33:56: SSH2 1: done calc MAC out #4
00:34:04: SSH2 1: ssh_receive: 160 bytes received
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #5 ok
00:34:04: SSH2 1: input: padlen 13
00:34:04: SSH2 1: received packet type 50
00:34:04: SSH2 1: send: len 16 (includes padlen 10)
00:34:04: SSH2 1: done calc MAC out #5
00:34:04: SSH2 1: authentication successful for lab
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #6 ok
00:34:04: SSH2 1: input: padlen 6
00:34:04: SSH2 1: received packet type 2
00:34:04: SSH2 1: ssh_receive: 64 bytes received
00:34:04: SSH2 1: input: packet len 48
00:34:04: SSH2 1: partial packet 16, need 32, maclen 16
00:34:04: SSH2 1: MAC #7 ok
00:34:04: SSH2 1: input: padlen 19
00:34:04: SSH2 1: received packet type 90
00:34:04: SSH2 1: channel open request
00:34:04: SSH2 1: send: len 32 (includes padlen 10)
00:34:04: SSH2 1: done calc MAC out #6
00:34:04: SSH2 1: ssh_receive: 192 bytes received
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #8 ok
00:34:04: SSH2 1: input: padlen 13
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: pty-req request
00:34:04: SSH2 1: setting TTY - requested: height 24, width 80; set: height 24,
width 80
00:34:04: SSH2 1: input: packet len 96
00:34:04: SSH2 1: partial packet 16, need 80, maclen 16
00:34:04: SSH2 1: MAC #9 ok
00:34:04: SSH2 1: input: padlen 11
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: x11-req request
00:34:04: SSH2 1: ssh_receive: 48 bytes received
00:34:04: SSH2 1: input: packet len 32
00:34:04: SSH2 1: partial packet 16, need 16, maclen 16
00:34:04: SSH2 1: MAC #10 ok
00:34:04: SSH2 1: input: padlen 12
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: shell request
00:34:04: SSH2 1: shell message received
00:34:04: SSH2 1: starting shell for vty
00:34:04: SSH2 1: send: len 48 (includes padlen 18)
00:34:04: SSH2 1: done calc MAC out #7
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
```

```
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #11 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #8
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #12 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #9
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #13 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #10
00:34:08: SSH2 1: ssh_receive: 48 bytes received
00:34:08: SSH2 1: input: packet len 32
00:34:08: SSH2 1: partial packet 16, need 16, maclen 16
00:34:08: SSH2 1: MAC #14 ok
00:34:08: SSH2 1: input: padlen 17
00:34:08: SSH2 1: received packet type 94
00:34:08: SSH2 1: send: len 32 (includes padlen 17)
00:34:08: SSH2 1: done calc MAC out #11
00:34:08: SSH2 1: ssh_receive: 48 bytes received
00:34:08: SSH2 1: input: packet len 32
00:34:08: SSH2 1: partial packet 16, need 16, maclen 16
00:34:08: SSH2 1: MAC #15 ok
00:34:08: SSH2 1: input: padlen 17
00:34:08: SSH2 1: received packet type 94
00:34:08: SSH2 1: send: len 32 (includes padlen 16)
00:34:08: SSH2 1: done calc MAC out #12
00:34:08: SSH2 1: send: len 48 (includes padlen 18)
00:34:08: SSH2 1: done calc MAC out #13
00:34:08: SSH2 1: send: len 16 (includes padlen 6)
00:34:08: SSH2 1: done calc MAC out #14
00:34:08: SSH2 1: send: len 16 (includes padlen 6)
00:34:08: SSH2 1: done calc MAC out #15
00:34:08: SSH1: Session terminated normally
```

Configuration Examples for Secure Shell Version 2 Support

Example: Configuring Secure Shell Version 2

```
Device# configure terminal
Device(config)# ip ssh version 2
```

Example: Starting an Encrypted Session with a Remote Device

```
Device# ssh -v 2 -c aes256-cbc -m hmac-sha1-160 -l shaship 10.76.82.24
```

Example: Configuring Server-Side SCP

The following example shows how to configure the server-side functionality for SCP. This example also configures AAA authentication and authorization on the device. This example uses a locally defined username and password.

```
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login default local
Device(config)# aaa authorization exec default local
Device(config)# username samplename privilege 15 password password1
Device(config)# ip ssh time-out 120
Device(config)# ip ssh authentication-retries 3
Device(config)# ip scp server enable
```

Example: Setting an SNMP Trap

The following example shows that an SNMP trap is set. The trap notification is generated automatically when the SSH session terminates. In the example, a.b.c.d is the IP address of the SSH client.

```
snmp-server
snmp-server host a.b.c.d public tty
```

The following is sample output from the **debug snmp packet** command. The output provides SNMP trap information for an SSH session.

```
Device1# debug snmp packet

SNMP packet debugging is on
Device1# ssh -l lab 10.0.0.2
Password:

Device2# exit

[Connection to 10.0.0.2 closed by foreign host]
Device1#
*Jul 18 10:18:42.619: SNMP: Queuing packet to 10.0.0.2
*Jul 18 10:18:42.619: SNMP: V1 Trap, ent cisco, addr 10.0.0.1, gentrap 6, spectrap 1
local.9.3.1.1.2.1 = 6
tcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 4
ltcpConnEntry.5.10.0.0.1.22.10.0.0.2.55246 = 1015
ltcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 1056
ltcpConnEntry.2.10.0.0.1.22.10.0.0.2.55246 = 1392
local.9.2.1.18.2 = lab
*Jul 18 10:18:42.879: SNMP: Packet sent via UDP to 10.0.0.2

Device1#
```

Examples: SSH Keyboard Interactive Authentication

Example: SNMP Debugging

The following is sample output from the **debug snmp packet** command. The output provides SNMP trap information for an SSH session.

```
Device1# debug snmp packet

SNMP packet debugging is on
Device1# ssh -l lab 10.0.0.2
Password:

Device2# exit

[Connection to 10.0.0.2 closed by foreign host]
Device1#
*Jul 18 10:18:42.619: SNMP: Queuing packet to 10.0.0.2
*Jul 18 10:18:42.619: SNMP: V1 Trap, ent cisco, addr 10.0.0.1, gentrap 6, spectrap 1
local.9.3.1.1.2.1 = 6
tcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 4
ltcpConnEntry.5.10.0.0.1.22.10.0.0.2.55246 = 1015
ltcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 1056
ltcpConnEntry.2.10.0.0.1.22.10.0.0.2.55246 = 1392
local.9.2.1.18.2 = lab
*Jul 18 10:18:42.879: SNMP: Packet sent via UDP to 10.0.0.2

Device1#
```

Examples: SSH Debugging Enhancements

The following is sample output from the **debug ip ssh detail** command. The output provides debugging information about the SSH protocol and channel requests.

```
Device# debug ip ssh detail

00:04:22: SSH0: starting SSH control process
00:04:22: SSH0: sent protocol version id SSH-1.99-Cisco-1.25
00:04:22: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
00:04:22: SSH2 0: SSH2_MSG_KEXINIT sent
00:04:22: SSH2 0: SSH2_MSG_KEXINIT received
00:04:22: SSH2:kex: client->server enc:aes128-cbc mac:hmac-sha1
00:04:22: SSH2:kex: server->client enc:aes128-cbc mac:hmac-sha1
00:04:22: SSH2 0: expecting SSH2_MSG_KEXDH_INIT
00:04:22: SSH2 0: SSH2_MSG_KEXDH_INIT received
00:04:22: SSH2: kex_derive_keys complete
00:04:22: SSH2 0: SSH2_MSG_NEWKEYS sent
00:04:22: SSH2 0: waiting for SSH2_MSG_NEWKEYS
00:04:22: SSH2 0: SSH2_MSG_NEWKEYS received
00:04:24: SSH2 0: authentication successful for lab
00:04:24: SSH2 0: channel open request
00:04:24: SSH2 0: pty-req request
00:04:24: SSH2 0: setting TTY - requested: height 24, width 80; set: height 24, width 80
00:04:24: SSH2 0: shell request
00:04:24: SSH2 0: shell message received
00:04:24: SSH2 0: starting shell for vty
00:04:38: SSH0: Session terminated normally
```

The following is sample output from the **debug ip ssh packet** command. The output provides debugging information about the SSH packet.

```
Device# debug ip ssh packet

00:05:43: SSH2 0: send:packet of  length 280 (length also includes padlen of 4)
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: input: total packet length of 280 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 24 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 4 bytes
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: input: total packet length of 144 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 16 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 6 bytes
00:05:43: SSH2 0: signature length 143
00:05:43: SSH2 0: send:packet of  length 448 (length also includes padlen of 7)
00:05:43: SSH2 0: send:packet of  length 16 (length also includes padlen of 10)
00:05:43: SSH2 0: newkeys: mode 1
00:05:43: SSH2 0: ssh_receive: 16 bytes received
00:05:43: SSH2 0: input: total packet length of 16 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 8 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 10 bytes
00:05:43: SSH2 0: newkeys: mode 0
00:05:43: SSH2 0: ssh_receive: 52 bytes received
00:05:43: SSH2 0: input: total packet length of 32 bytes
00:05:43: SSH2 0: partial packet length(block size)16 bytes,needed 16 bytes, maclen 20
00:05:43: SSH2 0: MAC compared for #3 :ok
```

Additional References for Secure Shell Version 2 Support

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

Standards

Standards	Title
IETF Secure Shell Version 2 Draft Standards	Internet Engineering Task Force website

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

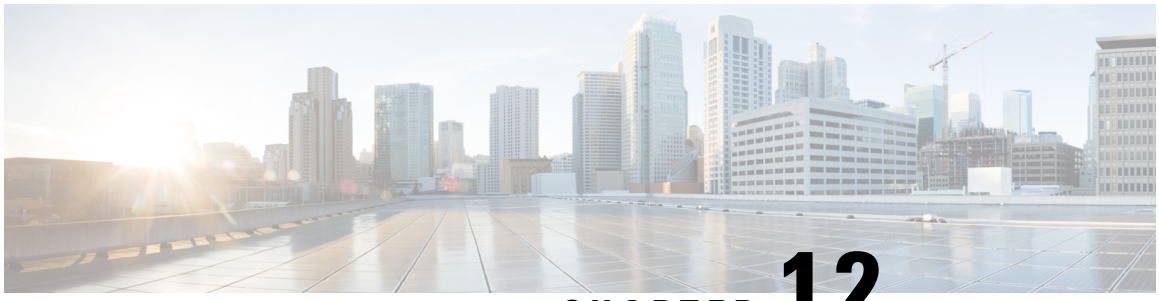
Feature History for Secure Shell Version 2 Support

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 Release 15.2(5)E	Secure Shell Version 2 Support	The Secure Shell Version 2 Support feature allows you to configure Secure Shell (SSH) Version 2 (SSH Version 1 support was implemented in an earlier Cisco IOS software release). SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. SSH version 2 also supports AES counter-based encryption 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 12

Configuring SSH File Transfer Protocol

Secure Shell (SSH) includes support for SSH File Transfer Protocol (SFTP), which is a new standard file transfer protocol introduced in SSHv2. This feature provides a secure and authenticated method for copying device configuration or device image files.

Prerequisites for SSH File Transfer Protocol

- SSH must be enabled.
- The **ip ssh source-interface** *interface-type interface-number* command must be configured.

Restrictions for SSH File Transfer Protocol

- The SFTP server is not supported.
- SFTP boot is not supported.
- The **sftp** option in the **install add** command is not supported.

Information About SSH File Transfer Protocol

The SFTP client functionality is provided as part of the SSH component and is always enabled on the corresponding device. Therefore, any SFTP server user with the appropriate permission can copy files to and from the device.

An SFTP client is VRF-aware; you can configure the secure FTP client to use the virtual routing and forwarding (VRF) associated with a particular source interface during connection attempts.

How to Configure SSH File Transfer Protocol

The following sections provide information about the various tasks that comprise an SFTP configuration.

Configuring SFTP

Perform the following steps:

Before you begin

To configure a Cisco device for SFTP client-side functionality, the **ip ssh source-interface** *interface-type interface-number* command must be configured first.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh source-interface** *interface-type interface-number*
4. **exit**
5. **show running-config**
6. **debug ip sftp**

DETAILED STEPS

	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 source-interface** *interface-type interface-number* **Example:** `Device(config)# ip ssh source-interface gigabitethernet 2/1`	Defines the source IP for the SSH session.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**show running-config** **Example:** `Device# show running-config`	(Optional) Displays the SFTP client-side functionality.
Step 6	**debug ip sftp** **Example:** `Device# debug ip sftp`	(Optional) Enables SFTP debugging.

Perform an SFTP Copy Operation

SFTP copy takes the IP or hostname of the corresponding server if Domain Name System (DNS) is configured. To perform SFTP copy operations, use the following commands in privileged EXEC mode:

Command	Purpose
Device# **copy ios-file-system:file sftp://user:pwd@server-ip//filepath** Or Device# **copy ios-file-system: sftp:**	Copies a file from the local Cisco IOS file system to the server. Specify the username, password, IP address, and filepath of the server.
Device# **copy sftp://user:pwd@server-ip //filepath ios-file-system:file** Or Device# **copy sftp: ios-file-system:**	Copies the file from the server to the local Cisco IOS file system. Specify the username, password, IP address, and filepath of the server.

Example: Configuring SSH File Transfer Protocol

The following example shows how to configure the client-side functionality of SFTP:

```
Device> enable
Device# configure terminal
Device(config)# ip ssh source-interface gigabitethernet 0/1
Device(config)# exit
```

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)
Secure Shell Version 1 and 2 Support	*Configuring Secure Shell*

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

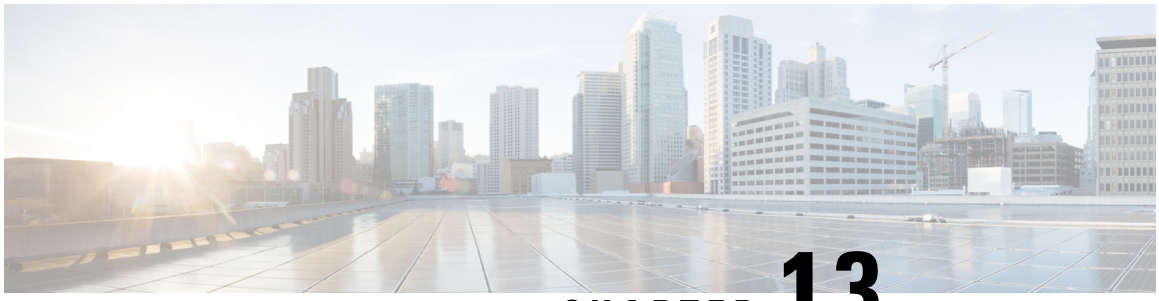
Feature History for SSH File Transfer Protocol

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 Release 15.2(5)E	SSH File Transfer Protocol	SSH includes support for SFTP, a new standard file transfer protocol introduced in SSHv2.

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

X.509v3 Certificates for SSH Authentication

The X.509v3 Certificates for SSH Authentication feature uses public key algorithm (PKI) for server and user authentication, and allows the Secure Shell (SSH) protocol to verify the identity of the owner of a key pair via digital certificates, signed and issued by a Certificate Authority (CA).

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 replaces the **ip ssh server authenticate user** command with the **ip ssh server algorithm authentication** command. Configure the **default ip ssh server authenticate user** command to remove the **ip ssh server authenticate user** command from the configuration. The IOS secure shell (SSH) server will start using the **ip ssh server algorithm authentication** command.

When you configure the **ip ssh server authenticate user** command, the following 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 the **default ip ssh server authenticate user** to make the CLI ineffective.

Restrictions for X.509v3 Certificates for SSH Authentication

- The X.509v3 Certificates for SSH Authentication feature implementation is applicable only on the Cisco IOS Secure Shell (SSH) server side.

- The Cisco IOS SSH server supports only the x509v3-ssh-rsa algorithm-based certificate for server and user authentication.

Information About X.509v3 Certificates for SSH Authentication

X.509v3 Certificates for SSH Authentication Overview

The Secure Shell (SSH) protocol provides a secure remote access connection to network devices. The communication between the client and server is encrypted.

There are two SSH protocols that use public key cryptography for authentication. The Transport Layer Protocol, uses a digital signature algorithm (called the public key algorithm) to authenticate the server to the client. And the User Authentication Protocol uses a digital signature to authenticate (public key authentication) the client to the server.

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, such as those in X.509 Version 3 (X.509v3), are used to provide identity management. X.509v3 uses a chain of signatures by a trusted root certification authority and intermediate certificate authorities to bind a public signing key to a specific digital identity. This implementation allows the use of a public key algorithm for server and user authentication, and allows SSH to verify the identity of the owner of a key pair via digital certificates, signed and issued by a Certificate Authority (CA).

Server and User Authentication Using X.509v3

For server authentication, the 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 IOS 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 IOS SSH server end.

OCSP Response Stapling

The Online Certificate Status Protocol (OCSP) enables applications to determine the (revocation) state of an identified certificate. This protocol specifies the data that needs to be exchanged between an application checking the status of a certificate and the server providing that status. An OCSP client issues a status request to an OCSP responder and suspends acceptance of the certificate until a response is received. An OCSP response at a minimum consists of a responseStatus field that indicates the processing status of the a request.

For the public key algorithms, the key format consists of a sequence of one or more X.509v3 certificates followed by a sequence of zero or more OCSP responses.

The X.509v3 Certificate for SSH Authentication feature uses OCSP Response Stapling. By using OCSP response stapling, a device obtains the revocation information of its own certificate by contacting the OCSP server and then stapling the result along with its certificates and sending the information to the peer rather than having the peer contact the OCSP responder.

How to Configure X.509v3 Certificates for SSH Authentication

Configuring Digital Certificates for Server Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh server algorithm hostkey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]}
4. **ip ssh server certificate profile**
5. **server**
6. **trustpoint sign** *PKI-trustpoint-name*
7. **ocsp-response include**
8. **end**
9. **line vty line_number** [*ending_line_number*]
10. **transport input ssh**

DETAILED STEPS

	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 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: • **x509v3-ssh-rsa**: certificate-based authentication • **ssh-rsa**: public key-based authentication
Step 4	**ip ssh server certificate profile** **Example:** `Device(config)# ip ssh server certificate profile`	Configures server and user certificate profiles and enters SSH certificate profile configuration mode.

	Command or Action	Purpose
Step 5	**server** **Example:** `Device(ssh-server-cert-profile)# server`	Configures server certificate profile and enters SSH server certificate profile server configuration mode. • The server profile is used to send out the certificate of the server to the SSH client during server authentication.
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, 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.
Step 9	**line vty line_number** [*ending_line_number*] **Example:** `Device(config)# line vty line_number [ending_line_number]`	Enters line configuration mode to configure the virtual terminal line settings. For line_number and ending_line_number, specify a pair of lines. The range is 0 to 15.
Step 10	**transport input ssh** **Example:** `Device(config-line)#transport input ssh`	Specifies that the device prevent non-SSH Telnet connections. This limits the router to only SSH connections.

Configuring Digital Certificates for User Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh server algorithm authentication** {**publickey** | **keyboard** | **password**}
4. **ip ssh server algorithm publickey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]}
5. **ip ssh server certificate profile**
6. **user**
7. **trustpoint verify** *PKI-trustpoint-name*
8. **ocsp-response required**

9. **end**
10. **line vty line_number** [*ending_line_number*]
11. **transport input ssh**

DETAILED STEPS

	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 IOS SSH server must have at least one configured user authentication algorithm. • To use the certificate method for user authentication, the **publickey** keyword must be configured.
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 IOS SSH client must have at least one configured public key algorithm: • **x509v3-ssh-rsa**: Certificate-based authentication • **ssh-rsa**: Public-key-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 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.
Step 10	**line vty line_number** [*ending_line_number*] **Example:** `Device(config)# line vty line_number [ending_line_number]`	Enters line configuration mode to configure the virtual terminal line settings. For line_number and ending_line_number, specify a pair of lines. The range is 0 to 15.
Step 11	**transport input ssh** **Example:** `Device(config-line)#transport input ssh`	Specifies that the device prevent non-SSH Telnet connections. This limits the router to only SSH connections.

Verifying the Server and User Authentication Using Digital Certificates

SUMMARY STEPS

1. **enable**
2. **show ip ssh**
3. **debug ip ssh detail**
4. **show log**
5. **debug ip packet**
6. **show log**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show ip ssh**

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.

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 : 1024 bits
```

Step 3 **debug ip ssh detail**

Turns on debugging messages for SSH details.

Example:

```
Device# debug ip ssh detail

ssh detail messages debugging is on
```

Step 4 **show log**

Shows the debug message log.

Example:

```
Device# show log

Syslog logging: enabled (0 messages dropped, 9 messages rate-limited, 0 flushes, 0 overruns, xml
disabled, filtering disabled)

No Active Message Discriminator.



No Inactive Message Discriminator.


    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 233 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    File logging: disabled
    Persistent logging: disabled
```

```
No active filter modules.

    Trap logging: level informational, 174 message lines logged
        Logging Source-Interface:       VRF Name:

Log Buffer (4096 bytes):
5 IST: SSH2 CLIENT 0: SSH2_MSG_KEXINIT sent
*Sep  6 14:44:08.496 IST: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
*Sep  6 14:44:08.496 IST: SSH2 0: kexinit sent: kex algo =
diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1
*Sep  6 14:44:08.496 IST: SSH2 0: Server certificate trustpoint not found. Skipping hostkey algo =
x509v3-ssh-rsa
*Sep  6 14:44:08.496 IST: SSH2 0: kexinit sent: hostkey algo = ssh-rsa
*Sep  6 14:44:08.496 IST: SSH2 0: kexinit sent: encryption algo = aes128-ctr,aes192-ctr,aes256-ctr
*Sep  6 14:44:08.496 IST: SSH2 0: kexinit sent: mac algo =
hmac-sha2-256,hmac-sha2-512,hmac-sha1,hmac-sha1-96
*Sep  6 14:44:08.496 IST: SSH2 0: SSH2_MSG_KEXINIT sent
*Sep  6 14:44:08.496 IST: SSH2 0: SSH2_MSG_KEXINIT received
*Sep  6 14:44:08.496 IST: SSH2 0: kex: client->server enc:aes128-ctr mac:hmac-sha2-256
*Sep  6 14:44:08.496 IST: SSH2 0: kex: server->client enc:aes128-ctr mac:hmac-sha2-256
*Sep  6 14:44:08.496 IST: SSH2 0: Using hostkey algo = ssh-rsa
*Sep  6 14:44:08.496 IST: SSH2 0: Using kex_algo = diffie-hellman-group-exchange-sha1
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: SSH2_MSG_KEXINIT received
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: kex: server->client enc:aes128-ctr mac:hmac-sha2-256
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: kex: client->server enc:aes128-ctr mac:hmac-sha2-256
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: Using hostkey algo = ssh-rsa
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: Using kex_algo = diffie-hellman-group-exchange-sha1
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: SSH2_MSG_KEX_DH_GEX_REQUEST sent
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: Range sent- 2048  < 2048  < 4096
*Sep  6 14:44:08.497 IST: SSH2 0: SSH2_MSG_KEX_DH_GEX_REQUEST received
*Sep  6 14:44:08.497 IST: SSH2 0: Range sent by client is - 2048 < 2048 < 4096
*Sep  6 14:44:08.497 IST: SSH2 0:  Modulus size established : 2048 bits
*Sep  6 14:44:08.510 IST: SSH2 0: expecting SSH2_MSG_KEX_DH_GEX_INIT
*Sep  6 14:44:08.510 IST: SSH2 CLIENT 0: SSH2_MSG_KEX_DH_GEX_GROUP received
*Sep  6 14:44:08.510 IST: SSH2 CLIENT 0: Server has chosen 2048 -bit dh keys
*Sep  6 14:44:08.523 IST: SSH2 CLIENT 0: expecting SSH2_MSG_KEX_DH_GEX_REPLY
*Sep  6 14:44:08.524 IST: SSH2 0: SSH2_MSG_KEXDH_INIT received
*Sep  6 14:44:08.555 IST: SSH2: kex_derive_keys complete
*Sep  6 14:44:08.555 IST: SSH2 0: SSH2_MSG_NEWKEYS sent
*Sep  6 14:44:08.555 IST: SSH2 0: waiting for SSH2_MSG_NEWKEYS
*Sep  6 14:44:08.555 IST: SSH2 CLIENT 0: SSH2_MSG_KEX_DH_GEX_REPLY received
*Sep  6 14:44:08.555 IST: SSH2 CLIENT 0: Skipping ServerHostKey Validation
*Sep  6 14:44:08.571 IST: SSH2 CLIENT 0: signature length 271
*Sep  6 14:44:08.571 IST: SSH2: kex_derive_keys complete
*Sep  6 14:44:08.571 IST: SSH2 CLIENT 0: SSH2_MSG_NEWKEYS sent
*Sep  6 14:44:08.571 IST: SSH2 CLIENT 0: waiting for SSH2_MSG_NEWKEYS
*Sep  6 14:44:08.571 IST: SSH2 CLIENT 0: SSH2_MSG_NEWKEYS received
*Sep  6 14:44:08.571 IST: SSH2 0: SSH2_MSG_NEWKEYS received
*Sep  6 14:44:08.571 IST: SSH2 0: Authentications that can continue =
publickey,keyboard-interactive,password
*Sep  6 14:44:08.572 IST: SSH2 0: Using method = none
*Sep  6 14:44:08.572 IST: SSH2 0: Authentications that can continue =
publickey,keyboard-interactive,password
*Sep  6 14:44:08.572 IST: SSH2 0: Using method = keyboard-interactive
*Sep  6 14:44:11.983 IST: SSH2 0: authentication successful for cisco
*Sep  6 14:44:11.984 IST: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: cisco] [Source:
192.168.121.40] [localport: 22] at 14:44:11 IST Thu Sep 6 2018
*Sep  6 14:44:11.984 IST: SSH2 0: channel open request
*Sep  6 14:44:11.985 IST: SSH2 0: pty-req request
*Sep  6 14:44:11.985 IST: SSH2 0: setting TTY - requested: height 24, width 80; set: height 24, width
 80
*Sep  6 14:44:11.985 IST: SSH2 0: shell request
*Sep  6 14:44:11.985 IST: SSH2 0: shell message received
*Sep  6 14:44:11.985 IST: SSH2 0: starting shell for vty
```

```
*Sep  6 14:44:22.066 IST: %SYS-6-LOGOUT: User cisco has exited tty session 1(192.168.121.40)
*Sep  6 14:44:22.166 IST: SSH0: Session terminated normally
*Sep  6 14:44:22.167 IST: SSH CLIENT0: Session terminated normally
```

Step 5 debug ip packet

Turns on debugging for IP packet details.

Example:

```
Device# debug ip packet
```

Step 6 show log

Shows the debug message log.

Example:

```
Device# show log

yslog logging: enabled (0 messages dropped, 9 messages rate-limited, 0 flushes, 0 overruns, xml
disabled, filtering disabled)

No Active Message Discriminator.



No Inactive Message Discriminator.


    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 1363 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    File logging: disabled
    Persistent logging: disabled

No active filter modules.

    Trap logging: level informational, 176 message lines logged
        Logging Source-Interface:       VRF Name:

Log Buffer (4096 bytes):
bleid=0, s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.177 IST: IP: s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1),
 len 40, sending
*Sep  6 14:45:45.177 IST: IP: s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1),
 len 40, output feature, NAT Inside(8), rtype 1, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.177 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1), d=192.168.121.40
 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.177 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1), d=192.168.121.40
 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.177 IST: IP: s=192.168.121.40 (local), d=192.168.121.40, len 40, local feature,
feature skipped, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1),
 len 40, sending
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1),
 len 40, output feature, NAT Inside(8), rtype 1, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1), d=192.168.121.40
```

```
 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40, len 40, local feature,
feature skipped, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1),
 len 40, sending
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1),
 len 40, output feature, NAT Inside(8), rtype 1, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1), d=192.168.121.40
 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1), d=192.168.121.40
 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1), d=192.168.121.40
 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40, len 40, local feature,
feature skipped, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1),
 len 40, sending
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1),
 len 40, output feature, NAT Inside(8), rtype 1, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.179 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1), d=192.168.121.40
 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.179 IST: IP: s=192.168.121.40 (local), d=192.168.121.40, len 40, local feature,
feature skipped, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.179 IST: IP: tableid=0, s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.179 IST: IP: s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1),
 len 40, sending
*Sep  6 14:45:45.179 IST: IP: s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1),
 len 40, output feature, NAT Inside(8), rtype 1, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.179 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1), d=192.168.121.40
 (FortyGigabitEthernet1/0/1), routed via RIB
```

Configuration Examples for X.509v3 Certificates for SSH Authentication

Example: Configuring 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)# exit
```

Example: Configuring 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
```

Additional References for X.509v3 Certificates for SSH Authentication

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)E (Catalyst 2960-L Switches)
PKI configuration	Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment

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

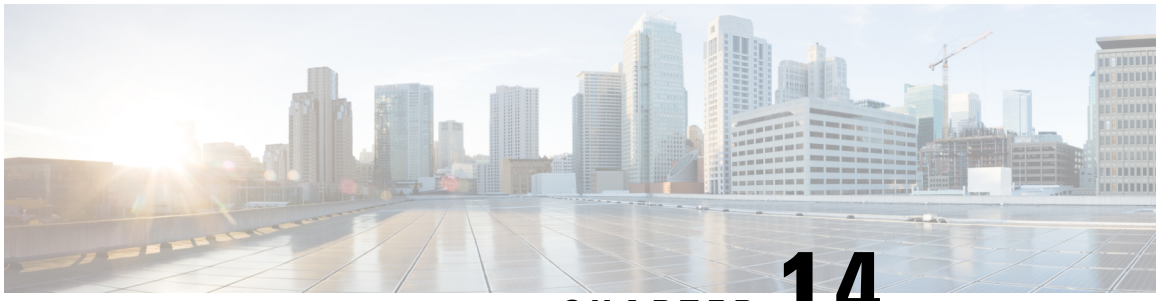
Feature History for X.509v3 Certificates for SSH Authentication

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 Release 15.2(5)E	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 14

Configuring Secure Socket Layer HTTP

This feature provides Secure Socket Layer (SSL) version 3.0 support for the HTTP 1.1 server and HTTP 1.1 client within Cisco IOS software. SSL provides server authentication, encryption, and message integrity to allow secure HTTP communications. SSL also provides HTTP client authentication. HTTP over SSL is abbreviated as HTTPS.

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 device is not configured with a hostname and a domain name, a temporary self-signed certificate is generated. If the device reboots, any temporary self-signed certificate is lost, and a new temporary new self-signed certificate is assigned.
- If the device has been configured with a host and domain name, a persistent self-signed certificate is generated. This certificate remains active if you reboot the device 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 device.

