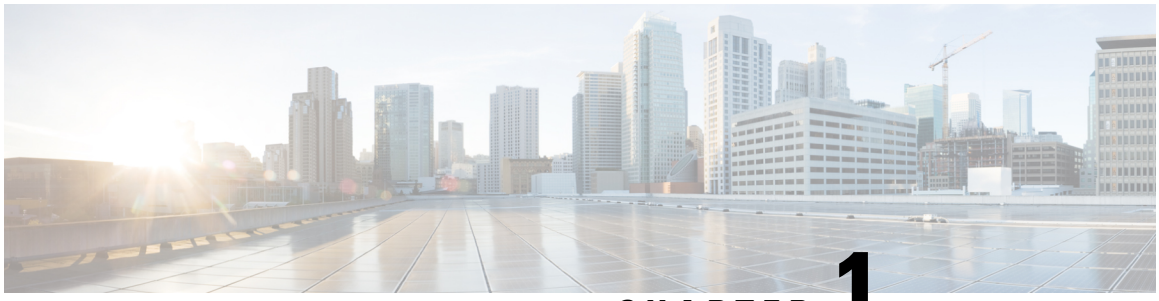
Cisco 8500 Series Secure Routers Software Configuration Guide

First Published: 2025-09-09

Last Modified: 2025-09-09

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



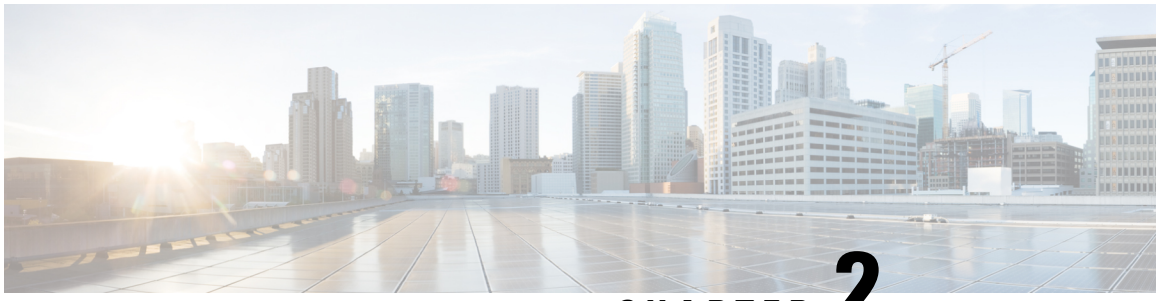
CHAPTER 1

Overview

The Cisco 8500 Series Secure Routers are compact 1RU platforms that are well-suited for Datacenter and Colocation deployments. The primary use-case for these routers is Enterprise WAN Aggregation.

This document covers configuration details for the following models:

- C8550-G2
- C8570-G2



CHAPTER 2

Software packaging and architecture

The Cisco 8500 Series Secure Routers introduces a new software packaging model and architecture.

This chapter discusses this new packaging and architecture and contains these sections.

Software packaging on the Cisco 8500 Series Secure Routers

This section covers the following topics:

Cisco 8500 Series Secure Routers software overview

The Cisco 8500 Series Secure Routers are high-performance WAN Aggregation platforms.

Consolidated packages

A consolidated package is a single image composed of individual software subpackage files. A single consolidated package file is a bootable file, and the Cisco 8500 Series Secure Routers can be run using the consolidated package.

Each consolidated package also contains a provisioning file. A provisioning file is used for booting in cases where the individual subpackages are extracted from the consolidated package, or optional subpackages are used to run the router. For additional information on the advantages and disadvantages of running a complete consolidated package, see the *Running the Cisco 8500 Series Secure Routers: An Overview*.

Important information about consolidated packages

The important information about consolidated packages include:

- A consolidated package file is a bootable file. If the router is configured to run using the complete consolidated package, boot the router using the consolidated package file. If the router is configured to run using individual subpackages, boot the router using the provisioning file. For additional information on the advantages and disadvantages of running a complete consolidated package, see the *Running the Cisco 8500 Series Secure Routers: An Overview* section .
- If you need to install optional subpackages, then you must boot the router using the individual subpackage provisioning file method.

Important notes about individual subpackages

The important information about individual subpackage include:

- Individual subpackages cannot be downloaded from Cisco.com individually. To get these individual subpackages, users must download a consolidated package and then extract the individual subpackages from the consolidated package using the command-line interface.
- If the router is being run using individual subpackages instead of being run using a complete consolidated package, the router must be booted using a provisioning file. A provisioning file is included in all consolidated packages and is extracted from the image along with the individual subpackages whenever individual subpackages are extracted.

Provisioning files

Note You must use the provisioning files to manage the boot process if you need to install optional subpackages.

Provisioning files manage the boot process when the Cisco 8500 Series Secure Routers is configured to run using individual subpackages or optional subpackages (such as the package for the Cisco WebEx Node Cisco 8500 Series Secure Routers Series). When individual subpackages are being used to run the Cisco 8500 Series Secure Routers, the router has to be configured to boot the provisioning file. The provisioning file manages the bootup of each individual subpackage and the Cisco 8500 Series Secure Routers assumes normal operation.

Provisioning files are extracted automatically when individual subpackage files are extracted from a consolidated package.

Provisioning files are not necessary for running the router using the complete consolidated package; if you want to run the router using the complete consolidated package, simply boot the router using the consolidated package file.

Important notes about provisioning files

The important information about provisioning files include:

- Each consolidated package contains two provisioning files. One of the provisioning files is always named “packages.conf”, while the other provisioning file will have a name based on the consolidated package naming structure. In any consolidated package, both provisioning files perform the exact same function.
- In most cases, the “packages.conf” provisioning file should be used to boot the router. Configuring the router to boot using this file is generally easier because the router can be configured to boot using “packages.conf”, so no changes have to be made to the boot statement when Cisco IOS XE is upgraded (the **boot system** *file-system*:**packages.conf** configuration command can remain unmodified before and after an upgrade).
- The provisioning file and individual subpackage files must be kept in the same directory. The provisioning file does not work properly if the individual subpackage files are in other directories.
- The provisioning filename can be renamed; the individual subpackage filenames cannot be renamed.
- After placing the provisioning file and the individual subpackage files in a directory and booting the router, it is highly advisable not to rename, delete, or alter any of these files. Renaming, deleting, or altering the files can lead to unpredictable router problems and behaviors.

File to upgrade field programmable hardware devices

A hardware programmable package file used to upgrade field programmable hardware devices is released as needed . A package file is provided for the field programmable device to customers in cases where a field upgrade is required. If the Cisco 8500 Series Secure Routers contains an incompatible version of the hardware programmable firmware, then that firmware may need to be upgraded.

Generally an upgrade is only necessary in cases where a system message indicates one of the field programmable devices on the Cisco 8500 Series Secure Routers needs an upgrade or a Cisco technical support representative suggests an upgrade.

Processes overview

Cisco IOS XE has numerous components that run entirely as separate processes on the Cisco 8500 Series Secure Routers. This modular architecture increases network resiliency by distributing operating responsibility among separate processes rather than relying on Cisco IOS software for all operations.

IOS as a process

In almost all previous Cisco router platforms, an overwhelming majority of the internal software processes are run using Cisco IOS memory.

The Cisco 8500 Series Secure Routers introduce a distributed software architecture that moves many operating system responsibilities out of the IOS process. In this architecture, IOS, which previously was responsible for almost all of the internal software processes, now runs as one of many Linux processes while allowing other Linux processes to share responsibility for running the router. This architecture allows for better allocation of memory so the router can run more efficiently.

Dual IOS processes

The Cisco 8500 Series Secure Routers support a dual IOS process model that allows for increased high availability at all times.

Using SSO, a second IOS process can be enabled on a Cisco 8500 Series Secure Routers.

The state of these dual IOS processes can be checked by entering the **show platform** command.

The advantages of a second IOS process includes:

- Increased fault tolerance—In the event of an active IOS failure, the second IOS process immediately becomes the active IOS process with little to no service disruption.

File systems on the Cisco 8500 Series Secure Routers

The following table provides a list of file systems that can be seen on the Cisco 8500 Series Secure Routers.

Table 1: File Systems

File System	Description
bootflash:	The boot flash memory file system on the active RP.

File System	Description
cns:	The Cisco Networking Services file directory.
nvram:	Router NVRAM. You can copy the startup configuration to NVRAM or from NVRAM.
obfl:	The file system for Onboard Failure Logging files.
system:	The system memory file system, which includes the running configuration.
tar:	The archive file system.
tmpsys:	The temporary system files file system.
usb:	The Universal Serial Bus (USB) flash drive file systems on the active RP.

If you run into a file system not listed in the above table, enter the**?** help option or see the **copy** command reference for additional information on that file system.

Autogenerated file directories and files

This section discusses the autogenerated files and directories that might appear on your Cisco 8500 Series Secure Routers, and how the files in these directories can be managed.

The following table provides a list and descriptions of autogenerated files on the Cisco 8500 Series Secure Routers.

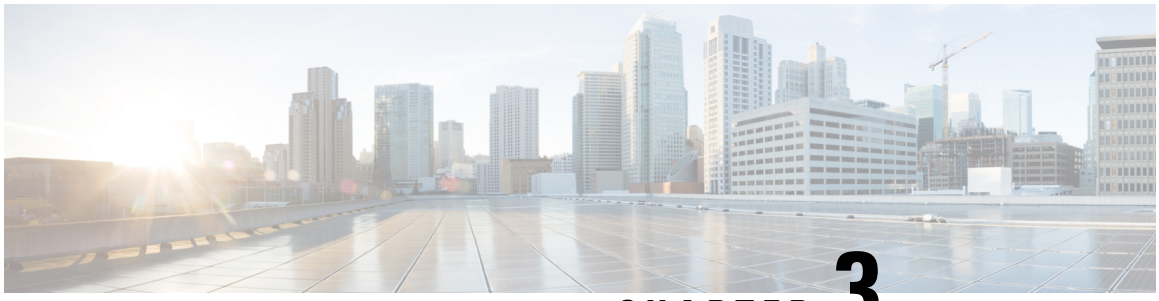
Table 2: Autogenerated Files

File or Directory	Description
crashinfo files	A crash info file may appear in the bootflash: file system. These files provide descriptive information of a crash and may be useful for tuning or troubleshooting purposes, but the files are not part of router operations and can be erased without impacting the functioning of the router.
core directory	The storage area for.core files. If this directory is erased, it will automatically regenerate itself at bootup. The .core files in this directory can be erased without impacting any router functionality, but the directory itself should not be erased.
lost+found directory	This directory is created on bootup if a system check is performed. Its appearance is completely normal and does not indicate any issues with the router.
tracelogs directory	The storage area for trace files. Trace files are useful for troubleshooting. Trace files, however, are not part of router operations and can be erased without impacting the router’s performance.

Important notes about autogenerated directories

The important information about autogenerated directories include:

- Any autogenerated file on the bootflash: directory should not be deleted, renamed, moved, or altered in any way unless directed by customer support. Altering autogenerating files on the bootflash: can have unpredictable consequences for system performance.
- Crashinfo, core, and trace files can be deleted, but the core and tracelog directories that are automatically part of the bootflash: file system should not be deleted.



CHAPTER 3

Deploy IOS-XE and SDWAN

Overview

The universalk9 image on Cisco 8500 Series Secure Router supports both Routing and SD-WAN.

Restrictions

Autonomous or controller mode

Access the Cisco IOS XE and Cisco IOS XE SD-WAN functionality through Autonomous and Controller execution modes, respectively. The Autonomous mode is the default mode for the routers and includes the Cisco IOS XE functionality. To access Cisco IOS XE SD-WAN functionality switch to the Controller mode.

For more information, see https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/sdwan-xe-gs-book/install-upgrade-17-2-later.html#Cisco_Concept.dita_42020dbf-1563-484f-8824-a0b3f468e787

Switch between controller and autonomous modes

The default mode of the device is autonomous mode. Use the **controller-mode** command in Privileged EXEC mode to switch between controller and autonomous modes.

The **controller-mode enable** command switches the device to controller mode

The **controller-mode disable** command switches the device to autonomous mode

For information see Cisco SD-WAN Getting Started Guide

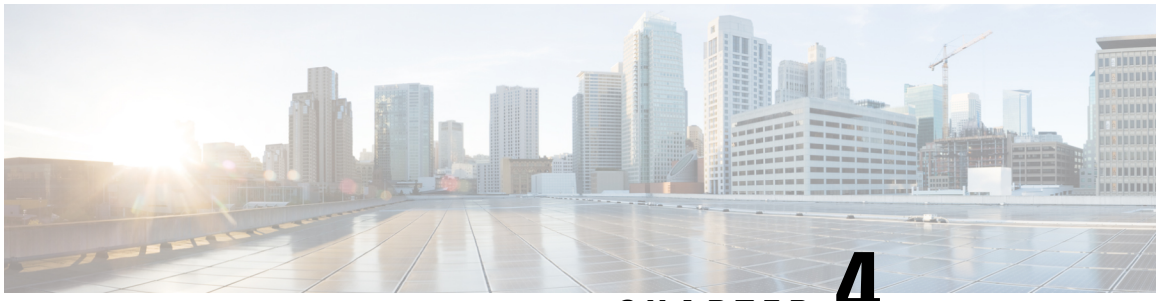
PnP discovery process

You can use the existing Plug and Play Workflow to determine the mode of the device.

The PnP-based discovery process determines the mode in which the device operates, based on the controller discovery and initiates a mode change, if required. This discovery is based on the controller profile attached to the device UID in the smart account/virtual account. The mode change results in a reboot of the device. Once reboot is complete, the device performs appropriate discovery process.

Plug and Play (PnP) deployment include the following discovery process scenarios:

Boot up Mode	Discovery Process	Mode Change
Autonomous	Plug and Play Connect Discovery or on-premise plug and play server discovery	No Mode change
Controller	Plug and Play Connect Discovery or on-premise plug and play server discovery	Mode change to autonomous mode



CHAPTER 4

Using Cisco IOS XE software

This chapter provides information to prepare you to configure the Cisco 8500 Series Secure Routers:

Accessing the CLI using a router console

The following sections describe how to access the command-line interface (CLI) using a directly-connected console or by using Telnet or a modem to obtain a remote console:

Accessing the CLI using a directly-connected console

This section describes how to connect to the console port on the router and use the console interface to access the CLI.

The console port on a Cisco 8500 Series Secure Routers is an EIA/TIA-232 asynchronous, serial connection with no flow control and an RJ-45 connector. The console port is located on the front panel of each Route Processor (RP).

Connecting to the console port

To connect to the console port, complete the following steps:

SUMMARY STEPS

1. Configure your terminal emulation software with the following settings:

2. Connect to the port using the RJ-45-to-RJ-45 cable and RJ-45-to-DB-25 DTE adapter or using the RJ-45-to-DB-9 DTE adapter (labeled “Terminal”).

DETAILED STEPS

Procedure

Step 1 Configure your terminal emulation software with the following settings:

- 9600 bits per second (bps)
- 8 data bits
- No parity
- 1 stop bit
- No flow control

Step 2 Connect to the port using the RJ-45-to-RJ-45 cable and RJ-45-to-DB-25 DTE adapter or using the RJ-45-to-DB-9 DTE adapter (labeled “Terminal”).

Using the console interface

To access the CLI using the console interface, complete the following steps:

SUMMARY STEPS

1. After you attach the terminal hardware to the console port on the router and you configure your terminal emulation software with the proper settings, the following prompt appears:
2. Press **Return** to enter user EXEC mode. The following prompt appears:
3. From user EXEC mode, enter the **enable** command as shown in the following example:
4. At the password prompt, enter your system password. If an enable password has not been set on your system, this step may be skipped.The following example shows entry of the password enablepass:
5. When your enable password is accepted, the privileged EXEC mode prompt appears:
6. You now have access to the CLI in privileged EXEC mode and you can enter the necessary commands to complete your desired tasks.
7. To exit the console session, enter the **quit** command as shown in the following example:

DETAILED STEPS

Procedure

Step 1 After you attach the terminal hardware to the console port on the router and you configure your terminal emulation software with the proper settings, the following prompt appears:

Example:

```
Press RETURN to get started.
```

Step 2 Press **Return** to enter user EXEC mode. The following prompt appears:

Example:

```
Router>
```

Step 3 From user EXEC mode, enter the **enable** command as shown in the following example:

Example:

```
Router>enable
```

Step 4 At the password prompt, enter your system password. If an enable password has not been set on your system, this step may be skipped.The following example shows entry of the password enablepass:

Example:

```
Password:enablepass
```

Step 5 When your enable password is accepted, the privileged EXEC mode prompt appears:

Example:

```
Router#
```

Step 6 You now have access to the CLI in privileged EXEC mode and you can enter the necessary commands to complete your desired tasks.

Step 7 To exit the console session, enter the **quit** command as shown in the following example:

Example:

```
Router#quit
```

Accessing the CLI from a remote console using Telnet

This section describes how to connect to the console interface on a router using Telnet to access the CLI.

Preparing to connect to the router console using Telnet

Before you can access the router remotely using Telnet from a TCP/IP network, you need to configure the router to support virtual terminal lines (vtys) using the **line vty** global configuration command. You also should configure the vtys to require login and specify a password.

Note To prevent disabling login on the line, be careful that you specify a password with the **password** command when you configure the **login** line configuration command. If you are using authentication, authorization, and accounting (AAA), you should configure the **login authentication** line configuration command. To prevent disabling login on the line for AAA authentication when you configure a list with the **login authentication** command, you must also configure that list using the **aaa authentication login** global configuration command. For more information about AAA services, see the *Cisco IOS XE Security Configuration Guide,* and the *Cisco IOS Security Command Reference Guide* .