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 device 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 Secure Hash Algorithm (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 default configuration of SSL is as follows:

- 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.

How to Configure Secure Socket Layer HTTP

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 device 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 device. 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.

SUMMARY STEPS

1. **show ip http server status**
2. **configure terminal**
3. **ip http secure-server**
4. **ip http secure-port** *port-number*
5. **ip http secure-ciphersuite** {[**3des-ede-cbc-sha**] [**rc4-128-md5**] [**rc4-128-sha**] [**des-cbc-sha**]}
6. **ip http secure-client-auth**
7. **ip http secure-trustpoint** *name*
8. **ip http path** *path-name*
9. **ip http access-class** *access-list-number*
10. **ip http access-class** { **ipv4** {*access-list-number* | *access-list-name*} | **ipv6** {*access-list-name*} }
11. **ip http max-connections** *value*
12. **ip http timeout-policy idle** *seconds* **life** *seconds* **requests** *value*
13. **end**

DETAILED STEPS

	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.

	Command or Action	Purpose
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:** `Device(config)# ip http secure-client-auth`	(Optional) Configures the HTTP server to request an X.509v3 certificate from the client 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:**	(Optional)Specifies an access list to use to allow access to the HTTP server.

	Command or Action	Purpose
	`Device(config)# ip http access-class ipv4 4`	
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). • **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

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 device. If a CA trustpoint is not configured and the remote HTTPS server requires client authentication, connections to the secure HTTP client fail.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http client secure-trustpoint** *name*
4. **ip http client secure-ciphersuite** {[**3des-ede-cbc-sha**] [**rc4-128-md5**] [**rc4-128-sha**] [**des-cbc-sha**]}
5. **end**

DETAILED STEPS

	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 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 4	**ip http client secure-ciphersuite** {[**3des-ede-cbc-sha**] [**rc4-128-md5**] [**rc4-128-sha**] [**des-cbc-sha**]} **Example:** `Device(config)# ip http client 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 particular CipherSuite, you should allow the server and client to negotiate a CipherSuite that they both support. This is the default.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *hostname*
4. **ip domain-name** *domain-name*
5. **crypto key generate rsa**
6. **crypto ca trustpoint** *name*
7. **enrollment url** *url*
8. **enrollment http-proxy** *host-name port-number*

9. **crl query** *url*
10. **primary** *name*
11. **exit**
12. **crypto ca authentication** *name*
13. **crypto ca enroll** *name*
14. **end**

DETAILED STEPS

	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`	Specifies the hostname of the device (required only if you have not previously configured a hostname). The hostname is required for security keys and certificates.
Step 4	**ip domain-name** *domain-name* **Example:** `Device(config)# ip domain-name your_domain`	Specifies the IP domain name of the device (required only if you have not previously configured an IP domain name). The domain name is required for security keys and certificates.
Step 5	**crypto key generate rsa** **Example:** `Device(config)# crypto key generate rsa`	(Optional) Generates an RSA key pair. RSA key pairs are required before you can obtain a certificate for the device. RSA key pairs are generated automatically. You can use this command to regenerate the keys, if needed.
Step 6	**crypto ca trustpoint** *name* **Example:** `Device(config)# crypto ca trustpoint your_trustpoint`	Specifies a local configuration name for the CA trustpoint and enter CA trustpoint configuration mode.
Step 7	**enrollment url** *url* **Example:** `Device(ca-trustpoint)# enrollment url`	Specifies the URL to which the device should send certificate requests.

	Command or Action	Purpose
	`http://your_server:80`	
Step 8	**enrollment http-proxy** *host-name port-number* **Example:** `Device(ca-trustpoint)# enrollment http-proxy your_host 49`	(Optional) Configures the device to obtain certificates from the CA through an HTTP proxy server. • For *host-name* , specify the proxy server used to get the CA. • For *port-number*, specify the port number used to access the CA.
Step 9	**crl query** *url* **Example:** `Device(ca-trustpoint)# crl query ldap://your_host:49`	Configures the device to request a certificate revocation list (CRL) to ensure that the certificate of the peer has not been revoked.
Step 10	**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 11	**exit** **Example:** `Device(ca-trustpoint)# exit`	Exits CA trustpoint configuration mode and return to global configuration mode.
Step 12	**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 the **crypto ca trustpoint** command.
Step 13	**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 14	**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 16: 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.

Configuration Examples for Secure Socket Layer HTTP

Example: Configuring Secure Socket Layer HTTP

The following example shows a configuration session in which the secure HTTP server is enabled, the port for the secure HTTP server is configured as 1025, and the remote CA trustpoint server *CA-trust-local* is used for certification.

```
Device# show ip http server status

HTTP server status: Disabled
HTTP server port: 80
HTTP server authentication method: enable
HTTP server access class: 0
HTTP server base path:
Maximum number of concurrent server connections allowed: 5
Server idle time-out: 600 seconds
Server life time-out: 600 seconds
Maximum number of requests allowed on a connection: 1
HTTP secure server capability: Present
HTTP secure server status: Disabled
HTTP secure server port: 443
HTTP secure server ciphersuite: 3des-ede-cbc-sha des-cbc-sha rc4-128-md5 rc4-12a
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint:


Device# configure terminal
Device(config)# ip http secure-server
Device(config)# ip http client secure-trustpoint CA-trust-local
Device(config)# ip http secure-port 1024
Invalid secure port value.
Device(config)# ip http secure-port 1025
Device(config)# ip http secure-ciphersuite rc4-128-sha rc4-128-md5
Device(config)# end



Device# show ip http serversecure status
```

```
HTTP secure server status: Enabled
HTTP secure server port: 1025
HTTP secure server ciphersuite: rc4-128-md5 rc4-128-sha
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint: CA-trust-local
```

In the following example, the CA trustpoint CA-trust-local is specified, and the HTTPS client is configured to use this trustpoint for client authentication requests:

```
Device# config terminal
Device(config)# crypto ca trustpoint CA-trust-local
Device(ca-trustpoint)# enrollment url http://example.com
Device(ca-trustpoint)# crl query ldap://example.com
Device(ca-trustpoint)# primary
Device(ca-trustpoint)# exit
Device(config)# ip http client secure-trustpoint CA-trust-local
Device(config)# end
Device# copy running-config startup-config
```

Additional References for Secure Socket Layer HTTP

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

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

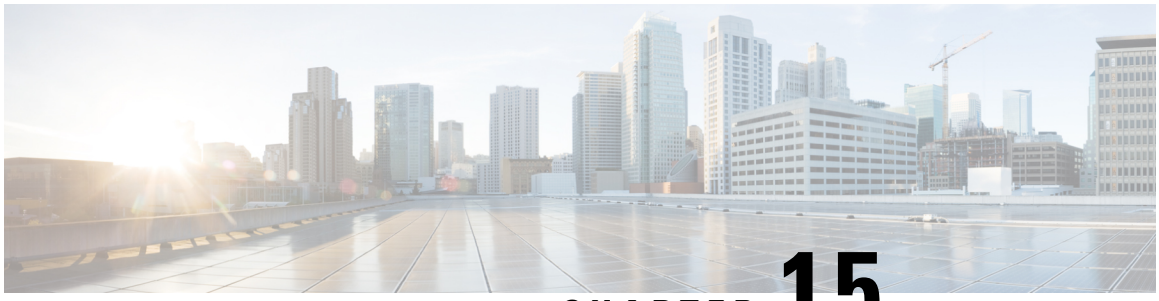
Feature History for Secure Socket Layer HTTP

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 Release 15.2(5)E	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.



CHAPTER 15

Certification Authority Interoperability

This chapter describes how to configure certification authority (CA) interoperability, which is provided in support of the IPSec protocol. CA interoperability permits Cisco IOS devices and CAs to communicate so that your Cisco IOS device can obtain and use digital certificates from the CA. Although IPSec can be implemented in your network without the use of a CA, using a CA provides manageability and scalability for IPSec.

Prerequisites For Certification Authority

You need to have a certification authority (CA) available to your network before you configure this interoperability feature. The CA must support the Public Key Infrastructure (PKI) protocol, and the Simple Certificate Enrollment Protocol (SCEP) .

Restrictions for Certification Authority

When configuring your CA, the following restrictions apply:

- This feature should be configured only when you also configure both IPsec and Internet Key Exchange (IKE) in your network.
- The Cisco IOS software does not support CA server public keys greater than 2048 bits.

Information About Certification Authority

This section provides information about Certification Authority.

CA Supported Standards

Without certification authority (CA) interoperability, Cisco IOS devices could not use CAs when deploying IPSec. CAs provide a manageable, scalable solution for IPSec networks.

Cisco supports the following standards with this feature:

- IPSec—IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer; it uses Internet Key Exchange to handle negotiation of protocols and algorithms based on local policy, and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.
- Internet Key Exchange (IKE)—A hybrid protocol that implements Oakley and Skeme key exchanges inside the Internet Security Association Key Management Protocol (ISAKMP) framework. Although IKE can be used with other protocols, its initial implementation is with the IPSec protocol. IKE provides authentication of the IPSec peers, negotiates IPSec keys, and negotiates IPSec security associations.
- Public-Key Cryptography Standard #7 (PKCS #7)—A standard from RSA Data Security, Inc., used to encrypt and sign certificate enrollment messages.
- Public-Key Cryptography Standard #10 (PKCS #10)—A standard syntax from RSA Data Security, Inc. for certificate requests.
- RSA Keys—RSA is the public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. RSA keys come in pairs: one public key and one private key.
- X.509v3 certificates—Certificate support that allows the IPSec-protected network to scale by providing the equivalent of a digital ID card to each device. When two devices wish to communicate, they exchange digital certificates to prove their identity (thus removing the need to manually exchange public keys with each peer or to manually specify a shared key at each peer). These certificates are obtained from a CA. X.509 is part of the X.500 standard of the ITU.

Purpose of CAs

Certificate authorities (CAs) are responsible for managing certificate requests and issuing certificates to participating IPSec network devices. These services provide centralized key management for the participating devices.

CAs simplify the administration of IPSec network devices. You can use a CA with a network containing multiple IPSec-compliant devices such as routers.

Digital signatures, enabled by public key cryptography, provide a means of digitally authenticating devices and individual users. In public key cryptography, such as the RSA encryption system, each user has a key pair containing both a public and a private key. The keys act as complements, and anything encrypted with one of the keys can be decrypted with the other. In simple terms, a signature is formed when data is encrypted with a user's private key. The receiver verifies the signature by decrypting the message with the sender's public key. The fact that the message could be decrypted using the sender's public key indicates that the holder of the private key, the sender, must have created the message. This process relies on the receiver's having a copy of the sender's public key and knowing with a high degree of certainty that it really does belong to the sender and not to someone pretending to be the sender.

Digital certificates provide the link. A digital certificate contains information to identify a user or device, such as the name, serial number, company, department, or IP address. It also contains a copy of the entity's public key. The certificate is itself signed by a certification authority (CA), a third party that is explicitly trusted by the receiver to validate identities and to create digital certificates.

In order to validate the signature of the CA, the receiver must first know the CA's public key. Normally this process is handled out-of-band or through an operation done at installation. For instance, most web browsers are configured with the public keys of several CAs by default. The Internet Key Exchange (IKE), an essential component of IPSec, can use digital signatures to scalably authenticate peer devices before setting up security associations.

Without digital signatures, one must manually exchange either public keys or secrets between each pair of devices that use IPSec to protect communications between them. Without certificates, every new device added to the network requires a configuration change on every other device with which it communicates securely. With digital certificates, each device is enrolled with a certification authority. When two devices wish to communicate, they exchange certificates and digitally sign data to authenticate each other. When a new device is added to the network, one simply enrolls that device with a CA, and none of the other devices needs modification. When the new device attempts an IPSec connection, certificates are automatically exchanged and the device can be authenticated.

Registration Authorities

Some CAs have a registration authority (RA) as part of their implementation. An RA is essentially a server that acts as a proxy for the CA so that CA functions can continue when the CA is offline.

Some of the configuration tasks described in this document differ slightly, depending on whether your CA supports an RA.

How to Configure Certification Authority

This section decribes how to configure certification authority.

Managing NVRAM Memory Usage

Certificates and certificate revocation lists (CRLs) are used by your device when a CA is used. Normally certain certificates and all CRLs are stored locally in the NVRAM of the device, and each certificate and CRL uses a moderate amount of memory.

The following certificates are normally stored at your device:

- Certificate of your device.
- Certificate of the CA
- Root certificates obtained from CA servers (all root certificates are saved in RAM after the device has been initialized)
- Two registration authority (RA) certificates (only if the CA supports an RA)

CRLs are normally stored at your device according to the following conditions:

- If your CA does not support an RA, only one CRL gets stored in the device.

- If your CA supports an RA, multiple CRLs can be stored in the device.

In some cases, storing these certificates and CRLs locally will not present any difficulty. In other cases, memory might become a problem—particularly if the CA supports an RA and a large number of CRLs have to be stored on the device. If the NVRAM is too small to store root certificates, only the fingerprint of the root certificate is saved.

To save NVRAM space, specify that certificates and CRLs should not be stored locally, but should be retrieved from the CA when needed. This alternative will save NVRAM space but could result in a slight performance impact. To specify that certificates and CRLs should not be stored locally on your device, but should be retrieved when required, enable query mode.

If you do not enable query mode now, you can do it later even if certificates and CRLs have are already stored on the device. In this case, when you enable query mode, the stored certificates and CRLs are deleted from the device after you save the configuration. (If you copy the configuration to a TFTP site prior to enabling query mode, you can save any stored certificates and CRLs at the TFTP site.)

Before disabling query mode, perform the **copy system:running-config nvram:startup-config** command to save all current certificates and CRLs to NVRAM. Otherwise they could be lost during a reboot.

To specify that certificates and CRLs should not be stored locally on your device, but should be retrieved when required, enable query mode by using the following command in global configuration mode:

Note Query mode may affect availability if the CA is down.

SUMMARY STEPS

1. **crypto ca certificate query**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto ca certificate query** **Example:** `Device(config)# crypto ca certificate query`	Enables query mode, which causes certificates and CRLs not to be stored locally.

Configuring the Device Host Name and IP Domain Name

You must configure the host name and IP domain name of a device if this has not already been done. This is required because the device assigns a fully qualified domain name (FQDN) to the keys and certificates used by IPsec, and the FQDN is based on the host name and IP domain name assigned to the device. For example, a certificate named "device20.example.com" is based on a device host name of "device20" and a device IP domain name of "example.com".

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*

4. **ip domain-name** *name*
5. **end**

DETAILED STEPS

	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** *name* **Example:** `Device(config)# hostname device1`	Configures the host name of the device.
Step 4	**ip domain-name** *name* **Example:** `Device(config)# ip domain-name domain.com`	Configures the IP domain name of the device.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration and returns to privileged EXEC mode.

Generating an RSA Key Pair

Rivest, Shamir, and Adelman (RSA) key pairs are used to sign and encrypt IKE key management messages and are required before obtaining a certificate for your device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** [**usage-keys**]
4. **end**

DETAILED STEPS

	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	**crypto key generate rsa** [**usage-keys**] **Example:** `Device(config)# crypto key generate rsa usage-keys`	Generates an RSA key pair. Use the **usage-keys** keyword to specify special-usage keys instead of general-purpose keys.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration and returns to privileged EXEC mode.

Declaring a Certification Authority

You should declare one certification authority (CA) to be used by the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*
5. **enrollment command**
6. **exit**
7. **crypto pki trustpoint** *name*
8. **crl query ldap**://*url*:[*port*]
9. **enrollment** {**mode ra** | **retry count** *number* | **retry period** *minutes* | **url** *url*}
10. **enrollment** {**mode ra** | **retry count** *number* | **retry period** *minutes* | **url** *url*}
11. **revocation-check** *method1* [*method2 method3*]
12. **end**

DETAILED STEPS

	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	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint ka`	Declares the certification authority (CA) that your device should use and enters the CA trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:** `Device(ca-trustpoint)# enrollment url http://entrust:81`	Specifies the URL of the CA server to which enrollment requests are sent.
Step 5	**enrollment command** **Example:** `Device(ca-trustpoint)# enrollment command`	Specifies the HTTP command that is sent to the CA for enrollment.
Step 6	**exit** **Example:** `Device(ca-trustpoint)# exit`	Exit CA profile enroll configuration mode and returns to global configuration mode.
Step 7	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint ka`	Declares the trustpoint that your device should use and enters Ca-trustpoint configuration mode.
Step 8	**crl query ldap**://*url*:[*port*] **Example:** `Device(ca-trustpoint)# crl query ldap://bar.cisco.com:3899`	Queries the certificate revocation list (CRL) to ensure that the certificate of the peer is not revoked.
Step 9	**enrollment** {**mode ra** | **retry count** *number* | **retry period** *minutes* | **url** *url*} **Example:** `Device(ca-trustpoint)# enrollment retry period 2`	Specifies the enrollment wait period between certificate request retries.
Step 10	**enrollment** {**mode ra** | **retry count** *number* | **retry period** *minutes* | **url** *url*} **Example:** `Device(ca-trustpoint)# enrollment retry count 8`	Specifies the number of times a device will resend a certificate request when it does not receive a response from the previous request.
Step 11	**revocation-check** *method1* [*method2 method3*] **Example:** `Device(ca-trustpoint)# revocation-check crl ocsp`	Checks the revocation status of a certificate.
Step 12	**end** **Example:** `Device(ca-trustpoint)# end`	Exit CA trustpoint configuration mode and returns to privileged EXEC mode.

Configuring a Root CA (Trusted Root)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **revocation-check** *method1* [*method2 method3*]
5. **root tftp** *server-hostname filename*
6. **enrollment http-proxy** *hostname port-number*
7. **end**

DETAILED STEPS

	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	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint ka`	Declares the trustpoint that your device should use and enters CA trustpoint configuration mode.
Step 4	**revocation-check** *method1* [*method2 method3*] **Example:** `Device(ca-trustpoint)# revocation-check ocsp`	Checks the revocation status of a certificate.
Step 5	**root tftp** *server-hostname filename* **Example:** `Device(ca-trustpoint)# root tftp server1 file1`	Obtains the certification authority (CA) certificate via TFTP.
Step 6	**enrollment http-proxy** *hostname port-number* **Example:** `Device(ca-trustpoint)# enrollment http-proxy host2 8080`	Accesses the certification authority (CA) by HTTP through the proxy server.
Step 7	**end** **Example:** `Device(ca-trustpoint)# end`	Exits CA trustpoint configuration mode and returns to privileged EXEC mode.

Authenticating the CA

The device must authenticate the certification authority (CA). It does this by obtaining the self-signed certificate of the CA, which contains the public key of the CA. Because the certificate of the CA is self-signed (the CA signs its own certificate) the public key of the CA should be manually authenticated by contacting the CA administrator to compare the fingerprint of the CA certificate when you perform this step.

Perform the following task to get the public key of the CA:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki authenticate***name*
4. **end**

DETAILED STEPS

	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	**crypto pki authenticate***name* **Example:** `Device(config)# crypto pki authenticate myca`	Authenticates the CA by getting the certificate of the CA.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Requesting Signed Certificates

You must obtain a signed certificate from the certification authority (CA) for each of the RSA key pairs on your device. If you generated general-purpose RSA keys, your device has only one RSA key pair and needs only one certificate. If you previously generated special-usage RSA keys, your device has two RSA key pairs and needs two certificates.

Perform the following task to request signed certificates from the CA:

Note If your device reboots after you have issued the **crypto pki enroll** command, but before you have received the certificates, you must reissue the command and notify the CA administrator.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki enroll** *number*
4. **end**

DETAILED STEPS

	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	**crypto pki enroll** *number* **Example:** `Device(config)# crypto pki enroll myca`	Obtains certificates for your device from the CA.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

What to do next

Saving Your Configuration

Always remember to save your work when you make configuration changes.

Use the **copy system:running-config nvram:startup-config** command to save your configuration. This command includes saving RSA keys to private NVRAM. RSA keys are not saved with your configuration when you use a **copy system:running-config rcp:** or copy **system:running-config tftp:** command.

Monitoring and Maintaining Certification Authority

This section provides information about monitoring and maintaining Certification Authority

Requesting a Certificate Revocation List

You can request a certificate revocation list (CRL) only if the certification authority (CA) does not support a registration authority (RA). The following task applies only when the CA does not support an RA.

When a device receives a certificate from a peer, your device will download a CRL from the CA. The device then checks the CRL to make sure the certificate that the peer sent has not been revoked. (If the certificate appears on the CRL, the device will not accept the certificate and will not authenticate the peer.)

A CRL can be reused with subsequent certificates until the CRL expires if query mode is off. If the device receives a peer's certificate after the applicable CRL has expired, the device will download the new CRL.

If the device has a CRL that has not yet expired, but you suspect that the contents of the CRL are out of date, you can request that the latest CRL be downloaded immediately to replace the old CRL.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki crl request** *name*
4. **end**

DETAILED STEPS

	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	**crypto pki crl request** *name* **Example:** `Device(config)# crypto pki crl request myca`	Requests that a new certificate revocation list (CRL) be obtained immediately from the CA.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Querying a Certification Revocation List

You can query a certificate revocation list (CRL) only when you configure your device with a trusted root. When your device receives a certificate from a peer from another domain (with a different CA), the CRL downloaded from the CA of the device will not include certificate information about the peer. Therefore, you should check the CRL published by the configured root with the LDAP URL to ensure that the certificate of the peer has not been revoked.

If you would like CRL of the root certificate to be queried when the device reboots, you must enter the **crl query** command.

Perform the following task to query the CRL published by the configured root with the LDAP URL:

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **crypto pki trustpoint** *name*
4. **crl query ldap** *://url* : [*port*]
5. **end**

DETAILED STEPS

	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	**crypto pki trustpoint** *name* **Example:** `Device(ca-trustpoint)# crypto pki trustpoint mytp`	Declares the trustpoint that your device should use and enters CA trustpoint configuration mode.
Step 4	**crl query ldap** *://url* : [*port*] **Example:** `Device(ca-trustpoint)# crl query ldap://url:[port]`	Queries the CRL to ensure that the certificate of the peer has not been revoked.
Step 5	**end** **Example:** `Device(ca-trustpoint)# end`	Exits CA trustpoint configuration mode and returns to privileged EXEC mode.

Deleting RSA Keys from a Device

Under certain circumstances you may want to delete RSA keys from your device. For example, if you believe the RSA keys were compromised in some way and should no longer be used, you should delete the keys.

]

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key zeroize rsa** [*key-pair-label*]
4. **end**

DETAILED STEPS

	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	**crypto key zeroize rsa** [*key-pair-label*] **Example:** `Device(config)# crypto key zeroize rsa`	Deletes all Rivest, Shamir, and Adelman (RSA) keys from your device.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

What to do next

After you delete RSA keys from the device, you should also complete the following two additional tasks:

- Ask the CA administrator to revoke the device certificates at the CA; you must supply the challenge password that you created when you originally obtained the device certificates with the **crypto pki enroll** command.
- Manually remove the device certificates from the device configuration.

Deleting Public Keys for a Peer

Under certain circumstances you may want to delete RSA public keys of peer devices from your device configuration. For example, if you no longer trust the integrity of the public key of a peer, you should delete the key.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key pubkey-chain rsa**
4. **no named key** *key-name* [**encryption** | **signature**]
5. **end**

DETAILED STEPS

	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	**crypto key pubkey-chain rsa** **Example:** `Device(config)# crypto key pubkey-chain rsa`	Enters public key chain configuration mode, so that you can manually specify other devices' RSA public keys.
Step 4	**no named key** *key-name* [**encryption** | **signature**] **Example:** `Device(config-pubkey-c)# no named-key otherpeer.example.com`	Deletes the RSA public key of a remote peer and enters public key configuration mode.
Step 5	**end** **Example:** `Device(config-pubkey)# end`	Exits public key configuration mode and returns to privileged EXEC mode.

Deleting Certificates from the Configuration

If the need arises, you can delete certificates that are saved in your device. Your devices saves its own certificates, the certificate of the CA, and any RA certificates .

To delete the CA's certificate, you must remove the entire CA identity, which also removes all certificates associated with the CA—your router's certificate, the CA certificate, and any RA certificates.

SUMMARY STEPS

1. **enable**
2. **show crypto pki certificates**
3. **configure terminal**
4. **crypto pki certificate chain** *name*
5. **no certificate** *certificate-serial-number*
6. **exit**
7. **no crypto pki import** *name* **certificate**
8. **exit**

DETAILED STEPS

	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	**show crypto pki certificates** **Example:** `Device# show crypto pki certificates`	Displays information about your device certificate, the certification authority (CA) certificate, and any registration authority (RA) certificates.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**crypto pki certificate chain** *name* **Example:** `Device(config)# crypto pki certificate chain myca`	Enters certificate chain configuration mode.
Step 5	**no certificate** *certificate-serial-number* **Example:** `Device(config-cert-chain)# no certificate 0123456789ABCDEF0123456789ABCDEF`	Deletes the certificate.
Step 6	**exit** **Example:** `Device(config-cert-chain)# exit`	Exits certificate chain configuration mode and returns to global configuration mode.
Step 7	**no crypto pki import** *name* **certificate** **Example:** `Device(config)# no crypto pki import MS certificate`	Deletes a certificate manually.
Step 8	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Viewing Keys and Certificates

Perform the following task toview keys and certificates:

SUMMARY STEPS

1. **enable**
2. **show crypto key mypubkey rsa** [*keyname*]
3. **show crypto key pubkey-chain rsa**
4. **show crypto key pubkey-chain rsa** [**name** *key-name* | **address** *key-address*]
5. **show crypto pki certificates**
6. **show crypto pki trustpoints**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**show crypto key mypubkey rsa** [*keyname*] **Example:** `Device# show crypto key mypubkey rsa [keyname]`	Displays the RSA public keys configured on a device.
Step 3	**show crypto key pubkey-chain rsa** **Example:** `Device# show crypto key pubkey-chain rsa`	Displays the RSA public keys of the peer that are stored on a device.
Step 4	**show crypto key pubkey-chain rsa** [**name** *key-name* | **address** *key-address*] **Example:** `Device# show crypto key pubkey-chain rsa address 209.165.202.129`	Displays the address of a specific key.
Step 5	**show crypto pki certificates** **Example:** `Device# show crypto pki certificates`	Displays information about the device certificate, the certification authority (CA) certificate, and any registration authority (RA) certificates
Step 6	**show crypto pki trustpoints** **Example:** `Device# show crypto pki certificates`	Displays trustpoints that are configured on a device.

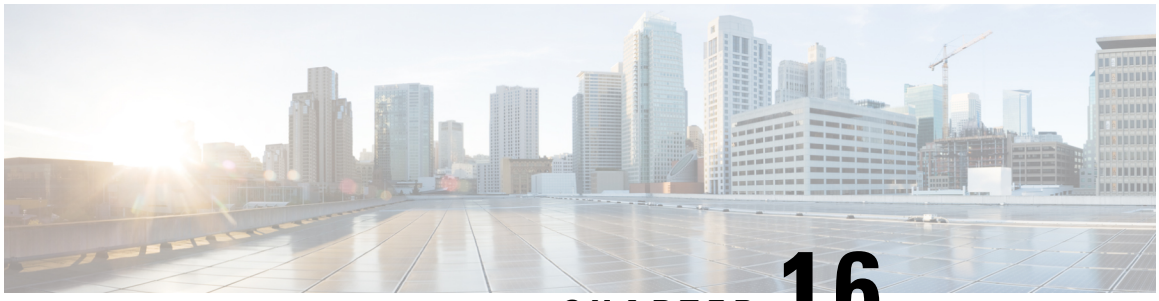
Feature History for Certification Authority Interoperability

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 Release 15.2(5)E	Certification Authority Interoperability	CA interoperability permits Cisco IOS devices and CAs to communicate so that your Cisco IOS device can obtain and use digital certificates from the CA.

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 16

Access Control List Overview

Access lists filter network traffic by controlling the forwarding or blocking of packets at the interface of a device. A device examines each packet to determine whether to forward or drop that packet, based on the criteria specified in access lists.

The criteria that can be specified in an access list include the source address of the traffic, the destination address of the traffic, and the upper-layer protocol.

Note Some users might successfully evade basic access lists because these lists require no authentication.

Information About Access Control Lists

Access lists filter network traffic by controlling the forwarding or blocking of packets at the interface of a device. A device examines each packet to determine whether to forward or drop that packet, based on the criteria specified in access lists.

The criteria that can be specified in an access list include the source address of the traffic, the destination address of the traffic, and the upper-layer protocol.

Note Some users might successfully evade basic access lists because these lists require no authentication.

Definition of an Access List

An access list is a sequential list consisting of at least one **permit** statement and possibly one or more **deny** statements. In the case of IP access lists, the statements can apply to IP addresses, upper-layer IP protocols, or other fields in IP packets. The access list is identified and referenced by a name or a number. Access list acts as a packet filter, filtering packets based on the criteria defined in the access list.

An access list may be configured, but it does not take effect until the access list is either applied to an interface, a virtual terminal line (vty), or referenced by some command that accepts an access list. Multiple commands can reference the same access list.

The following configuration example shows how to create an IP access list named branchoffices. The ACL is applied to gigabitEthernet on incoming packets. No sources other than those on the networks specified by each source address and mask pair can access this interface. The destinations for packets coming from sources on network 172.20.7.0 are unrestricted. The destination for packets coming from sources on network 172.29.2.0 must be 172.25.5.4.

```
ip access-list extended branchoffices
 10 permit 172.20.7.0 0.0.0.3 any
 20 permit 172.29.2.0 0.0.0.255 host 172.25.5.4
!
gigabitEthernet 0/1
 ip access-group branchoffices in
```

Functions of an Access Control List

There are many reasons to configure access lists; for example, to restrict contents of routing updates or to provide traffic flow control. One of the most important reasons to configure access lists is to provide security for your network, which is the focus of this module.

Use access lists to provide a basic level of security for accessing your network. If you do not configure access lists on your device, all packets passing through the device are allowed access to all parts of your network.

Access lists can allow a host to access a part of your network and prevent another host from accessing the same area. In the figure below, Host A is allowed to access the Human Resources network, but Host B is prevented from accessing the Human Resources network.

You can also use access lists to define the type of traffic that is forwarded or blocked at device interfaces. For example, you can permit e-mail traffic to be routed but at the same time block all Telnet traffic.

Purpose of IP Access Lists

Access lists perform packet filtering to control which packets move through the network and where. Such control can help limit network traffic and restrict the access of users and devices to the network. Access lists have many uses, and therefore many commands accept a reference to an access list in their command syntax. Access lists can be used to do the following:

- Filter incoming packets on an interface.
- Filter outgoing packets on an interface.
- Restrict the contents of routing updates.
- Limit debug output based on an address or protocol.
- Control virtual terminal line access.
- Identify or classify traffic for advanced features, such as congestion avoidance, congestion management, and priority and custom queuing.
- Trigger dial-on-demand routing (DDR) calls.

Reasons to Configure ACLs

There are many reasons to configure access lists; for example, you can use access lists to restrict contents of switching updates or to provide traffic flow control. One of the most important reasons to configure access lists is to provide a basic level of security for your network by controlling access to it. If you do not configure access lists on your device, all packets passing through the device could be allowed onto all parts of your network.

An access list can allow one host to access a part of your network and prevent another host from accessing the same area. For example, by applying an appropriate access list to interfaces of a device, Host A is allowed to access the human resources network and Host B is prevented from accessing the human resources network.

You can use access lists on a device that is positioned between two parts of your network, to control traffic entering or exiting a specific part of your internal network.

To provide some security benefits of access lists, you should at least configure access lists on border devices—devices located at the edges of your networks. Such an access list provides a basic buffer from the outside network or from a less controlled area of your own network into a more sensitive area of your network. On these border devices, you should configure access lists for each network protocol configured on the device interfaces. You can configure access lists so that inbound traffic or outbound traffic or both are filtered on an interface.

Access lists are defined on a per-protocol basis. In other words, you should define access lists for every protocol enabled on an interface if you want to control traffic flow for that protocol.

Software Processing of an Access List

The following general steps describe how the an access list is processed when it is applied to an interface, a vty, or referenced by any command. These steps apply to an access list that has 13 or fewer access list entries.

- The software receives an IP packet and tests parts of each packet being filtered against the conditions in the access list, one condition (**permit** or **deny** statement) at a time. For example, the software tests the source and destination addresses of the packet against the source and destination addresses in a **permit** or **deny** statement.
- If a packet does not match an access list statement, the packet is then tested against the next statement in the list.
- If a packet and an access list statement match, the rest of the statements in the list are skipped and the packet is permitted or denied as specified in the matched statement. The first entry that the packet matches determines whether the software permits or denies the packet. That is, after the first match, no subsequent entries are considered.
- If no conditions match, the software drops the packet. This is because each access list ends with an unwritten, implicit **deny** statement. That is, if the packet has not been permitted by the time it was tested against each statement, it is denied.

An access list with more than 13 entries is processed using a trie-based lookup algorithm. This process will happen automatically; it does not need to be configured.

Access List Rules

The following rules apply to access control lists (ACLs):

- Only one access list per interface, per protocol, and per direction is allowed.
- An access list must contain at least one **permit** statement or all packets are denied entry into the network.
- The order in which access list conditions or match criteria are configured is important. While deciding whether to forward or block a packet, Cisco software tests the packet against each criteria statement in the order in which these statements are created. After a match is found, no more criteria statements are checked. The same **permit** or **deny** statements specified in a different order can result in a packet being passed under one circumstance and denied in another circumstance.
- If an access list is referenced by a name, but the access list does not exist, all packets pass. An interface or command with an empty access list applied to it permits all traffic into the network.
- Standard access lists and extended access lists cannot have the same name.
- Inbound access lists process packets before packets are sent to an outbound interface. Inbound access lists that have filtering criteria that deny packet access to a network saves the overhead of a route lookup. Packets that are permitted access to a network based on the configured filtering criteria are processed for routing. For inbound access lists, when you configure a **permit** statement, packets are processed after they are received, and when you configure a **deny** statement, packets are discarded.
- Outbound access lists process packets before they leave the device. Incoming packets are routed to the outbound interface and then processed by the outbound access list. For outbound access lists, when you configure a **permit** statement, packets are sent to the output buffer, and when you configure a **deny** statement, packets are discarded.
- An access list can control traffic arriving at a device or leaving a device, but not traffic originating at a device.

Helpful Hints for Creating IP Access Lists

The following tips will help you avoid unintended consequences and help you create more efficient access lists.

- Create the access list before applying it to an interface (or elsewhere), because if you apply a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause you immediate access problems.
- Another reason to configure an access list before applying it is because an interface with an empty access list applied to it permits all traffic.
- All access lists need at least one **permit** statement; otherwise, all packets are denied and no traffic passes.
- Because the software stops testing conditions after it encounters the first match (to either a **permit** or **deny** statement), you will reduce processing time and resources if you put the statements that packets are most likely to match at the beginning of the access list. Place more frequently occurring conditions before less frequent conditions.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- Use the statement **permit any any** if you want to allow all other packets not already denied. Using the statement **permit any any** in effect avoids denying all other packets with the implicit deny statement at the end of an access list. Do not make your first access list entry **permit any any** because all traffic will

get through; no packets will reach the subsequent testing. In fact, once you specify **permit any any**, all traffic not already denied will get through.

- Although all access lists end with an implicit **deny** statement, we recommend use of an explicit **deny** statement (for example, **deny ip any any**). On most platforms, you can display the count of packets denied by issuing the **show access-list**command, thus finding out more information about who your access list is disallowing. Only packets denied by explicit **deny** statements are counted, which is why the explicit **deny** statement will yield more complete data for you.
- While you are creating an access list or after it is created, you might want to delete an entry.
  - You cannot delete an entry from a numbered access list; trying to do so will delete the entire access list. If you need to delete an entry, you need to delete the entire access list and start over.
  - You can delete an entry from a named access list. Use the **no permit** or **no deny** command to delete the appropriate entry.
- In order to make the purpose of individual statements more scannable and easily understood at a glance, you can write a helpful remark before or after any statement by using the **remark** command.
- If you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you.
- This hint applies to the placement of your access list. When trying to save resources, remember that an inbound access list applies the filter conditions before the routing table lookup. An outbound access list applies the filter conditions after the routing table lookup.

IP Packet Fields You Can Filter to Control Access

You can use an extended access list to filter on any of the following fields in an IP packet. Source address and destination address are the two most frequently specified fields on which to base an access list:

- Source address--Specifies a source address to control packets coming from certain networking devices or hosts.
- Destination address--Specifies a destination address to control packets being sent to certain networking devices or hosts.

Source and Destination Addresses

Source and destination address fields in an IP packet are two typical fields on which to base an access list. Specify source addresses to control the packets being sent from certain networking devices or hosts. Specify destination addresses to control the packets being sent to certain networking devices or hosts.

Wildcard Mask for Addresses in an Access List

Address filtering uses wildcard masking to indicate to the software whether to check or ignore corresponding IP address bits when comparing the address bits in an access list entry to a packet being submitted to the access list. By carefully setting wildcard masks, you can specify one or more IP addresses for permit or deny tests.

Wildcard masking for IP address bits uses the number 1 and the number 0 to specify how the software treats the corresponding IP address bits. A wildcard mask is sometimes referred to as an inverted mask because a 1 and 0 mean the opposite of what they mean in a subnet (network) mask.

- A wildcard mask bit 0 means check the corresponding bit value; they must match.
- A wildcard mask bit 1 means ignore that corresponding bit value; they need not match.

If you do not supply a wildcard mask with a source or destination address in an access list statement, the software assumes an implicit wildcard mask of 0.0.0.0, meaning all values must match.

Unlike subnet masks, which require contiguous bits indicating network and subnet to be ones, wildcard masks allow noncontiguous bits in the mask.

The table below shows examples of IP addresses and masks from an access list, along with the corresponding addresses that are considered a match.

Table 17: Sample IP Addresses, Wildcard Masks, and Match Results

Address	Wildcard Mask	Match Results
0.0.0.0	255.255.255.255	All addresses will match the access list conditions.
172.18.0.0/16	0.0.255.255	Network 172.18.0.0
172.18.5.2/16	0.0.0.0	Only host 172.18.5.2 matches
172.18.8.0	0.0.0.7	Only subnet 172.18.8.0/29 matches
172.18.8.8	0.0.0.7	Only subnet 172.18.8.8/29 matches
172.18.8.15	0.0.0.3	Only subnet 172.18.8.15/30 matches
10.1.2.0	0.0.254.255 (noncontiguous bits in mask)	Matches any even-numbered network in the range of 10.1.2.0 to 10.1.254.0

Access List Sequence Numbers

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If you wanted to insert an entry in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

This feature allows users to add sequence numbers to access list entries and resequence them. When you add a new entry, you specify the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry.

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.

Supported ACLs

The switch supports the following type of ACL 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 input direction to each access list type — IPv4 and MAC.

Port ACLs

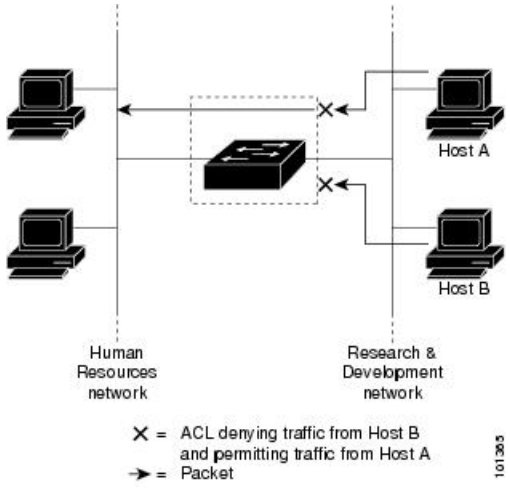
Port ACLs are ACLs that are applied to Layer 2 interfaces on a switch. Port ACLs are supported only on physical interfaces and not on EtherChannel interfaces. Port ACLs can be applied to the interface in inbound direction. The following access lists are supported:

- 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 4: 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 cannot 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.

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.

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.

Example: ACEs and Fragmented and Unfragmented Traffic

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.

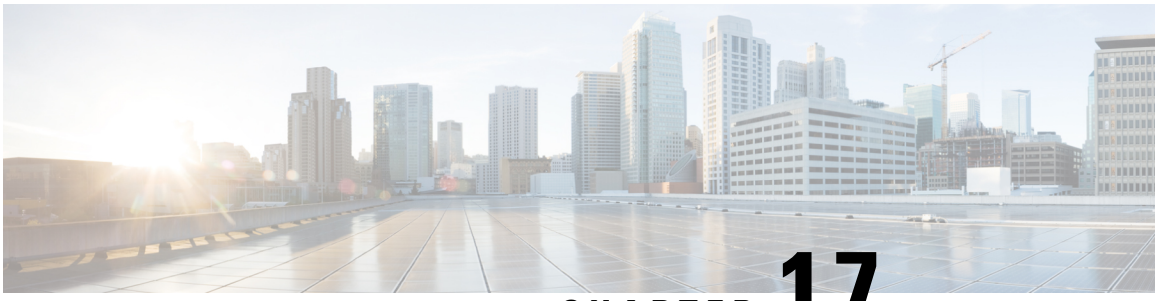
- Packet A is a TCP packet from host 10.2.2.2., port 65000, going to host 10.1.1.1 on the SMTP port. If this packet is fragmented, the first fragment matches the first ACE (a permit) as if it were a complete packet because all Layer 4 information is present. The remaining fragments also match the first 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.

Additional References for Access Control Lists Overview

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)
ACLs	For more information, see: • "Configuring IPv4 Access Control Lists" in the *Security Configuration Guide* • "Configuring IPv6 Access Control Lists" in the *Security Configuration Guide*



CHAPTER 17

Configuring IPv4 Access Control Lists

Restrictions for Configuring IPv4 Access Control Lists

General Network Security

The following are restrictions for configuring network security with ACLs:

- Router ACL and VLAN ACLs are not supported.
- 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.
- 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.
- For hardware ACL filtering, a maximum of 18 ACLs are supported. After crossing this limit, only software filtering takes place subject to low rate and CPU utilization.

  When the unique ACLs on a device reach 18, downloadable (DACLs) are not allowed, and an error message is displayed on the console. However, port ACLs (PACLs) are allowed because these use software forwarding.
- Per ASIC, 8 TCP port comparators and 8 UDP port comparators are supported, and each gt (greater than)/lt (less than)/neq (not equal) operator uses 1 port comparator, and each range operator uses 2 port comparators.

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.
- You do not have to enable routing to apply ACLs to Layer 2 interfaces.
- On Layer 3 ports and SVIs, ACLs are not supported.

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.

- MAC ACLs do not filter or block Address Resolution Protocol (ARP) traffic but allows all ARP traffic by default.

IP Access List Entry Sequence Numbering

- This feature does not support dynamic, reflexive, or firewall access lists.

TCAM Matching Priority

Table 18: TCAM matching priority for WebAuth with port ACL and download ACL

WebAuth	Port ACL or Download ACL	Final Action
Denied ACE	ACE present and denied	Packet is permitted
Denied ACE	ACE not present and implicit denied	Packet is permitted

- Due to hardware limitation, hardware TCAM match counters are not updated for permit ACEs. However, for deny ACEs they are updated.

Information About IPv4 Access Control Lists

Access control lists (ACLs) perform packet filtering to control which packets move through the network and where. Such control provides security by helping to limit network traffic, restrict the access of users and

devices to the network, and prevent traffic from leaving a network. IP access lists can reduce the chance of spoofing and denial-of-service attacks and allow dynamic, temporary user access through a firewall.

IP access lists can also be used for purposes other than security, such as bandwidth control, limiting debug output, and identifying or classifying traffic for quality of service (QoS) features. This module provides an overview of IP 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 or VLANs. 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, including packets bridged within a VLAN.

You configure access lists on a router 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. ACLs can be configured to block inbound traffic, outbound traffic, or both.

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 or bridge-group ACLs
- IP accounting
- Inbound and outbound rate limiting (except with QoS ACLs)

- Reflexive ACLs and dynamic ACLs are not supported. (except for some specialized dynamic ACLs used by the switch clustering feature)
- ACL logging for VLAN maps

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 19: 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.

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:

Note ICMP echo-reply cannot be filtered. All other ICMP codes or types can be filtered.

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. 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.

Benefits of Using the Named ACL Support for Noncontiguous Ports on an Access Control Entry Feature

The Named ACL Support for Noncontiguous Ports on an Access Control Entry feature allows you to specify noncontiguous ports in a single access control entry, which greatly reduces the number of entries required in an access control list when several entries have the same source address, destination address, and protocol, but differ only in the ports.

This feature greatly reduces the number of access control entries (ACEs) required in an access control list to handle multiple entries for the same source address, destination address, and protocol. If you maintain large numbers of ACEs, use this feature to consolidate existing groups of access list entries wherever it is possible and when you create new access list entries. When you configure access list entries with noncontiguous ports, you will have fewer access list entries to maintain.

Benefits of IP Access List Entry Sequence Numbering

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If a user wanted to insert an entry (statement) in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

This feature allows users to add sequence numbers to access list entries and resequence them. When a user adds a new entry, the user chooses the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry.

Sequence Numbering Behavior

- For backward compatibility with previous releases, if entries with no sequence numbers are applied, the first entry is assigned a sequence number of 10, and successive entries are incremented by 10. The maximum sequence number is 2147483647. If the generated sequence number exceeds this maximum number, the following message is displayed:

```
Exceeded maximum sequence number.
```

- If the user enters an entry without a sequence number, it is assigned a sequence number that is 10 greater than the last sequence number in that access list and is placed at the end of the list.
- If the user enters an entry that matches an already existing entry (except for the sequence number), then no changes are made.
- If the user enters a sequence number that is already present, the following error message is generated:

```
Duplicate sequence number.
```

- If a new access list is entered from global configuration mode, then sequence numbers for that access list are generated automatically.
- Distributed support is provided so that the sequence numbers of entries in the Route Processor (RP) and line card are in synchronization at all times.
- Sequence numbers are not nvgened. That is, the sequence numbers themselves are not saved. In the event that the system is reloaded, the configured sequence numbers revert to the default sequence starting number and increment. The function is provided for backward compatibility with software releases that do not support sequence numbering.
- This feature works with named and numbered, standard and extended IP access lists.

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.

The following is an example of a remark that describes function of the subsequent deny statement:

```
ip access-list extended telnetting
  remark Do not allow host1 subnet to telnet out
  deny tcp host 172.16.2.88 any eq telnet
```

Hardware and Software Treatment of IP ACLs

ACL processing is performed at the hardware side. If the hardware reaches its capacity to store ACL configurations, the packets are sent to the CPU, where ACL is processed at the software side. When sent for software ACL, the data packets are not sent at the line rate; instead, they are sent at a very low rate via rate limiting.

Note If an ACL configuration cannot be implemented in hardware due to an out-of-resource condition on a switch, then only the traffic in that VLAN arriving on that switch is affected. Software forwarding of packets might adversely impact the performance of the switch, depending on the number of CPU cycles that this consumes.

When traffic flows are both logged and forwarded, forwarding is done by hardware, but logging must be done by software. Because of the difference in packet handling capacity between hardware and software, if the sum of all flows being logged (both permitted flows and denied flows) is of great enough bandwidth, not all of the packets that are forwarded can be logged.

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. 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 an 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 in hardware.

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.

These are some benefits of using time ranges:

- You have more control over permitting or denying a user access to resources, such as an application (identified by an IP address/mask pair and a port number).
- You can control logging messages. ACL entries can be set to log traffic only at certain times of the day. Therefore, you can simply deny access without needing to analyze many logs generated during peak hours.

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.

Apply an Access Control List to an Interface

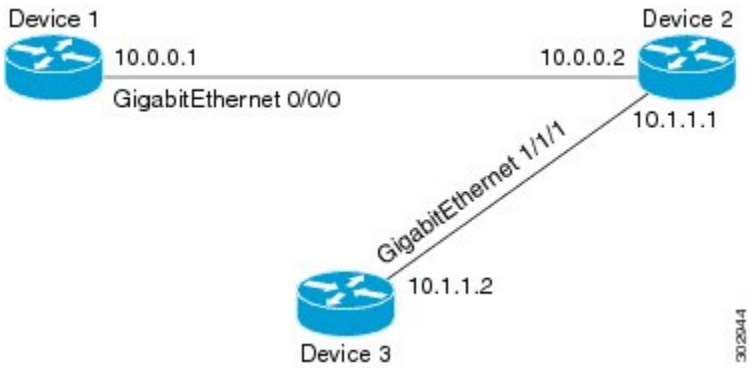
With some protocols, you can apply up to two access lists to an interface: one inbound access list and one outbound access list. With other protocols, you apply only one access list that checks both inbound and outbound packets.

With an inbound access list, when a device receives a packet, Cisco software checks the access list's criteria statements for a match. If the packet is permitted, the software continues to process the packet. If the packet is denied, the software discards the packet.

If the access list is outbound, after receiving and routing a packet to the outbound interface, Cisco software checks the access list's criteria statements for a match. If the packet is permitted, the software transmits the packet. If the packet is denied, the software discards the packet.

Note Access lists that are applied to interfaces on a device do not filter traffic that originates from that device.

Figure 5: Topology for Applying Access Control Lists



The figure above shows that Device 2 is a bypass device that is connected to Device 1 and Device 3. An outbound access list is applied to Gigabit Ethernet interface 0/0 on Device 1. When you ping Device 3 from Device 1, the access list does not check for packets going outbound because the traffic is locally generated.

The access list check is bypassed for locally generated packets, which are always outbound.

By default, an access list that is applied to an outbound interface for matching locally generated traffic will bypass the outbound access list check; but transit traffic is subjected to the outbound access list check.

Note The behavior described above applies to all single-CPU platforms that run Cisco software.

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 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.

How to Configure ACLs

This section provides information about how to configure ACLs.

Configuring IPv4 ACLs

Follow the procedure given below to use IP ACLs on the switch:

SUMMARY STEPS

1. Create an ACL by specifying an access list number or name and the access conditions.
2. Apply the ACL to interfaces.

DETAILED STEPS

Step 1 Create an ACL by specifying an access list number or name and the access conditions.

Step 2 Apply the ACL to interfaces.

Creating a Numbered Standard ACL (CLI)

Follow the procedure given below to create a numbered standard ACL:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *source source-wildcard* [**log**]
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	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* [**log**] **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.

	Command or Action	Purpose
		(Optional) The *source-wildcard* applies wildcard bits to the source. (Optional) Enter **log** to cause an informational logging message about the packet that matches the entry to be sent to the console. **Note** Logging is supported only on ACLs attached to Layer 3 interfaces.
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.

Creating a Numbered Extended ACL (CLI)

Follow the procedure given below to create a numbered extended ACL:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **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*]
4. **access-list** *access-list-number* {**deny** | **permit**} **tcp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*] [*flag*]
5. **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*]
6. **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*]

7. **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*]
8. **end**

DETAILED STEPS

	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**} *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. 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.

	Command or Action	Purpose
		• 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. **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 4	**access-list** *access-list-number* {**deny** | **permit**} **tcp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**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: • *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 5	**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** keyword is not valid for UDP.
Step 6	**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 7	**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 8	**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:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list standard** *name*
4. Use one of the following:
   - **deny** {*source* [*source-wildcard*] | **host** *source* | **any**} [**log**]
   - **permit** {*source* [*source-wildcard*] | **host** *source* | **any**} [**log**]
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	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.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-std-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.

Creating Extended Named ACLs

Follow the procedure given below to create an extended ACL using names:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *name*
4. {**deny** | **permit**} *protocol* {*source* [*source-wildcard*] | **host** *source* | **any**} {*destination* [*destination-wildcard*] | host *destination* | **any**} [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*]
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	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 access-list extended** *name* **Example:** `Device(config)# ip access-list extended 150`	Defines an extended IPv4 access list using a name, and enter access-list configuration mode. The name can be a number from 100 to 199.
Step 4	{**deny** | **permit**} *protocol* {*source* [*source-wildcard*] | **host** *source* | **any**} {*destination* [*destination-wildcard*] | host *destination* | **any**} [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] **Example:** `Device(config-ext-nacl)# permit 0 any any`	In access-list configuration mode, specify the conditions allowed or denied. Use the **log** keyword to get access list logging messages, including violations. • **host** *source*—A source and source wildcard of *source* 0.0.0.0. • **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.

Sequencing Access-List Entries and Revising the Access List

This task shows how to assign sequence numbers to entries in a named IP access list and how to add or delete an entry to or from an access list. When completing this task, keep the following points in mind:

- Resequencing the access list entries is optional. The resequencing step in this task is shown as required because that is one purpose of this feature and this task demonstrates that functionality.
- In the following procedure, the **permit** command is shown in Step 5 and the **deny** command is shown in Step 6. However, that order can be reversed. Use the order that suits the need of your configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list resequence** *access-list-name starting-sequence-number increment*
4. **ip access-list** {**standard**| **extended**} *access-list-name*
5. Do one of the following:
   - *sequence-number* **permit** *source source-wildcard*
   - *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
6. Do one of the following:
   - *sequence-number* **deny** *source source-wildcard*
   - *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
7. Do one of the following:
   - *sequence-number* **permit** *source source-wildcard*
   - *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
8. Do one of the following:
   - *sequence-number* **deny** *source source-wildcard*
   - *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
9. Repeat Step 5 and/or Step 6 to add sequence number statements, as applicable.
10. **end**
11. **show ip access-lists** *access-list-name*

DETAILED STEPS

	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 resequence** *access-list-name starting-sequence-number increment* **Example:** `Device(config)# ip access-list resequence kmd1 100 15`	Resequences the specified IP access list using the starting sequence number and the increment of sequence numbers.
Step 4	**ip access-list** {**standard**| **extended**} *access-list-name* **Example:** `Device(config)# ip access-list standard kmd1`	Specifies the IP access list by name and enters named access list configuration mode. • If you specify **standard**, make sure you subsequently specify **permit** and/or **deny** statements using the standard access list syntax. • If you specify **extended**, make sure you subsequently specify **permit** and/or **deny** statements using the extended access list syntax.
Step 5	Do one of the following: • *sequence-number* **permit** *source source-wildcard* • *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-std-nacl)# 105 permit 10.5.5.5 0.0.0 255`	Specifies a permit statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be Device(config-ext-nacl) and you would use the extended **permit** command syntax.
Step 6	Do one of the following: • *sequence-number* **deny** *source source-wildcard* • *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:**	(Optional) Specifies a deny statement in named IP access list mode. • This access list uses a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4,

	Command or Action	Purpose
	`Device(config-std-nacl)# 105 deny 10.6.6.7 0.0.0 255`	the prompt for this step would be Device(config-ext-nacl) and you would use the extended **deny** command syntax.
Step 7	Do one of the following: • *sequence-number* **permit** *source source-wildcard* • *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# 150 permit tcp any any log`	Specifies a permit statement in named IP access list mode. • This access list happens to use a **permit**statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the permit (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry.
Step 8	Do one of the following: • *sequence-number* **deny** *source source-wildcard* • *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# 150 deny tcp any any log`	(Optional) Specifies a deny statement in named IP access list mode. • This access list happens to use a **permit**statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the deny (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry.
Step 9	Repeat Step 5 and/or Step 6 to add sequence number statements, as applicable.	Allows you to revise the access list.
Step 10	**end** **Example:** `Device(config-std-nacl)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
Step 11	**show ip access-lists** *access-list-name* **Example:** `Device# show ip access-lists kmd1`	(Optional) Displays the contents of the IP access list.

Examples

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

```
Device# show ip access-lists kmd1
```

```
Standard IP access list kmd1
100 permit 10.4.4.0, wildcard bits 0.0.0.255
105 permit 10.5.5.0, wildcard bits 0.0.0.255
115 permit 10.0.0.0, wildcard bits 0.0.0.255
130 permit 10.5.5.0, wildcard bits 0.0.0.255
145 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Configuring Commented IP ACL Entries

Either use a named or numbered access list configuration. You must apply the access list to an interface or terminal line after the access list is created for the configuration to work.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} {*name* | *number*}
4. **remark** *remark*
5. **deny** *protocol* **host** *host-address* **any eq** *port*
6. **end**

DETAILED STEPS

	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**	**extended**} {*name*	*number*} **Example:** `Device(config)# ip access-list extended telnetting`	Identifies the access list by a name or number and enters extended named access list configuration mode.
Step 4	**remark** *remark* **Example:** `Device(config-ext-nacl)# remark Do not allow host1 subnet to telnet out`	Adds a remark for an entry in a named IP access list. • The remark indicates the purpose of the **permit** or **deny** statement.
Step 5	**deny** *protocol* **host** *host-address* **any eq** *port* **Example:** `Device(config-ext-nacl)# deny tcp host 172.16.2.88 any eq telnet`	Sets conditions in a named IP access list that denies packets.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-ext-nacl)# end`	Exits extended named access list configuration mode and enters privileged EXEC mode.

Configuring Time Ranges for ACLs

Follow these steps to configure a time-range parameter for an ACL:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **time-range** *time-range-name*
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*
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	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*]	Specifies when the function it will be applied to is operational.

	Command or Action	Purpose

	• **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`
		• 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.
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:

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **line** [**console** | **vty**] *line-number*
4. **access-class** *access-list-number* **in**
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	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:** `Devices(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.
Step 4	**access-class** *access-list-number* **in** **Example:** `Device(config-line)# access-class 10 in`	Restricts incoming 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:**	Verifies your entries.

	Command or Action	Purpose
	`Device# show running-config`	
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:

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **ip access-group** {*access-list-number* | *name*} {**in**}
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	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 2/1`	Identifies a specific interface for configuration, and enter interface configuration mode. The interface can be a Layer 2 interface (port ACL).
Step 3	**ip access-group** {*access-list-number* | *name*} {**in**} **Example:** `Device(config-if)# ip access-group 2 in`	Controls access to the specified interface.
Step 4	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	
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 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.

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 20: 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

This section provides configuration examples for IPv4 ACLs.

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 gigabitethernet 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 gigabitethernet 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. The secure system of the network always accepts mail connections on port 25.

```
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 gigabitethernet 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 **ACK** or **RST** keywords are used to match ACK or RST bits set, which show that the packet belongs to an existing connection.

```
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 RST
```

```
Device(config)# access-list 102 permit tcp any host 128.88.1.2 eq 25
Device(config)# interface gigabitethernet 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
```

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
```

Example Resequencing Entries in an Access List

The following example shows an access list before and after resequencing. The starting value is 1, and increment value is 2. The subsequent entries are ordered based on the increment values that users provide, and the range is from 1 to 2147483647.

When an entry with no sequence number is entered, by default it has a sequence number of 10 more than the last entry in the access list.

```
Router# show access-list carls
Extended IP access list carls
    10 permit ip host 10.3.3.3 host 172.16.5.34
    20 permit icmp any any
    30 permit tcp any host 10.3.3.3
    40 permit ip host 10.4.4.4 any
    50 Dynamic test permit ip any any
```

```
    60 permit ip host 172.16.2.2 host 10.3.3.12
    70 permit ip host 10.3.3.3 any log
    80 permit tcp host 10.3.3.3 host 10.1.2.2
    90 permit ip host 10.3.3.3 any
    100 permit ip any any
Router(config)# ip access-list extended carls
Router(config)# ip access-list resequence carls 1 2
Router(config)# end
Router# show access-list carls
Extended IP access list carls
    1 permit ip host 10.3.3.3 host 172.16.5.34
    3 permit icmp any any
    5 permit tcp any host 10.3.3.3
    7 permit ip host 10.4.4.4 any
    9 Dynamic test permit ip any any
    11 permit ip host 172.16.2.2 host 10.3.3.12
    13 permit ip host 10.3.3.3 any log
    15 permit tcp host 10.3.3.3 host 10.1.2.2
    17 permit ip host 10.3.3.3 any
    19 permit ip any any
```

Example Adding an Entry with a Sequence Number

In the following example, an new entry (sequence number 15) is added to an access list:

```
Router# show ip access-list
Standard IP access list tryon
2 permit 10.4.4.2, wildcard bits 0.0.255.255
5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
20 permit 10.0.0.2, wildcard bits 0.0.0.255
Router(config)# ip access-list standard tryon
Router(config-std-nacl)# 15 permit 10.5.5.5 0.0.0.255
Router# show ip access-list
Standard IP access list tryon
2 permit 10.4.0.0, wildcard bits 0.0.255.255
5 permit 10.0.0.0, wildcard bits 0.0.0.255
10 permit 10.0.0.0, wildcard bits 0.0.0.255
15 permit 10.5.5.0, wildcard bits 0.0.0.255
20 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Example Adding an Entry with No Sequence Number

The following example shows how an entry with no specified sequence number is added to the end of an access list. When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list.

```
Router(config)# ip access-list standard resources
Router(config-std-nacl)# permit 10.1.1.1 0.0.0.255
Router(config-std-nacl)# permit 10.2.2.2 0.0.0.255
Router(config-std-nacl)# permit 10.3.3.3 0.0.0.255
Router# show access-list
Standard IP access list resources
10 permit 10.1.1.1, wildcard bits 0.0.0.255
20 permit 10.2.2.2, wildcard bits 0.0.0.255
30 permit 10.3.3.3, wildcard bits 0.0.0.255
Router(config)# ip access-list standard resources
```

```
Router(config-std-nacl)# permit 10.4.4.4 0.0.0.255
Router(config-std-nacl)# end
Router# show access-list
Standard IP access list resources
10 permit 10.1.1.1, wildcard bits 0.0.0.255
20 permit 10.2.2.2, wildcard bits 0.0.0.255
30 permit 10.3.3.3, wildcard bits 0.0.0.255
40 permit 10.4.4.4, wildcard bits 0.0.0.255
```

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: 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: 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 gigabitethernet 0/1
Device(config-if)# ip access-group strict in
```

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 gigabitethernet 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 gigabitethernet 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
```

Examples: Troubleshooting ACLs

If this ACL manager message appears and [chars] is the access-list name,

```
ACLMGR-2-NOVMR: Cannot generate hardware representation of access list [chars]
```

The switch has insufficient resources to create a hardware representation of the ACL. The resources include hardware memory and label space but not CPU memory. A lack of available logical operation units or specialized hardware resources causes this problem. Logical operation units are needed for a TCP flag match or a test other than **eq** (**ne**, **gt**, **lt**, or **range**) on TCP, UDP, or SCTP port numbers.