In addition, before you can make a Telnet connection to the router, you must have a valid host name for the router or have an IP address configured on the router. For more information about requirements for connecting to the router using Telnet, information about customizing your Telnet services, and using Telnet key sequences, see the *Cisco IOS Configuration Fundamentals Configuration Guide.*

Using Telnet to access a console interface

To access a console interface using Telnet, complete the following steps:

SUMMARY STEPS

1. From your terminal or PC, enter one of the following commands:
2. At the password prompt, enter your login password. The following example shows entry of the password mypass:
3. From user EXEC mode, enter the **enable** command as shown in the following example:
4. At the password prompt, enter your system password. The following example shows entry of the password enablepass:
5. When the enable password is accepted, the privileged EXEC mode prompt appears:
6. You now have access to the CLI in privileged EXEC mode and you can enter the necessary commands to complete your desired tasks.
7. To exit the Telnet session, use the **exit** or **logout** command as shown in the following example:

DETAILED STEPS

Procedure

Step 1 From your terminal or PC, enter one of the following commands:

- **connect** *host* [*port*] [*keyword*]
- **telnet** *host* [*port*] [*keyword*]

In this syntax, *host* is the router hostname or an IP address, *port* is a decimal port number (23 is the default), and *keyword* is a supported keyword. For more information, see the *Cisco IOS Configuration Fundamentals Command Reference Guide* .

Note
If you are using an access server, then you will need to specify a valid port number such as **telnet 172.20.52.40 2004**, in addition to the hostname or IP address.

The following example shows the **telnet** command to connect to the router named router:

Example:

```
unix_host%telnet router
Trying 172.20.52.40...
Connected to 172.20.52.40.
Escape character is '^]'.
unix_host% connect
```

Step 2 At the password prompt, enter your login password. The following example shows entry of the password mypass:

Example:

```
User Access Verification
Password:mypass
```

Note
If no password has been configured, press **Return**.

Step 3 From user EXEC mode, enter the **enable** command as shown in the following example:

Example:

```
Router>enable
```

Step 4 At the password prompt, enter your system password. The following example shows entry of the password enablepass:

Example:

```
Password:enablepass
```

Step 5 When the enable password is accepted, the privileged EXEC mode prompt appears:

Example:

```
Router#
```

Step 6 You now have access to the CLI in privileged EXEC mode and you can enter the necessary commands to complete your desired tasks.

Step 7 To exit the Telnet session, use the **exit** or **logout** command as shown in the following example:

Example:

```
Router#logout
```

Using keyboard shortcuts

Commands are not case sensitive. You can abbreviate commands and parameters if the abbreviations contain enough letters to be different from any other currently available commands or parameters.

The following table lists the keyboard shortcuts for entering and editing commands.

Table 3: Keyboard Shortcuts

Keystrokes	Purpose
Ctrl-B or the **Left Arrow** key[1]	Move the cursor back one character
Ctrl-F orthe **Right Arrow** key1	Move the cursor forward one character
Ctrl-A	Move the cursor to the beginning of the command line
Ctrl-E	Move the cursor to the end of the command line
Esc B	Move the cursor back one word
Esc F	Move the cursor forward one word

[1] The arrow keys function only on ANSI-compatible terminals such as VT100s.

Using the history buffer to recall commands

The history buffer stores the last 20 commands you entered. History substitution allows you to access these commands without retyping them, by using special abbreviated commands.

The following table lists the history substitution commands.

Table 4: History Substitution Commands

Command	Purpose
Ctrl-P or the **Up Arrow** key[2]	Recall commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Ctrl-N or the **Down Arrow** key1	Return to more recent commands in the history buffer after recalling commands with **Ctrl-P** or the **Up Arrow** key.
Router# **show history**	While in EXEC mode, list the last several commands you have just entered.

[2] The arrow keys function only on ANSI-compatible terminals such as VT100s.

Understanding the command mode

The command modes available in the traditional Cisco IOS CLI are exactly the same as the command modes available in Cisco IOS XE.

You use the CLI to access Cisco IOS XE software. Because the CLI is divided into many different modes, the commands available to you at any given time depend on the mode that you are currently in. Entering a question mark (**?**) at the CLI prompt allows you to obtain a list of commands available for each command mode.

When you log in to the CLI, you are in user EXEC mode. User EXEC mode contains only a limited subset of commands. To have access to all commands, you must enter privileged EXEC mode, normally by using a password. From privileged EXEC mode, you can issue any EXEC command—user or privileged mode—or

you can enter global configuration mode. Most EXEC commands are one-time commands. For example, **show** commands show important status information, and **clear** commands clear counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and a variety of other modes, such as protocol-specific modes.

ROM monitor mode is a separate mode used when the Cisco IOS XE software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode.

The following table describes how to access and exit various common command modes of the Cisco IOS XE software. It also shows examples of the prompts displayed for each mode.

Table 5: Accessing and Exiting Command Modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** EXEC command.	`Router#`	To return to user EXEC mode, use the **disable** command.
Global configuration	From privileged EXEC mode, use the **configure terminal** privileged EXEC command.	`Router(config)#`	To return to privileged EXEC mode from global configuration mode, use the **exit** or **end** command.
Interface configuration	From global configuration mode, specify an interface using an **interface** command.	`Router(config-if)#`	To return to global configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command.

Command Mode	Access Method	Prompt	Exit Method
Diagnostic	The router boots up or accesses diagnostic mode in the following scenarios: In some cases, diagnostic mode will be reached when the IOS process or processes fail. In most scenarios, however, the router will. A user-configured access policy was configured using the **transport-map** command that directed the user into diagnostic mode. See the Chapter 4, "Console Port, Telnet, and SSH Handling" of this book for information on configuring access policies. The router was accessed using a Route Processor auxiliary port. A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) was entered and the router was configured to go into diagnostic mode when the break signal was received.	`Router(diag)#`	If the IOS process failing is the reason for entering diagnostic mode, the IOS problem must be resolved and the router rebooted to get out of diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or using a method that is configured to connect to the Cisco IOS CLI. If the router is accessed through the Route Processor auxiliary port, access the router through another port. Accessing the router through the auxiliary port is not useful for customer purposes anyway.
ROM monitor	From privileged EXEC mode, use the **reload** EXEC command. Press the **Break** key during the first 60 seconds while the system is booting.	>	To exit ROM monitor mode, use the **continue** command.

Getting help

Entering a question mark (**?**) at the CLI prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To get help specific to a command mode, a command, a keyword, or an argument, use one of the commands listed in the following table:

Table 6: Help Commands and Purpose

Command	Purpose
help	Provides a brief description of the help system in any command mode.
abbreviated-command-entry?	Provides a list of commands that begin with a particular character string. (No space between command and question mark.)
abbreviated-command-entry<Tab>	Completes a partial command name.
?	Lists all commands available for a particular command mode.
command ?	Lists the keywords or arguments that you must enter next on the command line. (Space between command and question mark.)

Finding command options

This section provides an example of how to display syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (**?**) at the configuration prompt or after entering part of a command followed by a space. The Cisco IOS XE software displays a list and brief description of available keywords and arguments. For example, if you were in global configuration mode and wanted to see all the keywords or arguments for the **arap** command, you would type **arap ?**.

The <cr> symbol in command help output stands for "carriage return." On older keyboards, the carriage return key is the Return key. On most modern keyboards, the carriage return key is the Enter key. The <cr> symbol at the end of command help output indicates that you have the option to press **Enter** to complete the command and that the arguments and keywords in the list preceding the <cr> symbol are optional. The <cr> symbol by itself indicates that no more arguments or keywords are available and that you must press **Enter** to complete the command.

The following table shows examples of how you can use the question mark (**?**) to assist you in entering commands.

Table 7: Finding Command Options

Command	Comment
Router>**enable** Password:*<password>* Router#	Enter the **enable** command and password to access privileged EXEC commands. You are in privileged EXEC mode when the prompt changes to a "# " from the "> "; for example, Router> to Router# .
Router# **configure terminal** Enter configuration commands, one per line. End with CNTL/Z. Router(config)#	Enter the **configure terminal** privileged EXEC command to enter global configuration mode. You are in global configuration mode when the prompt changes to Router(config)# .
Router(config)#**interface Ethernet ?** <0-6> Ethernet interface number Router(config)#**interface Ethernet 4 ?** / Router(config)#**interface Ethernet 4/ ?** <0-3> Ethernet interface number Router(config)#**interface Ethernet 4/0 ?** <cr> Router(config)#**interface Ethernet 4/0** Router(config-if)#	Enter interface configuration mode by specifying the ethernet interface that you want to configure using the **interface Ethernet** global configuration command. Enter **?** to display what you must enter next on the command line. In this example, you must enter the ethernet interface slot number and port number, separated by a forward slash. When the <cr> symbol is displayed, you can press Enter to complete the command. You are in interface configuration mode when the prompt changes to Router(config-if)# .

Command	Comment
Router(config-if)#? Interface configuration commands: . . . ip Interface Internet Protocol config commands keepalive Enable keepalive lan-name LAN Name command llc2 LLC2 Interface Subcommands load-interval Specify interval for load calculation for an interface locaddr-priority Assign a priority group logging Configure logging for interface loopback Configure internal loopback on an interface mac-address Manually set interface MAC address mls mls router sub/interface commands mpoa MPOA interface configuration commands mtu Set the interface Maximum Transmission Unit (MTU) netbios Use a defined NETBIOS access list or enable name-caching no Negate a command or set its defaults nrzi-encoding Enable use of NRZI encoding ntp Configure NTP . . . Router(config-if)#	Enter **?** to display a list of all the interface configuration commands available for the ethernet interface. This example shows only some of the available interface configuration commands.

Command	Comment
Router(config-if)#**ip ?** Interface IP configuration subcommands: access-group Specify access control for packets accounting Enable IP accounting on this interface address Set the IP address of an interface authentication authentication subcommands bandwidth-percent Set EIGRP bandwidth limit broadcast-address Set the broadcast address of an interface cgmp Enable/disable CGMP directed-broadcast Enable forwarding of directed broadcasts dvmrp DVMRP interface commands hello-interval Configures IP-EIGRP hello interval helper-address Specify a destination address for UDP broadcasts hold-time Configures IP-EIGRP hold time . . . Router(config-if)#**ip**	Enter the command that you want to configure for the interface. This example uses the **ip** command. Enter **?** to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands.
Router(config-if)#**ip address ?** A.B.C.D IP address negotiated IP Address negotiated over PPP Router(config-if)#**ip address**	Enter the command that you want to configure for the interface. This example uses the **ip address** command. Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP address or the **negotiated** keyword. A carriage return (<cr>) is not displayed; therefore, you must enter additional keywords or arguments to complete the command.
Router(config-if)#**ip address 172.16.0.1 ?** A.B.C.D IP subnet mask Router(config-if)#**ip address 172.16.0.1**	Enter the keyword or argument that you want to use. This example uses the 172.16.0.1 IP address. Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask. A <cr> is not displayed; therefore, you must enter additional keywords or arguments to complete the command.
Router(config-if)#**ip address 172.16.0.1 255.255.255.0 ?** secondary Make this IP address a secondary address <cr> Router(config-if)#**ip address 172.16.0.1 255.255.255.0**	Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask. Enter **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword, or you can press **Enter**. A <cr> is displayed; you can press **Enter** to complete the command, or you can enter another keyword.
Router(config-if)#**ip address 172.16.0.1 255.255.255.0** Router(config-if)#	In this example, **Enter** is pressed to complete the command.

Using the no and default forms of commands

Almost every configuration command has a **no** form. In general, use the **no** form to disable a function. Use the command without the **no** keyword to re-enable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** command; to re-enable IP routing, use the **ip routing** command. The Cisco IOS software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

Many CLI commands also have a **default** form. By issuing the command **default** *command-name* , you can configure the command to its default setting. The Cisco IOS software command reference publications describe the function of the **default** form of the command when the **default** form performs a different function than the plain and **no** forms of the command. To see what default commands are available on your system, enter **default ?** in the appropriate command mode.

Saving configuration changes

Use the **copy running-config startup-config** command to save your configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs. For example:

```
Router#copy running-config startup-config
Building configuration...
```

It might take a minute or two to save the configuration. After the configuration has been saved, the following output appears:

```
[OK]
Router#
```

This task saves the configuration to NVRAM.

Managing configuration files

On the Cisco 8500 Series Secure Routers, the startup configuration file is stored in the nvram: file system and the running-configuration files are stored in the system: file system. This configuration file storage setup is not unique to the Cisco 8500 Series Secure Routers and is used on several Cisco router platforms.

As a matter of routine maintenance on any Cisco router, users should backup the startup configuration file by copying the startup configuration file from NVRAM onto one of the router's other file systems and, additionally, onto a network server. Backing up the startup configuration file provides an easy method of recovering the startup configuration file in the event the startup configuration file in NVRAM becomes unusable for any reason.

The **copy** command can be used to backup startup configuration files. The following examples show the startup configuration file in NVRAM being backed up:

Example 1: Copying a Startup Configuration File to Bootflash

```
Router#dir bootflash:
Directory of bootflash:/
 11 drwx 16384 Sep 18 2020 15:16:35 +00:00 lost+found
1648321 drwx 4096 Oct 22 2020 12:08:47 +00:00 .installer
97921 drwx 4096 Sep 18 2020 15:18:00 +00:00 .rollback_timer
12 -rw- 1910 Oct 22 2020 12:09:09 +00:00 mode_event_log
1566721 drwx 4096 Sep 18 2020 15:33:23 +00:00 core
1215841 drwx 4096 Oct 22 2020 12:09:48 +00:00 .prst_sync
1289281 drwx 4096 Sep 18 2020 15:18:18 +00:00 bootlog_history
13 -rw- 133219 Oct 22 2020 12:09:34 +00:00 memleak.tcl
14 -rw- 20109 Sep 18 2020 15:18:39 +00:00 ios_core.p7b
15 -rwx 1314 Sep 18 2020 15:18:39 +00:00 trustidrootx3_ca.ca
391681 drwx 4096 Oct 6 2020 15:08:54 +00:00 .dbpersist
522241 drwx 4096 Sep 18 2020 15:32:59 +00:00 .inv
783361 drwx 49152 Oct 27 2020 08:36:44 +00:00 tracelogs
832321 drwx 4096 Sep 18 2020 15:19:17 +00:00 pnp-info
1207681 drwx 4096 Sep 18 2020 15:19:20 +00:00 onep
750721 drwx 4096 Oct 22 2020 12:09:57 +00:00 license_evlog
946561 drwx 4096 Sep 18 2020 15:19:24 +00:00 guest-share
383521 drwx 4096 Sep 18 2020 15:34:13 +00:00 pnp-tech
1583041 drwx 4096 Oct 22 2020 11:27:38 +00:00 EFI
16 -rw- 34 Oct 6 2020 13:56:03 +00:00 pnp-tech-time
17 -rw- 82790 Oct 6 2020 13:56:14 +00:00 pnp-tech-discovery-summary
18 -rw- 8425 Oct 6 2020 15:09:18 +00:00 1g_snake
19 -rw- 6858 Oct 7 2020 10:53:21 +00:00 100g_snake
20 -rw- 4705 Oct 22 2020 13:01:54 +00:00 startup-config

26975526912 bytes total (25538875392 bytes free)
Router#copy nvram:startup-config bootflash:
Destination filename [startup-config]?
3517 bytes copied in 0.647 secs (5436 bytes/sec)
```

Example 2: Copying a Startup Configuration File to USB Flash Disk

```
Router#dir usb0:
Directory of usb0:/
43261 -rwx 208904396 Oct 27 2020 14:10:20 -07:00
c8000aep-universalk9.17.02.01.SPA.bin
255497216 bytes total (40190464 bytes free)
Router#copy nvram:startup-config usb0:
Destination filename [startup-config]?
3172 bytes copied in 0.214 secs (14822 bytes/sec)
Router#dir usb0:
Directory of usb0:/
43261 -rwx 208904396 Oct 27 2020 14:10:20 -07:00
c8000aep-universalk9.17.02.01.SPA.bin
15:40:45 -07:00 startup-config255497216 bytes total (40186880 bytes free)
```

Example 3: Copying a Startup Configuration File to a TFTP Server

```
Router#copy bootflash:startup-config tftp:
Address or name of remote host []? 172.17.16.81
Destination filename [pe24_asr-1002-confg]? /auto/tftp-users/user/startup-config
!!
3517 bytes copied in 0.122 secs (28828 bytes/sec)
```

For more detailed information on managing configuration files, see the *Managing Configuration Files* section in the *Cisco IOS XE Configuration Fundamentals Configuration Guide*

Filtering the output of the show and more commands

You can search and filter the output of **show** and **more** commands. This functionality is useful if you need to sort through large amounts of output or if you want to exclude output that you need not see.