Use one of these workarounds:

- Modify the ACL configuration to use fewer resources.
- Rename the ACL with a name or number that alphanumerically precedes the ACL names or numbers.

For more information about configuring ACLs with insufficient resources, see CSCsq63926 in the Bug Toolkit.

For example, if you apply this ACL to an interface:

```
permit tcp source source-wildcard destination destination-wildcard range 5 60
permit tcp source source-wildcard destination destination-wildcard range 15 160
permit tcp source source-wildcard destination destination-wildcard range 115 1660
permit tcp source source-wildcard destination destination-wildcard
```

And if this message appears:

```
ACLMGR-2-NOVMR: Cannot generate hardware representation of access list [chars]
```

The flag-related operators are not available. To avoid this issue,

- Move the fourth ACE before the first ACE by using **ip access-list resequence** global configuration command:

  ```
  permit tcp source source-wildcard destination destination-wildcard
  permit tcp source source-wildcard destination destination-wildcard range 5 60
  permit tcp source source-wildcard destination destination-wildcard range 15 160
  permit tcp source source-wildcard destination destination-wildcard range 115 1660
  ```

  or

- Rename the ACL with a name or number that alphanumerically precedes the other ACLs (for example, rename ACL *79* to ACL *1*).

You can now apply the first ACE in the ACL to the interface. The switch allocates the ACE to available mapping bits in the Opselect index and then allocates flag-related operators to use the same bits in the hardware memory.

Additional References for IPv4 Access Control Lists

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)
ACLs	For more information, see: • "Access Control Lists Overview" in the *Security Configuration Guide* • "Configuring IPv6 Access Control Lists" in the *Security Configuration Guide*

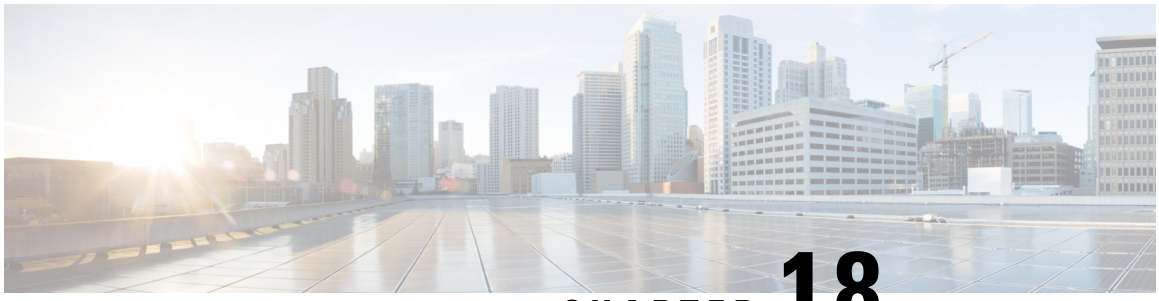
Feature History for IPv4 Access Control Lists

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 Release 15.2(5)E	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.



CHAPTER 18

IPv6 Access Control Lists

Access lists determine what traffic is blocked and what traffic is forwarded at device interfaces and allow filtering of traffic based on source and destination addresses, and inbound and outbound traffic to a specific interface. Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control. Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control.

This module describes how to configure IPv6 traffic filtering and to control access to virtual terminal lines.

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 does not support router ACL 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), 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 IPv6) are not 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 processed to the CPU, and the ACLs are applied in software.
- The switch supports IPv6 address-matching for a full range of prefix-lengths.

Information About Configuring IPv6 ACLs

Access lists determine what traffic is blocked and what traffic is forwarded at device interfaces and allow filtering of traffic based on source and destination addresses, and inbound and outbound traffic to a specific interface. Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control. Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control.

This module describes how to configure IPv6 traffic filtering and to control access to virtual terminal lines.

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 or VLANs. 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, including packets bridged within a VLAN.

You configure access lists on a router 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. ACLs can be configured to block inbound traffic, outbound traffic, or both.

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 apply both IPv4 and IPv6 ACLs to an interface.

Interactions with Other Features and Switches

- 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, 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 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, the packets associated with the ACL are processed to the CPU, and the ACLs are applied in software.

Default Configuration for IPv6 ACLs

The default IPv6 ACL configuration is as follows:

```
Device# 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
```

Supported ACL Features

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 TCAM space, packets associated with the ACL label are forwarded to the CPU, and the ACLs are applied in software.

IPv6 Port-Based Access Control List Support

The IPv6 PACL feature provides the ability to provide access control (permit or deny) on Layer 2 switch ports for IPv6 traffic. IPv6 PACLs are similar to IPv4 PACLs, which provide access control on Layer 2 switch ports for IPv4 traffic. They are supported only in the ingress direction and in hardware.

ACLs and Traffic Forwarding

The IPv6 ACL Extensions for Hop by Hop Filtering feature allows you to control IPv6 traffic that might contain hop-by-hop extension headers. You can configure an access control list (ACL) to deny all hop-by-hop traffic or to selectively permit traffic based on protocol.

IPv6 access control lists (ACLs) determine what traffic is blocked and what traffic is forwarded at device interfaces. ACLs allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Use the **ipv6 access-list** command to define an IPv6 ACL, and the **deny** and **permit** commands to configure its conditions.

The IPv6 ACL Extensions for Hop by Hop Filtering feature implements RFC 2460 to support traffic filtering in any upper-layer protocol type.

How to Configure IPv6 ACLs

This section provides information about how to configure IPv6 ACLs.

Configuring IPv6 ACLs

To filter IPv6 traffic, you perform these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. {**ipv6 access-list** *list-name*
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*]
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*] [**fin**] [**log**] [**log-input**] [**neq** {**port** | protocol}] [**psh**] [**range** {**port** | protocol}] [**rst**] [**sequence** *value*] [**syn**] [**time-range** *name*] [**urg**]
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*]]
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*]
8. **end**
9. **show ipv6 access-list**
10. **show running-config**
11. **copy running-config startup-config**

DETAILED STEPS

	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	{**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 Internet protocol: **ahp**, **esp**, **icmp**, **ipv6**, **pcp**, **stcp**, **tcp**, or **udp**, or an integer in the range 0 to 255 representing an IPv6 protocol number. • 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

	Command or Action	Purpose
		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 the packet that matches the entry. Enter **log-input** to include the input interface in the log entry. • (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*] [**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. • **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

	Command or Action	Purpose
		that the [operator [*port*]] port number or name must be a UDP port number or name.
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.

What to do next

Attach the IPv6 ACL to an Interface

Attaching an IPv6 ACL to an Interface

You can apply an ACL to inbound traffic on Layer 2 interfaces.

Follow these steps to control access to an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ipv6 address** *ipv6-address*
5. **ipv6 traffic-filter** *access-list-name* **in**
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

DETAILED STEPS

	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 on which to apply an access list, and enter interface configuration mode.
Step 4	**ipv6 address** *ipv6-address*	This command is not required on Layer 2 interfaces or if the interface has already been configured with an explicit IPv6 address.
Step 5	**ipv6 traffic-filter** *access-list-name* **in**	Apply the access list to incoming traffic on the interface. **Note** The out keyword is not supported for Layer 2 interfaces (port ACLs).
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.

	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.

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:

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.

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.

```
Device# 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.

```
Device# 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
```

Configuring PACL Mode and Applying IPv6 PACL on an Interface

Before you begin

Before you configure the IPv6 PACL feature, you must configure an IPv6 access list. Once you have configured the IPv6 access list, you must configure the port-based access control list (PACL) mode on the specified IPv6 Layer 2 interface.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. **exit**
5. **interface** *type number*
6. **ipv6 traffic-filter** *access-list-name* **in**
7. **end**

DETAILED STEPS

	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** *access-list-name* **Example:** `Device(config)# ipv6 access-list list1`	Defines an IPv6 ACL and enters IPv6 access list configuration mode.
Step 4	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 5	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Specifies an interface type and number and enters interface configuration mode.
Step 6	**ipv6 traffic-filter** *access-list-name* **in** **Example:** `Device(config-if)# ipv6 traffic-filter list1 in`	Filters incoming IPv6 traffic on an interface.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring IPv6 ACL Extensions for Hop by Hop Filtering

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ipv6 access-list** *access-list-name*
4. **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*] [**hbh**] [**log**] [**log-input**] [**reflect** *name* [**timeout** *value*]] [**sequence** *value*] [**time-range** *name*]
5. **deny** *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*] [**hbh**] [**log**] [**log-input**] [**sequence** *value*] [**time-range** *name*]
6. **end**

DETAILED STEPS

	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** *access-list-name* **Example:** `Device(config)# ipv6 access-list hbh-acl`	Defines an IPv6 ACL and enters IPv6 access list configuration mode.
Step 4	**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*] [**hbh**] [**log**] [**log-input**] [**reflect** *name* [**timeout** *value*]] [**sequence** *value*] [**time-range** *name*] **Example:** `Device(config-ipv6-acl)# permit icmp any any`	Sets permit conditions for the IPv6 ACL.
Step 5	**deny** *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*] [**hbh**] [**log**] [**log-input**] [**sequence** *value*] [**time-range** *name*] **Example:** `Device(config-ipv6-acl)# deny icmp any any`	Sets deny conditions for the IPv6 ACL.
Step 6	**end** **Example:** `Device (config-ipv6-acl)# end`	Returns to privileged EXEC configuration mode.

Configuration Examples for IPv6 ACLs

This section provides configuration examples for IPv6 ACLs.

Example: Configuring IPv6 ACLs

This example configures the IPv6 access list named CISCO. The first deny entry in the list denies all packets that have a destination TCP port number greater than 5000. The second deny entry denies packets that have a source UDP port number less than 5000. The second deny also logs all matches to the console. The first permit entry in the list permits all ICMP packets. The second permit entry in the list permits all other traffic. The second permit entry is necessary because an implicit deny -all condition is at the end of each IPv6 access list.

```
Device(config)# ipv6 access-list CISCO
Device(config-ipv6-acl)# deny tcp any any gt 5000
Device(config-ipv6-acl)# deny ::/0 lt 5000 ::/0 log
Device(config-ipv6-acl)# permit icmp any any
Device(config-ipv6-acl)# permit any any
```

Example: Configuring PACL Mode and Applying IPv6 PACL on an Interface

```
Device# configure terminal

Device(config)# ipv6 access-list list1

Device(config-ipv6-acl)# exit

Device(config-if)# ipv6 traffic-filter list1 in
```

Example: IPv6 ACL Extensions for Hop by Hop Filtering

```
Device(config)# ipv6 access-list hbh_acl
Device(config-ipv6-acl)# permit tcp any any hbh
Device(config-ipv6-acl)# permit tcp any any
Device(config-ipv6-acl)# permit udp any any
Device(config-ipv6-acl)# permit udp any any hbh
Device(config-ipv6-acl)# permit hbh any any
Device(config-ipv6-acl)# permit any any
Device(config-ipv6-acl)# exit

! Assign an IP address and add the ACL on the interface.

Device(config)# interface gigabitethernet 0/1
Device(config-if)# ipv6 address 1001::1/64
Device(config-if)# ipv6 traffic-filter hbh_acl in
Device(config-if)# exit
Device(config)# exit
Device# clear counters
Clear "show interface" counters on all interfaces [confirm]
Device#

! Verify the configurations.

Device# show running-config interface gigabitethernet 0/1
```

```
Building configuration...

Current configuration : 114 bytes
!
interface gigabitethernet 0/1
no switchport
ipv6 address 1001::1/64
ipv6 traffic-filter hbh_acl
end
```

Additional References for IPv6 Access Control Lists

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)
ACLs	For more information, see: • "Access Control Lists Overview" in the *Security Configuration Guide* • "Configuring IPv4 Access Control Lists" in the *Security Configuration Guide*

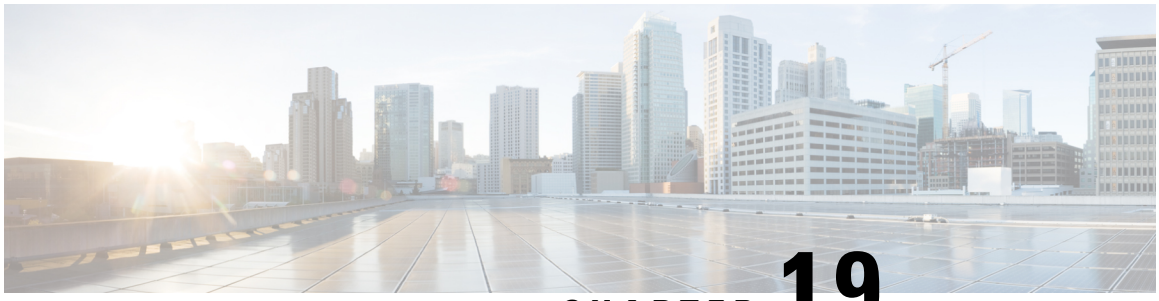
Feature History for IPv6 Access Control Lists

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 Release 15.2(5)E	IPv6 Access Control Lists	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.



CHAPTER 19

Configuring IPv6 RA Guard

Restrictions for IPv6 Router Advertisement Guard

- The IPv6 Router Advertisement Guard feature does not offer protection in environments where IPv6 traffic is tunneled.
- This feature is supported only in hardware when the ternary content addressable memory (TCAM) is programmed.
- This feature can be configured on a switch port interface in the ingress direction.
- This feature supports host mode and router mode.
- This feature is supported only in the ingress direction; it is not supported in the egress direction.
- This feature is not supported on EtherChannel and EtherChannel port members.
- This feature is not supported on trunk ports with merge mode.
- This feature is supported on auxiliary VLANs.
- Packets dropped by the IPv6 Router Advertisement Guard feature can be spanned.

Information About IPv6 Router Advertisement Guard

The following sections provide information on IPv6 global policies and IPv6 router advertisement guard.

About IPv6 Global Policies

IPv6 global policies provide storage and access policy database services. IPv6 neighbor discovery inspection and IPv6 router advertisement guard are IPv6 global policies features. Every time an neighbor discovery

inspection or router advertisement guard is configured globally, the policy attributes are stored in the software policy database. The policy is then applied to an interface, and the software policy database entry is updated to include this interface to which the policy is applied.

About IPv6 Router Advertisement Guard

The IPv6 Router Advertisement Guard feature provides support for allowing the network administrator to block or reject unwanted or rogue router advertisement guard messages that arrive at the network device platform. Router Advertisements are used by devices to announce themselves on the link. The IPv6 Router Advertisement Guard feature analyzes these router advertisements and filters out router advertisements that are sent by unauthorized devices. In host mode, all router advertisement and router redirect messages are disallowed on the port. The router advertisement guard feature compares configuration information on the Layer 2 device with the information found in the received router advertisement frame. Once the Layer 2 device has validated the content of the router advertisement frame and router redirect frame against the configuration, it forwards the router advertisement to its unicast or multicast destination. If the router advertisement frame content is not validated, the router advertisement is dropped.

In the wireless deployment router advertisements coming on wireless ports are dropped as routers cannot reside on these interfaces.

How to Configure IPv6 Router Advertisement Guard

The following section provides information on configuring IPv6 router advertisement guard policy on a device and configuring router advertisement on an interface

Configuring the IPv6 Router Advertisement Guard Policy on the Device

To configure IPv6 router advertisement guard policy on the device, 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 nd raguard policy** *policy-name* **Example:** `Device(config)# ipv6 nd raguard policy policy1`	Defines the router advertisement guard policy name and enters router advertisement guard policy configuration mode.
Step 4	**device-role** {**host** | **router**} **Example:**	Specifies the role of the device attached to the port.

	Command or Action	Purpose
	`Device(config-ra-guard)# device-role router`	
Step 5	**hop-limit** {**maximum** | **minimum** *limit*} **Example:** `Device(config-ra-guard)# hop-limit minimum 3`	(Optional) Enables verification of the advertised hop count limit. If not configured, this check will be bypassed.
Step 6	**managed-config-flag** {**on** | **off**} **Example:** `Device(config-ra-guard)# managed-config-flag on`	(Optional) Enables verification that the advertised managed address configuration flag is on. If not configured, this check will be bypassed.
Step 7	**match ipv6 access-list** *ipv6-access-list-name* **Example:** `Device(config-ra-guard)# match ipv6 access-list list1`	(Optional) Enables verification of the sender's IPv6 address in inspected messages from the configured authorized device source access list. If not configured, this check will be bypassed.
Step 8	**match ra prefix-list** *ipv6-prefix-list-name* **Example:** `Device(config-ra-guard)# match ra prefix-list listname1`	(Optional) Enables verification of the advertised prefixes in inspected messages from the configured authorized prefix list. If not configured, this check will be bypassed.
Step 9	**other-config-flag** {**on** | **off**} **Example:** `Device(config-ra-guard)# other-config-flag on`	(Optional) Enables verification of the advertised "other" configuration parameter.
Step 10	**router-preference maximum** {**high** | **low** | **medium**} **Example:** `Device(config-ra-guard)# router-preference maximum high`	(Optional) Enables verification that the advertised default router preference parameter value is lower than or equal to a specified limit.
Step 11	**trusted-port** **Example:** `Device(config-ra-guard)# trusted-port`	(Optional) Specifies that this policy is being applied to trusted ports. All router advertisement guard policing will be disabled.
Step 12	**exit** **Example:** `Device(config-ra-guard)# exit`	Exits router advertisement guard policy configuration mode and returns to global configuration mode.

Configuring IPv6 Router Advertisement Guard on an Interface

To configure IPv6 router advertisement guard on an interface, perform this procedure:

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	**interface** *type number* **Example:** `Device(config)# interface fastethernet 3/13`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 nd raguard attach-policy** [*policy-name* [**vlan** {**add** | **except** | **none** | **remove** | **all**} *vlan* [*vlan1, vlan2, vlan3...*]]] **Example:** `Device(config-if)# ipv6 nd raguard attach-policy`	Applies the IPv6 Router Advertisement Guard feature to a specified interface.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 6	**show ipv6 nd raguard policy** [*policy-name*] **Example:** `Device# show ipv6 nd raguard policy raguard1`	Displays the router advertisement guard policy on all interfaces configured with the router advertisement guard.
Step 7	**debug ipv6 snooping raguard** [*filter* | *interface* | *vlanid*] **Example:** `Device# debug ipv6 snooping raguard`	Enables debugging for IPv6 router advertisement guard snooping information.

Configuration Examples for IPv6 Router Advertisement Guard

The following sections provide configuration examples for IPv6 router advertisement guard.

Example: Configuring IPv6 Router Advertisement Guard

The following example shows how to configure IPv6 router advertisement guard:

```
Device> enable
Device# configure terminal
Device(config)# interface fastethernet 3/13
Device(config-if)# ipv6 nd raguard attach-policy
Device# show running-config interface fastethernet 3/13

Building configuration...
Current configuration : 129 bytes
!
interface FastEthernet3/13
```

```
 switchport
 switchport access vlan 222
 switchport mode access
 access-group mode prefer port
 ipv6 nd raguard
end
```

Example: Viewing IPv6 Neighbor Discovery Inspection and Router Advertisement Guard Configurations on an Interface

The following example shows information about an interface on which both the neighbor discovery inspection and router advertisement guard are configured.

```
Device> enable
Device# show ipv6 snooping capture-policy interface ethernet 0/0

Hardware policy registered on Ethernet 0/0
Protocol     Protocol value   Message   Value     Action    Feature
ICMP         58               RS        85        punt      RA Guard
                                                  punt      ND Inspection
ICMP         58               RA        86        drop      RA guard
                                                  punt      ND Inspection
ICMP         58               NS        87        punt      ND Inspection
ICM          58               NA        88        punt      ND Inspection
ICMP         58               REDIR     89        drop      RA Guard
                                                  punt      ND Inspection
```

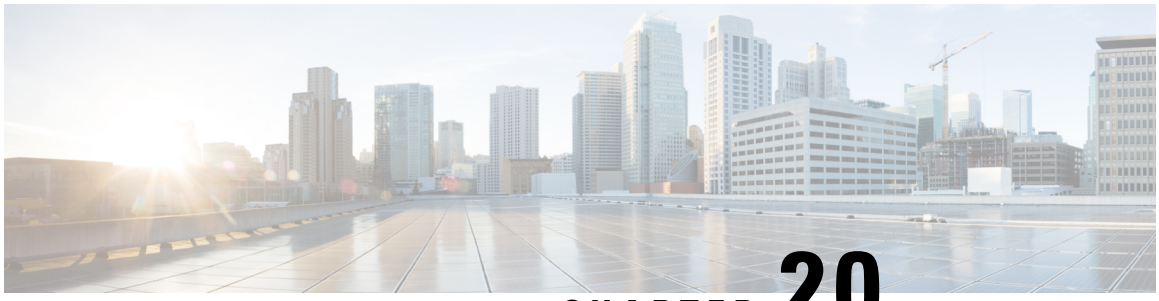
Feature Information for Configuring IPv6 Router Advertisement Guard

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 21: Feature Information for Configuring IPv6 Router Advertisement Guard

Feature Name	Releases	Feature Information
Configuring IPv6 Router Advertisement Guard	Cisco IOS Release 15.2(5)E	The feature was introduced.



CHAPTER 20

Configuring IP Source Guard

Information About IP Source Guard

This section provides 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.

You can configure IP Source Guard on EtherChannel interfaces.

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. When you enter the **show ip device tracking all** EXEC 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 vendor 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

This section provides information about how to configure IP source guard.

Enabling IP Source Guard

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip verify source** [**port-security**]
5. **exit**
6. **ip source binding** *mac-address* **vlan** *vlan-id ip-address* **interface** *interface-id*
7. **end**
8. **show running-config**
9. **copy running-config startup-config**

DETAILED STEPS

	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 0/1`	Specifies the interface to be configured, and enters interface configuration mode.
Step 4	**ip verify source** [**port-security**] **Example:** `Device(config-if)# ip verify source`	Enables IP source guard with source IP address filtering. (Optional) **port-security**: Enables IP Source Guard with source IP address and MAC address filtering.

	Command or Action	Purpose
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 gigabitethernet 2/1`	Adds a static IP source binding. Enter this command for each static binding.
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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip device tracking**
4. **interface** *interface-id*
5. **switchport mode access**
6. **switchport access vlan** *vlan-id*
7. **ip verify source**[**tracking**] [**port-security**]

8. **ip device tracking maximum** *number*
9. **end**

DETAILED STEPS

	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.
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 0/1`	Enters interface configuration mode.
Step 5	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Configures a port as access.
Step 6	**switchport access vlan** *vlan-id* **Example:** `Device(config-if)# switchport access vlan 10`	Configures the VLAN for this port.
Step 7	**ip verify source**[**tracking**] [**port-security**] **Example:** `Device(config-if)# ip verify source tracking port-security`	Enables IP source guard with source IP address filtering. (Optional) **tracking**—Enables IP source guard for static hosts. (Optional) **port-security**—Enables MAC address filtering. The command **ip verify source tracking port-security**enables IP source guard for static hosts with MAC address filtering.

	Command or Action	Purpose
Step 8	**ip device tracking maximum** *number* **Example:** `Device(config-if)# ip device tracking maximum 8`	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. **Note** You must configure the **ip device tracking maximum** *limit-number* interface configuration command.
Step 9	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Monitoring IP Source Guard

Table 22: 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 23: 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.

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
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

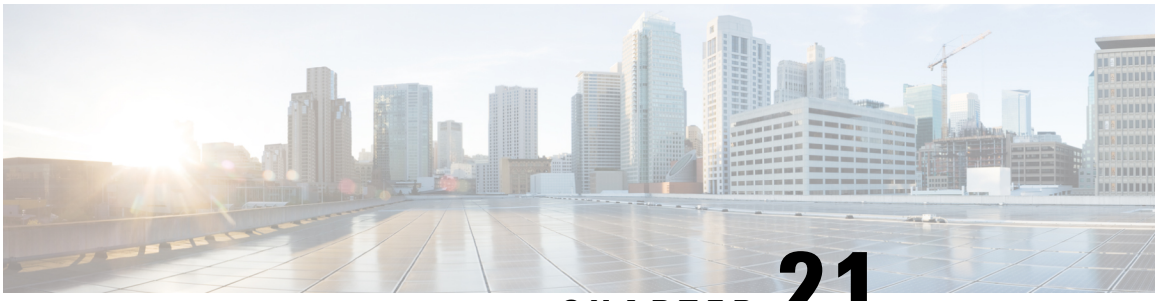
Feature Information for IP Source Guard

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 24: Feature Information for AAA-SERVER-MIB Set Operation

Feature Name	Releases	Feature Information
IP Source Guard	Cisco IOS Release 15.2(5)E	The feature was introduced.



CHAPTER 21

Configuring IEEE 802.1x Port-Based Authentication

Prerequisites for 802.1x Port-Based Authentication

The following tasks must be completed before implementing the IEEE 802.1X Port-Based Authentication feature:

- IEEE 802.1X must be enabled on the device port.
- The device must have a RADIUS configuration and be connected to the Cisco secure access control server (ACS). You should understand the concepts of the RADIUS protocol and have an understanding of how to create and apply access control lists (ACLs).
- EAP support must be enabled on the RADIUS server.
- You must configure the IEEE 802.1X supplicant to send an EAP-logoff (Stop) message to the device when the user logs off. If you do not configure the IEEE 802.1X supplicant, an EAP-logoff message is not sent to the device and the accompanying accounting Stop message is not sent to the authentication server.
- Authentication, authorization, and accounting (AAA) must be configured on the port for all network-related service requests. The authentication method list must be enabled and specified. A method list describes the sequence and authentication method to be queried to authenticate a user.
- The port must be successfully authenticated.

Information About IEEE 802.1x Port-Based Authentication

802.1x Port-Based Authentication Overview

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 device 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, 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.

The following table below the maximum number of each client session supported:

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 device 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 device can use the client MAC address for authorization. If the client MAC address is valid and the authorization succeeds, the device grants the client access to the network. If the client MAC address is invalid and the authorization fails, the device assigns the client to a guest VLAN that provides limited services if a guest VLAN is configured.

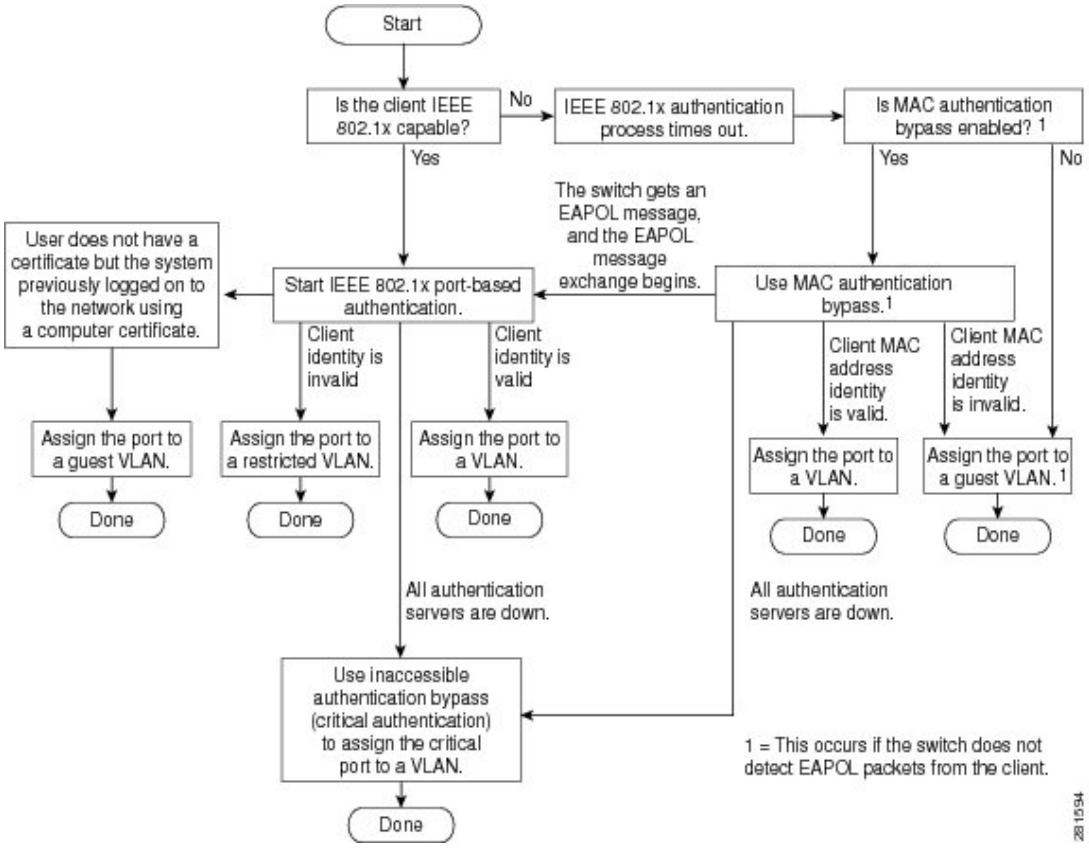
- If the device gets an invalid identity from an 802.1x-capable client and a restricted VLAN is specified, the device can assign the client to a restricted VLAN that provides limited services.
- If the RADIUS authentication server is unavailable (down) and inaccessible authentication bypass is enabled, the device 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 6: Authentication Flowchart

This figure shows the authentication process.



The device 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 device-specific value or to be based on values from the RADIUS server.

After 802.1x authentication using a RADIUS server is configured, the device 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 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 device or the client can initiate authentication. If you enable authentication on a port by using the **authentication port-control auto** interface configuration command, the device initiates authentication when the link state changes from down to up or periodically as long as the port remains up and unauthenticated. The device 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 device, the client can initiate authentication by sending an EAPOL-start frame, which prompts the device to request the client's identity.

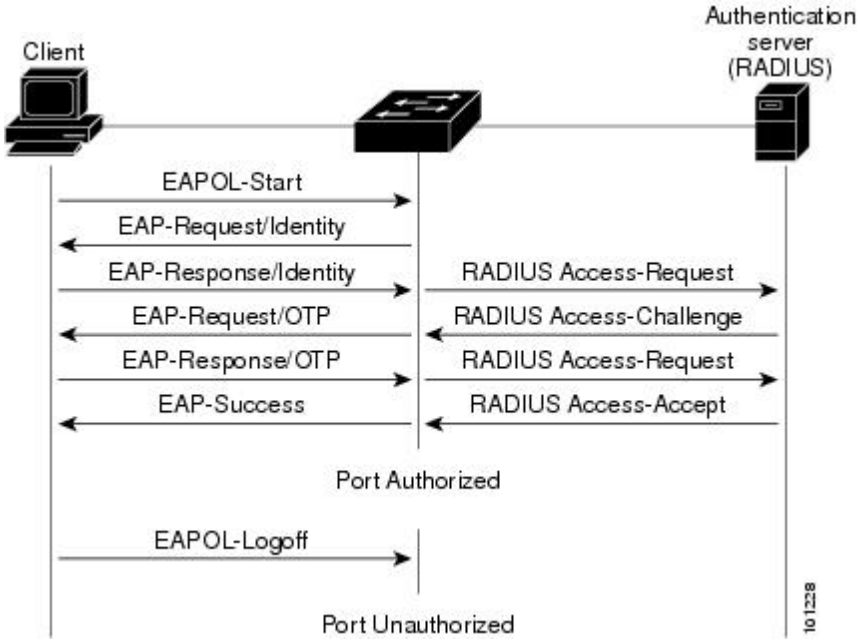
Note If 802.1x authentication is not enabled or supported on the network access device, any EAPOL frames from the client are dropped. If the client does not receive an EAP-request/identity frame after three attempts to start authentication, the client sends frames as if the port is in the authorized state. A port in the authorized state effectively means that the client has been successfully authenticated.

When the client supplies its identity, the device 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 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 7: Message Exchange

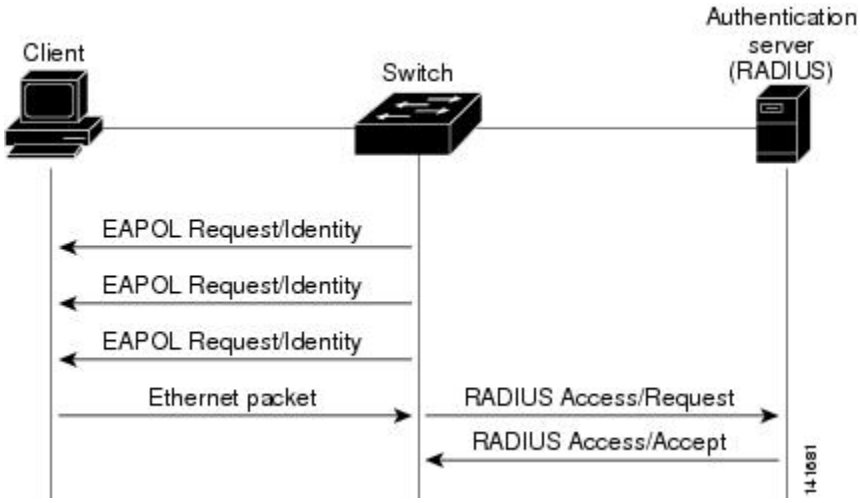
This figure shows a message exchange initiated by the client when the client uses the One-Time-Password (OTP) authentication method with a RADIUS server.



If 802.1x authentication times out while waiting for an EAPOL message exchange and MAC authentication bypass is enabled, the device can authorize the client when the device detects an Ethernet packet from the client. The device 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 device the RADIUS-access/accept frame (authorization is successful), the port becomes authorized. If authorization fails and a guest VLAN is specified, the device assigns the port to the guest VLAN. If the device detects an EAPOL packet while waiting for an Ethernet packet, the device stops the MAC authentication bypass process and starts 802.1x authentication.

Figure 8: Message Exchange During MAC Authentication Bypass

This figure shows the message exchange during MAC authentication bypass.



Port-Based Authentication Methods

Table 25: 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
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 **Note** For clients that do not support 802.1x authentication.	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

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.

These 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.

To disable dot1x on a device, 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

Per-User ACLs and Filter-IDs

Note You can only set **any** as the source in the ACL.

Note For any ACL that is configured for multiple-host mode, the source portion of statement must be *any*. (For example, **permit icmp** *any* **host 10.10.1.1**.)

Note Using role-based ACLs as filter-ID is not recommended.

You must specify **any** in the source ports of any defined ACL. Otherwise, the ACL cannot be applied and authorization fails. Single host is the only exception to support backward compatibility.

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.

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, Cisco Discovery Protocol,

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 Cisco Discovery Protocol 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.

802.1x Host Mode

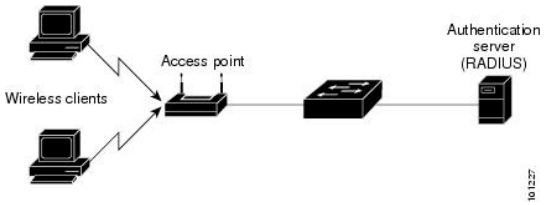
You can configure an 802.1x port for single-host or for multiple-hosts mode. In single-host mode, only one client can be connected to the 802.1x-enabled port. The device detects the client by sending an EAPOL frame when the port link state changes to the up state. If a client leaves or is replaced with another client, the device 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 device denies network access to all of the attached clients.

In this topology, the wireless access point is responsible for authenticating the clients attached to it, and it also acts as a client to the device.

Figure 9: Multiple Host Mode Example



Note For all host modes, the line protocol stays up before authorization when port-based authentication is configured.

The device 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 device port.

802.1x Multiple Authentication Mode

Multiple-authentication (multi-auth) 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 multi-auth port.

If a hub or access point is connected to an 802.1x-enabled port, each connected client must be authenticated. For non-802.1x devices, you can use MAC authentication bypass or web authentication as the per-host authentication fallback method to authenticate different hosts with different methods on a single port.

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.

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 re-authenticated on the new port. When a host moves to a second port, the session on the first port is deleted, and the host is re-authenticated 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 multidomain 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 device 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 device that is configured for 802.1x accounting. Three types of RADIUS accounting packets are sent by a device:

- START: Sent when a new user session starts
- INTERIM: Sent during an existing session for updates
- STOP: Sent when a session terminates

You can view the AV pairs that are being sent by the device by entering the **debug radius accounting** privileged EXEC command.

This table lists the AV pairs and when they are sent are sent by the device.

Table 26: 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[5]	NAS-Port	Always	Always	Always
Attribute[8]	Framed-IP-Address	Never	Sometimes[1]	Sometimes
Attribute[25]	Class	Always	Always	Always
Attribute[30]	Called-Station-ID	Always	Always	Always
Attribute[31]	Calling-Station-ID	Always	Always	Always

Attribute Number	AV Pair Name	START	INTERIM	STOP
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

[1] 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.

Device-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

These are the 802.1x authentication configuration guidelines:

- 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 device. 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) destination ports: You can enable 802.1x authentication on a port that is a SPAN destination port. However, 802.1x authentication is disabled until the port is removed as a SPAN destination port. You can enable 802.1x authentication on a SPAN source port.

- Before globally enabling 802.1x authentication on a device 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.

Default 802.1x Authentication Configuration

Table 27: Default 802.1x Authentication Configuration

Feature	Default Setting
Device 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.
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 device restarts the authentication process before the port changes to the unauthorized state).

Feature	Default Setting
Quiet period	60 seconds (number of seconds that the device remains in the quiet state following a failed authentication exchange with the client).
Retransmission time	30 seconds (number of seconds that the device 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 device 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 device 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 device 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.

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 these features, 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: Multi-authentication 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.

802.1x Authentication with VLAN Assignment

The device 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 device port. The RADIUS server database maintains the username-to-VLAN mappings, assigning the VLAN based on the username of the client connected to the device port. You can use this feature to limit network access for certain users.

Voice device authentication is supported with multidomain host mode. 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 device 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 multidomain host mode.

When the port is in the force authorized, force unauthorized, unauthorized, or shutdown state, it is put into the configured access 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, 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 multidomain 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 device:
  - [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 Guest VLAN

You can configure a guest VLAN for each 802.1x port on the device 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 device assigns clients to a guest VLAN when the device does not receive a response to its EAP request/identity frame or when EAPOL packets are not sent by the client.

The device maintains the EAPOL packet history. If an EAPOL packet is detected on the interface during the lifetime of the link, the device 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 device 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 device authorizes the voice device. However, the device 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 device during the lifetime of the link, the device 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 device 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.

The device supports *MAC authentication bypass*. When MAC authentication bypass is enabled on an 802.1x port, the device 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 device waits for an Ethernet packet from the client. The device sends the authentication server a RADIUS-access/request frame with a username and password based on the MAC address. If authorization succeeds, the device grants the client access to the network. If authorization fails, the device 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 device 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 device port remains in the spanning-tree blocking state. With this feature, you can configure the device 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.

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 Auth Fail VLAN

You can configure an auth fail VLAN for each 802.1X port on a device 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. An auth fail 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 auth fail VLAN.

Note You can configure a VLAN to be both the guest VLAN and the auth fail 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 device port remains in the spanning-tree blocking state. With this feature, you can configure the device port to be in the auth fail 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 auth fail 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 auth fail VLAN, the failed attempt counter resets.

Users who fail authentication remain in the auth fail VLAN until the next re-authentication attempt. A port in the auth fail VLAN tries to re-authenticate at configured intervals (the default is 60 seconds). If re-authentication fails, the port remains in the auth fail 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. It is recommended 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 auth fail VLAN, a simulated EAP success message is sent to the client. This prevents clients from indefinitely attempting authentication.

As a prerequisite, the device must be connected to a Cisco secure Access Control System (ACS) and RADIUS authentication, authorization, and accounting (AAA) must be configured for Web authentication. If appropriate, you must enable ACL download.

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.

Limiting Login for Users

The Limiting Login feature helps network administrators to limit the login attempt of users to a network. When a user fails to successfully login to a network within a configurable number of attempts within a configurable time limit, this user can be blocked. This feature is enabled only for local users and not for remote users. You need to configure the **aaa authentication rejected** command in global configuration mode to enable this feature.

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 device cannot reach the configured RADIUS servers and new hosts cannot be authenticated. You can configure the device 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 device tries to authenticate a host connected to a critical port, the device checks the status of the configured RADIUS server. If a server is available, the device can authenticate the host. However, if all the RADIUS servers are unavailable, the device 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 device. If the *critical vlan* is inactive (or) down, *critical authentication* session will keep trying to enable the inactive VLAN and fail repeatedly. This can lead to large amount of memory holding.

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 device puts the port in the critical-authentication state in the RADIUS-configured or user-specified access VLAN.
- If the port is already authorized and re-authentication occurs, the device 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 device 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 device assigns a client to a guest VLAN when the device 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 device 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 device 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 device 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 device 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.
- 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.

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 multihost mode and moved to the critical VLAN. To support this inaccessible bypass on multiple-authentication (multi-auth) 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.

VLAN Assignment, Guest VLAN, Restricted VLAN, and Inaccessible Authentication Bypass

The following are 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.
- The 802.1x authentication with VLAN assignment feature is not supported on trunk ports, or dynamic ports.
- You can configure any VLAN except 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 device 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.

- You can configure the inaccessible authentication bypass feature and the restricted VLAN on an 802.1x port. If the device tries to re-authenticate a critical port in a restricted VLAN and all the RADIUS servers are unavailable, the device changes the port state to the critical authentication state and remains in the restricted VLAN.

- You can configure any VLAN except 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.

IEEE 802.1x Authentication with MAC Authentication Bypass

You can configure the device 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 device 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 device 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 device waits for an Ethernet packet from the client. The device sends the authentication server a RADIUS-access/request frame with a username and password based on the MAC address. If authorization succeeds, the device grants the client access to the network. If authorization fails, the device 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 device determines that the device connected to that interface is an 802.1x-capable supplicant and uses 802.1x authentication (not MAC authentication bypass) to authorize the interface. EAPOL history is cleared if the interface link status goes down.

If the device already authorized a port by using MAC authentication bypass and detects an IEEE 802.1x supplicant, the device does not unauthorize the client connected to the port. When re-authentication occurs, the device 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 device keeps the port in the same VLAN. If re-authentication fails, the device 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 device 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).”

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 device 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
- 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.

MAC Authentication Bypass Guidelines

This section describes 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 device 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

The maximum number of devices allowed on an 802.1x-enabled port are as follows:

- 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.

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 device 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 Cisco Discovery Protocol message from the IP phone. Cisco IP phones do not relay Cisco Discovery Protocol messages from other devices. As a result, if several IP phones are connected in series, the device recognizes only the one directly connected to it. When IEEE 802.1x authentication is enabled on a voice VLAN port, the device drops packets from unrecognized IP phones more than one hop away.

When IEEE 802.1x authentication is enabled on a device port, you can configure an access port VLAN that is also a voice VLAN.

When IP phones are connected to an 802.1x-enabled device port that is in single host mode, the device 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 device for up to 30 seconds.

IEEE 802.1x Authentication with Port Security

IEEE 802.1x enforces a single MAC address per port (or per VLAN when MDA is configured for IP telephony), port security is redundant and in some cases may interfere with expected IEEE 802.1x operations.

We do not recommend enabling port security when IEEE 802.1x is enabled.

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 device 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 device can use the client MAC address for authorization. If the client MAC address is valid and the authorization succeeds, the device grants the client access to the network. If the client

MAC address is invalid and the authorization fails, the device assigns the client to a guest VLAN that provides limited services if a guest VLAN is configured.

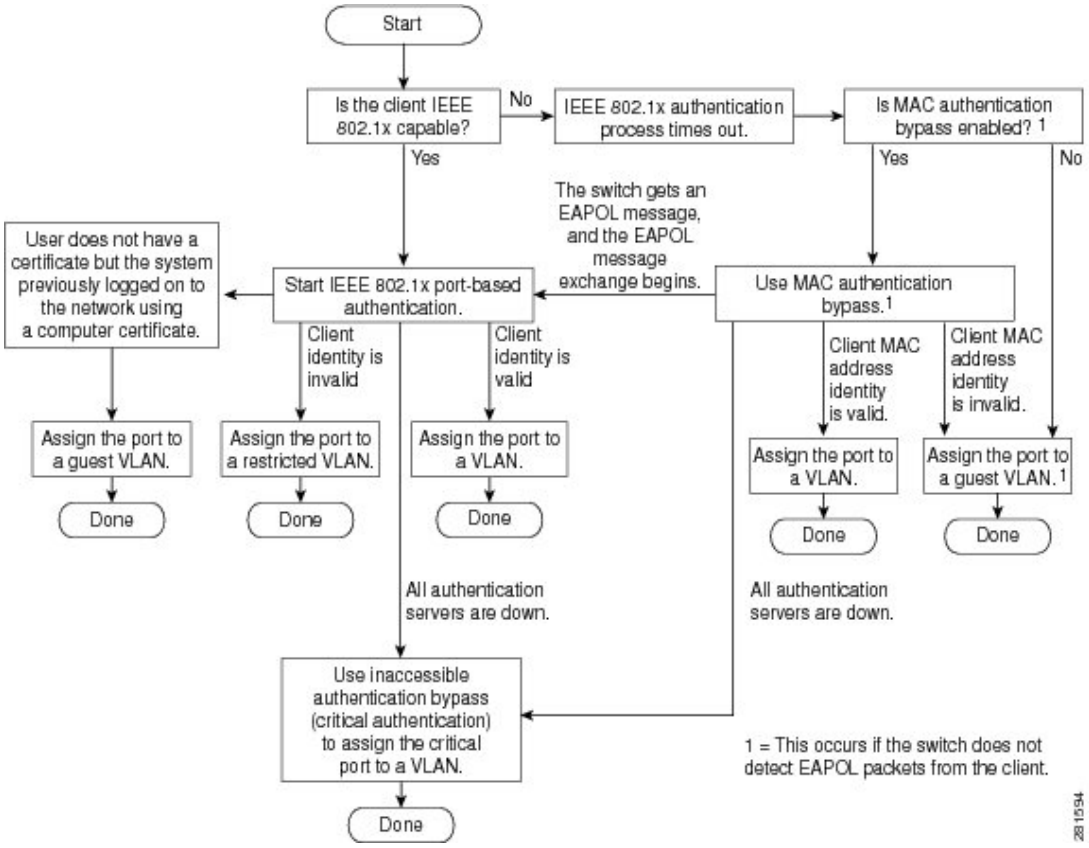
- If the device gets an invalid identity from an 802.1x-capable client and a restricted VLAN is specified, the device can assign the client to a restricted VLAN that provides limited services.
- If the RADIUS authentication server is unavailable (down) and inaccessible authentication bypass is enabled, the device 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 10: Authentication Flowchart

This figure shows the authentication process.



The device 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 device-specific value or to be based on values from the RADIUS server.

After 802.1x authentication using a RADIUS server is configured, the device 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 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 device or the client can initiate authentication. If you enable authentication on a port by using the **authentication port-control auto** interface configuration command, the device initiates authentication when the link state changes from down to up or periodically as long as the port remains up and unauthenticated. The device 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 device, the client can initiate authentication by sending an EAPOL-start frame, which prompts the device to request the client's identity.

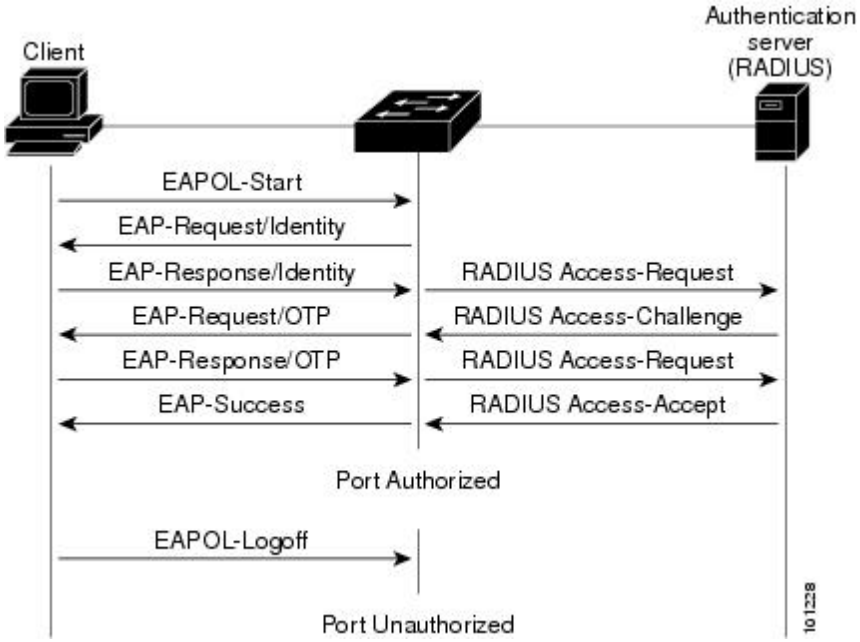
Note If 802.1x authentication is not enabled or supported on the network access device, any EAPOL frames from the client are dropped. If the client does not receive an EAP-request/identity frame after three attempts to start authentication, the client sends frames as if the port is in the authorized state. A port in the authorized state effectively means that the client has been successfully authenticated.

When the client supplies its identity, the device 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 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 11: Message Exchange

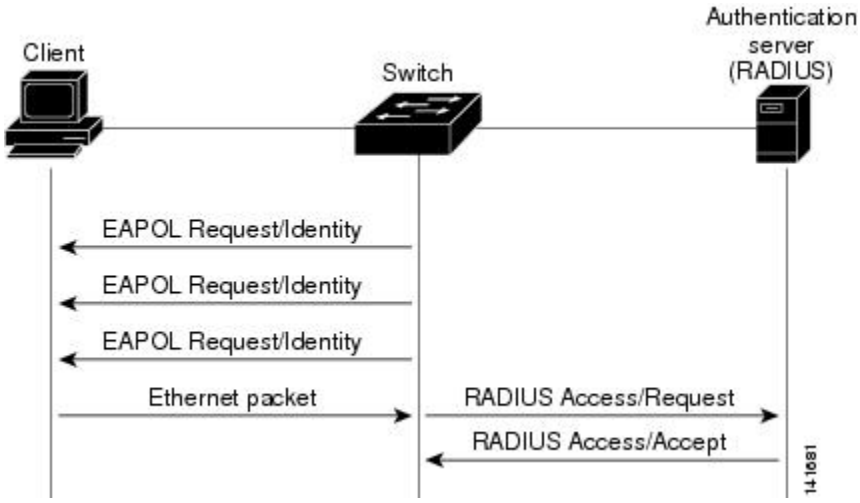
This figure shows a message exchange initiated by the client when the client uses the One-Time-Password (OTP) authentication method with a RADIUS server.



If 802.1x authentication times out while waiting for an EAPOL message exchange and MAC authentication bypass is enabled, the device can authorize the client when the device detects an Ethernet packet from the client. The device 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 device the RADIUS-access/accept frame (authorization is successful), the port becomes authorized. If authorization fails and a guest VLAN is specified, the device assigns the port to the guest VLAN. If the device detects an EAPOL packet while waiting for an Ethernet packet, the device stops the MAC authentication bypass process and starts 802.1x authentication.

Figure 12: Message Exchange During MAC Authentication Bypass

This figure shows the message exchange during MAC authentication bypass.



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 device 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 device 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.

802.1x Supplicant and Authenticator Devices with Network Edge Access Topology

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 device by using the 802.1x supplicant feature. This configuration is helpful in a scenario, where, for example, a device is outside a wiring closet and is connected to an upstream device through a trunk port. A device configured with the 802.1x device supplicant feature authenticates with the upstream device for secure connectivity.

Once the supplicant device authenticates successfully the port mode changes from access to trunk in an authenticator device. In a supplicant device you must manually configure trunk when enabling CISP.

- If the access VLAN is configured on the authenticator device, it becomes the native VLAN for the trunk port after successful authentication.

In the default state, when you connect a supplicant device to an authenticator device 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 device has authenticated. 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 transient** command on a supplicant device when BPDU guard is enabled on the authenticator device port with the **spanning-tree bpduguard enable** interface configuration command.

Note If you globally enable BPDU guard on the authenticator device by using the **spanning-tree portfast bpduguard default** global configuration command, entering the **dot1x supplicant controlled transient** command does not prevent the BPDU violation.

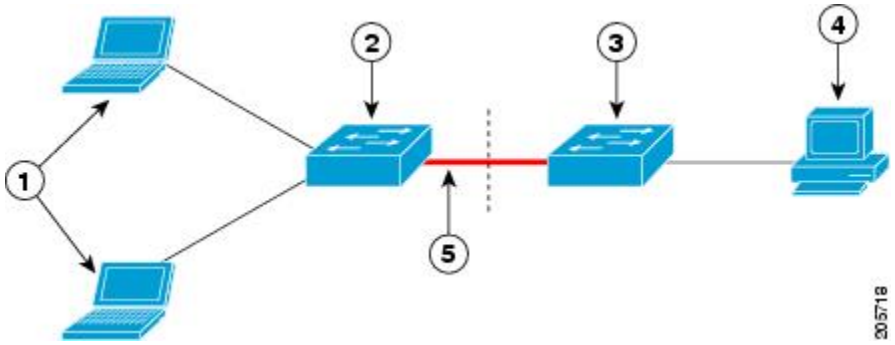
You can enable MDA or multi-auth mode on the authenticator device interface that connects to one more supplicant devices. Multihost mode is not supported on the authenticator device interface.

When you reboot an authenticator device 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 device for Network Edge Access Topology (NEAT) to work in all host modes.

- Host Authorization: Ensures that only traffic from authorized hosts (connecting to the device with supplicant) is allowed on the network. The devices use Client Information Signalling Protocol (CISP) to send the MAC addresses connecting to the supplicant device to the authenticator device.
- Auto enablement: Automatically enables trunk configuration on the authenticator device, allowing user traffic from multiple VLANs coming from supplicant devices. 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 13: Authenticator and Supplicant Device using CISP



1	Workstations (clients)	2	Supplicant device (outside wiring closet)
3	Authenticator device	4	Cisco ISE
5	Trunk port		

Note The **switchport nonegotiate** command is not supported on supplicant and authenticator devices 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.

Per-User ACLs and Filter-IDs

Note You can only set **any** as the source in the ACL.

Note For any ACL that is configured for multiple-host mode, the source portion of statement must be *any*. (For example, **permit icmp** *any* **host 10.10.1.1**.)

Note Using role-based ACLs as filter-ID is not recommended.

You must specify **any** in the source ports of any defined ACL. Otherwise, the ACL cannot be applied and authorization fails. Single host is the only exception to support backward compatibility.

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.

Per-User ACLs Authentication through 802.1x/MAB/WebAuth Users

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 device. The device applies the attributes to the 802.1x port for the duration of the user session. The device removes the per-user ACL configuration when the session is over, if authentication fails, or if a link-down condition occurs. The device does not save RADIUS-specified ACLs in the running configuration. When the port is unauthorized, the device removes the ACL from the port.

RADIUS supports per-user attributes, including vendor-specific attributes. These vendor-specific attributes (VSAs) are in octet-string format and are passed to the device 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 device 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 device. 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 device, 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.

Voice-Aware 802.1x Security

Use the Voice-Aware 802.1x security feature to configure the device to disable only the VLAN on which a security violation occurs, whether it is a data or voice VLAN. Prior to this feature, when an attempt to authenticate the data client caused a security violation, the entire port shut down, resulting in a complete loss of connectivity.

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 device without interruption.

How to Configure IEEE 802.1x Port-Based Authentication

Configuring 802.1x Port-Based Authentication

Beginning in privileged EXEC mode, follow these steps to configure 802.1x port-based authentication:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication dot1x** {**default**} *method1*
5. **dot1x system-auth-control**
6. **aaa authorization network** {**default**} **group radius**
7. **radius server** *server-name*
8. **address ipv4** *ip address* **auth-port port** *number* **acct-port port** *number*
9. **key** *string*
10. **exit**
11. **interface** *type number*
12. **switchport mode access**
13. **authentication port-control auto**
14. **dot1x pae authenticator**
15. **end**

DETAILED STEPS

	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, use

	Command or Action	Purpose
	`Device(config)# aas authentication dot1x default group radius`	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 device.
Step 6	**aaa authorization network** {**default**} **group radius** **Example:** `Device(config)# aaa authorization network default group radius`	(Optional) Configures the device 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 server1`	(Optional) Specifies the name for the RADIUS server configuration, and enters RADIUS server configuration mode.
Step 8	**address ipv4** *ip address* **auth-port port** *number* **acct-port port** *number* **Example:** `Device(config-radius-server)# address ipv4 10.1.10.1 auth-port 1645 acct-port 1682`	(Optional) Specifies the RADIUS server.
Step 9	**key** *string* **Example:** `Device(config-radius-server)# key rad123`	(Optional) Specifies the authentication and encryption key used between the device and the RADIUS daemon running on the RADIUS server.
Step 10	**exit** **Example:** `Device(config-radius-server)# exit`	Exits RADIUS server configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 11	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port connected to the client that is to be enabled for IEEE 802.1x authentication, and enters 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.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **switchport mode access**
5. **no dot1x pae authenticator**
6. **end**

DETAILED STEPS

	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** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**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 5	**no dot1x pae authenticator** **Example:** `Device(config-if)# no dot1x pae authenticator`	Disables 802.1x authentication on the port.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **dot1x default**
5. **end**

DETAILED STEPS

	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** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**dot1x default** **Example:** `Device(config-if)# dot1x default`	Resets the 802.1x parameters to the default values.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **authentication periodic**
5. **authentication timer** {{[**inactivity** | **reauthenticate** | **restart** | **unauthorized**]} {*value*}}
6. **end**

DETAILED STEPS

	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** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**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 device use a RADIUS-provided session timeout, enter the **authentication timer reauthenticate** command.
Step 5	**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 device only if periodic re-authentication is enabled.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Setting the Re-Authentication Number

You can also change the number of times that the device restarts the authentication process before the port changes to the unauthorized state.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **switchport mode access**
5. **dot1x max-req** *count*
6. **end**

DETAILED STEPS

	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** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Sets the port to access mode only if you previously configured the RADIUS server.

	Command or Action	Purpose
Step 5	**dot1x max-req** *count* **Example:** `Device(config-if)# dot1x max-req 4`	Sets the number of times that the device restarts the authentication process before the port changes to the unauthorized state. The range is 0 to 10; the default is 2.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Setting the Device-to-Client Frame-Retransmission Number

In addition to changing the device-to-client retransmission time, you can change the number of times that the device 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 device-to-client frame-retransmission number. This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **dot1x max-reauth-req** *count*
5. **end**

DETAILED STEPS

	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	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**dot1x max-reauth-req** *count* **Example:** `Device(config-if)# dot1x max-reauth-req 5`	Sets the number of times that the device 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 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Changing the Switch-to-Client Retransmission Time

The client responds to the EAP-request/identity frame from the device with an EAP-response/identity frame. If the device 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.

Follow these steps to change the amount of time that the device waits for client notification. This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **authentication timer reauthenticate** *seconds*
5. **end**
6. **show authentication sessions interface** *type number*

DETAILED STEPS

	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	**interface** *interface-type interface-number* **Example:** **Example:** `Device(config)# interface gigabitethernet 0/1`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**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 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show authentication sessions interface** *type number* **Example:** **Example:** `Device# show authentication sessions gigabitethernet 0/1`	Displays information about current Auth-Manager sessions for the specified interface.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **authentication host-mode** [**multi-auth** | **multi-domain** | **multi-host** | **single-host**]

5. **end**

DETAILED STEPS

	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** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**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**: Allows 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. • **multi-host**: Allows multiple hosts on an 802.1x-authorized port after a single host has been authenticated. • **multi-domain**:Allows 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 5	**end** **Example:**	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	

Enabling MAC Move

MAC move allows an authenticated host to move from one port on the device to another.

Beginning in privileged EXEC mode, follow these steps to globally enable MAC move on the device. This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **authentication mac-move permit**
4. **end**

DETAILED STEPS

	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	**authentication mac-move permit** **Example:** `Device(config)# authentication mac-move permit`	Enables MAC move on the device. 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 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **authentication violation** {**protect** | **replace** | **restrict** | **shutdown**}
5. **end**

DETAILED STEPS

	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** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**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 5	**end** **Example:**	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	

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.

Because RADIUS uses the unreliable UDP transport protocol, accounting messages might be lost due to poor network conditions. If the device 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **aaa accounting dot1x default start-stop group radius**
5. **aaa accounting system default start-stop group radius**
6. **end**

DETAILED STEPS

	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	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/3`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**aaa accounting dot1x default start-stop group radius** **Example:** `Device(config-if)# aaa accounting dot1x default start-stop group radius`	Enables 802.1x accounting using the list of all RADIUS servers.
Step 5	**aaa accounting system default start-stop group radius** **Example:** `Device(config-if)# 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 device reloads.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the Device-to-RADIUS-Server Communication

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.

You can globally configure the timeout, retransmission, and encryption key values for all RADIUS servers by using the **radius server** global configuration command. If you want to configure these options on a per-server basis, use the **radius-server timeout**, **radius-server key**, and **radius-server retransmit** global configuration commands.

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. For more information, see the RADIUS server documentation.

Follow these steps to configure the RADIUS server parameters on the device. This procedure is required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius server** *server-name*
4. **address ipv4** *ip address* **auth-port port** *number* **acct-port port** *number*
5. **key** *string*
6. **end**

DETAILED STEPS

	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 server1`	(Optional) Specifies the name for the RADIUS server configuration, and enters RADIUS server configuration mode.
Step 4	**address ipv4** *ip address* **auth-port port** *number* **acct-port port** *number* **Example:** `Device(config-radius-server)# address ipv4 10.1.10.1 auth-port 1645 acct-port 1682`	(Optional) Specifies the RADIUS server.
Step 5	**key** *string* **Example:** `Device(config-radius-server)# key rad123`	(Optional) Specifies the authentication and encryption key used between the device and the RADIUS daemon running on the RADIUS server.
Step 6	**end** **Example:** `Device(config-radius-server)# end`	Exits global configuration mode and 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 device 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.

SUMMARY STEPS

1. A user connects to a port on the device.
2. Authentication is performed.
3. VLAN assignment is enabled, as appropriate, based on the RADIUS server configuration.
4. The device sends a start message to an accounting server.
5. Re-authentication is performed, as necessary.
6. The device sends an interim accounting update to the accounting server that is based on the result of re-authentication.
7. The user disconnects from the port.
8. The device sends a stop message to the accounting server.

DETAILED STEPS

	Command or Action	Purpose
Step 1	A user connects to a port on the device.	
Step 2	Authentication is performed.	
Step 3	VLAN assignment is enabled, as appropriate, based on the RADIUS server configuration.	
Step 4	The device sends a start message to an accounting server.	
Step 5	Re-authentication is performed, as necessary.	
Step 6	The device 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 device sends a stop message to the accounting server.	

Configuring the Number of Authentication Retries

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.

Perform this optional task to configure the maximum number of allowed authentication attempts.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **access-session port-control auto**
5. **authentication event fail action authorize vlan** *vlan-id*
6. **authentication event failretry** *retry-count*
7. **end**

DETAILED STEPS

	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** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**access-session port-control auto** **Example:** `Device(config-if)# access-session 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 40`	Specifies an active VLAN as an 802.1X auth-fail VLAN. The range is 1 to 4094.
Step 6	**authentication event failretry** *retry-count* **Example:** `Device(config-if)# authentication event fail retry 4`	Specifies a number of authentication attempts before a port moves to the auth-fail VLAN. The range is 0 to 5, and the default is 2 attempts after the initial failed event.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Example

The following example shows how to set 2 as the number of authentication attempts allowed before the port moves to the auth-fail VLAN:

```
Device(config-if)# authentication event retry 2
```

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.

Note Before changing the default order and priority of these authentication methods, however, you should understand the potential consequences of those changes. See http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/application_note_c27-573287_ps6638_Products_White_Paper.html for details.

Beginning in privileged EXEC mode, follow these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **switchport mode access**
5. **authentication order** [**dot1x** | **mab**] | {**webauth**}
6. **authentication priority** [**dot1x** | **mab**] | {**webauth**}
7. **end**

DETAILED STEPS

	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** *type number* **Example:**	Specifies the port to be configured, and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface gigabitethernet 0/1`	
Step 4	**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 5	**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 6	**authentication priority** [**dot1x** | **mab**] | {**webauth**} **Example:** `Device(config-if)# authentication priority mab dot1x`	(Optional) Adds an authentication method to the port-priority list.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

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 device 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **authentication event no-response action authorize vlan** *vlan-id*
5. **end**

DETAILED STEPS

	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** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Specifies the port to be configured, and enters interface configuration mode.
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`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring a Restricted VLAN

When you configure a restricted VLAN on a device, 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 device 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **authentication port-control auto**
5. **authentication event fail action authorize vlan** *vlan-id*
6. **end**

DETAILED STEPS

	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** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Specifies the port to be configured, and enters interface configuration mode.
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`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring 802.1X Auth-Fail VLAN

Perform this task to configure an auth-fail VLAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **access-session port-control auto**
5. **authentication event fail action authorize vlan** *vlan-id*

6. **end**
7. **show access-session interface** *interface-id*
8. **copy running-config startup-config**

DETAILED STEPS

	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** *type slot/port* **Example:** `Device(config)# interface gigabitethernet 0/1`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**access-session port-control auto** **Example:** `Device(config-if)# access-session 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 40`	Specifies an active VLAN as an 802.1X auth fail VLAN. The range is 1 to 4094.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show access-session interface** *interface-id* **Example:** `Device# show access-session interface gigabitethernet 0/1`	(Optional) Verify your entries.
Step 8	**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 disable and remove the auth-fail VLAN, use the **no authentication event fail** interface configuration command. The port returns to the default state.

Configuring Open1x

Beginning in privileged EXEC mode, follow these steps to enable manual control of the port authorization state:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **switchport mode access**
5. **authentication control-direction** {**both** | **in**}
6. **authentication fallback** *name*
7. **authentication host-mode** [**multi-auth** | **multi-domain** | **multi-host** | **single-host**]
8. **authentication open**
9. **authentication order** [**dot1x** | **mab**] | {**webauth**}
10. **authentication periodic**
11. **authentication port-control** {**auto** | **force-authorized** | **force-un authorized**}
12. **end**

DETAILED STEPS

	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** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Sets the port to access mode only if you configured the RADIUS server.

	Command or Action	Purpose
Step 5	**authentication control-direction** {**both** | **in**} **Example:** `Device(config-if)# authentication control-direction both`	(Optional) Configures the port control as unidirectional or bidirectional.
Step 6	**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.
Step 7	**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 8	**authentication open** **Example:** `Device(config-if)# authentication open`	(Optional) Enables or disable open access on a port.
Step 9	**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 10	**authentication periodic** **Example:** `Device(config-if)# authentication periodic`	(Optional) Enables or disable reauthentication on a port.
Step 11	**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.

	Command or Action	Purpose
Step 12	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Limiting Login for Users

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login default local**
5. **aaa authentication rejected** *n* **in** *m* **ban** *x*
6. **end**
7. **show aaa local user blocked**
8. **clear aaa local user blocked username** *username*

DETAILED STEPS

	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 the authentication, authorization, and accounting (AAA) access control model.
Step 4	**aaa authentication login default local** **Example:** `Device(config)# aaa authentication login default local`	Sets the authentication, authorization, and accounting (AAA) authentication by using the default authentication methods.
Step 5	**aaa authentication rejected** *n* **in** *m* **ban** *x* **Example:** `Device(config)# aaa authentication rejected 3 in 20 ban 300`	Configures the time period for which an user is blocked, if the user fails to successfully login within the specified time and login attempts. • *n*: Specifies the number of times a user can try to login.

	Command or Action	Purpose
		• *m*: Specifies the number of seconds within which an user can try to login. • *x*: Specifies the time period an user is banned if the user fails to successfully login.
Step 6	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 7	**show aaa local user blocked** **Example:** `Device# show aaa local user blocked`	Displays the list of local users who were blocked.
Step 8	**clear aaa local user blocked username** *username* **Example:** `Device# clear aaa local user blocked username user1`	Clears the information about the blocked local user.

Example

The following is sample output from the **show aaa local user blocked** command:

```
Device# show aaa local user blocked

       Local-user                 State

       user1                      Watched (till 11:34:42 IST Feb 5 2015)
```

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **radius-server dead-criteria**{**time** *seconds* } [**tries** *number*]
5. **radius-server deadtime** *minutes*
6. **radius server** *server-name*
7. **address ipv4** *ip address* **auth-port port** *number* **acct-port port** *number*
8. **key** *string*

9. **dot1x critical** {**eapol** | **recovery delay** *milliseconds*}
10. **interface** *type number*
11. **authentication event server dead action** {**authorize** | **reinitialize**} **vlan** *vlan-id*]
12. **switchport voice vlan** *vlan-id*
13. **authentication event server dead action authorize voice**
14. **end**
15. **show authentication interface** *type number*

DETAILED STEPS

	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	**radius-server dead-criteria**{**time** *seconds* } [**tries** *number*] **Example:** `Device(config)# radius-server dead-criteria time 20 tries 10`	Sets the conditions that determine when a RADIUS server is considered unavailable or down (dead). • **time**:1 to 120 seconds. The device dynamically determines a default *seconds* value between 10 and 60. • **number**: 1 to 100 tries. The device dynamically determines a default **tries** *number* between 10 and 100.
Step 5	**radius-server deadtime** *minutes* **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 server1`	(Optional) Specifies the name for the RADIUS server configuration, and enters RADIUS server configuration mode.

	Command or Action	Purpose
Step 7	**address ipv4** *ip address* **auth-port port** *number* **acct-port port** *number* **Example:** `Device(config-radius-server)# address ipv4 10.1.10.1 auth-port 1645 acct-port 1682`	(Optional) Specifies the RADIUS server.
Step 8	**key** *string* **Example:** `Device(config-radius-server)# key rad123`	(Optional) Specifies the authentication and encryption key used between the device and the RADIUS daemon running on the RADIUS server.

Step 9	**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 device sends an EAPOL-Success message when the device successfully authenticates the critical port.
		• **recovery delay** *milliseconds*: Set the recovery delay period during which the device waits to re-initialize a critical port 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 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Specify the port to be configured, and enters interface configuration mode.

Step 11	**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 12	**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.

	Command or Action	Purpose
Step 13	**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 14	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 15	**show authentication interface** *type number* **Example:** `Device# show authentication interface gigabitethernet 0/1`	(Optional) Verify your entries.

What to do next

To return to the RADIUS server default settings, use the **no radius-server dead-criteria**, **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.

Configuring MAC Authentication Bypass

Beginning in privileged EXEC mode, follow these steps to enable MAC authentication bypass. This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **authentication port-control auto**
5. **mab** [**eap**]
6. **end**

DETAILED STEPS

	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	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**authentication port-control auto** **Example:** `Device(config-if)# authentication port-control auto`	Enables 802.1x authentication on the port.
Step 5	**mab** [**eap**] **Example:** `Device(config-if)# mab`	Enables MAC authentication bypass. (Optional) Use the **eap** keyword to configure the device to use EAP for authorization.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Formatting a MAC Authentication Bypass Username and Password

Use the optional **mab request format** command to format the MAB username and password in a style accepted by the authentication server. The username and password are usually the MAC address of the client. Some authentication server configurations require the password to be different from the username.

Beginning in privileged EXEC mode, follow these steps to format MAC authentication bypass username and passwords.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mab request format attribute 1 groupsize** {**1** | **2** | **4** |**12**} [**separator** {**-** | **:** | **.**} {**lowercase** | **uppercase**}]
4. **mab request format attribute2** {**0** | **7**} *text*
5. **end**

DETAILED STEPS

	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	**mab request format attribute 1 groupsize** {**1** | **2** | **4** |**12**} [**separator** {**-** | **:** | **.**} {**lowercase** | **uppercase**}] **Example:** `Device(config)# mab request format attribute 1 groupsize 12`	Specifies the format of the MAC address in the User-Name attribute of MAB-generated Access-Request packets. • **1**: Sets the username format of the 12 hex digits of the MAC address. • **groupsize**: The number of hex nibbles to concatenate before insertion of a separator. A valid groupsize must be either 1, 2, 4, or 12. • **separator**: The character that separates the hex nibbles according to group size. A valid separator must be either a hyphen, colon, or period. No separator is used for a group size of 12. • {**lowercase** | **uppercase**}: Specifies if non-numeric hex nibbles should be in lowercase or uppercase.
Step 4	**mab request format attribute2** {**0** | **7**} *text* **Example:** `Device(config)# mab request format attribute 2 7 A02f44E18B12`	• **2**: Specifies a custom (nondefault) value for the User-Password attribute in MAB-generated Access-Request packets. • **0**: Specifies a clear-text password to follow. • **7**: Specifies an encrypted password to follow. • *text*: Specifies the password to be used in the User-Password attribute. **Note** When you send configuration information in e-mail, remove type 7 password information. The **show tech-support** command removes this information from its output by default.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **authentication port-control auto**
5. **authentication event fail action authorize vlan** *vlan-id*
6. **authentication event retry** *retry count*
7. **end**

DETAILED STEPS

	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** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port to be configured, and enters interface configuration mode.
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* **Example:** `Device(config-if)# authentication event retry 2`	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.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring VLAN ID-Based MAC Authentication

Beginning in privileged EXEC mode, follow these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mab request format attribute 32 vlan access-vlan**
4. **end**

DETAILED STEPS

	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	**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 4	**end** **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

Configuring a Supplicant Device with NEAT

You can also use an Auto Smartports user-defined macro instead of the device VSA to configure the authenticator device.

Beginning in privileged EXEC mode, follow these steps to configure a device as a supplicant:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cisp enable**
4. **dot1x credential**s *profile*
5. **username** *suppswitch*
6. **password** *password*
7. **dot1x supplicant force-multicast**
8. **interface** *type number*
9. **switchport mode trunk**
10. **dot1x pae supplicant**
11. **dot1x credentials** *profile-name*
12. **end**

DETAILED STEPS

	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	**cisp enable** **Example:** `Device(config)# cisp enable`	Enables CISP.
Step 4	**dot1x credential**s *profile* **Example:**	Creates 802.1x credentials profile. This must be attached to the port that is configured as supplicant.

	Command or Action	Purpose
	`Device(config)# dot1x credentials test`	
Step 5	**username** *suppswitch* **Example:** `Device(config)# username suppswitch`	Creates a username.
Step 6	**password** *password* **Example:** `Device(config)# password myswitch`	Creates a password for the new username.
Step 7	**dot1x supplicant force-multicast** **Example:** `Device(config)# dot1x supplicant force-multicast`	Forces the device to send only multicast EAPOL packets when it receives either unicast or multicast packets. This also allows NEAT to work on the supplicant device in all host modes.
Step 8	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port to be configured, and enters interface configuration 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:** `Device(config-if)# dot1x pae supplicant`	Configures the interface as a port access entity (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`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring an Authenticator Device with NEAT

Configuring this feature requires that one device outside a wiring closet is configured as a supplicant and is connected to an authenticator device.

Note

- The authenticator device 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 Cisco 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 device as an authenticator:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cisp enable**
4. **interface** *type number*
5. **switchport mode access**
6. **authentication port-control auto**
7. **dot1x pae authenticator**
8. **spanning-tree portfast**
9. **end**
10. **copy running-config startup-config**

DETAILED STEPS

	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	**cisp enable** **Example:** `Device(config)# cisp enable`	Enables CISP.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port to be configured, and enters interface configuration mode.
Step 5	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Sets the port mode to **access**.
Step 6	**authentication port-control auto** **Example:** `Device(config-if)# authentication port-control auto`	Sets the port-authentication mode to auto.
Step 7	**dot1x pae authenticator** **Example:** `Device(config-if)# dot1x pae authenticator`	Configures the interface as a port access entity (PAE) authenticator.
Step 8	**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..
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
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.

Changing the Quiet Period

When a device cannot authenticate the client, the device 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **authentication timer restart** *seconds*
5. **end**
6. **show authentication sessions interface** *interface-type interface-number*

DETAILED STEPS

	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-type interface-number* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	**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 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show authentication sessions interface** *interface-type interface-number* **Example:** **Example:** `Device# show authentication sessions interface gigabitethernet 0/2`	Displays information about current Auth-Manager sessions.

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 device:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication dot1x** {**default**} *method1*
5. **interface** *type number*
6. **switchport mode access**
7. **authentication violation** {**shutdown** | **restrict** | **protect** | **replace**}
8. **end**

DETAILED STEPS

	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 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. • For *method1*, enter the **group radius** keywords to use the list of all RADIUS servers for authentication.
Step 5	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port connected to the client that is to be enabled for IEEE 802.1x authentication, and enters interface configuration mode.
Step 6	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Sets the port to access mode.
Step 7	**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**: Generates a syslog error. • **protect**: Drops packets from any new device that sends traffic to the port. • **replace**: Removes the current session and authenticates with the new host.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Voice-Aware 802.1x Security

You use the voice-aware 802.1x security feature on the device 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 device without interruption.

Follow these guidelines to configure voice-aware 802.1x voice security on the device:

- 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 device.

  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:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **errdisable detect cause security-violation shutdown vlan**
4. **exit**
5. **clear errdisable interface***interface-type interface-number* **vlan** *[vlan-list]*
6. **show errdisable detect**

DETAILED STEPS

	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	**errdisable detect cause security-violation shutdown vlan** **Example:** `Device(config)# errdisable detect cause security-violation shutdown vlan`	Shuts 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 4	**exit** **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# exit`	
Step 5	**clear errdisable interface***interface-type interface-number* **vlan** *[vlan-list]* **Example:** `Device(config)# clear errdisable interface gigabitethernet 0/2 vlan`	(Optional) Reenables individual VLANs that have been error disabled. • For the *interface-type interface-number* arguments, specify the port on which to reenable the individual VLANs. • (Optional) For the*[vlan-list]* argument, specify a list of VLANs to be re-enabled. If the VLAN list is not specified, all VLANs are re-enabled.
Step 6	**show errdisable detect** **Example:** `Device# show errdisable detect`	Displays the error-disable detection status.

Configuration Examples for IEEE 802.1x Port-Based Authentication

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 10.1.10.1 auth-port 1645 acct-port 1682
Device(config-radius-server)# key rad123
Device(config-radius-server)# exit
Device(config)# dot1x critical eapol
Device(config)# dot1x critical recovery delay 2000
Device(config)# interface gigabitethernet 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
```

Example: Per-User ACLs Authentication through 802.1x/MAB/WebAuth Users

The following example shows how to configure a device for a downloadable policy:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization network default local group radius
Device(config)# ip device tracking
Device(config)# ip access-list extended default_acl
Device(config-ext-nacl)# permit ip any any
Device(config-ext-nacl)# exit
Device(config)# radius-server vsa send authentication
Device(config)# interface fastEthernet 2/13
Device(config-if)# ip access-group default_acl in
Device(config-if)# end
```

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

Standards and RFCs

Standard/RFC	Title
RFC 3580	*IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*

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

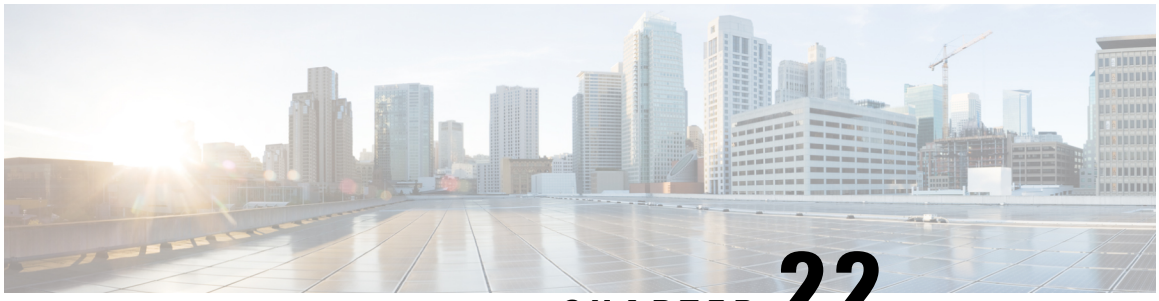
Feature History for IEEE 802.1x Port-Based Authentication

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 Release 15.2(5)E	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

Configuring IPv6 First Hop Security

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 https://cfnng.cisco.com/. An account on Cisco.com is not required.

Prerequisites for First Hop Security in IPv6

- You have configured the necessary IPv6 enabled SDM template.
- QoS should be enabled on the switch before configuring CoPP policies using **mls qos** command.

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 for IPv6 (DHCPv6) 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.
- The following restrictions apply for CoPP policies with IPv6 SISF-based device tracking policies due to limitation reported in CSCvk32439:
  - CoPP policies are required to limit IPv6 NDP traffic when IPv6 SISF policies are configured on the switch.
  - After NDP CoPP policies are configured, limited traffic hits CPU. To accommodate the total end points connected, the number of NDP CoPP policies should be slightly more than the number of users connected to each switch in a stack. If you configure NDP CoPP policies less than the number of end points connected to the switch, the IP allocation to the end point is delayed but is not ignored completely.

  Note For example, if a stack of 5 switches has approximately 300 users, the NDP CoPP policies should be more than 300.
  - The DHCPv6 (server-to-client and client-to-server) CoPP policies are required only if Lightweight DHCPv6 Relay Agent (LDRA) is configured under IPv6 SISF-based device tracking policies on the switch.

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.
- 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

Beginning in privileged EXEC mode, follow these steps to configure IPv6 Snooping Policy :

SUMMARY STEPS

1. **configure terminal**
2. **ipv6 snooping policy** *policy-name*
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**] }
4. **end**
5. **show ipv6 snooping policy** *policy-name*

DETAILED STEPS

	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 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

	Command or Action	Purpose
		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:

SUMMARY STEPS

1. **configure terminal**
2. **interface** Interface_type *stack/module/port*
3. **switchport**
4. **ipv6 snooping** [**attach-policy** *policy_name* [**vlan** {*vlan_id* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids*}] | **vlan** {*vlan_id* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
5. **do show running-config**

DETAILED STEPS

	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:**	Enters the Switchport mode.

	Command or Action	Purpose
	`Device(config-if)# switchport`	**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* | **except***vlan_ids* | **none** | **remove** *vlan_ids*}] | **vlan** {*vlan_id* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`Device(config-if)# ipv6 snooping`
	or
	`Device(config-if)# ipv6 snooping 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`
		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.**

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* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	Example:
	`Device(config-if-range)# ipv6 snooping attach-policy example_policy`
	`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`
		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.

Step 4	**do show running-config interface***portchannel_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 Configure the IPv6 Binding Table Content

Beginning in privileged EXEC mode, follow these steps to configure IPv6 Binding Table Content :

SUMMARY STEPS

1. **configure terminal**
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**] }]

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-limit***number*]]]]
4. **ipv6 neighbor binding logging**
5. **exit**
6. **show ipv6 neighbor binding**

DETAILED STEPS

	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.
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-limit***number*]]]] **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:

SUMMARY STEPS

1. **configure terminal**
2. [**no**]**ipv6 nd inspection policy** *policy-name*
3. **device-role** {**host** | **monitor** | **router** | **switch**}
4. **drop-unsecure**
5. **limit address-count** *value*
6. **sec-level minimum** *value*
7. **tracking** {**enable** [**reachable-lifetime** {*value* | **infinite**}] | **disable** [**stale-lifetime** {*value* | **infinite**}]}
8. **trusted-port**
9. **validate source-mac**
10. **no** {**device-role** | **drop-unsecure** | **limit address-count** | **sec-level minimum** | **tracking** | **trusted-port** | **validate source-mac**}
11. **default** {**device-role** | **drop-unsecure** | **limit address-count** | **sec-level minimum** | **tracking** | **trusted-port** | **validate source-mac**}
12. **do show ipv6 nd inspection policy** *policy_name*

DETAILED STEPS

	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**	**monitor**	**router**	**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	**drop-unsecure** **Example:** `Device(config-nd-inspection)# drop-unsecure`	Drops messages with no or invalid options or an invalid signature.
Step 5	**limit address-count** *value* **Example:** `Device(config-nd-inspection)# limit address-count 1000`	Enter 1–10,000.

	Command or Action	Purpose
Step 6	**sec-level minimum** *value* **Example:** `Device(config-nd-inspection)# limit address-count 1000`	Specifies the minimum security level parameter value when Cryptographically Generated Address (CGA) options are used.
Step 7	**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 8	**trusted-port** **Example:** `Device(config-nd-inspection)# trusted-port`	Configures a port to become a trusted port.
Step 9	**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 10	**no** {**device-role** | **drop-unsecure** | **limit address-count** | **sec-level minimum** | **tracking** | **trusted-port** | **validate source-mac**} **Example:** `Device(config-nd-inspection)# no validate source-mac`	Remove the current configuration of a parameter with the **no** form of the command.
Step 11	**default** {**device-role** | **drop-unsecure** | **limit address-count** | **sec-level minimum** | **tracking** | **trusted-port** | **validate source-mac**} **Example:** `Device(config-nd-inspection)# default limit address-count`	Restores configuration to the default values.
Step 12	**do show ipv6 nd inspection policy** *policy_name* **Example:** `Device(config-nd-inspection)# do show ipv6 nd inspection policy example_policy`	Verifies the ND Inspection Configuration without exiting ND inspection configuration mode.

How to Attach an IPv6 Neighbor Discovery Inspection Policy to an Interface

Starting with Cisco IOS XE Amsterdam 17.1.1 the IPv6 ND Inspection feature is deprecated and the SISF-based device tracking feature replaces it. For the corresponding replacement task, see *Attaching a Device Tracking Policy to an Interface* under the *Configuring SISF-Based Device Tracking* chapter in this document.

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 ND Inspection policy to an interface or VLANs on an interface :

SUMMARY STEPS

1. **configure terminal**
2. **interface** Interface_type *stack/module/port*
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* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
4. **do show running-config**

DETAILED STEPS

	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 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* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	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
	`Device(config-if)# ipv6 nd inspection 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.

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

Starting with Cisco IOS XE Amsterdam 17.1.1 the IPv6 ND Inspection feature is deprecated and the SISF-based device tracking feature replaces it. For the corresponding replacement task, see *Attaching a Device Tracking Policy to an Interface* under the *Configuring SISF-Based Device Tracking* chapter in this document.

Beginning in privileged EXEC mode, follow these steps to attach an IPv6 Neighbor Discovery Inspection policy on an EtherChannel interface or VLAN:

SUMMARY STEPS

1. **configure terminal**
2. **interface range** *Interface_name*
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* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
4. **do show running-config interface***portchannel_interface_name*

DETAILED STEPS

	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* | **except***vlan_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
	`Device(config-if-range)#ipv6 nd inspection vlan 222, 223,224`
		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
Step 4	**do show running-config interface***portchannel_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 Multicast Suppress Policy on a Device

To attach an IPV6 Neighbor Discovery Multicast Suppress policy on a device, complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nd suppress policy** *policy-name*
4. **mode dad-proxy**
5. **mode full-proxy**
6. **mode mc-proxy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters the global configuration mode.
Step 3	**ipv6 nd suppress policy** *policy-name*	Defines the Neighbor Discovery suppress policy name and enters Neighbor Discovery suppress policy configuration mode.
Step 4	**mode dad-proxy**	Enables Neighbor Discovery suppress in IPv6 DAD proxy mode.
Step 5	**mode full-proxy**	Enables Neighbor Discovery suppress to proxy multicast and unicast Neighbor Solicitation messages.
Step 6	**mode mc-proxy**	Enables Neighbor Discovery suppress to proxy multicast Neighbor Solicitation messages.

How to Attach an IPv6 Neighbor Discovery Multicast Suppress Policy on an Interface

To attach an IPv6 Neighbor Discovery Multicast Suppress policy on an interface, complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Perform one of the following tasks:
   - **interface** *type number*
   - **ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** { **add** | **except** | **none** | **remove** | **all**} *vlan* [*vlan1, vlan2, vlan3...*]]]

     OR

   - **vlan configuration** *vlan-id*
   - **ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** { **add** | **except** | **none** | **remove** | **all**} *vlan* [*vlan1, vlan2, vlan3...*]]]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters the global configuration mode.
Step 3	Perform one of the following tasks: • **interface** *type number* • **ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** { **add** | **except** | **none** | **remove** | **all**} *vlan* [*vlan1, vlan2, vlan3...*]]] OR • **vlan configuration** *vlan-id* • **ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** { **add** | **except** | **none** | **remove** | **all**} *vlan* [*vlan1, vlan2, vlan3...*]]]	Specifies an interface type and number, and places the device in interface configuration mode. Attaches the IPv6 Neighbor Discovery Multicast Policy to an interface or a VLAN.
Step 4	**exit**	Exists the interface configuration mode.

How to Attach an IPv6 Neighbor Discovery Multicast Suppress Policy to a Layer 2 EtherChannel Interface

To attach an IPv6 Neighbor Discovery Multicast Suppress policy on an EtherChannel interface, complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Perform one of the following tasks:
   - **interface port-channel** *port-channel-number*
   - **ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** { **add** | **except** | **none** | **remove** | **all**} *vlan* [*vlan1, vlan2, vlan3...*]]]

     OR

   - **vlan configuration** *vlan-id*
   - **ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** { **add** | **except** | **none** | **remove** | **all**} *vlan* [*vlan1, vlan2, vlan3...*]]]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters the global configuration mode.
Step 3	Perform one of the following tasks: • **interface port-channel** *port-channel-number* • **ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** { **add** | **except** | **none** | **remove** | **all**} *vlan* [*vlan1, vlan2, vlan3...*]]] OR • **vlan configuration** *vlan-id* • **ipv6 nd inspection** [**attach-policy** *policy_name* [**vlan** { **add** | **except** | **none** | **remove** | **all**} *vlan* [*vlan1, vlan2, vlan3...*]]]	Specifies an interface type and port number and places the switch in the port channel configuration mode. Attaches the IPv6 Neighbor Discovery Multicast Policy to an interface or a VLAN.
Step 4	**exit**	Exists the interface 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 :

SUMMARY STEPS

1. **configure terminal**
2. [**no**]**ipv6 nd raguard policy** *policy-name*
3. [**no**]**device-role** {**host** | **monitor** | **router** | **switch**}
4. [**no**]**hop-limit** {**maximum** | **minimum**} *value*
5. [**no**]**managed-config-flag** {**off** | **on**}
6. [**no**]**match** {**ipv6 access-list** *list* | **ra prefix-list** *list*}
7. [**no**]**other-config-flag** {**on** | **off**}
8. [**no**]**router-preference maximum** {**high** | **medium** | **low**}
9. [**no**]**trusted-port**
10. **default** {**device-role** | **hop-limit** {**maximum** | **minimum**} | **managed-config-flag** | **match** {**ipv6 access-list** | **ra prefix-list** } | **other-config-flag** | **router-preference maximum**| **trusted-port**}
11. **do show ipv6 nd raguard policy** *policy_name*

DETAILED STEPS

	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**.
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

	Command or Action	Purpose
		lower than the value you specify. Configure **maximum**to block RA messages with Hop Limit values greater than the value you specify.
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** {**ipv6 access-list** | **ra**	Restores a command to its default value.

	Command or Action	Purpose
	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 :

SUMMARY STEPS

1. **configure terminal**
2. **interface** Interface_type *stack/module/port*
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* \| **except***vlan_ids* \| **none** \| **remove** *vlan_ids* \| **all**}]
4. **do show running-config**

DETAILED STEPS

	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* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] **Example:** `Device(config-if)# ipv6 nd raguard attach-policy example_policy` or	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

	`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`	

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:

SUMMARY STEPS

1. **configure terminal**
2. **interface range** *Interface_name*
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* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
4. **do show running-config interface***portchannel_interface_name*

DETAILED STEPS

	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*	**except***vlan_ids*	**none**	**remove** *vlan_ids*	**all**}]	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

	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`	

Step 4	**do show running-config interface***portchannel_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 Configure an IPv6 DHCP Guard Policy

Beginning in privileged EXEC mode, follow these steps to configure an IPv6 DHCP (DHCPv6) Guard policy:

SUMMARY STEPS

1. **configure terminal**
2. [**no**]**ipv6 dhcp guard policy** *policy-name*
3. [**no**]**device-role** {**client** | **server**}
4. [**no**] **match server access-list** *ipv6-access-list-name*
5. [**no**] **match reply prefix-list** *ipv6-prefix-list-name*
6. [**no**]**preference**{ **max** *limit* | **min** *limit* }
7. [**no**] **trusted-port**
8. **default** {**device-role** | **trusted-port**}
9. **do show ipv6 dhcp guard policy** *policy_name*

DETAILED STEPS

	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:**	Specifies the DHCPv6 Guard policy name and enters DHCPv6 Guard Policy configuration mode.

	Command or Action	Purpose
	`Device(config)# ipv6 dhcp guard policy example_policy`	
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.

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 **server**to 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.

	Command or Action	Purpose
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 :

SUMMARY STEPS

1. **configure terminal**
2. **interface** Interface_type *stack/module/port*
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* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
4. **do show running-config interface** Interface_type *stack/module/port*

DETAILED STEPS

	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* | **except***vlan_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:

SUMMARY STEPS

1. **configure terminal**
2. **interface range** *Interface_name*
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* \| **except***vlan_ids* \| **none** \| **remove** *vlan_ids* \| **all**}]

4. **do show running-config interface***portchannel_interface_name*

DETAILED STEPS

	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* | **except***vlan_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 interface***portchannel_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 Configure IPv6 Source Guard

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. [**no**] **ipv6 source-guard policy** *policy_name*
4. [**deny global-autoconf**] [**permit link-local**] [**default**{. . . }] [**exit**] [**no**{. . . }]
5. **end**
6. **show ipv6 source-guard policy** *policy_name*

DETAILED STEPS

	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	[**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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** Interface_type *stack/module/port*
4. **ipv6 source-guard** [**attach-policy** <*policy_name*>]
5. **show ipv6 source-guard policy** *policy_name*

DETAILED STEPS

	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 Source Guard Policy to a Layer 2 EtherChannel Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *port-channel-number*

4. **ipv6 source-guard** [**attach-policy** <*policy_name*>]
5. **show ipv6 source-guard policy** *policy_name*

DETAILED STEPS

	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.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. [**no**] **ipv6 source-guard policy** *source-guard-policy*
4. [**no**] **validate address**
5. **validate prefix**
6. **exit**

7. **show ipv6 source-guard policy** [*source-guard-policy*]

DETAILED STEPS

	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.
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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** Interface_type *stack/module/port*
4. **ipv6 source-guard attach-policy** *policy_name*

5. **show ipv6 source-guard policy** *policy_name*

DETAILED STEPS

	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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *port-channel-number*
4. **ipv6 source-guard** [**attach-policy** <*policy_name*>]
5. **show ipv6 source-guard policy** *policy_name*

DETAILED STEPS

	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 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
```