To use this functionality, enter a **show** or **more** command followed by the "pipe" character (|); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case sensitive):

show *command* | {**append** | **begin** | **exclude | include** | **redirect** | **section** | **tee**} *regular-expression*

The output matches certain lines of information in the configuration file. The following example illustrates how to use output modifiers with the **show interface** command when you want the output to include only lines in which the expression "protocol" appears:

```
Router# show interface | include protocol
FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

Disabling front-panel USB ports

SUMMARY STEPS

1. enable
2. configure terminal
3. platform usb disable
4. end
5. write memory

DETAILED STEPS

Procedure

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

	Command or Action	Purpose
Step 3	platform usb disable **Example:** `Device # platform usb disable`	Disables USB ports. **Note** For re-enabling of front-panel usb ports, use the no form of command (**no platform usb disable**).
Step 4	end **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 5	write memory	Save to configuration.

Configuration examples for disabling of front-panel USB ports

Example: Disabling Front-Panel USB Ports On Autonomous, Controller and vManage Mode

The following example shows the configuration of disabling front-panel USB ports on autonomous, controller and vManage mode:

```
Router# sh run | inc usb
platform usb disable
Router#
```

Verifying disabling of front panel USB ports

To verify the disabling of USB ports on your device, use the following show command:

show platform usb status

```
Router#show platform usb status
USB enabled
Router#
```

Powering off a router

Before you turn off a power supply, make certain the chassis is grounded and you perform a soft shutdown on the power supply. Not performing a soft shutdown will often not harm the router, but may cause problems in certain scenarios.

To perform a soft shutdown before powering off the router, enter the **reload** command to halt the system and then wait for ROM Monitor to execute before proceeding to the next step.

The following screenshot shows an example of this process:

```
Router#reload
Proceed with reload? [confirm]
...(Some messages are omitted here)
Initializing Hardware...
```

Place the power supply switch in the Off position after seeing this message.

Finding support information for platforms and Cisco software images

Cisco software is packaged in feature sets consisting of software images that support specific platforms. The feature sets available for a specific platform depend on which Cisco software images are included in a release. To identify the set of software images available in a specific release or to find out if a feature is available in a given Cisco IOS XE software image, you can use Cisco Feature Navigator or the software release notes.

Using the Cisco feature navigator

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS XE software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn . An account on Cisco.com is not required.

Using the software advisor

To see if a feature is supported by a Cisco IOS XE release, to locate the software document for that feature, or to check the minimum software requirements of Cisco IOS XE software with the hardware installed on your router, Cisco maintains the Software Advisor tool on Cisco.com at http://www.cisco.com/cgi-bin/Support/CompNav/Index.pl.

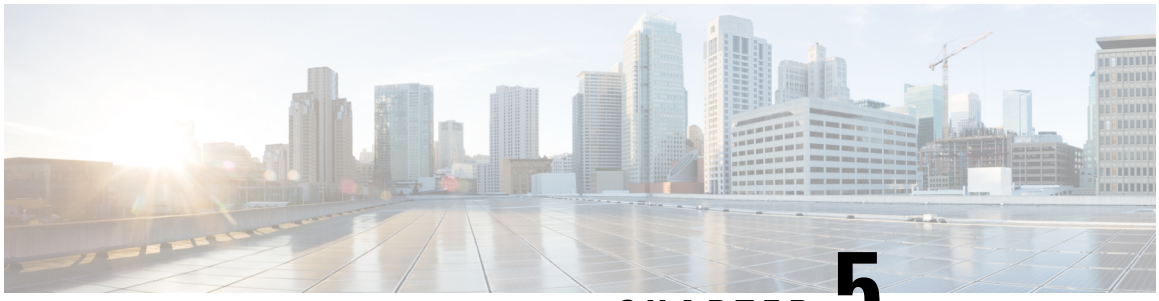
You must be a registered user on Cisco.com to access this tool.

Using the software release notes

Cisco IOS XE software releases include release notes that provide the following information:

- Platform support information
- Memory recommendations
- New feature information
- Open and resolved severity 1 and 2 caveats for all platforms

Release notes are intended to be release-specific for the most current release, and the information provided in these documents may not be cumulative in providing information about features that first appeared in previous releases. Refer to Cisco Feature Navigator for cumulative feature information.



CHAPTER 5

Tamper Detection

Tamper detection is a security feature implemented in Cisco 8000 Series Secure Routers to identify potential tampering events. Each router is shipped from the manufacturer with its chassis cover securely fastened. If the chassis cover is opened after shipment, the hardware records all chassis cover open and close events in tamper-proof memory, regardless of whether the device is powered on or off.

On boot up, the software reads the latest event index and compares it with previously known indices. When there is a discrepancy, the software generates a SYSlog message during startup to report the tamper event. When the device is in fully powered up state, software will generate syslog message and SNMP trap immediately.

Benefits

Tamper Detection notification detects unauthorized physical access or attempts to compromise the device, helping protect sensitive data and network integrity.

Limitations

Tamper Detection feature is currently not supported on SDWAN/SD-Routing mode.

Configure Tamper Detection

Tamper detection is enabled by default on the router. The open or close events of the chassis cover are auto recorded.

The tamper event notification can be enabled or disabled using these commands in config mode.

- Enable the Tamper Detection notification using this command in config mode (enabled by default):

```
Router(config)#platform tamper detection
```

- Disable the Tamper Detection notification using the command in config mode:

```
Router(config)#no platform tamper detection
```

Mark tamper detection events

A Tamper event marking is a feature that

- allows authorized users to prevent specific activities from appearing as tamper events,
- enables the system to distinguish between authorized and unauthorized access, and
- maintains the accuracy of tamper event logs.

Configure tamper detection events

To prevent authorized activities from appearing as tamper events in logs. Follow these steps.

Procedure

Step 1 Generate a challenge to obtain the consent token using the below command.

Example:

```
request consent-token generate-challenge tamper-auth auth-timeout <mins>
```

This step ensures that the consent token is generated only by an authorized user.

Step 2 Once the challenge is verified and the consent token is generated, accept the consent token using this command.

Example:

```
request consent-token accept-response tamper-auth <consent token>
```

Step 3 Mark the tamper detection event using the command below

Example:

```
request platform hardware tamper-detection event-mark
```

request platform hardware tamper-detection event-mark command is supported from Cisco IOS XE 17.17.1a

The last known event index and timestamp are marked, preventing authorized activities from appearing as tamper events in the log.

Verify the Tamper Detection events

```
Router# show platform tamper-detection event [power-off | power-on] [all | lastx | new]
```

Option	Description
power off	The **power off** option specifies the tampering events when router doesn't have power cable connected.

Option	Description
power on	The **power on** option specifies the tampering events when the router was powered on.
all	The **all** option specifies all the recorded tampering events. System can show maximum 500 entries; every 500 entries will increase one rollover counter.
lastx	The **lastx** option specifies the number of events to display. For example, "lastx 10" will show the last 10 events.
new	The **new** option specifies the new tampering events since last known events index.

The show command provides an event log which contains these details:

- Current event index
- Current time
- Rollover Status and Rollover Count:
  - When tamper event count <= 500, the Rollover Count is 0, Rollover Status: No
  - For every 500 logs, the Rollover Count will be increased by 1:
  - When 501-1000 overwrites 1-500, the Rollover Count is 1, Rollover Status: Yes
  - When 1001-1500 overwrites 501-1000, the Rollover Count is 2, Rollover Status: Yes
- The events indicating event type and timestamp

Verify events when system is powered off

When the system power is off, the router uses battery power to record any tampering events. The router records the first chassis cover open event between last power off and next power on. If chassis cover was open or closed many times during power off, only the first open event is recorded. An example shows the event log when the system is powere off :

```
Router#show platform tamper-detection event power-off all
Current Time: 2025/04/25 19:55:03       Rollover Status: No     Rollover Count: 0
-------------------------------------------------------------------------------------------------
Tamper event index      |       Tamper event timestamp  |       Tamper events description
-------------------------------------------------------------------------------------------------

    #2                          2024/08/08 02:36:41             Chassis is opened
    #1                         2000/00/00 00:00:00            Battery not present or used
 up
```

Verify events when system is partially or fully powered on

When system power is available, the router records all chassis cover open or close events. An example of the event log when the system is partially or fully powered on:

```
Router show platform tamper-detection event power-on lastx 10
Current Time: 2025/04/25 19:54:46      Rollover Status: No     Rollover Count: 0
-------------------------------------------------------------------------------------------------------
Tamper event index      |       Tamper event timestamp  |       Tamper events description
-------------------------------------------------------------------------------------------------------

    #2                  025/04/24 22:10:14             Chassis is opened
    #1                   025/04/24 22:02:33             Chassis is closed
```

Tamper Detection SYSlog

When the router boots up, it reads the current event index from the event log and compares it with the last known index stored previously. If there is a mismatch between the indices or the timestamps differ, IOS will generate a warning-level SYSlog message and send a notification to the controller in controller mode.

This section provides examples of distinct SYSlog events.

- SYSlog for power-on events

  When Tamper Detection is enabled and the system is powered on, a power-on SYSlog message will be displayed during boot-up.

  ```
  Apr 25 20:15:01.064: %CMRP-4-TAMPER_DETECTION_EVENT_MSG: R0/0: cmand: System cover was
   opened 0 times and closed 0 times during power up since last known event index 7 at
  2025/04/24 22:11:51
  ```

- SYSlog for power-off events

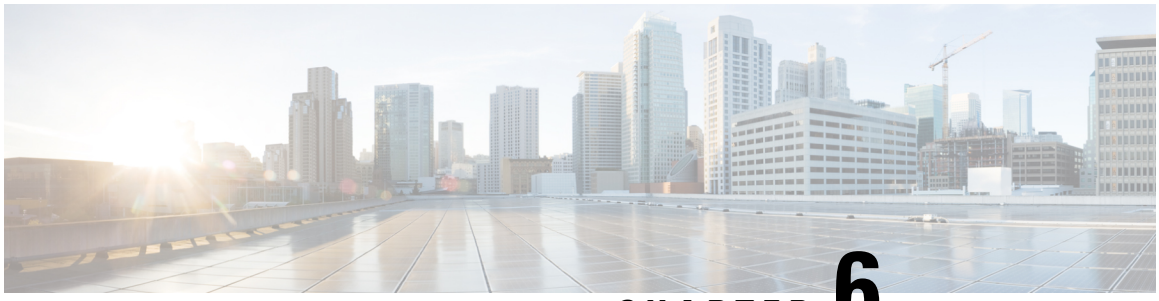
  When Tamper Detection is enabled and the system is powered off, a power-off SYSlog message will be displayed during boot-up.

  ```
  Apr 25 20:15:01.064: %CMRP-4-TAMPER_DETECTION_EVENT_MSG: R0/0: cmand: System cover was
   opened 0 times and closed 0 times during power down since last known event index 5 at
   2025/04/24 22:11:51
  ```

- SYSlog for runtime events

  ```
  *Aug 29 06:56:34.560: %CMRP-4-INTRUSION_ALERT: R0/0: cmand: The system cover has been
  opened !!
  *Aug 29 06:56:36.130: %CMRP-4-TAMPER_DETECTION_EVENT_MSG: R0/0: cmand: System cover was
   opened 2 times and closed 0 times during power down since last known event index 50
  at 2025/06/04 08:03:06
  *Aug 29 06:56:36.130: %CMRP-4-TAMPER_DETECTION_EVENT_MSG: R0/0: cmand: System cover was
   opened 20 times and closed 20 times during power up since last known event index 1638
   at 2025/06/05 07:08:12

  *Aug 29 06:57:04.563: %CMRP-4-INTRUSION_ALERT: R0/0: cmand: The system cover has been
  closed !!
  *Aug 29 06:57:06.137: %CMRP-4-TAMPER_DETECTION_EVENT_MSG: R0/0: cmand: System cover was
   opened 2 times and closed 0 times during power down since last known event index 50
  at 2025/06/04 08:03:06
  *Aug 29 06:57:06.137: %CMRP-4-TAMPER_DETECTION_EVENT_MSG: R0/0: cmand: System cover was
   opened 20 times and closed 21 times during power up since last known event index 1638
   at 2025/06/05 07:08:12
  ```

Note If the Tamper Detection feature is disabled, the SYSlog messages will not be displayed at boot up.



CHAPTER 6

Bay Configuration

Bay configuration C8570-G2

On C8570-G2 there are three built-in EPAs that are configurable.

The following table describes the port details:

Bay Number	EPA	Port Configuration	Interface numbers
Bay 0 8xSFP+	1/10G EPA	Eight 1/10G interfaces - TE0 - TE7 Disabled when 100G port in used in Bay 1	0/0/0 0/0/1 0/0/2 0/0/3 0/0/4 0/0/5 0/0/6 0/0/7

Bay Number	EPA	Port Configuration	Interface numbers
Bay 1 4xSFP+/1xQSFP	1/10/40/100G EPA	Four 1/10G interfaces active - TE0 - TE3 (interfaces 0/1/0 … 0/1/3) The bay can be used in the following modes: • Four 1/10G interfaces • One 40G interface active • One 100G interface. This utilizes the eight 1/10G ports of Bay 0	0/1/0 0/1/1 0/1/1 0/1/3
Bay 2 3xQSFP	40/100G EPA	Three 40G interfaces (0/2/0, 0/2/4, 0/2/8) One 100G interface (0/2/0)	0/2/0 0/2/4 0/2/8

Note The speed of a 10G interface can be 1G or 10G based on the SFP transceiver plugged into to the port. Even when the speed changes the interface name is still indicated as TenGigabitEthernet.

By default , C8570-G2 operates Bay 1 in 10G mode and Bay 2 in 40G mode. The Bay 1 mode can be changed from 10G to 40G to 100G and vice versa. But if Bay 1 is set to 100G, all ports of Bay 0 move to *admin down* state and the ports are no longer functional.

The Bay 2 mode can be changed from 40G to 100G and vice versa. The mode change on Bay 2 does not impact traffic on Bay 1.

Use the **show platform** and **show ip interface** commands to view the bay and interface details:

```
Router#show platform
Chassis type: C8570-G2

Slot      Type                State                 Insert time (ago)
--------- ------------------- --------------------- -----------------
0         C8570-G2            ok                    2w6d
 0/0      8xSFP+              ok                    2w6d
 0/1      4xSFP+/1xQSFP       ok                    2w6d
 0/2      3xQSFP              ok                    2w6d
R0        C8570-G2            ok, active            2w6d
F0        C8570-G2            ok, active            2w6d
P0        PWR-CH1-750WACR     ok                    2w6d
P1        Unknown             empty                 never
P2        C8500-FAN-1R        ok                    2w6d

Slot      CPLD Version        Firmware Version
--------- ------------------- ------------------------------------
```

```
0          23122108            17.15(5r)
R0         23122108            17.15(5r)
F0         23122108            17.15(5r)


Router#show ip interface
Te0/0/0                unassigned      YES NVRAM  down                    down
Te0/0/1                unassigned      YES NVRAM  down                    down
Te0/0/2                unassigned      YES NVRAM  down                    down
Te0/0/3                unassigned      YES NVRAM  down                    down
Te0/0/4                unassigned      YES NVRAM  down                    down
Te0/0/5                unassigned      YES NVRAM  down                    down
Te0/0/6                unassigned      YES NVRAM  down                    down
Te0/0/7                unassigned      YES NVRAM  down                    down
Te0/1/0                unassigned      YES NVRAM  down                    down
Te0/1/1                unassigned      YES NVRAM  down                    down
Te0/1/2                unassigned      YES NVRAM  down                    down
Te0/1/3                unassigned      YES NVRAM  down                    down
Fo0/2/0                unassigned      YES unset  down                    down
Fo0/2/4                unassigned      YES unset  down                    down
Fo0/2/8                unassigned      YES unset  down                    down
GigabitEthernet0       10.104.33.213   YES NVRAM  up                      up
Router#
```

Bay configuration examples

The following examples show how mode can be changed on C8570-G2 to achieve different traffic speeds:

Examples

The following example shows how to change to 40G mode on Bay 1 of C8570-G2:

```
Router(config)# hw-module subslot 0/1 mode 40G
Present configuration of this subslot will be erased and will not be restored.
CLI will not be available until mode change is complete and EPA returns to OK state.
Do you want to proceed? [confirm]

*Jul  7 08:46:56.550: 4xSFP+/1xQSFP[0/1] : config for spa port 0 would be lost
*Jul  7 08:46:56.556: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:46:56.556: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:46:56.557: 4xSFP+/1xQSFP[0/1] : TenGigabitEthernet0/1/0 moved to default config
*Jul  7 08:46:56.557: 4xSFP+/1xQSFP[0/1] : config for spa port 1 would be lost
*Jul  7 08:46:56.561: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:46:56.562: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:46:56.562: 4xSFP+/1xQSFP[0/1] : TenGigabitEthernet0/1/1 moved to default config
*Jul  7 08:46:56.562: 4xSFP+/1xQSFP[0/1] : config for spa port 2 would be lost
*Jul  7 08:46:56.566: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:46:56.567: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:46:56.567: 4xSFP+/1xQSFP[0/1] : TenGigabitEthernet0/1/2 moved to default config
*Jul  7 08:46:56.567: 4xSFP+/1xQSFP[0/1] : config for spa port 3 would be lost
*Jul  7 08:46:56.571: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:46:56.572: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
```

```
*Jul  7 08:46:56.572: 4xSFP+/1xQSFP[0/1] : TenGigabitEthernet0/1/3 moved to default config
*Jul  7 08:46:57.572: 4xSFP+/1xQSFP[0/1] : Received mode change request from 10G to 40G!
system_configured TRUE
*Jul  7 08:46:57.586: %IOSXE_OIR-6-SOFT_RELOADSPA: SPA(4xSFP+/1xQSFP) reloaded on subslot
0/1
*Jul  7 08:46:57.588: 4xSFP+/1xQSFP[0/1] : EPA moving from 10G mode to 40G mode
*Jul  7 08:46:57.588: 4xSFP+/1xQSFP[0/1] : config for spa port 0 would be lost
*Jul  7 08:46:57.589: 4xSFP+/1xQSFP[0/1] : config for spa port 1 would be lost
*Jul  7 08:46:57.589: 4xSFP+/1xQSFP[0/1] : config for spa port 2 would be lost
*Jul  7 08:46:57.590: 4xSFP+/1xQSFP[0/1] : config for spa port 3 would be lost
*Jul  7 08:46:57.590: 4xSFP+/1xQSFP[0/1] : Old mode cleanup done!
*Jul  7 08:46:57.593: %SPA_OIR-6-OFFLINECARD: SPA (4xSFP+/1xQSFP) offline in subslot 0/1
*Jul  7 08:47:02.828: 4xSFP+/1xQSFP[0/1] : Number of ports 1
Encore(config)#
*Jul  7 08:47:10.402: %SPA_OIR-6-ONLINECARD: SPA (4xSFP+/1xQSFP) online in subslot 0/1
```

The following example shows how to change to 40G mode to 100G on Bay 1 of C8570-G2:

```
Router(config)# hw-module subslot 0/1 mode 100G
Present configuration of this subslot will be erased and will not be restored.
CLI will not be available until mode change is complete and EPA returns to OK state.
Do you want to proceed? [confirm]

*Jul  7 08:39:21.152: 4xSFP+/1xQSFP[0/1] : config for spa port 0 would be lost
*Jul  7 08:39:21.165: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:21.165: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:21.166: 4xSFP+/1xQSFP[0/1] : FortyGigabitEthernet0/1/0 moved to default config
*Jul  7 08:39:22.165: 8xSFP+[0/0] : config for spa port 0 would be lost
*Jul  7 08:39:22.171: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.172: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.172: 8xSFP+[0/0] : TenGigabitEthernet0/0/0 moved to default config
*Jul  7 08:39:22.172: 8xSFP+[0/0] : config for spa port 1 would be lost
*Jul  7 08:39:22.176: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.177: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.177: 8xSFP+[0/0] : TenGigabitEthernet0/0/1 moved to default config
*Jul  7 08:39:22.177: 8xSFP+[0/0] : config for spa port 2 would be lost
*Jul  7 08:39:22.181: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.182: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.182: 8xSFP+[0/0] : TenGigabitEthernet0/0/2 moved to default config
*Jul  7 08:39:22.182: 8xSFP+[0/0] : config for spa port 3 would be lost
*Jul  7 08:39:22.186: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.186: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.187: 8xSFP+[0/0] : TenGigabitEthernet0/0/3 moved to default config
*Jul  7 08:39:22.187: 8xSFP+[0/0] : config for spa port 4 would be lost
*Jul  7 08:39:22.193: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.194: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.194: 8xSFP+[0/0] : TenGigabitEthernet0/0/4 moved to default config
*Jul  7 08:39:22.194: 8xSFP+[0/0] : config for spa port 5 would be lost
*Jul  7 08:39:22.199: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.199: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
```