Additional References

Related Documents

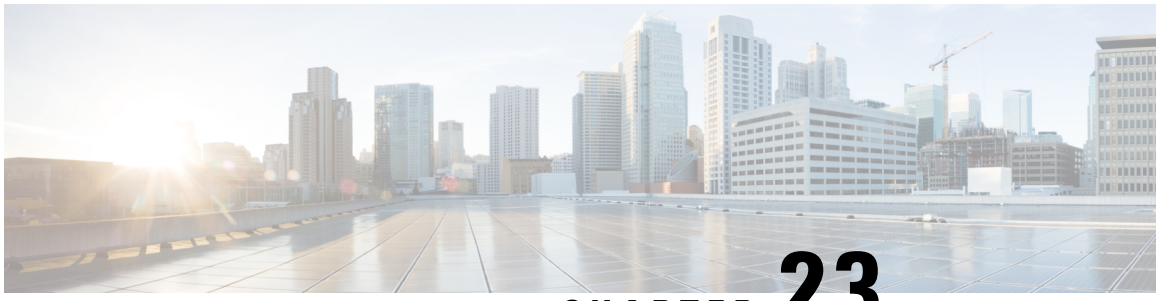
Related Topic	Document Title
Implementing IPv6 Addressing and Basic Connectivity	[illegible]
IPv6 network management and security topics	IPv6 Configuration Library, Cisco IOS XE Release 3SE (Catalyst 3850 Switches) [illegible]
IPv6 Command Reference	IPv6 Command Reference, Cisco IOS XE Release 3SE (Catalyst 3850 Switches) [illegible]

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

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



CHAPTER 23

Per-User ACL Support for 802.1X/MAB/Webauth Users

This feature allows per-user ACLs to be downloaded from the Cisco Access Control Server (ACS) as policy enforcement after authentication using IEEE 802.1X, MAB authentication bypass, or web authentication.

Prerequisites for Per-User ACL Support for 802.1X/MAB/Webauth Users

- AAA authentication must be enabled.
- AAA authorization must be enabled by using the **network** keyword to allow interface configuration from the RADIUS server.
- 802.1X authentication must be enabled.
- The user profile and VSAs must be configured on the RADIUS server.

Restrictions for Per-User ACL Support for 802.1X/MAB/Webauth Users

- Per-user Access Control Lists (ACLs) are supported only in single-host mode.
- This feature does not support standard ACLs on the switch port.

- Only one 802.1X-authenticated user is supported on a port. If the multiple-hosts mode is enabled on the port, the per-user ACL attribute is disabled for the associated port.
- 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.

Information About Per-User ACL Support for 802.1X/MAB/Webauth Users

802.1X Authentication with Per-User ACLs

Per-user access control lists (ACLs) can be configured to provide different levels of network access and service to an 802.1X-authenticated user. When the RADIUS server authenticates a user that is 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.

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. MAB 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. For more information, see the “Configuring Network Security with ACLs|” module.

The extended ACL syntax style should be used to define the per-user configuration that is 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 the Filter-Id attribute is used, it can point to a standard ACL.

The Filter-Id attribute can be used 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. Because of limited support of Cisco IOS access lists on the switch, the Filter-Id attribute is supported only for IP ACLs numbered 1 to 199 and 1300 to 2699 (IP standard and IP extended ACLs).

Only one 802.1X-authenticated user is supported on a port. If the multiple-hosts mode is enabled on the port, the per-user ACL attribute is disabled for the associated port.

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.

How to Configure Per-User ACL Support for 802.1X/MAB/Webauth Users

Configuring Downloadable ACLs

To configure a device to accept downloadable ACLs or redirect URLs from the RADIUS server during authentication of an attached host, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip device tracking**
4. **aaa new-model**
5. **aaa authorization network default group radius**
6. **radius-server vsa send authentication**
7. **interface** *interface-id*
8. **ip access-group** *acl-id* **in**
9. end
10. **show running-config interface***interface-id*
11. **copy running-config startup-config**

DETAILED STEPS

	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`	Enables the IP device tracking table.
Step 4	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA.

	Command or Action	Purpose
Step 5	**aaa authorization network default group radius** **Example:** `Device(config)# aaa authorization network default group radius`	Sets the authorization method. To remove the authorization method, use the **no aaa authorization network default group radius** command.
Step 6	**radius-server vsa send authentication** **Example:** `Device(config)# radius-server vsa send autentication`	Configures the network access server.
Step 7	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the port to be configured, and enters interface configuration mode.
Step 8	**ip access-group** *acl-id* **in** **Example:** `Device(config-if)# ip access-group 99 in`	Configures the default ACL on the port in the input direction. **Note** The ACL ID is an access list name or number.
Step 9	end	`Device(config-if)# end` Returns to Privileged EXEC mode.
Step 10	**show running-config interface***interface-id* **Example:** `Device# show running-config interface interface-id`	Displays the specific interface configuration for verification.
Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Save entries in the configuration file.

Configuration Examples for Per-User ACL Support for 802.1X/MAB/Webauth Users

Example: Configuring a Switch for a Downloadable Policy

The following example shows how to configure a switch for a downloadable policy:

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization network default local group radius
Device(config)# ip device tracking
Device(config)# ip access-list extended default_acl
```

```
Device(config-ext-nacl)# permit ip any any
Device(config-ext-nacl)# exit
Device(config)# radius-server vsa send authentication
Device(config)# interface fastEthernet 2/13
Device(config-if)# ip access-group default_acl in
Device(config-if)# end
```

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

Standards and RFCs

Standard/RFC	Title
IEEE 802.1X protocol	—
RFC 3580	*IEEE 802.1x Remote Authentication Dial In User Service (RADIUS)*

MIBs

MIB	MIBs Link
• CISCO–AUTH–FRAMEWORK–MIB • CISCO–MAB–AUTH–BYPASS–MIB • CISCO–PAE–MIB • IEEE8021–PAE–MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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

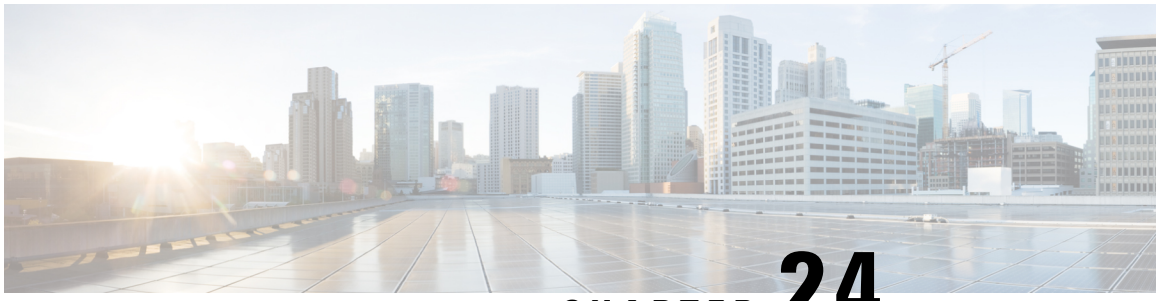
Feature Information for Per-User ACL Support for 802.1X/MAB/Webauth Users

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 28: Feature Information for Per-User ACL Support for 802.1X/MAB/Webauth Users

Feature Name	Releases	Feature Information
Per-User ACL Support for 802.1X/MAB/Webauth Users	Cisco IOS Release 15.2(5)ECisco IOS Release 15.2(7)E1	This feature allows per-user ACLs to be downloaded from the Cisco Access Control Server (ACS) as policy enforcement after authentication using IEEE 802.1X, MAB authentication bypass, or web authentication.



CHAPTER 24

Web Authentication Redirection to Original URL

The Web Authentication Redirection to Original URL feature enables networks to redirect guest users to the URL that they had originally requested. This feature is enabled by default and requires no configuration. This module provides information about this feature.

Web Authentication Redirection to Original URL Overview

The Web Authentication Redirection to Original URL feature enables networks to redirect guest users to the URL that they had originally requested. This feature is enabled by default and requires no configuration.

Guest networks are network connections provided by an enterprise to allow their guests to gain access to the Internet and to their own enterprise networks without compromising the security of the host enterprise. Guest users of an enterprise network can connect to the guest access network through either a wired Ethernet connection or a wireless connection.

Guest access uses a captive portal to gather all web requests made by guests and redirect these requests to one of the guest on-boarding web pages. When guests successfully complete the guest workflow, they are redirected to the page that they had originally requested.

The originally requested URL is passed as metadata along with the Cisco Identity Services Engine (ISE) guest access redirect URL. The Cisco ISE is a security policy management and control platform. It automates and simplifies access control and security compliance for wired, wireless, and VPN connectivity. The requested URL is added at the end of the Cisco ISE guest URL so that the device can send the redirect URL to the guest client. The Cisco ISE parses the URL and redirects the guest to the original URL after completing the on-boarding.

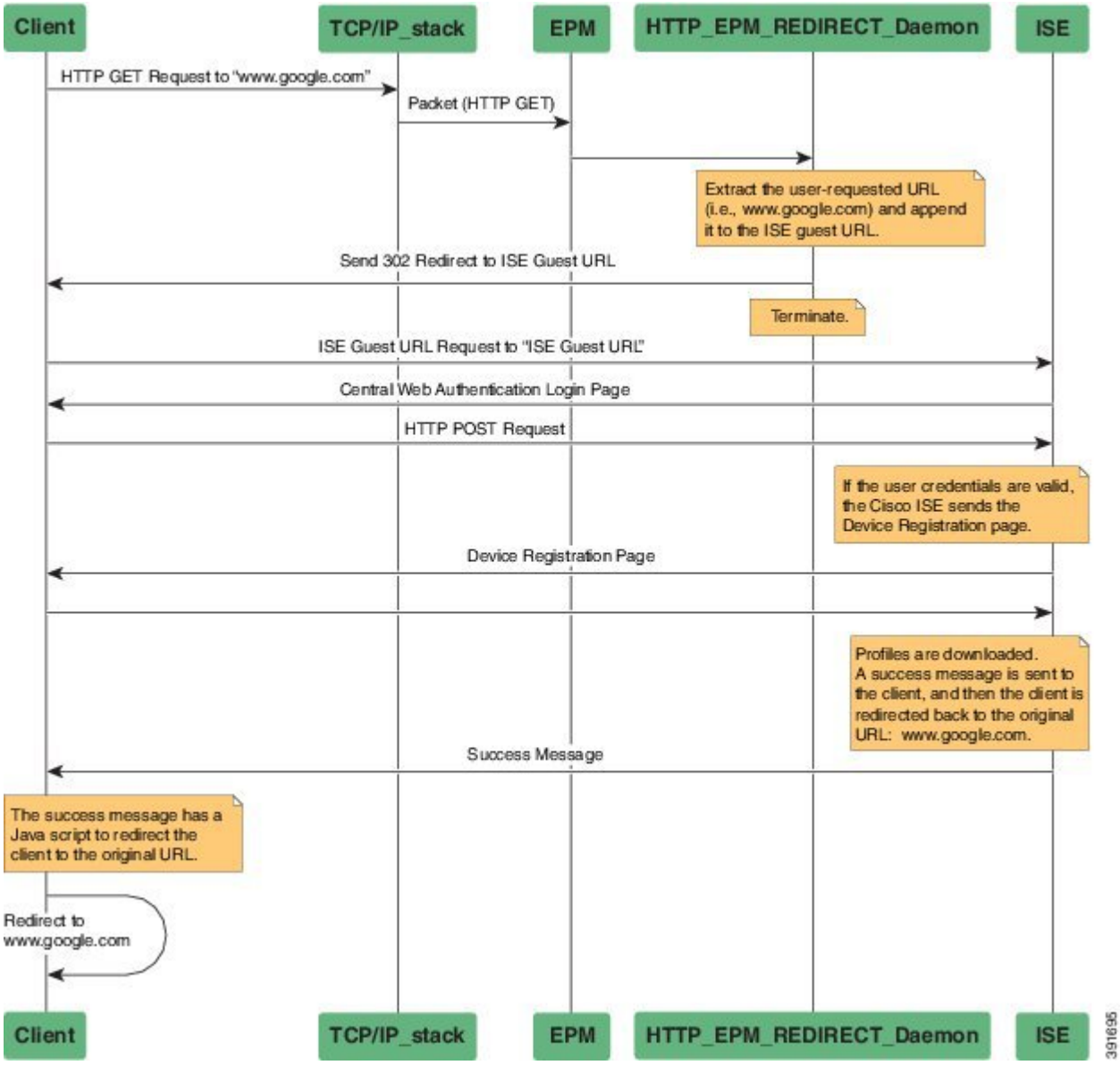
The following is an example of a redirect URL along with the original requested URL:

```
https://10.64.67.92:8443/guestportal/gateway?sessionId=0920269E0000000B0002426B&action=cwa&redirect_
url=http://www.cisco.com/
```

In this example, the URL, https://10.64.67.92:8443/guestportal/gateway?sessionId=0920269E0000000B0002426B&action=cwa is the URL for the guest portal, “&” tells the browser that what follows is a list of name value pairs, and redirect_url=http://www.cisco.com identifies the URL that the user originally requested and to which the user is redirected after completing the guest workflow.

This illustration displays the packet flow that redirects a user to the originally requested URL:

Figure 14: Original URL Redirection Packet Flow



1. A user accesses a network for the first time and sends an HTTP request to access www.google.com. When the user first accesses the network, a MAC authentication bypass (MAB) is triggered and the MAC address is sent to the Cisco ISE.
2. The Cisco ISE returns a RADIUS access-accept message (even if the MAC address is not received) along with the redirect access control list (ACL), the ACL-WEBAUTH-REDIRECT message, and the guest web portal URL to the device.

   The RADIUS message instructs the device to open a port that is restricted based on the configured port and the redirect ACLs, for regular network traffic.

3. When the user launches a web browser, the device intercepts the HTTP traffic and redirects the browser to the Cisco ISE central web authentication (CWA) guest web portal URL; the user-requested URL is extracted and appended to the Cisco ISE guest URL.
4. When the user is authenticated, the Cisco ISE sends the Device Registration page to the user. The user enters the required information, and the page is returned to the Cisco ISE. The Cisco ISE downloads user profiles and redirects the user to the originally requested URL: www.google.com.

Additional References for Web Authentication Redirection to Original URL

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

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

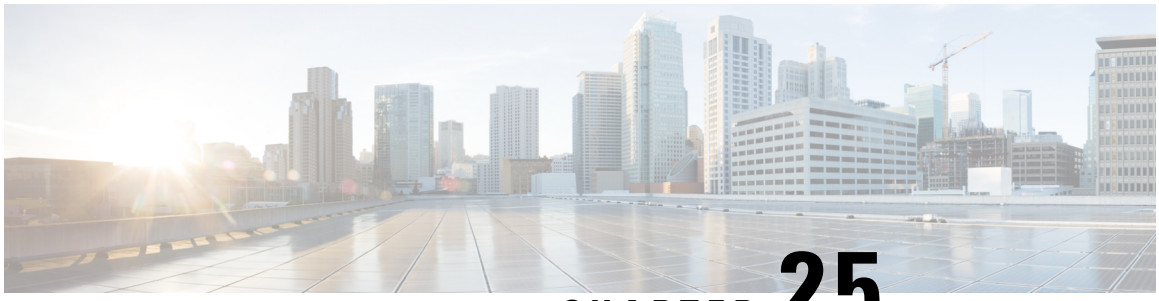
Feature Information for Web Authentication Redirection to Originial URL

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 29: Feature Information for Storm Control

Feature Name	Releases	Feature Information
Web Authentication Redirection to Original URL	Cisco IOS Release 15.2(6)E	This feature was introduced.



CHAPTER 25

Configuring Web-Based Authentication

The Web-Based Authentication feature, also known as web authentication proxy, authenticates end users on host systems that do not run the IEEE 802.1x supplicant.

Restrictions for Web-Based Authentication

- Web-based authentication and URL-redirect are not supported on the same port at the same time.

Information About Web-Based Authentication

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 authentication can be categorized 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

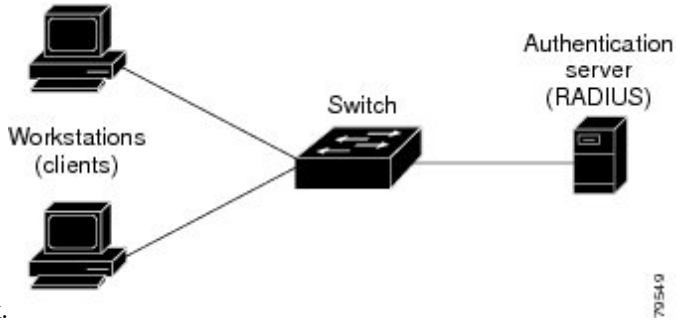
With web-based authentication, the devices in the network have these specific roles:

- *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 15: 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.

Authentication Proxy Interaction with the Client

The authentication proxy feature requires some user interaction on the client host. The table below describes the interaction of the authentication proxy with the client host.

Table 30: Authentication Proxy Interaction with the Client Host

Authentication Proxy Action with Client	Description
Triggering on HTTP connections	If a user is not currently authenticated at the firewall router, any HTTP connection initiated by the user triggers the authentication proxy. If the user is already authenticated, the authentication proxy is transparent to the user.
Logging in using the login page	Triggering the authentication proxy generates an HTML-based login page. The user must enter a username and password to be authenticated with the AAA server. The Authentication Proxy Login Page figure, in the How the Authentication Proxy Works module, illustrates the authentication proxy login page.

Authentication Proxy Action with Client	Description
Authenticating the user at the client	Following the login attempt, the authentication proxy action can vary depending on whether JavaScript is enabled in the browser. If JavaScript is enabled, and authentication is successful, the authentication proxy displays a message indicating the status of the authentication as shown in the Authentication Proxy Login Status Message figure, in the How the Authentication Proxy Works module. After the authentication status is displayed, the proxy automatically completes the HTTP connection. If JavaScript is disabled, and authentication is successful, the authentication proxy generates a popup window with additional instructions for completing the connection. See the Authentication Proxy Login Status Message with JavaScript Disabled figure, in the Secure Authentication module. If authentication is unsuccessful in any case, the user must log in again from the login page.

When to Use the Authentication Proxy

The following are some situations in which you can use the authentication proxy:

- To manage access privileges on an individual (per-user) basis using the services provided by the authentication servers instead of configuring access control based on host IP address or global access policies. Authenticating and authorizing users from any host IP address also allows network administrators to configure host IP addresses using DHCP.
- To authenticate and authorize local users before permitting access to intranet or Internet services.
- To authenticate and authorize remote users before permitting access to local services.
- To control access for specific extranet users. For example, you might want to authenticate and authorize the financial officer of a corporate partner with one set of access privileges while authorizing the technology officer for that same partner to use another set of access privileges.
- To use the authentication proxy in conjunction with the VPN client software to validate users and to assign specific access privileges.
- To use the authentication proxy in conjunction with AAA accounting to generate start and stop accounting records that can be used for billing, security, or resource allocation purposes, thereby allowing users to track traffic from the authenticated hosts.

Applying Authentication Proxy

Apply the authentication proxy in the inbound direction at any interface on the router where you want per-user authentication and authorization. Applying the authentication proxy inbound at an interface causes it to intercept the initial connection request from an user, before that request is subjected to any other processing. If the user fails to gain authentication with the AAA server, the connection request is dropped.

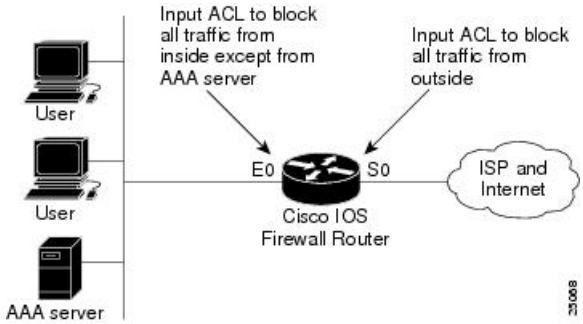
How you apply the authentication proxy depends on your security policy. For example, you can block all traffic through an interface and enable the authentication proxy feature to require authentication and

authorization for all user-initiated HTTP connections. Users are authorized for services only after successful authentication with the AAA server.

The authentication proxy feature also allows you to use standard access lists to specify a host or group of hosts whose initial HTTP traffic triggers the proxy.

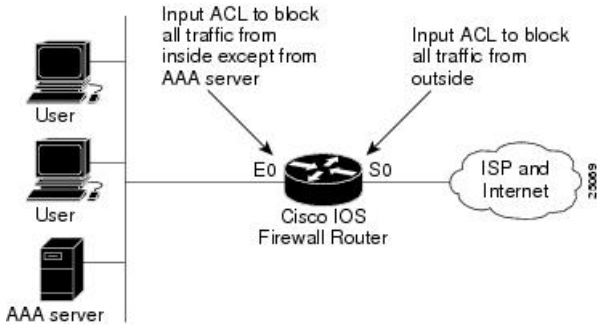
The figure below shows the authentication proxy applied at the LAN interface with all network users required to be authenticated upon the initial connection (all traffic is blocked at each interface).

Figure 16: Applying the Authentication Proxy at the Local Interface



The figure below shows the authentication proxy applied at the dial-in interface with all network traffic blocked at each interface.

Figure 17: Applying the Authentication Proxy at an Outside Interface



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:

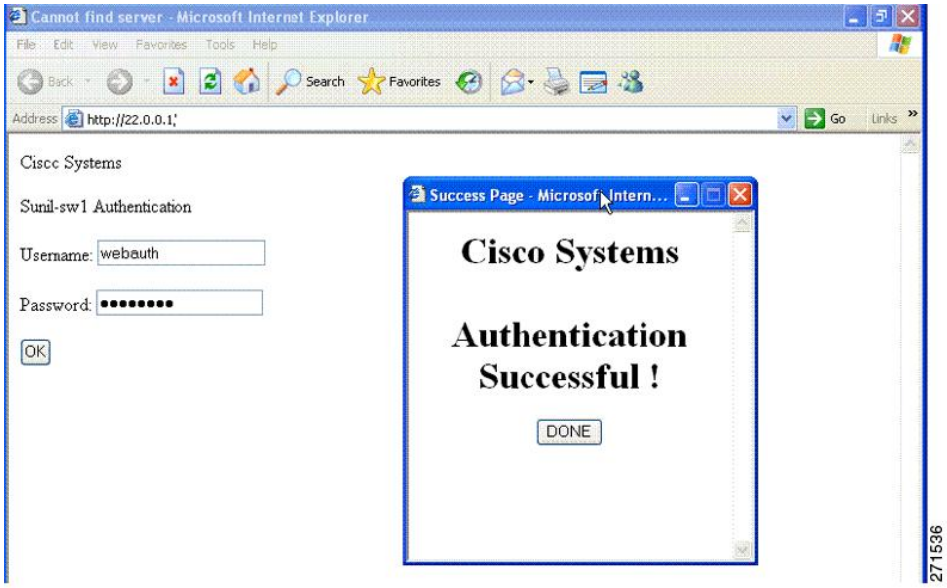
- *Authentication Successful*
- *Authentication Failed*
- *Authentication Expired*

The Local Web Authentication Banner can be configured in legacy and new-style (Session-aware) 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.

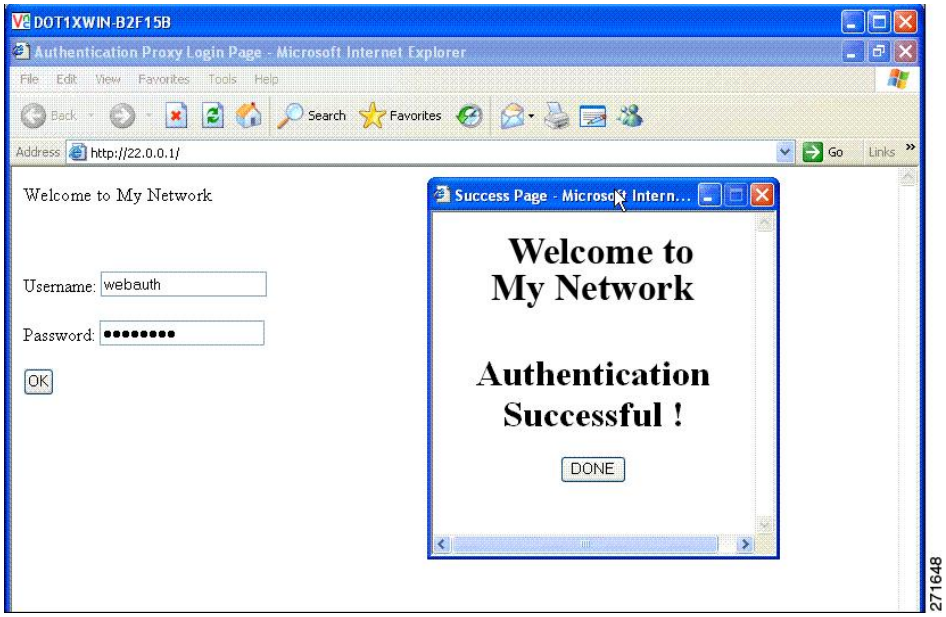
Figure 18: Authentication Successful Banner



The banner can be customized as follows:

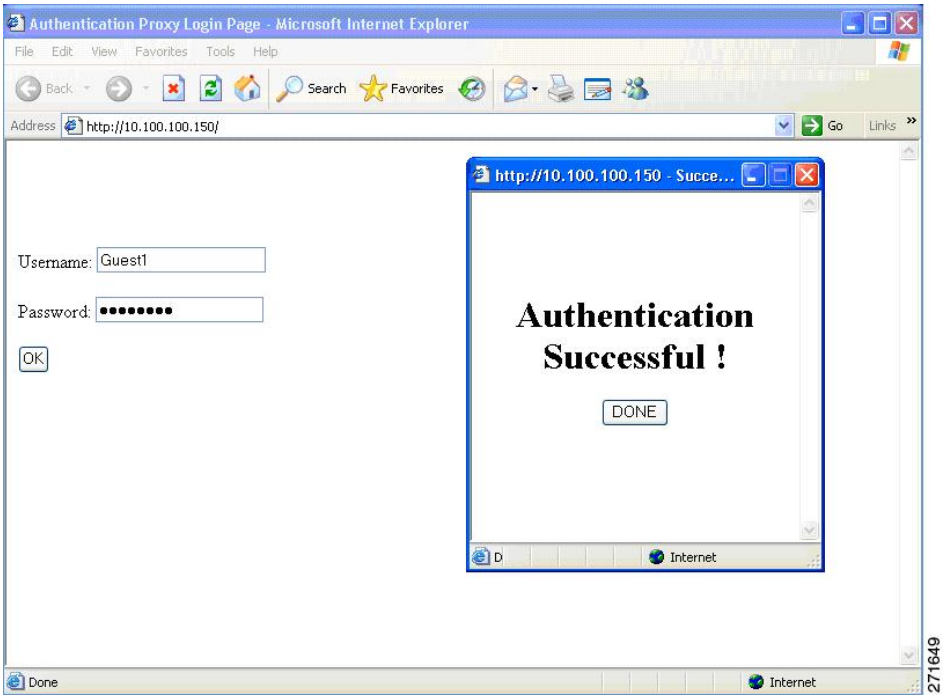
- Add a message, such as switch, router, or company name to the banner:
  - Legacy mode: Use the **ip admission auth-proxy-banner http** *banner-text*global configuration command.
  - New-style mode: Use the **parameter-map type webauth global banner** global configuration command.
- Add a logo or text 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 19: 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 20: 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:

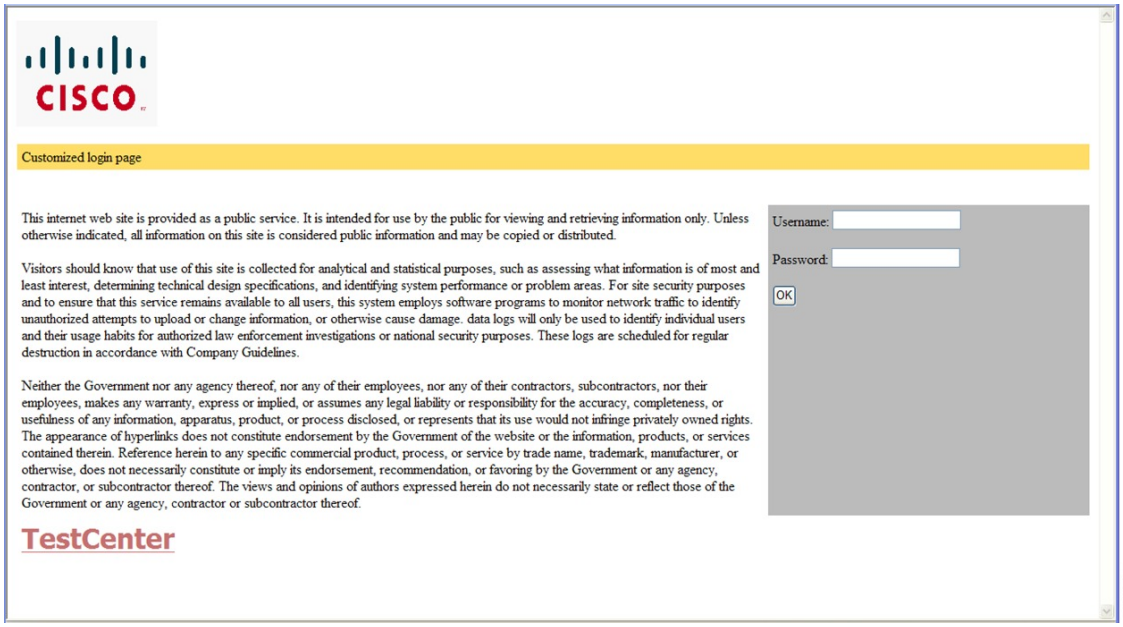
- Login: Your credentials are requested.
- Success: The login was successful.
- Fail: The login failed.
- Expire: The login session has expired because of excessive login failures.

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.
- 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 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 redirecting users to specific URL after authentication occurs is entered and then the command configuring web pages is entered, the CLI 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 stack's active switch or a member).
- 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 21: 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.

Web-Based Authentication Interactions with Other Features

AAA Accounting with Authentication Proxy

Using the authentication proxy, you can generate start and stop accounting records with enough information to be used for billing and security auditing purposes. Thus, you can monitor the actions of authenticated hosts that use the authentication proxy service.

When an authentication proxy cache expires and is deleted, additional data, such as elapsed time, is added to the accounting information and a stop record is sent to the server. At this point, the information is deleted from the data structure.

The accounting records for the authentication proxy user session are related to the cache and the dynamic ACL usage.

ACLs

You must configure port ACLs on interfaces for web-based authentication.

Ensure that sufficient TCAM space is available to enable web-based authentication.

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.

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.

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.

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.

Default Web-Based Authentication Configuration

The following table shows the default web-based authentication configuration.

Table 31: Default Web-based Authentication Configuration

Feature	Default Setting
AAA	Disabled
RADIUS server • IP address • UDP authentication port • Key	• None specified • 1645 • 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 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 key**, and **radius-server retransmit** 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.

How to Configure Web-Based Authentication

Configuring the Authentication Rule and Interfaces

Follow these steps to configure the authentication rule and interfaces:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip admission name** *name* **proxy http**
4. **interface** *type slot/port*

5. **ip access-group** *name*
6. **ip admission name**
7. **exit**
8. **ip device tracking**
9. **end**
10. **show ip admission**

DETAILED STEPS

	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 0/2`	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, GigabitEthernet, or TenGigabitEthernet.
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	**exit** **Example:**	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-if)# exit`	
Step 8	**ip device tracking** **Example:** `Device(config)# ip device tracking`	Enables the IP device tracking table.
Step 9	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 10	**show ip admission** **Example:** `Device# show ip admission`	Displays the network admission cache entries and information about web authentication sessions.

Configuring AAA Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login default group** {**tacacs+** | **radius**}
5. **aaa authorization auth-proxy default group** {**tacacs+** | **radius**}
6. **tacacs-server host** {*hostname* | *ip_address*}
7. **tacacs-server key** {*key-data*}
8. **end**

DETAILED STEPS

	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 functionality.
Step 4	**aaa authentication login default group** {**tacacs+** | **radius**} **Example:** `Device(config)# aaa authentication login default group tacacs+`	Defines the list of authentication methods at login. • **named_authentication_list** refers to any name that is not greater than 31 characters. • **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**} **Example:** `Device(config)# aaa authorization auth-proxy default group tacacs+`	Creates an authorization method list for web-based authorization.
Step 6	**tacacs-server host** {*hostname* | *ip_address*} **Example:** `Device(config)# tacacs-server host 10.1.1.1`	Specifies an AAA server.
Step 7	**tacacs-server key** {*key-data*} **Example:** `Device(config)# tacacs-server key`	Configures the authorization and encryption key used between the device and the TACACS server.
Step 8	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Switch-to-RADIUS-Server Communication

Follow these steps to configure the RADIUS server parameters:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip radius source-interface**
4. **radius-server host** {*hostname* | *ip-address*} **test username** *username*
5. **radius-server key** *string*
6. **radius-server dead-criteria tries** *num-tries*
7. **end**

DETAILED STEPS

	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** **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 host** {*hostname* | *ip-address*} **test username** *username* **Example:** `Device(config)# radius-server host 172.120.39.46 test username user1`	Specifies the host name or IP address of the remote RADIUS server. • The **test username** *username* option enables automated testing of the RADIUS server connection. The specified *username* does not need to be a valid user name. • To use multiple RADIUS servers, reenter this command for each server.
Step 5	**radius-server key** *string* **Example:** `Device(config)# radius-server key rad123`	Configures the authorization and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.

	Command or Action	Purpose
Step 6	**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 7	**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 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:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **ip http secure-server**
5. **end**

DETAILED STEPS

	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 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`	Exits global configuration mode and 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip admission proxy http login page file** *device:login-filename*
4. **ip admission proxy http success page file** *device:success-filename*
5. **ip admission proxy http failure page file** *device:fail-filename*
6. **ip admission proxy http login expired page file** *device:expired-filename*
7. **end**

DETAILED STEPS

	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	**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.
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`	Exits global configuration mode and 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:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip admission max-login-attempts** *number*
4. **end**

DETAILED STEPS

	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	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Web Authentication Local Banner

Beginning in privileged EXEC mode, follow the procedure given below to configure a local banner on a device that has web authentication configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip auth-proxy auth-proxy-banner http** [*banner-text* | *file-path*]
4. **end**

DETAILED STEPS

	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 auth-proxy auth-proxy-banner http** [*banner-text*	*file-path*] **Example:** `Device(config)# aaa ip auth-proxy auth-proxy-banner 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 a file-path indicates a file (for example, a logo or text file) that appears in the banner.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Central Web Authentication

Central Web Authentication (CWA) is a process where a policy server, like Cisco Identity Services Engine (ISE), is used to centrally authenticate users via Web Authentication. Having a central policy server for Web Authentication makes it easier to implement operationally. CWA supports both ACL and VLAN-based enforcement. Additionally, RADIUS CoA is also supported. This allows for posture assessment and enforcement based on profiling.

Note CWA is introduced for the Catalyst 2960-L switch from Cisco IOS Release 15.2(5)E1.

For details on how to configure Central Web Authentication for all Catalyst switches, refer to the Central Web Authentication with a Switch and Identity Services Engine Configuration Example document.

Removing Web-Based Authentication Cache Entries

Follow these steps to remove web-based authentication cache entries:

SUMMARY STEPS

1. **enable**

2. **clear ip admission cache** {* | *host ip address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**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 32: 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.

Displaying Web-Based Authentication Status

Perform this task to display the web-based authentication settings for all interfaces or for specific ports:

SUMMARY STEPS

1. **show authentication sessions** {**interface***type/ slot*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show authentication sessions** {**interface***type/ slot*} **Example:**	Displays the web-based authentication settings.

	Command or Action	Purpose

	This example shows how to view only the global web-based authentication status:
	`Device# show authentication sessions`
	Example:
	This example shows how to view the web-based authentication settings for GigabitEthernet interface 3/27:
	`Device# show authentication sessions interface gigabitethernet 3/27`
		• (Optional) Use the **interface** keyword to display the web-based authentication settings for a specific interface
		• The *type* argument can be FastEthernet, GigabitEthernet, or TenGigabitEthernet

Monitoring HTTP Authentication Proxy

Perform the following task to troubleshoot your HTTP authentication proxy configuration:

SUMMARY STEPS

1. **enable**
2. **debug ip admission all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip admission all** **Example:** `Device# debug ip admission all`	Displays all IP admission debugging information for web-based authentication.

Verifying HTTPS Authentication Proxy

To verify your HTTPS authentication proxy configuration, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show ip auth-proxy cache**
3. **show ip admission { status | cache }**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip auth-proxy cache** **Example:** `Device# show ip auth-proxy cache`	Displays the list of user authentication entries. The authentication proxy cache lists the host IP address, the source port number, the timeout value for the authentication proxy, and the state of the connection. If the authentication proxy state is HTTP_ESTAB, the user authentication was successful.
Step 3	**show ip admission { status | cache }** **Example:** `Device# show ip admission cache`	Display the network admission configuration status and cache entries for web authentication sessions.

Configuration Examples for Web-Based Authentication

Example: Configuring the Authentication Rule and Interfaces

This example shows how to enable web-based authentication on Fast Ethernet port 5/1 :

```
Device> enable
Device# configure terminal
Device(config)# ip admission name webauth1 proxy http
Device(config)# interface fastethernet 5/1
Device(config-if)# ip admission webauth1
Device(config-if)# exit
Device(config)# ip device tracking
Device(config)# end
```

This example shows how to verify the configuration:

```
Device# show ip admission
IP admission status:
  Enabled interfaces            0
  Total sessions                0
  Init sessions                 0     Max init sessions allowed     100
    Limit reached               0     Hi watermark                  0
  TCP half-open connections     0     Hi watermark                  0
  TCP new connections           0     Hi watermark                  0
  TCP half-open + new           0     Hi watermark                  0
  HTTPD1 Contexts               0     Hi watermark                  0

  Parameter Map: Global
    Custom Pages
      Custom pages not configured
```

```
    Banner
      Banner not configured
```

Example: AAA Configuration

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login default group tacacs group radius
! Set up the aaa new model to use the authentication proxy.
Device(config)# aaa authorization auth-proxy default group tacacs group radius
! Define the AAA servers used by the device.
Device(config)# aaa accounting auth-proxy default start-stop group tacacs+
! Set up authentication proxy with accounting.
Device(config)# tacacs-server host 172.31.54.143
Device(config)# tacacs-server key cisco
Device(config)# radius-server host 172.31.54.143
Device(config)# radius-server key cisco
Device(config)# end
```

Example: HTTP Server Configuration

```
Device> enable
Device# configure terminal
! Enable the HTTP server on the device.
Device(config)# ip http server
! Set the HTTP server authentication method to AAA.
Device(config)# ip http authentication aaa
! Define standard access list 61 to deny any host.
Device(config)# access-list 61 deny any
! Use ACL 61 to deny connections from any host to the HTTP server.
Device(config)# ip http access-class 61
Device(config)# end
```

Example: Customizing the Authentication Proxy Web Pages

This example shows how to configure custom authentication proxy web pages:

```
Device> enable
Device# configure terminal
Device(config)# ip admission proxy http login page file flash:login.htm
Device(config)# ip admission proxy http success page file flash:success.htm
Device(config)# ip admission proxy http fail page file flash:fail.htm
Device(config)# ip admission proxy http login expired page flash flash:expired.htm
Device(config)# end
```

The following output displays the host IP addresses, the session timeout, and the posture states. If the posture statue is POSTURE ESTAB, the host validation was successful.

```
Device# show ip admission cache eapoudp
Posture Validation Proxy Cache
Total Sessions: 3 Init Sessions: 1
 Client IP 10.0.0.112, timeout 60, posture state POSTURE ESTAB
 Client IP 10.0.0.142, timeout 60, posture state POSTURE INIT
 Client IP 10.0.0.205, timeout 60, posture state POSTURE ESTAB
```

Example: Specifying a Redirection URL for Successful Login

Configuring redirection URL for successful login

```
Device> enable
Device# configure terminal
Device(config)# ip admission proxy http success redirect www.cisco.com
Device(config)# end
```

Additional References for Web-Based Authentication

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

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

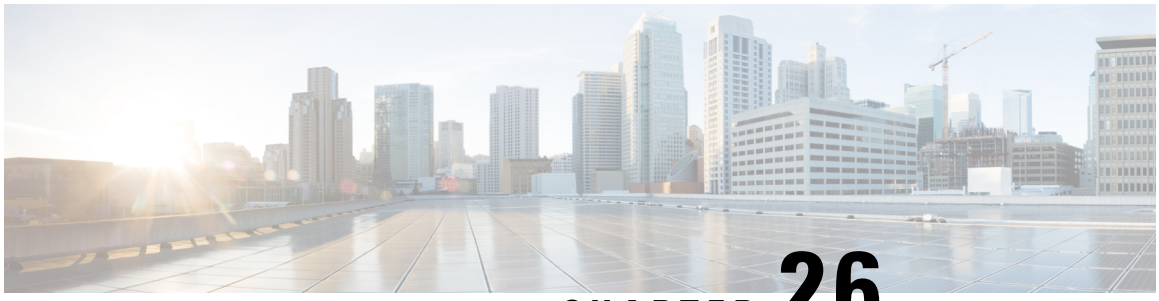
Feature Information for Web-Based Authentication

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 33: Feature Information for Web-Based Authentication

Feature Name	Releases	Feature Information
Web-Based Authentication	Cisco IOS Release 15.2(5)ECisco IOS Release 15.2(7)E1	This feature was introduced.



CHAPTER 26

Port Security

Prerequisites for Port Security

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.

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 device 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 34: Security Violation Mode Actions

Violation Mode	Traffic is forwarded [2]	Sends SNMP trap	Sends syslog message	Displays error message [3]	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 [4]

[2] Packets with unknown source addresses are dropped until you remove a sufficient number of secure MAC addresses.
[3] The switch returns an error message if you manually configure an address that would cause a security violation.
[4] 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.

Default Port Security Configuration

Table 35: 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 device does not support port security aging of sticky secure MAC addresses.

This table summarizes port security compatibility with other port-based features.

Table 36: Port Security Compatibility with Other Features

Type of Port or Feature on Port	Compatible with Port Security
DTP [5] port [6]	No
Trunk port	Yes
Dynamic-access port [7]	No
Routed port	No
SPAN source port	Yes
SPAN destination port	No
EtherChannel	Yes
Tunneling port	Yes
Protected port	Yes
IEEE 802.1x port	Yes
Voice VLAN port [8]	Yes
IP source guard	Yes
Dynamic Address Resolution Protocol (ARP) inspection	Yes

[5] DTP=Dynamic Trunking Protocol
[6] A port configured with the **switchport mode dynamic** interface configuration command.
[7] A VLAN Query Protocol (VQP) port configured with the **switchport access vlan dynamic** interface configuration command.
[8] 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.

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:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **port-security mac-address forbidden** *mac address*
4. **interface** *interface-id*
5. **switchport mode** {**access** | **trunk**}
6. **switchport voice vlan** *vlan-id*
7. **switchport port-security**
8. **switchport port-security [maximum** *value* [**vlan** {*vlan-list* | {**access** | **voice**}}]]
9. **switchport port-security violation** {**protect** | **restrict** | **shutdown** | **shutdown vlan**}
10. **switchport port-security [mac-address** *mac-address* [**vlan** {*vlan-id* | {**access** | **voice**}}]
11. **switchport port-security mac-address sticky**
12. **switchport port-security mac-address sticky** [*mac-address* | **vlan** {*vlan-id* | {**access** | **voice**}}]
13. **switchport port-security mac-address forbidden** *mac address*
14. **end**
15. **show port-security**

DETAILED STEPS

	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	**port-security mac-address forbidden** *mac address* **Example:** `Device(config)# port-security mac-address forbidden 2.2.2`	Specifies a MAC address that should be forbidden by port-security on all the interfaces.
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the interface to be configured, and enter interface configuration mode.

	Command or Action	Purpose
Step 5	**switchport mode** {**access** | **trunk**} **Example:** `Device(config-if)# switchport mode access`	Sets the interface switchport mode as access or trunk; an interface in the default mode (dynamic auto) cannot be configured as a secure port.
Step 6	**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 7	**switchport port-security** **Example:** `Device(config-if)# switchport port-security`	Enable port security on the interface.
Step 8	**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 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. **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.

	Command or Action	Purpose
Step 9	**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, 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 10	**switchport port-security [mac-address** *mac-address* [**vlan** {*vlan-id* | {**access** | **voice**}}] **Example:**	(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.

	Command or Action	Purpose
	`Device(config-if)# switchport port-security mac-address 00:A0:C7:12:C9:25 vlan 3 voice`	**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. **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 11	**switchport port-security mac-address sticky** **Example:** `Device(config-if)# switchport port-security mac-address sticky`	(Optional) Enables sticky learning on the interface.
Step 12	**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.

	Command or Action	Purpose
		• **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.
Step 13	**switchport port-security mac-address forbidden** *mac address* **Example:** `Device(config-if)# switchport port-security mac-address forbidden 2.2.2`	Specifies a MAC address that should be forbidden by port-security on the particular interface.
Step 14	**end** **Example:** `Device(config-f)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 15	**show port-security** **Example:** `Device# show port-security`	Displays information about the port-security settings.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **switchport port-security aging** {**static** | **time** *time* | **type** {**absolute** | **inactivity**}}
5. **end**
6. **show port-security**

DETAILED STEPS

	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 0/2`	Specifies the interface to be configured, and enter interface configuration mode.

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 device does not support port security aging of sticky secure addresses.
		• Enter the **static** keyword to enable aging for statically configured secure addresses on this port.
		• The *time* argument specifies the aging time for this port. The valid values are from 0 to 1440 minutes.
		• For the**type** keyword, 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-f)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show port-security** **Example:** `Device# show port-security`	Displays information about the port-security settings.

Monitoring Port Security

This table displays port security information.

Table 37: Commands for Displaying Port Security Status and Configuration

Command	Purpose
show port-security [**interface** *interface-id*]	Displays port security settings for the switch or for the specified interface, including the maximum allowed number of secure MAC addresses for each interface, the number of secure MAC addresses on the interface, the number of security violations that have occurred, and the violation mode.
show port-security [**interface** *interface-id*] **address**	Displays all secure MAC addresses configured on all switch interfaces or on a specified interface with aging information for each address.
show port-security interface *interface-id* **vlan**	Displays the number of secure MAC addresses configured per VLAN on the specified interface.

Configuration Examples for Port Security

Example: Enabling and Configuring 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> enable
Device# configure terminal
Device(config)# interface gigabitethernet 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
Device(config-if)# end
```

This example shows how to configure a static secure MAC address on VLAN 3 on a port:

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 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
Device(config-if)# end
```

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> enable
Device# configure terminal
Device(config)# interface tengigabitethernet 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
Device(config-if)# end
```

Example: Enabling and Configuring Port Security Aging

The following example shows how to enable and configure port security aging:

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 0/1
Device(config-if)# switchport port-security aging time 120
evice(config-f)# end
```

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

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

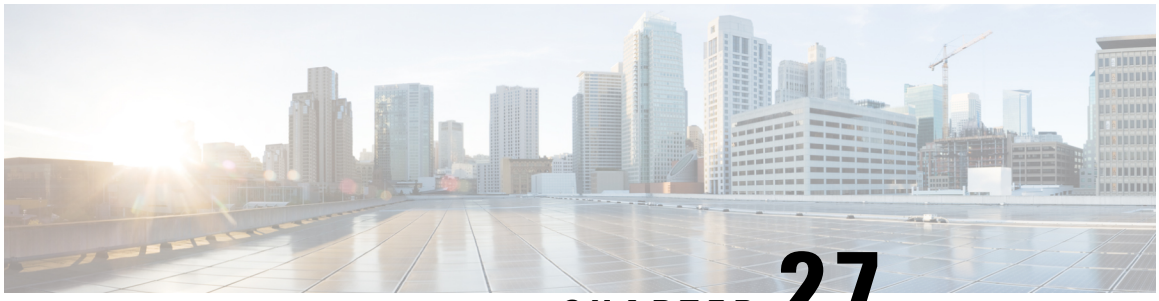
Feature History for Port Security

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 Release 15.2(5)E	Port Security	The Port Security feature restricts input to an interface by limiting and identifying MAC addresses of the stations allowed to access the port.

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 27

Port Blocking

Information About 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.

Blocking Flooded Traffic on an Interface

To block flooded traffic on n interface, perform this procedure:

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **switchport block multicast**
5. **switchport block unicast**
6. **end**
7. **show interfaces** *interface-id* **switchport**
8. **show running-config**
9. **copy running-config startup-config**

DETAILED STEPS

	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 0/2`	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. **Note** Pure Layer 2 multicast traffic as well as multicast packets that contain IPv6 information in the header are blocked.
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-line)# end`	Returns to privileged EXEC mode.
Step 7	**show interfaces** *interface-id* **switchport** **Example:** `Device# show interfaces gigabitethernet 0/2 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 38: 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.

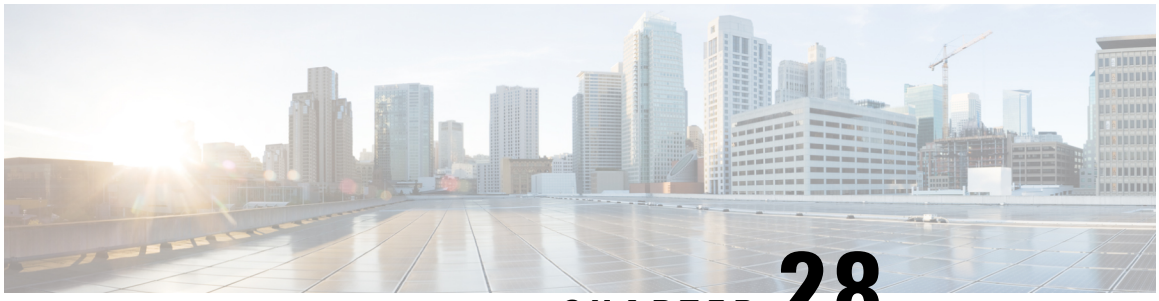
Feature History for Port Blocking

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 Release 15.2(5)E	Port Blocking	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.

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 28

Protected Ports

Information About Protected Ports

The following sections provide 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

The following section provides information on configuring protected ports.

Configuring a Protected Port

To configure a protected port, perform this procedure:

Before you begin

Protected ports are not pre-defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **switchport protected**
5. **end**
6. **show interfaces** *interface-id* **switchport**
7. **show running-config**
8. **copy running-config startup-config**

DETAILED STEPS

	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 0/2`	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-if)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show interfaces** *interface-id* **switchport** **Example:** `Device(config)# show interfaces gigabitethernet 0/2 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 39: 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.

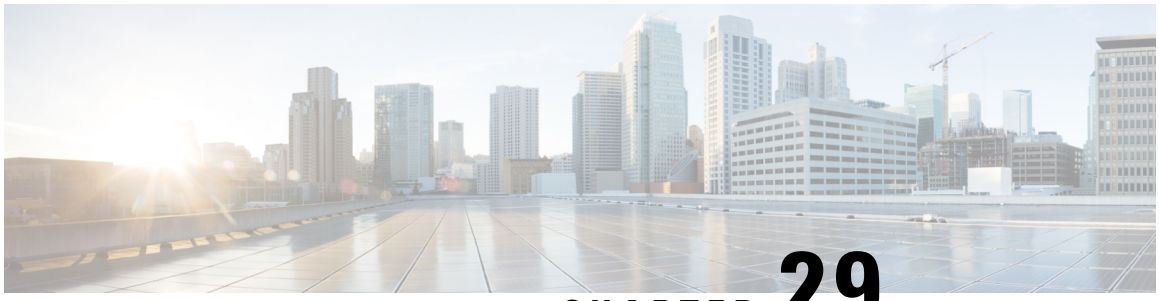
Feature History for Protected Ports

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 Release 15.2(5)E	Protected Ports	Protected ports ensures that there is no exchange of unicast, broadcast, or multicast traffic between ports on the switch.

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 29

Protocol Storm Protection

Restrictions for Configuring Protocol Storm Protection

Virtual port error disabling is not supported for EtherChannel .

Information About 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.

Protocol storm protection is disabled by default. When it is enabled, auto-recovery of the virtual port is disabled by default.

How to Enable Protocol Storm Protection

To enable protocol storm protection, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **psp** {**arp** | **dhcp** | **igmp**} pps *value*
4. **errdisable detect cause psp**
5. **errdisable recovery interval** *time*
6. **end**
7. **show psp config** {**arp** | **dhcp** | **igmp**}

DETAILED STEPS

	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	**psp** {**arp**	**dhcp**	**igmp**} pps *value* **Example:** `Device(config)# psp dhcp pps 35`	Configures protocol storm protection for ARP, IGMP, or DHCP. *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:**	(Optional) Configures an auto-recovery time (in seconds) for error-disabled virtual ports. When a virtual port is

	Command or Action	Purpose
	`Device(config)# errdisable recovery interval 100`	error-disabled, the switch auto-recovers after this time. The range is from 30 to 86400 seconds.
Step 6	**end** **Example:** `Device(config-line)# 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

Table 40: Commands for Verfying Entries

Command	Purpose
show psp config {**arp** | **dhcp** | **igmp**}	Verify your entries.

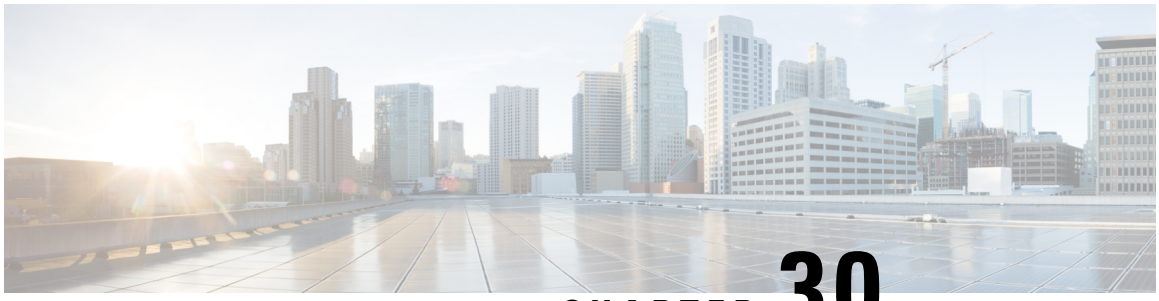
Feature History for Protocol Storm Protection

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 Release 15.2(5)E	Protocol Storm Protection	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.

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 30

Storm Control

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 device 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

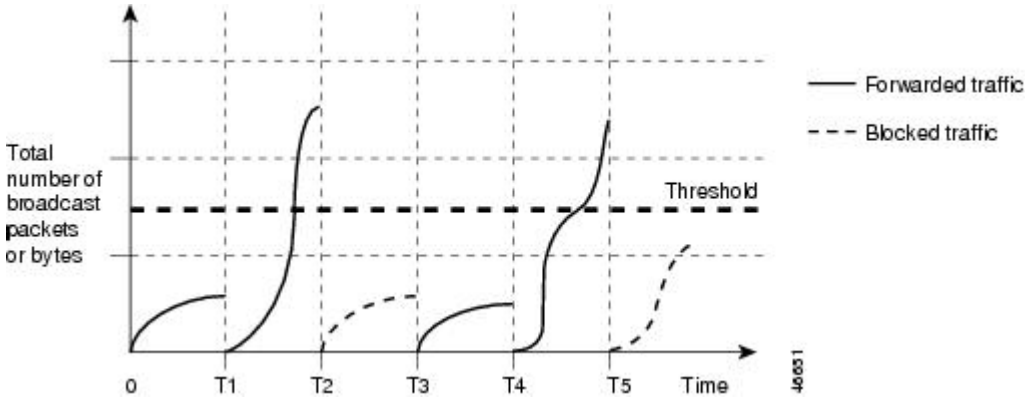
With each method, the port blocks traffic when the rising threshold is reached. The port remains blocked until the traffic rate drops below the falling threshold (if one is specified) and then resumes normal forwarding. If the falling suppression level is not specified, the device blocks all traffic 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 device does not differentiate between routing updates, such as OSPF, and regular multicast data traffic, so both types of traffic are blocked.

Traffic Patterns

Figure 22: 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **storm-control action** {**shutdown** | **trap**}
5. **storm-control** {**broadcast** | **multicast** | **unicast**} **level** {*level* [*level-low*] | **bps** *bps* [*bps-low*] | **pps** *pps* [*pps-low*]}
6. **end**
7. **show storm-control** [*interface-id*] [**broadcast** | **multicast** | **unicast**]

DETAILED STEPS

	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	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 0/2`	Specifies the interface to be configured, and enter interface configuration mode.
Step 4	**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 5	**storm-control** {**broadcast** | **multicast** | **unicast**} **level** {*level* [*level-low*] | **bps** *bps* [*bps-low*] | **pps** *pps* [*pps-low*]} **Example:** `Device(config-if)# storm-control unicast level 87 65`	Configures broadcast, multicast, or unicast storm control. By default, storm control is disabled. The keywords have these meanings: • 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.

	Command or Action	Purpose
		• 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 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 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 gigabitethernet 0/2 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.

Configuration Examples for Storm Control

Example: Configuring Storm Control and Threshold Levels

The following example shows how to configure storm control and threshold levels:

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 0/1
Device(config-if)# storm-control action trap
Device(config-if)# storm-control unicast level 87 65
Device(config-if)# end
Device# show storm-control gigabitethernet 0/1 unicast
```

Additional References for Storm Control

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 2960-L Switches)

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 Storm 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 Release 15.2(5)E	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.

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.