```
*Jul  7 08:39:22.200: 8xSFP+[0/0] : TenGigabitEthernet0/0/5 moved to default config
*Jul  7 08:39:22.200: 8xSFP+[0/0] : config for spa port 6 would be lost
*Jul  7 08:39:22.204: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.204: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.205: 8xSFP+[0/0] : TenGigabitEthernet0/0/6 moved to default config
*Jul  7 08:39:22.205: 8xSFP+[0/0] : config for spa port 7 would be lost
*Jul  7 08:39:22.209: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.209: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:39:22.210: 8xSFP+[0/0] : TenGigabitEthernet0/0/7 moved to default config
*Jul  7 08:39:23.210: 4xSFP+/1xQSFP[0/1] : Received mode change request from 40G to 100G!
system_configured TRUE
*Jul  7 08:39:23.210: %SPA_OIR-6-SHUTDOWN: subslot 0/0 is administratively shutdown; Use
'no hw-module  shutdown' to enable
*Jul  7 08:39:23.244: %SPA_OIR-6-OFFLINECARD: SPA (8xSFP+) offline in subslot 0/0
*Jul  7 08:39:23.250: %IOSXE_OIR-6-SOFT_RELOADSPA: SPA(4xSFP+/1xQSFP) reloaded on subslot
0/1
*Jul  7 08:39:23.251: 4xSFP+/1xQSFP[0/1] : EPA moving from 40G mode to 100G mode
*Jul  7 08:39:23.251: 4xSFP+/1xQSFP[0/1] : config for spa port 0 would be lost
*Jul  7 08:39:23.252: 4xSFP+/1xQSFP[0/1] : Old mode cleanup done!
*Jul  7 08:39:23.252: %SPA_OIR-6-OFFLINECARD: SPA (4xSFP+/1xQSFP) offline in subslot 0/1
*Jul  7 08:39:28.599: 4xSFP+/1xQSFP[0/1] : Number of ports 1
*Jul  7 08:39:38.023: %SPA_OIR-6-ONLINECARD: SPA (4xSFP+/1xQSFP) online in subslot 0/1
```

The following example shows how to change to 10G mode from 100G on Bay 1 of C8570-G2:

```
Router(config)# hw-module subslot 0/1 mode 10G
Present configuration of this subslot will be erased and will not be restored.
CLI will not be available until mode change is complete and EPA returns to OK state.
Do you want to proceed? [confirm]

*Jul  7 08:45:59.779: 4xSFP+/1xQSFP[0/1] : config for spa port 0 would be lost
*Jul  7 08:45:59.785: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:45:59.785: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:45:59.786: 4xSFP+/1xQSFP[0/1] : FortyGigabitEthernet0/1/0 moved to default config
*Jul  7 08:46:00.785: 4xSFP+/1xQSFP[0/1] : Received mode change request from 40G to 10G!
system_configured TRUE
*Jul  7 08:46:00.790: %IOSXE_OIR-6-SOFT_RELOADSPA: SPA(4xSFP+/1xQSFP) reloaded on subslot
0/1
*Jul  7 08:46:00.791: 4xSFP+/1xQSFP[0/1] : EPA moving from 40G mode to 10G mode
*Jul  7 08:46:00.791: 4xSFP+/1xQSFP[0/1] : config for spa port 0 would be lost
*Jul  7 08:46:00.791: 4xSFP+/1xQSFP[0/1] : Old mode cleanup done!
*Jul  7 08:46:00.792: %SPA_OIR-6-OFFLINECARD: SPA (4xSFP+/1xQSFP) offline in subslot 0/1
*Jul  7 08:46:06.025: 4xSFP+/1xQSFP[0/1] : Number of ports 4
Encore(config)#
*Jul  7 08:46:13.676:  Dot3 Stats : 0/3 not valid intf
*Jul  7 08:46:13.684: %SPA_OIR-6-ONLINECARD: SPA (4xSFP+/1xQSFP) online in subslot 0/1
*Jul  7 08:46:15.675: %LINK-3-UPDOWN: Interface TenGigabitEthernet0/1/0, changed state to
down
*Jul  7 08:46:15.676: %LINK-3-UPDOWN: Interface TenGigabitEthernet0/1/1, changed state to
down
*Jul  7 08:46:15.677: %LINK-3-UPDOWN: Interface TenGigabitEthernet0/1/2, changed state to
down
*Jul  7 08:46:15.678: %LINK-3-UPDOWN: Interface TenGigabitEthernet0/1/3, changed state to
down
*Jul  7 08:46:15.687: %LINK-3-UPDOWN: SIP0/1: Interface TenGigabitEthernet0/1/0, changed
state to down
*Jul  7 08:46:19.254: %LINK-3-UPDOWN: Interface TenGigabitEthernet0/1/0, changed state to
up
```

```
*Jul  7 08:46:20.254: %LINEPROTO-5-UPDOWN: Line protocol on Interface TenGigabitEthernet0/1/0,
 changed state to up
*Jul  7 08:46:19.254: %LINK-3-UPDOWN: SIP0/1: Interface TenGigabitEthernet0/1/0, changed
state to up
```

The following example shows how to change to 100G mode from 100G on Bay 2 of C8570-G2:

```
Router(config)# hw-module subslot 0/2 mode 100G
Present configuration of this subslot will be erased and will not be restored.
CLI will not be available until mode change is complete and EPA returns to OK state.
Do you want to proceed? [confirm]

*Jul  7 08:48:15.432: 3xQSFP[0/2] : config for spa port 0 would be lost
*Jul  7 08:48:15.462: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:48:15.463: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:48:15.463: 3xQSFP[0/2] : FortyGigabitEthernet0/2/0 moved to default config
*Jul  7 08:48:15.463: 3xQSFP[0/2] : config for spa port 1 would be lost
*Jul  7 08:48:15.469: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:48:15.470: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:48:15.470: 3xQSFP[0/2] : FortyGigabitEthernet0/2/4 moved to default config
*Jul  7 08:48:15.470: 3xQSFP[0/2] : config for spa port 2 would be lost
*Jul  7 08:48:15.475: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:48:15.476: %SYS-5-CONFIG_P: Configured programmatically by process Exec from
console as console
*Jul  7 08:48:15.476: 3xQSFP[0/2] : FortyGigabitEthernet0/2/8 moved to default config
*Jul  7 08:48:16.476: 3xQSFP[0/2] : Received mode change request from 40G to 100G!
system_configured TRUE
*Jul  7 08:48:16.487: %IOSXE_OIR-6-SOFT_RELOADSPA: SPA(3xQSFP) reloaded on subslot 0/2
*Jul  7 08:48:16.489: 3xQSFP[0/2] : EPA moving from 40G mode to 100G mode
*Jul  7 08:48:16.489: 3xQSFP[0/2] : config for spa port 0 would be lost
*Jul  7 08:48:16.490: 3xQSFP[0/2] : config for spa port 1 would be lost
*Jul  7 08:48:16.490: 3xQSFP[0/2] : config for spa port 2 would be lost
*Jul  7 08:48:16.491: 3xQSFP[0/2] : Old mode cleanup done!
*Jul  7 08:48:16.493: %SPA_OIR-6-OFFLINECARD: SPA (3xQSFP) offline in subslot 0/2
*Jul  7 08:48:21.731: 3xQSFP[0/2] : Number of ports 1
*Jul  7 08:48:21.733: 3xQSFP[0/2] : XCVR namestring create: Maximum number of XCVR = 1
Encore(config)#
Encore(config)#
*Jul  7 08:48:35.865: %SPA_OIR-6-ONLINECARD: SPA (3xQSFP) online in subslot 0/2
```

Breakout support

Understand breakout support

Breakout support for a port helps to split a higher density port to multiple independent and logical ports. Breakout support is introduced in Bay 2 of C8570-G2 that supports breakout capable 40G native ports. The breakout support is of 4X10G and uses a 3-tuple approach.

The table below explains the interface names when breakout is configured.

Table 8: Interface Names when Breakout is Configured

Sr. No	Interface names	Description
	Te0/2/0, Te0/2/1, Te0/2/2, Te0/2/3, Te0/2/4, Te0/2/5, Te0/2/6, Te0/2/7, Te0/2/8, Te0/2/9, Te0/2/10, Te0/2/11	All three 40 G native ports working in 10G breakout mode
	Fo0/2/0,Fo0/2/4, Te0/2/8, Te0/2/9, Te0/2/10, Te0/2/11	1st native port in 40G mode 2nd native port in 40G mode 3rd native port in 10G breakout mode
	Fo0/2/0, Te0/2/4, Te0/2/5, Te0/2/6, Te0/2/7 Fo0/2/8	1st native port in 40G mode 2nd native port 10G breakout mode 3rd native port in 40G mode
	Te0/2/0, Te0/2/1, Te0/2/2, Te0/2/3, Fo0/2/4, Fo0/2/8	1st native port in 10G breakout mode 2nd native port in 40G mode 3rd native port in 40G mode
	Fo0/2/0, Te0/2/4, Te0/2/5, Te0/2/6, Te0/2/7, Te0/2/8, Te0/2/9, Te0/2/10, Te0/2/11	1st native port in 40G mode 2nd native port in 10G breakout mode 3rd native port in 10G breakout mode
	Te0/2/0, Te0/2/1, Te0/2/2, Te0/2/3, Te0/2/4, Te0/2/5, Te0/2/6, Te0/2/7, Fo0/2/8	1st native port in 10G breakout mode 2nd native port in 10G breakout mode 3rd native port in 40G mode
	Te0/2/0, Te0/2/1, Te0/2/2, Te0/2/3, Fo0/2/4, Te0/2/8, Te0/2/9, Te0/2/10, Te0/2/11	1st native port in 10G breakout mode 2nd native port in 40G mode 3rd native port in 10G breakout mode

Breakout support

Note Before using the breakout capability, ensure that Bay 2 is configured in 40G mode

```
Router(config)#hw-module subslot 0/2 breakout 10G port ?

all                configure all native ports in breakout mode
native_port_0      configure native port 0 in breakout mode
native_port_4      configure native port 4 in breakout mode
native_port_8      configure native port 8 in breakout mode
```

Sample commands to configure breakout support

When native_port 0 and 8 are in 10G breakout and native_port 4 is running in 40G mode

```
hw-module subslot 0/2 breakout 10g port native_port_0
hw-module subslot 0/2 breakout 10g port native_port_8
```

When all three native 40G ports have same breakout config

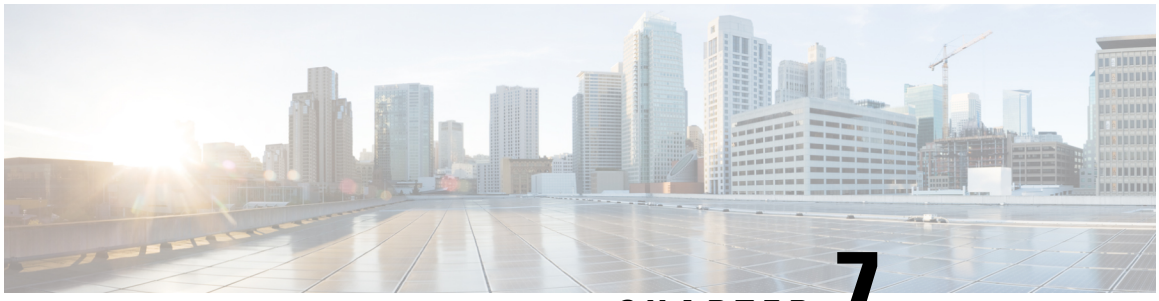
```
hw-module subslot 0/2 breakout 10g port all
hw-module subslot 0/2 breakout none port all
```

When you want to remove breakout configuration from all ports

```
hw-module subslot 0/2 breakout none port all
```

Bay configuration C8550-G2

On C8550-G2 there is one built-in EPA that supports ports TE0 - TE11 for SFP/SFP+ transceivers.



CHAPTER 7

Consolidated Package Management

This chapter discusses how consolidated packages are managed and are used to run the Cisco 8500 Series Secure Routers.

It contains the following sections:

Running the Cisco 8500 Series Secure Routers: an overview

The Cisco 8500 Series Secure Routers can be run using a complete consolidated package.

This section covers the following topics:

Running the Cisco 8500 Series Secure Routers using a consolidated package: an overview

The Cisco 8500 Series Secure Routers can be configured to run using a consolidated package.

When the router is configured to run using a consolidated package, the entire consolidated package file is copied onto the router or accessed by the router via TFTP or another network transport method. The router runs using the consolidated package file.

When a Cisco 8500 Series Secure Routers is configured to run using the consolidated package file, more memory is required to process router requests because the router has to search one larger file for every request. The peak amount of memory available for passing network traffic is therefore lower when the router is configured to run using a consolidated package.

A Cisco 8500 Series Secure Routers configured to run using a consolidated package is booted by booting the consolidated package file.

A consolidated package can be booted and utilized using TFTP or another network transport method.Running the router using a consolidated package may be the right method of running the router in certain networking environments.

The consolidated package should be stored on bootflash:, usb[0]:, or a remote file system when this method is used to run the router.

Running the Cisco 8500 Series Secure Routers: a summary

This section summarizes the advantages and disadvantages of each method of running your Cisco 8500 Series Secure Routers.

The advantages of running your router using a consolidated package include:

- Simplified installation—Only one software file needs to be managed instead of several separate images.
- Storage—A consolidated package can be used to run the router while being stored in bootflash:, on a USB Flash disk, or on a network server. A consolidated package can be booted and utilized using TFTP or another network transport method.

Software file management using command sets

Software files can be managed on the Cisco 8500 Series Secure Routers: using three distinct command sets. This section provides overviews of the following command sets:

The request platform Command Set

The **request platform software package** command is part of the larger **request platform** command set being introduced on the isco 8500 Series Secure Routers. For additional information on each **request platform** command and the options available with each command, see the *Cisco IOS Configuration Fundamentals Command Reference* .

The **request platform software package** command, which can be used to upgrade individual subpackages and a complete consolidated package, is used to upgrade software on the Cisco 8500 Series Secure Routers. Notably, the **request platform software package** command is the recommended way of performing an individual subpackage upgrade, and also provides the only method of no-downtime upgrades of individual subpackages on the router when the router is running individual subpackages.

The **request platform software package** command requires that the destination device or process be specified in the command line, so the commands can be used to upgrade software on both an active or a standby processor. The **request platform software package** command allows for no downtime software upgrades in many scenarios.

The basic syntax of the command is **request platform software package install rp** *rp-slot-number* **file** *file-URL*, where *rp-slot-number* is the number of the RP slot and *file-URL* is the path to the file being used to upgrade the Cisco 8500 Series Secure Routers. The command has other options; see the **request platform software package** command references for information on all of the options available with this command set.

The copy command

To upgrade a consolidated package on the isco 8500 Series Secure Routers, copy the consolidated package onto a file system, usually bootflash: or usb[0-1]: on the router, using the **copy** command as you would on most other Cisco routers. After making this copy, configure the router to boot using the consolidated package file.

See the **copy** command reference for a list of the options that are available with the **copy** command.

Managing and configuring the router to tun using consolidated packages

This section discusses the following topics:

Quick start software upgrade

The following instructions provide a quick start version of upgrading the software running the Cisco 8500 Series Secure Routers. These instructions assume you have access to the consolidated package and that the files will be stored in a bootflash: file system and has enough room for the file or files.

For more detailed installation examples, see the other sections of this chapter.

To upgrade the software using a quick start version, perform the following steps:

SUMMARY STEPS

1. Copy the consolidated package into bootflash: using the **copy** *URL-to-image* **bootflash:** command.
2. Enter the **dir bootflash:** command to verify your consolidated package in the directory.
3. Set up the boot parameters for your boot. Set the configuration register to 0x2 by entering the **config-register 0x2102** global configuration command, and enter the **boot system flash bootflash:***image-name*
4. Enter **copy running-config startup-config** to save your configuration.
5. Enter the **reload** command to reload the router and finish the boot. The upgraded software should be running when the reload completes.

DETAILED STEPS

Procedure

Step 1	Copy the consolidated package into bootflash: using the **copy** *URL-to-image* **bootflash:** command.
Step 2	Enter the **dir bootflash:** command to verify your consolidated package in the directory.
Step 3	Set up the boot parameters for your boot. Set the configuration register to 0x2 by entering the **config-register 0x2102** global configuration command, and enter the **boot system flash bootflash:***image-name*
Step 4	Enter **copy running-config startup-config** to save your configuration.
Step 5	Enter the **reload** command to reload the router and finish the boot. The upgraded software should be running when the reload completes.

Managing and configuring a router to run using a consolidated package

This section documents the following procedures:

Managing and configuring a consolidated package using the copy command

To upgrade a consolidated package on the Cisco 8500 Series Secure Routers using the **copy** command, copy the consolidated package into the bootflash: directory on the router using the **copy** command as you would on most other Cisco routers. After making this copy, configure the router to boot using the consolidated package file.

In the following example, the consolidated package file is copied onto the bootflash: file system from TFTP. The config-register is then set to boot using **boot system** commands, and the **boot system** commands instruct the router to boot using the consolidated package stored in the bootflash: file system. The new configuration is then saved using the **copy running-config startup-config** command, and the system is then reloaded to complete the process.

```
Router# dir bootflash:
Directory of bootflash:/

2203649  drwx            40960   Jul 7 2025 14:06:44 +05:30  tracelogs
1630209  drwx             4096   Jul 7 2025 13:28:19 +05:30  memaudit_log
90113   drwx             8192   Jul 7 2025 10:29:48 +05:30  license_evlog
17506305  drwx             4096   Jul 7 2025 10:28:06 +05:30  sdavc
13      -rw-           144302   Jul 7 2025 10:27:43 +05:30  memleak.tcl
7946241  drwx             4096   Jul 7 2025 10:27:29 +05:30  .inv
12      -rwx            32397   Jul 7 2025 10:27:27 +05:30  mode_event_log
12558337  drwx             4096   Jul 6 2025 19:49:31 +05:30  sysboot
9609217  drwx             4096   Jul 4 2025 10:12:42 +05:30  .sdp_install
11886593  drwx             4096   Jul 3 2025 13:41:56 +05:30  core
4890625  drwx             4096   Jul 2 2025 11:48:30 +05:30  system_report_stage
11206657  drwx             4096   Jul 2 2025 11:48:26 +05:30  .prst_sync
13099009  drwx             4096   Jul 2 2025 08:35:30 +05:30  .rollback_timer
28      -rw-             1376   Jul 2 2025 08:35:28 +05:30  packages.conf
30      -rw-             1376   Jul 2 2025 08:35:23 +05:30
c8500x_COMPUTE_ASR1K.image.BLD_LUX_DEV_S2C_20250629_135017-2-g6d472f6332eb.SSA.conf
15990792  -rw-         66959470   Jul 2 2025 08:35:22 +05:30
c8500x_COMPUTE_ASR1K.rpboot.BLD_LUX_DEV_S2C_20250629_135017-2-g6d472f6332eb.SSA.pkg
15990785  drwx             4096   Jul 2 2025 08:35:21 +05:30  .images
29      -rw-        947406255   Jul 2 2025 08:34:33 +05:30  c8000aep.bin
21      -rw-       1232501326   Jul 2 2025 08:33:11 +05:30
c8500x_COMPUTE_ASR1K.image.BLD_LUX_DEV_S2C_20250629_135017-2-g6d472f6332eb.SSA.bin
15990805  -rw-             5284   Jul 1 2025 23:09:02 +05:30
c8500x_COMPUTE_ASR1K.empty.BLD_LUX_DEV_S2C_20250629_135017-2-g6d472f6332eb.SSA.pkg
32      -rw-         945615620   Jul 1 2025 11:22:38 +05:30
c8000aep-universalk9.BLD_POLARIS_DEV_LATEST_20250628_033228_V17_19_0_21.SSA.bin
15589377  drwx             4096  Jun 26 2025 19:40:17 +05:30  core_temp
17      -rw-           583287  Jun 26 2025 15:58:31 +05:30
CRFT__uut__000001__C8570-G2_2025-06-26_10-28-31.tar.gz
73750   -rwx            64520  Jun 20 2025 16:19:02 +05:30  tam_util_tool
4923393  drwx             4096  Jun 16 2025 20:38:50 +05:30  EFI
26656769  drwx             4096  May 21 2025 23:08:19 +05:30  orch_test_logs
73744   -rw-        145309462  May 20 2025 21:22:55 +05:30  IOSD_start.undo
73743   -rw-             6526  May 20 2025 19:56:45 +05:30  undo-engine.json
73742   -rwx         21387416  May 20 2025 14:15:44 +05:30  live-record_x64
73741   -rwx            10645  May 20 2025 14:14:54 +05:30  live-record_polaris
73733   -rw-            98614   May 6 2025 09:49:32 +05:30  collated_log_20250506-041923
73736   -rw-          1207993   May 5 2025 15:18:45 +05:30  mcp_diag.image
24      -rw-            16319  Apr 29 2025 09:58:52 +05:30  test_rumack.txt
31      -rw-             3566  Apr 23 2025 10:44:06 +05:30  Rum-Report-v0010.txt
73731   -rw-             4785   Apr 2 2025 13:53:10 +05:30  backup_config.cfg
73730   -rw-             2689  Mar 17 2025 14:06:29 +05:30  ipconfig.config
28147713  drwx             4096   Mar 8 2025 14:58:59 +05:30  pnp-tech
18      -rw-              257   Mar 8 2025 14:58:16 +05:30  .iox_dir_list
14516225  drwx             4096  Jan 30 2025 15:04:49 +05:30  pcap
```

```
27074561  drwx             4096  Jan 30 2025 15:04:49 +05:30  SHARED-IOX
11        drwx            4096  Jan 30 2025 15:04:32 +05:30  lost+found
1998849   drwx             4096   Dec 4 2024 13:55:50 +05:30  .geo

468237697024 bytes total (434919051264 bytes free)
```

Installing the software using install commands

Cisco 8500 Series Secure Routers are shipped in install mode by default. Users can boot the platform, and upgrade or downgrade to Cisco IOS XE software versions using a set of **install** commands.

Restrictions for Installing the Software Using install Commands

- ISSU is not covered in this feature.
- Install mode requires a reboot of the system.

Information about installing the software using install commands

This table describes the differences between Bundle mode and Install mode:

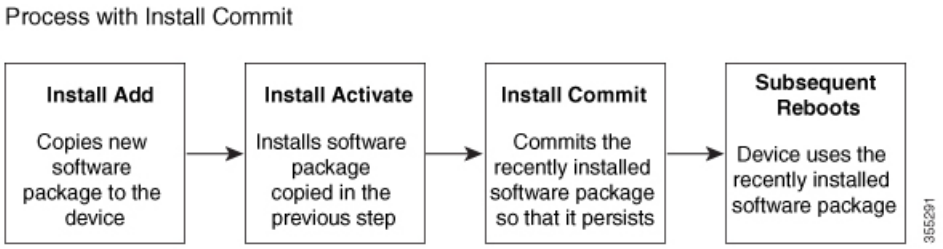
Table 9: Bundle Mode vs Install Mode

Bundle Mode	Install Mode
This mode provides a consolidated boot process, using local (hard disk, flash) or remote (TFTP) .bin image. **Note** Bundle boot from USB and TFTP Boot is not supported.	This mode uses the local (bootflash) packages.conf file for the boot process.
This mode uses a single .bin file.	.bin file is replaced with expanded .pkg files in this mode.
CLI: `#boot system file <filename>`	CLI: `#install add file bootflash: [activate commit]`
To upgrade in this mode, point the boot system to the new image.	To upgrade in this mode, use the **install** commands.
Image Auto-Upgrade: When a new Field-Replaceable Unit (FRU) is inserted in a modular chassis, manual intervention is required to get the new FRU running with the same version as the active FRUs.	Image Auto-Upgrade: When a new FRU is inserted in a modular chassis, the joining FRU is auto-upgraded to the image version in sync with the active FRUs.
Rollback: Rollback to the previous image with multiple Software Maintenance Updates (SMUs) may require multiple reloads.	Rollback: Enables rollback to an earlier version of Cisco IOS XE software, including multiple patches in single reload.

Install Mode Process Flow

The install mode process flow comprises three commands to perform installation and upgrade of software on platforms–**install add**, **install activate**, and **install commit**.

The following flow chart explains the install process with **install** commands:



The **install add** command copies the software package from a local or remote location to the platform. The location can be FTP, HTTP, HTTPs, or TFTP. The command extracts individual components of the .package file into subpackages and packages.conf files. It also validates the file to ensure that the image file is specific to the platform on which it is being installed.

The **install activate** command performs the required validations and provisions the packages previously added using the **install add** command. It also triggers a system reload.

The **install commit** command confirms the packages previously activated using the **install activate** command, and makes the updates persistent over reloads.

Note Installing an update replaces any previously installed software image. At any time, only one image can be installed in a device.

The following set of install commands is available:

Table 10: List of install Commands

Command	Syntax	Purpose
install add	**install add file** *location:filename.bin*	Copies the contents of the image, package, and SMUs to the software repository. File location may be local or remote. This command does the following: • Validates the file–checksum, platform compatibility checks, and so on. • Extracts individual components of the package into subpackages and packages.conf • Copies the image into the local inventory and makes it available for the next steps.
install activate	**install activate**	Activates the package added using the **install add** command. • Use the **show install summary** command to see which image is inactive. This image will get activated. • System reloads on executing this command. Confirm if you want to proceed with the activation. Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts.

Command	Syntax	Purpose
(install activate) auto abort-timer	**install activate auto-abort timer** *<30-1200>*	The **auto-abort timer** starts automatically, with a default value of 120 minutes. If the **install commit** command is not executed within the time provided, the activation process is terminated, and the system returns to the last-committed state. • You can change the time value while executing the **install activate** command. • The **install commit** command stops the timer, and continues the installation process. • The **install activate auto-abort timer stop** command stops the timer without committing the package. • Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts. • This command is valid only in the three-step install variant.
install commit	**install commit**	Commits the package activated using the **install activate** command, and makes it persistent over reloads. • Use the **show install summary** command to see which image is uncommitted. This image will get committed.

Command	Syntax	Purpose
install abort	**install abort**	Terminates the installation and returns the system to the last-committed state. • This command is applicable only when the package is in activated status (uncommitted state). • If you have already committed the image using the **install commit** command, use the **install rollback to** command to return to the preferred version.
install remove	**install remove {file** <*filename*> **| inactive}**	Deletes inactive packages from the platform repository. Use this command to free up space. • **file**: Removes specified files. • **inactive**: Removes all the inactive files.
install rollback to	**install rollback to {base | label | committed | id}**	Rolls back the software set to a saved installation point or to the last-committed installation point. The following are the characteristics of this command: • Requires reload. • Is applicable only when the package is in committed state. • Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts. **Note** If you are performing install rollback to a previous image, the previous image must be installed in install mode. Only SMU rollback is possible in bundle mode.

Command	Syntax	Purpose
install deactivate	**install deactivate file** <*filename*>	Removes a package from the platform repository. This command is supported only for SMUs. • Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts.

The following show commands are also available:

Table 11: List of show Commands

Command	Syntax	Purpose
show install log	**show install log**	Provides the history and details of all install operations that have been performed since the platform was booted.
show install package	**show install package** <*filename*>	Provides details about the .pkg/.bin file that is specified.
show install summary	**show install summary**	Provides an overview of the image versions and their corresponding install states for all the FRUs. • The table that is displayed will state for which FRUs this information is applicable. • If all the FRUs are in sync in terms of the images present and their state, only one table is displayed. • If, however, there is a difference in the image or state information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table.
show install active	**show install active**	Provides information about the active packages for all the FRUs. If there is a difference in the information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table.

Command	Syntax	Purpose
show install inactive	**show install inactive**	Provides information about the inactive packages, if any, for all the FRUs. If there is a difference in the information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table.
show install committed	**show install committed**	Provides information about the committed packages for all the FRUs. If there is a difference in the information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table.
show install uncommitted	**show install uncommitted**	Provides information about uncommitted packages, if any, for all the FRUs. If there is a difference in the information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table.
show install rollback	**show install rollback {point-id | label}**	Displays the package associated with a saved installation point.
show version	**show version [rp-slot] [installed [user-interface] | provisioned | running]**	Displays information about the current package, along with hardware and platform information.

Booting the Platform in Install Mode

You can install, activate, and commit a software package using a single command (one-step install) or multiple separate commands (three-step install).

If the platform is working in bundle mode, the one-step install procedure must be used to initially convert the platform from bundle mode to install mode. Subsequent installs and upgrades on the platform can be done with either one-step or three-step variants.

One-Step Installation or Converting from Bundle Mode to Install Mode

Note

- All the CLI actions (for example, add, activate, and so on) are executed on all the available FRUs.
- The configuration save prompt will appear if an unsaved configuration is detected.
- The reload prompt will appear after the second step in this workflow. Use the **prompt-level none** keyword to automatically ignore the confirmation prompts.
- If the prompt-level is set to None, and there is an unsaved configuration, the install fails. You must save the configuration before reissuing the command.

Use the one-step install procedure described below to convert a platform running in bundle boot mode to install mode. After the command is executed, the platform reboots in install boot mode.

Later, the one-step install procedure can also be used to upgrade the platform.

This procedure uses the **install add file activate commit** command in privileged EXEC mode to install a software package, and to upgrade the platform to a new version.

SUMMARY STEPS

1. **enable**
2. **install add file location:** *filename* [**activate commit**]
3. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**install add file location:** *filename* [**activate commit**] **Example:** `Device#install add file bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_15_4o_117.SSA.bin activate commit`	Copies the software install package from a local or remote location (through FTP, HTTP, HTTPs, or TFTP) to the platform and extracts the individual components of the .package file into subpackages and packages.conf files. It also performs a validation and compatibility check for the platform and image versions, activates the package, and commits the package to make it persistent across reloads. The platform reloads after this command is run.
Step 3	**exit** **Example:** `Device#exit`	Exits privileged EXEC mode and returns to user EXEC mode.

Three-Step Installation

Note

- All the CLI actions (for example, add, activate, and so on) are executed on all the available FRUs.
- The configuration save prompt will appear if an unsaved configuration is detected.
- The reload prompt will appear after the install activate step in this workflow. Use the **prompt-level none** keyword to automatically ignore the confirmation prompts.

The three-step installation procedure can be used only after the platform is in install mode. This option provides more flexibility and control to the customer during installation.

This procedure uses individual **install add**, **install activate**, and **install commit** commands for installing a software package, and to upgrade the platform to a new version.

SUMMARY STEPS

1. **enable**
2. **install add file location:** *filename*
3. **show install summary**
4. **install activate** [**auto-abort-timer** <*time*>]
5. **install abort**
6. **install commit**
7. **install rollback to committed**
8. **install remove** {**file** *filesystem: filename* | **inactive**}
9. **show install summary**
10. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**install add file location:** *filename* **Example:** `Device#install add file bootflash:c8000aep-universalk9.17.15.04a.SPA.bin`	Copies the software install package from a remote location (through FTP, HTTP, HTTPs, or TFTP) to the platform, and extracts the individual components of the .package file into subpackages and packages.conf files.
Step 3	**show install summary** **Example:** `Device#show install summary`	(Optional) Provides an overview of the image versions and their corresponding install state for all the FRUs.

	Command or Action	Purpose
Step 4	**install activate** [**auto-abort-timer** <*time*>] **Example:** `Device# install activate auto-abort-timer 120`	Activates the previously added package and reloads the platform. • When doing a full software install, do not provide a package filename. • In the three-step variant, **auto-abort-timer** starts automatically with the **install activate** command; the default for the timer is 120 minutes. If the **install commit** command is not run before the timer expires, the install process is automatically terminated. The platform reloads and boots up with the last committed version.
Step 5	**install abort** **Example:** `Device#install abort`	(Optional) Terminates the software install activation and returns the platform to the last committed version. • Use this command only when the image is in activated state, and not when the image is in committed state.
Step 6	**install commit** **Example:** `Device#install commit`	Commits the new package installation and makes the changes persistent over reloads.
Step 7	**install rollback to committed** **Example:** `Device#install rollback to committed`	(Optional) Rolls back the platform to the last committed state.
Step 8	**install remove** {**file** *filesystem: filename* | **inactive**} **Example:** `Device#install remove inactive`	(Optional) Deletes software installation files. • **file**: Deletes a specific file • **inactive**: Deletes all the unused and inactive installation files.
Step 9	**show install summary** **Example:** `Device#show install summary`	(Optional) Displays information about the current state of the system. The output of this command varies according to the **install** commands run prior to this command.
Step 10	**exit** **Example:** `Device#exit`	Exits privileged EXEC mode and returns to user EXEC mode.

Upgrading in Install Mode

Use either the one-step installation or the three-step installation to upgrade the platform in install mode.

Downgrading in Install Mode

Use the **install rollback** command to downgrade the platform to a previous version by pointing it to the appropriate image, provided the image you are downgrading to was installed in install mode.

The **install rollback** command reloads the platform and boots it with the previous image.

Note The **install rollback** command succeeds only if you have not removed the previous file using the **install remove inactive** command.

Alternatively, you can downgrade by installing the older image using the **install** commands.

Terminating a Software Installation

You can terminate the activation of a software package in the following ways:

- When the platform reloads after activating a new image, the auto-abort-timer is triggered (in the three-step install variant). If the timer expires before issuing the **install commit** command, the installation process is terminated, and the platform reloads and boots with the last committed version of the software image.

  Alternatively, use the **install auto-abort-timer stop** command to stop this timer, without using the **install commit** command. The new image remains uncommitted in this process.

- Using the **install abort** command returns the platform to the version that was running before installing the new software. Use this command before issuing the **install commit** command.

Configuration Examples for Installing the Software Using install Commands

The following is an example of the one-step installation or converting from bundle mode to install mode:

```
Router #install add file
bootflash:c8000aep-universalk9.BLD_POLARIS_DEV_LATEST_20250628_033228_V17_19_0_21.SSA.bin
activate commit
install_add_activate_commit: START Mon Jul 07 14:22:07 IST 2025
install_add: START Mon Jul 07 14:22:07 IST 2025
install_add: Adding IMG
--- Starting initial file syncing ---
Copying
bootflash:c8000aep-universalk9.BLD_POLARIS_DEV_LATEST_20250628_033228_V17_19_0_21.SSA.bin
from  R0 to  R0
Info: Finished copying to the selected
Finished initial file syncing

--- Starting Add ---
Performing Add on all members

*Jul  7 08:52:07.326: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install
add_activate_commit
bootflash:c8000aep-universalk9.BLD_POLARIS_DEV_LATEST_20250628_033228_V17_19_0_21.SSA.binChecking
 status of Add on [R0]
Add: Passed on [R0]
Image added. Version: 17.19.01.0.224220
```

```
Finished Add

install_activate: START Mon Jul 07 14:22:16 IST 2025
install_activate: Activating IMG
Following packages shall be activated:
/bootflash/c8000aep-firmware_ngwic_t1e1.BLD_POLARIS_DEV_LATEST_20250628_033228_V17_19_0_21.SSA.pkg
/bootflash/c8000aep-firmware_nim_ssd.BLD_POLARIS_DEV_LATEST_20250628_033228_V17_19_0_21.SSA.pkg
/bootflash/c8000aep-mono-universalk9.BLD_POLARIS_DEV_LATEST_20250628_033228_V17_19_0_21.SSA.pkg
/bootflash/c8000aep-rpboot.BLD_POLARIS_DEV_LATEST_20250628_033228_V17_19_0_21.SSA.pkg

This operation may require a reload of the system. Do you want to proceed? [y/n]
*Jul  7 08:52:16.603: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install
activate NONEy

--- Starting Activate ---
Performing Activate on all members
 [1] Activate package(s) on  R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate

--- Starting Commit ---
Performing Commit on all members
 [1] Commit package(s) on  R0
 [1] Finished Commit on  R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit operation

SUCCESS: install_add_activate_commit Mon Jul 07 14:22:41 IST 2025
Encore#
*Jul  7 08:52:41.750: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_mgr: Completed install
 add_activate_commitJul  7 14:22:48.332: %PMAN-5-E




Initializing Hardware ...


System integrity status: 90170200  21030106
Procyon RSM done


System Bootstrap, Version Private [sajjha-blue_pqc 109], DEVELOPMENT SOFTWARE
Copyright (c) 1994-2025 by cisco Systems, Inc.
Compiled Fri Jun 13 14:17:01 2025 by sajjha

Current image running: Boot ROM0
Last reset cause: LocalSoft

Disk ID:#0,MSA281400HY-Micron_7450_MTFDKBA480TFR    - Disk already unlocked
C8570-G2 platform with 33554432 Kbytes of main memory

Enc_5_P2B 1 >
```

The following is an example of the three-step installation:

```
Router #install add file boo
Encore#$rsalk9.BLD_POLARIS_DEV_LATEST_20250625_033136_V17_19_0_20.SSA.bin
install_add: START Mon Jul 07 14:53:11 IST 2025
install_add: Adding IMG
```

```
--- Starting initial file syncing ---
Copying
bootflash:c8000aep-universalk9.BLD_POLARIS_DEV_LATEST_20250625_033136_V17_19_0_20.SSA.bin
from  R0 to  R0
Info: Finished copying to the selected
Finished initial file syncing

--- Starting Add ---
Performing Add on all members

*Jul  7 09:23:11.416: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install add

bootflash:c8000aep-universalk9.BLD_POLARIS_DEV_LATEST_20250625_033136_V17_19_0_20.SSA.binChecking
 status of Add on [R0]
Add: Passed on [R0]
Image added. Version: 17.19.01.0.223976

Finished Add

SUCCESS: install_add
/bootflash/c8000aep-universalk9.BLD_POLARIS_DEV_LATEST_20250625_033136_V17_19_0_20.SSA.bin
 Mon Jul 07 14:53:19 IST 2025

Encore#
*Jul  7 09:23:19.987: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_mgr: Completed install
 add
bootflash:/c8000aep-universalk9.BLD_POLARIS_DEV_LATEST_20250625_033136_V17_19_0_20.SSA.bin
Encore#
Encore#install activate
install_activate: START Mon Jul 07 14:54:14 IST 2025
install_activate: Activating IMG
Following packages shall be activated:
/bootflash/c8000aep-firmware_ngwic_t1e1.BLD_POLARIS_DEV_LATEST_20250625_033136_V17_19_0_20.SSA.pkg
/bootflash/c8000aep-firmware_nim_ssd.BLD_POLARIS_DEV_LATEST_20250625_033136_V17_19_0_20.SSA.pkg
/bootflash/c8000aep-mono-universalk9.BLD_POLARIS_DEV_LATEST_20250625_033136_V17_19_0_20.SSA.pkg
/bootflash/c8000aep-rpboot.BLD_POLARIS_DEV_LATEST_20250625_033136_V17_19_0_20.SSA.pkg

This operation may require a reload of the system. Do you want to proceed? [y/n]
                                                                                *Jul  7
09:24:14.874: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install activate
NONEy

--- Starting Activate ---
Performing Activate on all members
 [1] Activate package(s) on  R0

*Jul  7 09:25:18.674: %INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: R0/0: rollback_timer:
Install auto abort timer will expire in 7200 seconds [1] Finished Activate on  R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate

SUCCESS: install_activate Mon Jul 07 14:55:25 IST 2025

Encore#
*Jul  7 09:25:25.208: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_mgr: Completed install
 activateJul  7 14:55:31.791: %PMAN-5-EXITAC
Encore#install commit
install_commit: START Mon Jul 07 14:59:12 IST 2025
--- Starting Commit ---
Performing Commit on all members
 [1] Commit packages(s) on  R0

*Jul  7 09:29:12.013: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install
```

```
commit [1] Finished Commit packages(s) on  R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit operation

SUCCESS: install_commit Mon Jul 07 14:59:13 IST 2025

Encore#
*Jul  7 09:29:13.749: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_mgr: Completed install
 commit
```

The following is an example of downgrading in install mode:

```
Router# install activate file bootflash:c8000be-universalk9.17.06.01a.SPA.bin activate
commit

install_add_activate_commit: START Fri Dec 10 18:07:17 GMT 2021

*Dec 10 18:07:18.405 GMT: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started
install one-shot bootflash:c8000be-universalk9.17.06.01a.SPA.bininstall_add_activate_commit:
 Adding PACKAGE
install_add_activate_commit: Checking whether new add is allowed ....

--- Starting Add ---
Performing Add on Active/Standby
  [1] Add package(s) on R0
  [1] Finished Add on R0
Checking status of Add on [R0]
Add: Passed on [R0]
Finished Add

Image added. Version: 17.06.01a.0.298
install_add_activate_commit: Activating PACKAGE
Following packages shall be activated:
/bootflash/c8000be-rpboot.17.06.01a.SPA.pkg
/bootflash/c8000be-mono-universalk9.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_sm_nim_adpt.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_sm_dsp_sp2700.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_sm_async.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_sm_1t3e3.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_sm_10g.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_prince.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_xdsl.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_ssd.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_shdsl.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_ge.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_cwan.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_bri_st_fw.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_async.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_ngwic_t1e1.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_dsp_tilegx.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_dsp_sp2700.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_dsp_analogbri.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_dreamliner.17.06.01a.SPA.pkg

This operation may require a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on Active/Standby
  [1] Activate package(s) on R0
  [1] Finished Activate on R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate
```

```
--- Starting Commit ---
Performing Commit on Active/Standby
  [1] Commit package(s) on R0
Building configuration...

  [1] Finished Commit on R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit

[OK]
*Dec 10 18:14:57.782 GMT: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private
config fileSend model notification for install_add_activate_commit before reload
/usr/binos/conf/install_util.sh: line 164: /bootflash/.prst_sync/reload_info: No such file
 or directory
/usr/binos/conf/install_util.sh: line 168: /bootflash/.prst_sync/reload_info: No such file
 or directory
cat: /bootflash/.prst_sync/reload_info: No such file or directory
Install will reload the system now!
SUCCESS: install_add_activate_commit  Fri Dec 10 18:15:23 GMT 2021

ROUTER#
*Dec 10 18:15:23.955 GMT: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
 install one-shot PACKAGE bootflash:c8000be-universalk9.17.06.01a.SPA.binDec 10 18:15:27.708:
 %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting: reload action requested

Initializing Hardware ...

Checking for PCIe device presence...done
System integrity status: 0x610
Rom image verified correctly

System Bootstrap, Version 17.3(5r), RELEASE SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.

Current image running: Boot ROM0

Last reset cause: LocalSoft
ROUTER platform with 8388608 Kbytes of main memory

□

Press RETURN to get started!

□

ROUTER#
ROUTER# show version
Cisco IOS XE Software, Version 17.06.01a
Cisco IOS Software [Bengaluru], c8000be Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version
 17.6.1a, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2021 by Cisco Systems, Inc.
Compiled Sat 21-Aug-21 03:27 by mcpre

Cisco IOS-XE software, Copyright (c) 2005-2021 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
```

```
software.

ROM: 17.3(5r)

ROUTER uptime is 0 minutes
Uptime for this control processor is 2 minutes
System returned to ROM by LocalSoft
System image file is "bootflash:packages.conf"
Last reload reason: LocalSoft

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

Technology Package License Information:

------------------------------------------------------------------
Technology     Type         Technology-package Technology-package
                            Current           Next Reboot
------------------------------------------------------------------
Smart License  Perpetual    None              None
Smart License  Subscription None              None

The current crypto throughput level is 250000 kbps

Smart Licensing Status: Registration Not Applicable/Not Applicable

cisco ROUTER (1RU) processor with 3747220K/6147K bytes of memory.
Processor board ID FDO2521M27S
Router operating mode: Autonomous
5 Gigabit Ethernet interfaces
2 2.5 Gigabit Ethernet interfaces
2 Cellular interfaces
32768K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
7573503K bytes of flash memory at bootflash:.
1875361792K bytes of NVMe SSD at harddisk:.
16789568K bytes of USB flash at usb0:.

Configuration register is 0x2102
```

The following is an example of terminating a software installation:

```
Router# install abort
install_abort: START Fri Oct 29 02:42:51 UTC 2021

This install abort would require a reload. Do you want to proceed? [y/n]
                                                                    *Oct 29 02:42:52.789:
 %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install aborty
--- Starting Abort ---
Performing Abort on Active/Standby
```

```
  [1] Abort package(s) on R0
  [1] Finished Abort on R0
Checking status of Abort on [R0]
Abort: Passed on [R0]
Finished Abort

Send model notification for install_abort before reload
Install will reload the system now!
SUCCESS: install_abort  Fri Oct 29 02:44:47 UTC 2021

Router#
*Oct 29 02:44:47.866: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
install abort PACKAGEOct 29 02:44:51.577: %PMAN-5-EXITACTION: R0/0: pvp: Process manager
is exiting: reload action requested

Initializing Hardware ...

Checking for PCIe device presence...done
System integrity status: 0x610

System Bootstrap, Version 17.3(4.1r), RELEASE SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.

Current image running   : Boot ROM1
Last reset cause        : LocalSoft
C8300-2N2S-6T platform with 8388608 Kbytes of main memory

□

Press RETURN to get started!

□
```

The following are sample outputs for show commands:

show install log

```
Device# show install log
[0|install_op_boot]: START Thu Oct 28 22:09:29 Universal 2021
[0|install_op_boot(INFO, )]: Mount IMG INI state base image
[0|install_op_boot]: END SUCCESS  Thu Oct 28 22:09:30 Universal 2021
```

show install summary

```
Device# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------
IMG   C    17.07.01.0.1515

--------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------
```

show install package *filesystem: filename*

```
Device# show install package
bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.bin
Package: c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.bin
```

```
  Size: 831447859
  Timestamp: 2021-10-23 17:08:14 UTC
  Canonical path:
/bootflash/c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.bin


  Raw disk-file SHA1sum:
    5c4e7617a6c71ffbcc73dcd034ab58bf76605e3f
  Header size:     1192 bytes
  Package type:    30000
  Package flags:   0
  Header version:  3

  Internal package information:
    Name: rp_super
    BuildTime: 2021-10-21_13.00
    ReleaseDate: 2021-10-21_03.11
    BootArchitecture: i686
    RouteProcessor: radium
    Platform: C8000BE
    User: mcpre
    PackageName: universalk9
    Build: BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117
    CardTypes:

  Package is bootable from media and tftp.
  Package contents:

  Package:
c8000be-firmware_nim_ge.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
    Size: 2966620
    Timestamp: 2021-10-21 20:10:44 UTC

    Raw disk-file SHA1sum:
      501d59d5f152ca00084a0da8217bf6f6b95dddb1
    Header size:     1116 bytes
    Package type:    40000
    Package flags:   0
    Header version:  3

    Internal package information:
      Name: firmware_nim_ge
      BuildTime: 2021-10-21_13.00
      ReleaseDate: 2021-10-21_03.11
      BootArchitecture: none
      RouteProcessor: radium
      Platform: C8000BE
      User: mcpre
      PackageName: firmware_nim_ge
      Build: BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117
      CardTypes:

    Package is not bootable.
  Package:
c8000be-firmware_prince.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
    Size: 10204252
    Timestamp: 2021-10-21 20:10:43 UTC

    Raw disk-file SHA1sum:
      a57bed4ddecfd08af3b456f69d11aaeb962865ea
    Header size:     1116 bytes
    Package type:    40000
    Package flags:   0
    Header version:  3
```

```
    Internal package information:
      Name: firmware_prince
      BuildTime: 2021-10-21_13.00
      ReleaseDate: 2021-10-21_03.11
      BootArchitecture: none
      RouteProcessor: radium
      Platform: C8000BE
      User: mcpre
      PackageName: firmware_prince
      Build: BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117
      CardTypes:

    Package is not bootable.
```

show install active

```
Device# show install active
[ R0 ] Active Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------
IMG   C    17.07.01.0.1515

--------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------
```

show install inactive

```
Device# show install inactive
[ R0 ] Inactive Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------
No Inactive Packages
```

show install committed

```
Device# show install committed
[ R0 ] Committed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------
IMG   C    17.07.01.0.1515

--------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------
```

show install uncommitted

```
Device# show install uncommitted
[ R0 ] Uncommitted Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------
No Uncommitted Packages
```

Troubleshooting Software Installation Using install Commands

Problem Troubleshooting the software installation

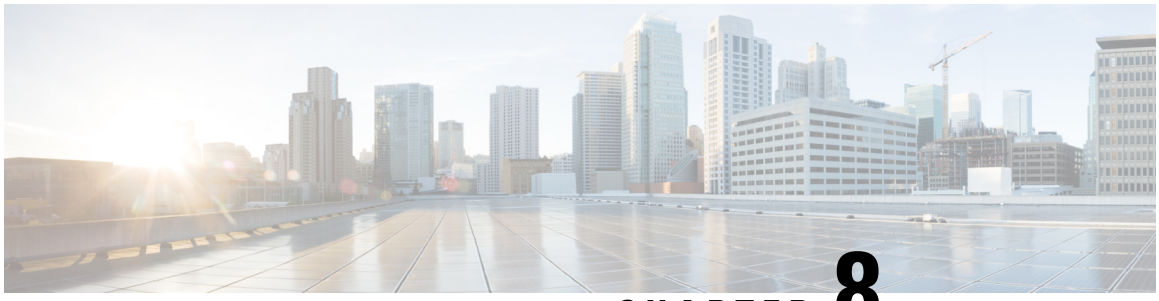
Solution Use the following show commands to view installation summary, logs, and software versions.

- **show install summary**
- **show install log**
- **show version**
- **show version running**

Problem Other installation issues

Solution Use the following commands to resolve installation issue:

- **dir** <*install directory*>
- **more location:***packages.conf*
- **show tech-support install**: this command automatically runs the **show** commands that display information specific to installation.
- **request platform software trace archive target bootflash** <*location*>: this command archives all the trace logs relevant to all the processes running on the system since the last reload, and saves this information in the specified location.

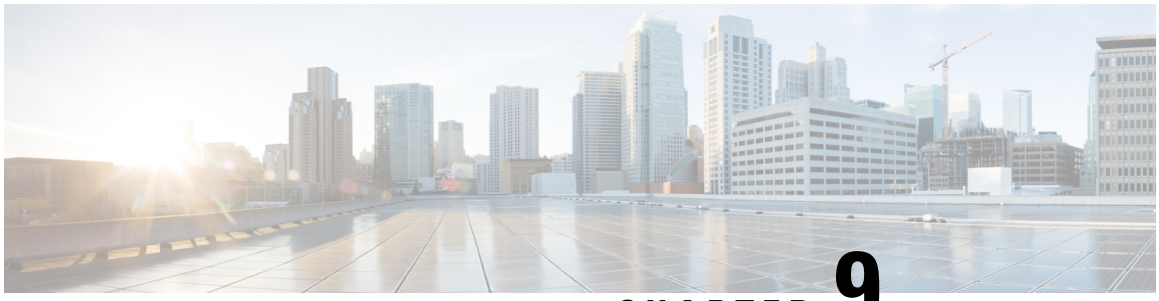


CHAPTER 8

Software upgrade processes

If you want to upgrade the ROMMON and IOS at the same time, perform the steps given below:

- Copy the XE image to the router and configure the boot system to point to the new image.
- Copy the ROMMON package to the router and perform the ROMMON upgrade.
- Reload the router and verify that it boots to the IOS prompt on the new XE image.
- Verify that the new ROMMON image was successfully installed using a show platform.



CHAPTER 9

Factory Reset

This chapter describes Factory Reset feature and how it can be used to protect or restore a router to an earlier, fully functional state.

Feature information for factory reset

Table 12: Feature Information for Factory Reset

Feature Name	Releases	Feature Information
Factory Reset	Cisco IOS XE 17.15.4a	From Cisco IOS XE 17.15.4a, Cisco 8500 Series Secure Routers support the following commands: **factory-reset all** **factory-reset all secure** **factory-reset keep-licensing-info** **factory-reset sed**

Information about factory reset

Factory Reset is a process of clearing the current running and start-up configuration information on a device, and resetting the device to an earlier, fully-functional state.

The factory reset process uses the **factory-reset all** command to take backup of existing configuration, and then reset the router to an earlier, fully functional state. The duration of the factory reset process is dependent on the storage size of the router. It can vary between 30 minutes on a consolidated platform, and up to 3 hours on a high availability setup.

You can use the **factory-reset all secure** command to reset the router and erase the persistent storages of the router as per the NIST PURGE/CLEAR standards. Also, the image with which the router was booted wont be retained and the router will fall back to the rommon prompt. This process takes between 5 minutes to 2 hours.

Table 13: Data Erased or Retained during Factory Reset

Command Name	Data Erased	Data Retained
factory-reset all secure	Non-volatile random-access memory (NVRAM) data	Data from remote field-replaceable units (FRUs).
	OBFL (Onboard Failure Logging) logs	Contents of USB
	Licenses	Credentials (Secure Unique Device Identifier [SUDI] certificates, public key infrastructure (PKI) keys, and FIPS-related keys)
	User data, startup, and running configuration	
	ROMMON variables	
	All writeable file systems and personal data. **Note** After the completion of factory-reset, the router has to be booted from a image stored in a remote storage or a USB as everything is erased from the non-volatile storage.	
	Value of configuration register **Important** The value of the configuration register can be erased using the **factory-reset all secure** command on Cisco 8500 Series Secure Routers.	

Command Name	Data Erased	Data Retained
factory-reset all	Non-volatile random-access memory (NVRAM) data	Data from remote field-replaceable units (FRUs).
	OBFL (Onboard Failure Logging) logs	Contents of USB
	Licenses	Credentials (Secure Unique Device Identifier [SUDI] certificates, public key infrastructure (PKI) keys, and FIPS-related keys)
	User data, startup, and running configuration	Value of configuration register
	ROMMON variables	
	All writeable file systems and personal data.	
factory-reset keep-licensing-info	License Boot level configuration	Real User Monitoring (RUM) Reports (open/unacknowledged license usage report)
	Throughput level configuration	Usage reporting details (last ACK received, next ACK scheduled, last/next report push)
	Smart license transport type	Unique Device Identification (UDI) trust codes
	Smart license URL data	Customer policy received from CSSM
		SLAC, SLR authorization codes return codes
		Factory installed purchase information

Command Name	Data Erased	Data Retained
factory-reset sed	Non-volatile random-access memory (NVRAM) data	Data from remote field-replaceable units (FRUs).
	OBFL (Onboard Failure Logging) logs	Contents of USB
	Licenses	Credentials (Secure Unique Device Identifier [SUDI] certificates, public key infrastructure (PKI) keys, and FIPS-related keys)
	User data, startup, and running configuration	ROMMON variables
	All the data on the sed enabled disk.	Value of configuration register

The table below outlines the supported platforms for the **factory-reset sed** command

Platform	Bootflash	Harddisk
C8550-G2	SED enabled	NA
C8570-G2	SED enabled	NA

The **factory-reset all secure** command always boots into ROMMON. For other commands it depends on the config-register value. If you have the zero-touch provisioning (ZTP) capability setup, after the router completes the factory reset procedure, the router reboots with ZTP configuration.

Software and hardware support for factory reset

- Factory Reset process is supported on standalone routers as well as on routers configured for high availability.

Prerequisites for performing factory reset

- Ensure that all the software images, configurations and personal data is backed up before performing factory reset.
- Ensure that there is uninterrupted power supply when factory reset is in progress.
- The **factory-reset all** command takes a backup of the boot image if the system is booted from an image stored locally (bootflash or hard disk).
- The **factory-reset all secure** command erases all files, including the boot image, even if the image is stored locally. You would need to boot the router using an image stored in TFTP or USB in this case.
- Ensure that ISSU/ISSD (In- Service Software Upgrade or Downgrade) is not in progress before performing factory reset.

Restrictions for performing a factory reset

- Any software patches that are installed on the router are not restored after the factory reset operation.
- If the factory reset command is issued through a Virtual Teletype (VTY) session, the session is not restored after the completion of the factory reset process.
- The **factory-reset all secure** command is not supported through a Virtual Teletype (VTY) session.

When to perform factory reset

- Return Material Authorization (RMA): If a router is returned back to Cisco for RMA, it is important that all sensitive information is removed.
- Router is compromised: If the router data is compromised due to a malicious attack, the router must be reset to factory configuration and then reconfigured once again for further use.
- Repurposing: The router needs to be moved to a new topology or market from the existing site to a different site.

How to perform a factory reset

Before you begin

Refer Table 2 to determine which information is going to be deleted and retained. Based on the information you require, execute the appropriate command mentioned below.

Procedure

Step 1 Log in to a Cisco 8500 Series Secure Router

Important
If the current boot image is a remote image or is stored in a USB, ensure that you take a backup of the image before starting the factory reset process.

Step 2 This step is divided into four parts (a,b,c and d). If you want all the data to be erased as per the NIST standards without retaining the boot image, follow step 2.a. If you want to erase the data with the configuration register value and local boot image retained, follow step 2.b. If you want to just erase the sed drive, follow step 2.c. If you need to retain the licensing information while performing the factory-reset command, follow step 2.d

a) Execute **factory-reset all secure** command to erase as per the NIST standards.

The system displays the following message when you use the **factory-reset all secure** command:

```
Router# factory-reset all secure
The factory reset operation is irreversible for securely reset all. Are you sure? [confirm]
 This is a NIST CLEAR/PURGE.
 The following will be deleted as a part of factory reset:
```

```
 1: All writable file systems and personal data
 2: OBFL logs
 3: Licenses
 4: Userdata and Startup config
 5: Rommon variables
 6: User Credentials
 The system will reload to perform factory reset.
 This operation can take anywhere between 30 minutes to 3 hours
 DO NOT UNPLUG THE POWER OR INTERRUPT THE OPERATION
 Are you sure you want to continue? [confirm]
 The image saved on the router would be lost. The router will fall to the rommon prompt
 Are you sure you want to continue? [confirm]
Mar

Enabling factory reset for this reload cycle


Enabling factory reset for this reloa
*Mar 24 08:19:02.634: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): % FACTORYRESET : Starting ACT2/AIKIDO
CLEANUP

*Mar 24 08:19:17.289: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): % FACTORYRESET : ACT2/AIKIDO Cleanup
done

*Mar 24 08:19:17.413: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): % FACTORYRESET : Erasing Rommon Variables
 and Config Register

*Mar 24 08:19:18.400: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): % FACTORYRESET : Successfully erased
the rommon variables and config register

*Mar 24 08:19:18.568: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): % FACTORYRESET : Starting Datawipe

*Mar 24 08:19:18.685: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): % FACTORYRESET : Erasing the storage
devices as per NIST-SP-800-88-r standard:
Executing Data Sanitization...
bootflash
NVMe Data Sanitization started ...
!!! Please, wait - NVMe sanitizing /dev/nvme0n1 !!!
NVME Sanitize Status: Successful
NVMe Data Sanitization completed ...
Data Sanitization Success! Exiting...

*Mar 24 08:19:48.948: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): Purge non-volatile storage done.
========================
#CISCO C8500-G2 DATA SANITIZATION REPORT#
START : 24-03-2025, 08:19:21
  END : 24-03-2025, 08:19:44
-NVMe-
PNM : MSA281400FR
PRV : E2MU200
SN : /dev/ng0n1
Status : SUCCESS
NIST : PURGE
========================

*Mar 24 08:19:49.357: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): % FACTORYRESET : Datawipe Completed

*Mar 24 08:20:02.980: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): % FACTORYRESET : Report save done.

*Mar 24 08:20:03.097: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): Factory reset successfull. Continuing
with reboot...
```

b) Execute **factory-reset all** command to erase the data.

The system displays the following message when you use the **factory-reset all** command:

```
Router# factory-reset all
The factory reset operation is irreversible for all operations. Are you sure? [confirm]
 The following will be deleted as a part of factory reset:
 1: All writable file systems and personal data
 2: OBFL logs
 3: Licenses
 4: Userdata and Startup config
 5: Rommon variables
 6: User Credentials
 The system will reload to perform factory reset.
 This operation can take anywhere between 30 minutes to 3 hours
 DO NOT UNPLUG THE POWER OR INTERRUPT THE OPERATION
 Are you sure you want to continue? [confirm]
 Factory reset will take a backup of the boot image if the system is currently booted from an
image stored locally. If the current boot image is a remote image or stored on a usb/nim-ssd,
please take a backup of the image before executing this command.
 Are you sure you want to continue? [confirm]
Mar

Enabling factory reset for this reload cycle


Enabling factory reset for this reload cycle

*Mar 24 10:36:43.858: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): Taking a backup of
bootflash:packages.conf. It will be restored to bootflash: after factory reset is complete

*Mar 24 10:36:45.571: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): % FACTORYRESET : Starting ACT2/AIKIDO
CLEANUP

*Mar 24 10:37:00.098: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): % FACTORYRESET : ACT2/AIKIDO Cleanup
done

*Mar 24 10:37:00.596: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): Erasing file system

*Mar 24 10:37:01.458: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): Erasing file system

*Mar 24 10:37:01.796: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): Erasing file system
 . . . . . .
 . . . . . .
 . . . . . .
 . . . . . .
. . . . . .
. . . . . .
 . . . . . .
 . . . . . .
. . . . . .
. . . . . .
 . . . . . .
 . . . . . .
. . . . . .

*Mar 24 11:32:04.869: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): Erasing file system
 .

*Mar 24 11:32:07.004: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): Erasing file system
 .

*Mar 24 11:32:09.133: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): Erasing file system
 .

*Mar 24 11:32:11.262: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): Erasing file system
```

```
 .

*Mar 24 11:32:15.568: %IOSXEBOOT-4-FACTORY_RESET: (rp/0): Factory reset successfull. Continuing
with reboot...


Initializing Hardware ...
```

c) Execute **factory-reset sed** command to erase the sed drive.

The system displays the following message when you use the **factory-reset sed** command:

```
Router# factory-reset sed
% Warning: factory reset on SED drive reloads router
Do you want to continue? (yes/[no]): yes
SUCCESS

Router#
Router#


***
*** --- SHUTDOWN NOW ---
***

*Jan 14 00:48:41.482: %SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload due to
factory reset on SED.Jan 14 00:48:48.499: %PMAN-5-EXITACTION: R


Initializing Hardware ...
```

d) Execute **factory-reset keep-licensing-info** command to retain the licensing data.

The system displays the following message when you use the **factory-reset keep-licensing-info** command:

```
Router# factory-reset keep-licensing-info

The factory reset operation is irreversible for Keeping license usage. Are you sure? [confirm]
This operation may take 20 minutes or more. Please do not power cycle.

Dec 1 20:58:38.205: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting: process exit with
reload chassis code
/bootflash failed to mount
Dec 01 20:59:44.264: Factory reset operation completed.
Initializing Hardware ...

Current image running: Boot ROM1

Last reset cause: LocalSoft

ISR4331/K9 platform with 4194304 Kbytes of main memory
rommon 1
```

Step 3 Enter **confirm** to proceed with the factory reset.

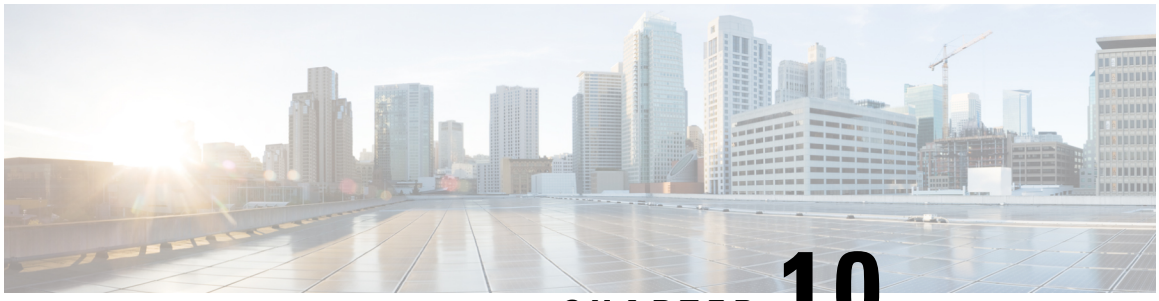
Note
The duration of the factory reset process depends on the storage size of the router. It can extend between 5 minutes and up to 3 hours on a high availability setup. If you want to quit the factory reset process, press the **Escape** key.

What happens after a factory reset

After the factory reset is successfully completed, the router boots up. However, before the factory reset process started, if the configuration register was set to manually boot from ROMMON, the router stops at ROMMON.

After you configure Smart Licensing, execute the **#show license status** command, to check whether Smart Licensing is enabled for your instance.

Note If you had Specific License Reservation enabled before you performed the factory reset, use the same license and enter the same license key that you received from the smart agent.



CHAPTER 10

Support for Security-Enhanced Linux

This chapter describes the SELinux feature, and includes the following sections:

Overview

Security-Enhanced Linux (SELinux) is a solution composed of Linux kernel security module and system utilities to incorporate a strong, flexible Mandatory Access Control (MAC) architecture into Cisco IOS-XE platforms.

SELinux provides an enhanced mechanism to enforce the separation of information, based on confidentiality and integrity requirements, which addresses threats of tampering and bypassing of application security mechanisms and enables the confinement of damage that malicious or flawed applications can cause.

Prerequisites for SELinux

There are no specific prerequisites for this feature.

Restrictions for SELinux

There are no specific restrictions for this feature.

Information About SELinux

SELinux enforces mandatory access control policies that confine user programs and system services to the minimum privilege required to perform their assigned functionality. This reduces or eliminates the ability of

these programs and daemons to cause harm when compromised (for example, through buffer overflows or misconfigurations). This is a practical implementation of principle of least privilege by enforcing MAC on Cisco IOS-XE platforms. This confinement mechanism works independently of the traditional Linux access control mechanisms. SELinux provides the capability to define policies to control the access from an application process to any resource object, thereby allowing for the clear definition and confinement of process behavior.

SELinux can operate either in **Permissive mode** or **Enforcing mode** when enabled on a system.

- In Permissive mode, SELinux does not enforce the policy, and only generates system logs for any denials caused by violation of the resource access policy. The operation is not denied, but only logged for resource access policy violation.
- In Enforcing mode, the SELinux policy is enabled and enforced. It denies resource access based on the access policy rules, and generates system logs.

SELinux is enabled in Enforcing mode by default on supported Cisco IOS XE platforms. In the Enforcing mode, any system resource access that does not have the necessary allow policy is treated as a violation, and the operation is denied. The violating operation fails when a denial occurs, and system logs are generated. In Enforcing mode, the solution works in access-violation prevention mode.

Configuring SELinux

The are no additional requirements or configuration steps needed to enable or use the SELinux feature in Enforcing mode.

The following commands are introduced as part of the SELinux feature:

```
platform security selinux {enforcing | permissive}
```

```
show platform software selinux
```

Configuring SELinux (EXEC Mode)

Use the **set platform software selinux** command to configure SELinux in EXEC mode.

The following example shows SELinux configuration in EXEC mode:

```
Device# set platform software selinux ?

default  Set SELinux mode to default
enforcing  Set SELinux mode to enforcing
permissive  Set SELinux mode to permissive
```

Configuring SELinux (CONFIG Mode)

Use the **platform security selinux** command to configure SELinux in configuration mode.

The following example shows SELinux configuration in CONFIG mode:

```
Device(config)# platform security selinux

enforcing  Set SELinux policy to Enforcing mode
permissive  Set SELinux policy to Permissive mode

Device(config)# platform security selinux permissive
```

```
Device(config)#
*Oct 20 21:52:45.155: %IOSXE-1-PLATFORM: R0/0:
SELINUX_MODE_PROG: Platform Selinux confinement mode downgraded to permissive!

Device(config)#
```

Examples for SELinux

The following example shows the output for changing the mode from Enforcing to Permissive:

```
"*Oct 20 21:44:03.609: %IOSXE-1-PLATFORM: R0/0:
SELINUX_MODE_PROG: Platform Selinux confinement mode downgraded to permissive!"
```

The following example shows the output for changing the mode from Permissive to Enforcing:

```
"*Oct 20 21:44:34.160: %IOSXE-1-PLATFORM: R0/0:
SELINUX_MODE_PROG: Platform Selinux confinement mode upgraded to enforcing!"
```

Note If the SELinux mode is changed, this change is considered a system security event, and a system log message is generated.

SysLog Message Reference

Facility-Severity-Mnemonic	**%SELINUX-1-VIOLATION**
Severity-Meaning	Alert Level Log
Message	N/A
Message Explanation	Resource access was made by the process for which a resource access policy does not exist. The operation was flagged, and resource access was denied. A system log was generated with information that process resource access has been denied.
Component	SELINUX
Recommended Action	Contact Cisco TAC with the following relevant information as attachments: • The exact message as it appears on the console or in the system • Output of the **show tech-support** command (text file) • Archive of Btrace files from the box using the following command: **request platform software trace archive target <URL>** • Output of the **show platform software selinux** command

The following examples demonstrate sample syslog messages:

Example 1:

```
*Nov 14 00:09:04.943: %SELINUX-1-VIOLATION: R0/0: audispd: type=AVC
msg=audit(1699927057.934:129): avc: denied { getattr } for pid=5899 comm="ls"
path="/root/test" dev="rootfs" ino=25839
scontext=system_u:system_r:polaris_iosd_t:s0
tcontext=system_u:object_r:admin_home_t:s0 tclass=file permissive=0
```

Example 2:

```
*Nov 14 00:09:04.947: %SELINUX-1-VIOLATION: R0/0: audispd: t type=AVC
msg=audit(1699927198.486:130): avc: denied { write } for pid=6012 comm="echo"
path="/root/test" dev="rootfs" ino=25839
scontext=system_u:system_r:polaris_iosd_t:s0
tcontext=system_u:object_r:admin_home_t:s0 tclass=file permissive= 0
```

Verifying SELinux Enablement

Use the **show platform software selinux** command to view the SELinux configuration mode:

```
Device# show platform software selinux
========================================
IOS-XE SELINUX STATUS
========================================
SElinux Status :    Enabled
Current Mode :      Enforcing
Config file Mode :  Enforcing
```

Troubleshooting SELinux

If there is an instance of an SELinux violation on your device or network, please reach out to Cisco TAC with the following details:

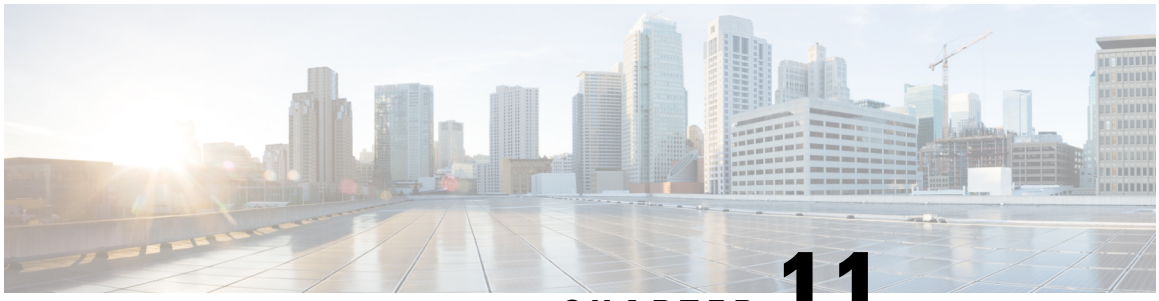
- The message exactly as it appears on the console or in the system log. For example:

  ```
  device#request platform software trace archive target
      flash:selinux_btrace_logs
  ```

- Output of the **show tech-support** command (text file)
- Archive of Btrace files from the box using the following command:

  request platform software trace archive target <URL>

- Output of the **show platform software selinux** command



CHAPTER 11

High Availability Overview

Cisco High Availability (HA) enables network-wide protection by providing fast recovery from faults that may occur in any part of the network. With Cisco High Availability, network hardware and software work together and enable rapid recovery from disruptions to ensure fault transparency to users and network applications.

The unique hardware and software architecture of the Cisco 8500 Series Secure Router is designed to maximize router uptime during any network event, and thereby provide maximum uptime and resilience within any network scenario.

This guide covers the aspects of High Availability that are unique to the Cisco 8500 Series Secure Router. It is not intended as a comprehensive guide to High Availability, nor is it intended to provide information on High Availability features that are available on other Cisco routers that are configured and implemented identically on the Cisco 8500 Series Secure Routers. The Cisco IOS feature documents and guides should be used in conjunction with this chapter to gather information about High Availability-related features that are available on multiple Cisco platforms and work identically on the Cisco 8500 Series Secure Router.

Finding feature information in this module

Your software release might not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

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

Contents

This section discusses various aspects of High Availability on the Cisco 8500 Series Secure Router and contains the following sections:

Software redundancy on the Cisco 8500 Series Secure Router

This section covers the following topics:

Software redundancy overview

On the Cisco 8500 Series Secure Routers, IOS runs as one of many processes within the operating system. This is different than on traditional Cisco IOS, where all processes are run within Cisco IOS. See the "IOS as a Process" section on page 2-7 for more information regarding IOS as a process on the Cisco 8500 Series Secure Routers.

This architecture allows for software redundancy opportunities that are not available on other platforms that run Cisco IOS software. Specifically, a standby IOS process can be available on the same Route Processor as the active IOS process. This standby IOS process can be switched to in the event of an IOS failure.

Configuring two Cisco IOS processes

On the Cisco 8500 Series Secure Routers, Cisco IOS runs as one of the many processes. This architecture supports software redundancy opportunities. Specifically, a standby Cisco IOS process is available on the same Route Processor as the active Cisco IOS process. In the event of a Cisco IOS failure, the system switches to the standby Cisco IOS process.

SUMMARY STEPS

1. enable
2. **configure terminal**
3. redundancy
4. mode SSO
5. **exit**
6. reload

DETAILED STEPS

Procedure

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

	Command or Action	Purpose
Step 3	redundancy **Example:** `Router(config)# redundancy`	Enters redundancy configuration mode.
Step 4	mode SSO **Example:** `Router(config)# mode SSO`	Configures SSO. When this command is entered, the redundant supervisor engine is reloaded and begins to work in SSO mode.

Step 5	**exit**
	Example:
	`Router(config)# exit`
	Example:
	`Router #`
		Exits configuration mode and returns to global configuration mode.

Step 6	reload **Example:** `Router # reload`	Reloads IOS.

Example

```
Router# configure terminal
Router(config)# redundancy
Router(config)# mode SSO
Router(config)# exit
Router# reload
```

Stateful switchover

On the Cisco 8500 Series Secure Routers, Stateful Switchover (SSO) can be used to enable a second IOS process.

Stateful Switchover is particularly useful in conjunction with Nonstop Forwarding. SSO allows the dual IOS processes to maintain state at all times, and Nonstop Forwarding lets a switchover happen seamlessly when a switchover occurs

For additional information on NSF/SSO, see the Cisco Nonstop Forwarding document.

SSO-Aware Protocol and applications

SSO-supported line protocols and applications must be SSO-aware. A feature or protocol is SSO-aware if it maintains, either partially or completely, undisturbed operation through an RP switchover. State information

for SSO-aware protocols and applications is synchronized from active to standby to achieve stateful switchover for those protocols and applications.

The dynamically created state of SSO-unaware protocols and applications is lost on switchover and must be reinitialized and restarted on switchover.

To see which protocols are SSO-aware on your router, use the following commands **show redundancy client** or **show redundancy history**.

IPsec failover

IPSec failover is a feature that increases the total uptime (or availability) of a customer's IPSec network. Traditionally, this is accomplished by employing a redundant (standby) router in addition to the original (active) router. If the active router becomes unavailable for any reason, the standby router takes over the processing of IKE and IPSec. IPSec failover falls into two categories: stateless failover and stateful failover.

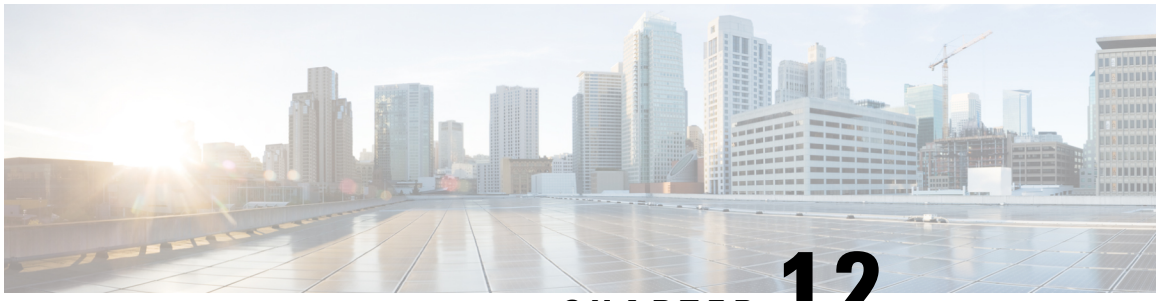
The IPsec on the Cisco 8500 Series Secure Routers supports only stateless failover. Stateless failover uses protocols such as the Hot Standby Router Protocol (HSRP) to provide primary to secondary cutover and also allows the active and standby VPN gateways to share a common virtual IP address.

Bidirectional forwarding detection

Bidirectional Forwarding Detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning is easier, and reconvergence time is consistent and predictable.

On the Cisco 8500 Series Secure Routers, BFD for IPv4 Static Routes and BFD for BGP are fully supported.

For more information on BFD, see the Bidirectional Forwarding Detection document.



CHAPTER 12

Using the management ethernet interface

The Cisco 8500 Series Secure Routers have one Gigabit Ethernet Management Ethernet interface.

Finding feature information in this module

Your software release might not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

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

Contents

This guide covers the following topics:

Gigabit ethernet management interface overview

The purpose of this interface is to allow users to perform management tasks on the router; it is basically an interface that should not and often cannot forward network traffic but can otherwise access the router, often via Telnet and SSH, and perform most management tasks on the router. The interface is most useful before a router has begun routing, or in troubleshooting scenarios when the SPA interfaces are inactive.

The following aspects of the Management Ethernet interface should be noted:

- IPv4, IPv6, and ARP are the only routed protocols supported for the interface.
- The Ethernet Management Interface cannot be used as a Lawful Intercept MD source interface.

- The Management Ethernet interface is part of its own VRF. This is discussed in more detail in the Gigabit ethernet management interface VRF, on page 84.

Gigabit ethernet port numbering

The Gigabit Ethernet Management port is always GigabitEthernet0.

The port can be accessed in configuration mode like any other port on the Cisco 8500 Series Secure Routers:

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface gigabitethernet0
Router(config-if)#
```

IP Address handling in ROMmon and the management ethernet port

On the Cisco 8500 Series Secure Routers, IP addresses can be configured in ROMmon (the **IP_ADDRESS=** and **IP_SUBNET_MASK=** commands) and through the use of the IOS command-line interface (the **ip address** command in interface configuration mode).

Assuming the IOS process has not begun running on the Cisco 8500 Series Secure Routers, the IP address that was set in ROMmon acts as the IP address of the Management Ethernet interface. In cases where the IOS process is running and has taken control of the Management Ethernet interface, the IP address specified when configuring the Gigabit Ethernet 0 interface in the IOS CLI becomes the IP address of the Management Ethernet interface. The ROMmon-defined IP address is only used as the interface address when the IOS process is inactive.

For this reason, the IP addresses specified in ROMmon and in the IOS CLI can be identical and the Management Ethernet interface will function properly in single RP configurations.

Gigabit ethernet management interface VRF

The Gigabit Ethernet Management interface is automatically part of its own VRF. This VRF, which is named “Mgmt-intf,” is automatically configured on the Cisco 8500 Series Secure Routers and is dedicated to the Management Ethernet interface; no other interfaces can join this VRF. Therefore, this VRF does not participate in the MPLS VPN VRF or any other network-wide VRF. The Mgmt-intf VRF supports loopback interface.

Placing the management ethernet interface in its own VRF has the following effects on the Management Ethernet interface:

- Many features must be configured or used inside the VRF, so the CLI may be different for certain Management Ethernet functions on the Cisco 8500 Series Secure Routers than on Management Ethernet interfaces on other routers.
- Prevents transit traffic from traversing the router. Because all built-in portd and the Management Ethernet interface are automatically in different VRFs, no transit traffic can enter the Management Ethernet interface and leave a built-in port, or vice versa.

- Improved security of the interface. Because the Mgmt-intf VRF has its own routing table as a result of being in its own VRF, routes can only be added to the routing table of the Management Ethernet interface if explicitly entered by a user.

The Management Ethernet interface VRF supports both IPv4 and IPv6 address families.

Common ethernet management tasks

Because users can perform most tasks on a router through the Management Ethernet interface, many tasks can be done by accessing the router through the Management Ethernet interface.

This section documents tasks that might be common or slightly tricky on the Cisco 8500 Series Secure Routers. It is not intended as a comprehensive list of all tasks that can be done using the Management Ethernet interface.

This section covers the following processes:

Viewing the VRF configuration

The VRF configuration for the Management Ethernet interface is viewable using the **show running-config vrf** command.

This example shows the default VRF configuration:

```
Router#show running-config vrf
Building configuration...
Current configuration : 351 bytes
vrf definition Mgmt-intf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
(some output removed for brevity)
```

Viewing detailed VRF information for the management ethernet VRF

To see detailed information about the Management Ethernet VRF, enter the **show vrf detail Mgmt-intf** command:

```
Router#show vrf detail Mgmt-intf
```

Setting a default route in the management ethernet interface VRF

To set a default route in the Management Ethernet Interface VRF, enter the following command

ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 *next-hop-IP-address*

Setting the management ethernet IP address

The IP address of the Management Ethernet port is set like the IP address on any other interface.

Below are two simple examples of configuring an IPv4 adress and an IPv6 address on the Management Ethernet interface.

IPv4 Example

```
Router(config)#interface GigabitEthernet 0
Router(config-if)#ip address
A.B.C.D A.B.C.D
```

IPv6 Example

```
Router(config)#interface GigabitEthernet 0
```

Router(config-if)# **ipv6 address** *X:X:X:X::X*

Telnetting over the management ethernet interface

Telnetting can be done through the VRF using the Management Ethernet interface.

In the following example, the router telnets to 172.17.1.1 through the Management Ethernet interface VRF:

```
Router#telnet 172.17.1.1 /vrf Mgmt-intf
```

Pinging over the management ethernet interface

Pinging other interfaces using the Management Ethernet interface is done through the VRF.

In the following example, the router pings the interface with the IP address of 172.17.1.1 through the Management Ethernet interface:

```
Router#ping vrf Mgmt-intf 172.17.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.17.1.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

Copy using TFTP or FTP

To copy a file using TFTP through the Management Ethernet interface, the **ip tftp source-interface GigabitEthernet 0** command must be entered before entering the **copy tftp** command because the **copy tftp** command has no option of specifying a VRF name.

Similarly, to copy a file using FTP through the Management Ethernet interface, the **ip ftp source-interface GigabitEthernet 0** command must be entered before entering the **copy ftp** command because the **copy ftp** command has no option of specifying a VRF name.

TFTP Example

```
Router(config)#ip tftp source-interface gigabitethernet 0
```

FTP Example

```
Router(config)#ip ftp source-interface gigabitethernet 0
```

NTP server

To allow the software clock to be synchronized by a Network Time Protocol (NTP) time server over the Management Ethernet interface, enter the **ntp server vrf Mgmt-intf** command and specify the IP address of the device providing the update.

The following CLI provides an example of this procedure.

```
Router(config)#ntp server vrf Mgmt-intf 172.17.1.1
```

SYSLOG server

To specify the Management Ethernet interface as the source IP or IPv6 address for logging purposes, enter the **logging host <ip-address> vrf Mgmt-intf** command.

The following CLI provides an example of this procedure.

```
Router(config)#logging host <ip-address> vrf Mgmt-intf
```

SNMP-related services

To specify the Management Ethernet interface as the source of all SNMP trap messages, enter the **snmp-server source-interface traps gigabitEthernet 0** command.

The following CLI provides an example of this procedure:

```
Router(config)#snmp-server source-interface traps gigabitEthernet 0
```

Domain name assignment

The IP domain name assignment for the Management Ethernet interface is done through the VRF.

To define the default domain name as the Management Ethernet VRF interface, enter the **ip domain-name vrf Mgmt-intf** *domain* command.

```
Router(config)#ip domain-name vrf Mgmt-intf cisco.com
```

DNS service

To specify the Management Ethernet interface VRF as a name server, enter the **ip name-server vrf Mgmt-intf** *IPv4-or-IPv6-address* command.

```
Router(config)# ip name-server vrf Mgmt-intf
 IPv4-or-IPv6-address
```

RADIUS or TACACS+ server

To group the Management VRF as part of a AAA server group, enter the **ip vrf forward Mgmt-intf** command when configuring the AAA server group.

The same concept is true for configuring a TACACS+ server group. To group the Management VRF as part of a TACACS+ server group, enter the **ip vrf forwarding Mgmt-intf** command when configuring the TACACS+ server group.

RADIUS Server Group Configuration

```
Router(config)#aaa group server radius hello
Router(config-sg-radius)#ip vrf forwarding Mgmt-intf
```

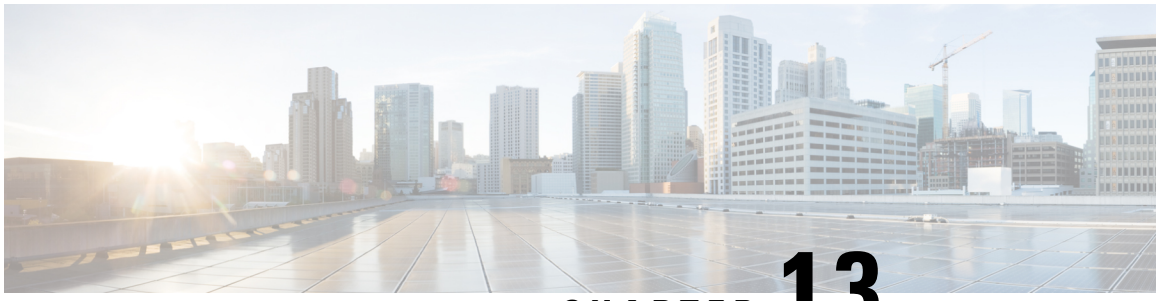
TACACS+ Server Group Example

```
outer(config)#aaa group server tacacs+ hello
Router(config-sg-tacacs+)#ip vrf forwarding Mgmt-intf
```

VTY lines with ACL

To ensure an access control list (ACL) is attached to vty lines that are and are not using VRF, use the **vrf-also** option when attaching the ACL to the vty lines.

```
Router(config)# line vty 0 4
Router(config-line)# access-class 90 in vrf-also
```



CHAPTER 13

Configuring bridge domain interfaces

The Cisco 8500 Series Secure Routers support the bridge domain interface (BDI) feature for packaging Layer 2 Ethernet segments into Layer 3 IP.

Restrictions for bridge domain interfaces

The following are the restrictions pertaining to bridge domain interfaces:

- Only 4096 bridge domain interfaces are supported per system.
- For a bridge domain interface, the maximum transmission unit (MTU) size can be configured between 1500 and 9216 bytes.
- Bridge domain interfaces support only the following features:
  - IPv4 Multicast
  - QoS marking and policing. Shaping and queuing are not supported
  - IPv4 VRF
  - IPv6 unicast forwarding
  - Dynamic routing such as BGP, OSPF, EIGRP, RIP, IS-IS, and STATIC
  - Hot Standby Router Protocol (HSRP)
  - Virtual Router Redundancy Protocol (VRRP) from IOS XE 3.8.0 onwards.
- Bridge domain interfaces do not support the following features:
  - PPP over Ethernet (PPPoE)
  - Bidirectional Forwarding Detection (BFD) protocol
  - QoS
  - Network-Based Application Recognition (NBAR) or Advanced Video Coding (AVC)

Information about bridge domain interface

Bridge domain interface is a logical interface that allows bidirectional flow of traffic between a Layer 2 bridged network and a Layer 3 routed network traffic. Bridge domain interfaces are identified by the same index as the bridge domain. Each bridge domain represents a Layer 2 broadcast domain. Only one bridge domain interface can be associated with a bridge domain.

Bridge domain interface supports the following features:

- IP termination
- Layer 3 VPN termination
- Address Resolution Protocol (ARP), G-ARP, and P-ARP handling
- MAC address assignment

Prior to configuring a bridge domain interface, you must understand the following concepts:

- Ethernet Virtual Circuit Overview
- Bridge Domain Interface Encapsulation
- Assigning a MAC Address
- Support for IP Protocols
- Support for IP Forwarding
- Packet Forwarding
- Bridge Domain Interface Statistics

Ethernet virtual circuit overview

An Ethernet Virtual Circuit (EVC) is an end-to-end representation of a single instance of a Layer 2 service that is offered by a provider. It embodies the different parameters on which the service is being offered. In the Cisco EVC Framework, the bridge domains are made up of one or more Layer 2 interfaces known as service instances. A service instance is the instantiation of an EVC on a given port on a given router. Service instance is associated with a bridge domain based on the configuration.

An incoming frame can be classified as service instance based on the following criteria:

- Single 802.1Q VLAN tag, priority-tagged, or 802.1ad VLAN tag
- Both QinQ (inner and outer) VLAN tags, or both 802.1ad S-VLAN and C-VLAN tags
- Outer 802.1p CoS bits, inner 802.1p CoS bits, or both
- Payload Ethernet type (five choices are supported: IPv4, IPv6, PPPoE-all, PPoE-discovery, and PPPoE-session)

Service instance also supports alternative mapping criteria:

- Untagged—Mapping to all the frames lacking a 802.1Q or 802.1ad header
- Default—Mapping to all the frames

Bridge domain interface encapsulation

Security Group classification includes both Source and Destination Group, which is specified by source SGT and DGT. SGT Based PBR feature provides the PBR route-map match clause for SGT/DGT based packet classification. SGT Based PBR feature supports configuration of unlimited number of tags, but it is recommended to configure the tags based on memory available in the platform.

An EVC provides the ability to employ different encapsulations on each Ethernet flow point (EFP) present in a bridge domain. A BDI egress point may not be aware of the encapsulation of an egress packet because the packet may have egressed from one or more EFPs with different encapsulations.

In a bridge domain, if all the EFPs have different encapsulations, the BDI must be untagged (using the no 802.1Q tag). Encapsulate all the traffic in the bridge domain (popped or pushed) at the EFPs. Configure rewrite at each EFP to enable encapsulation of the traffic on the bridge domain.

In a bridge domain, if all the EFPs have the same encapsulation, configure the encapsulations on the BDI using the encapsulation command. Enabling encapsulation at the BDI ensures effective pushing or popping of tags, thereby eliminating the need for configuring the rewrite command at the EFPs. For more information on configuring the encapsulations on the BDI, see the How to Configure a Bridge Domain Interface.

Assigning a MAC address

All the bridge domain interfaces on the Cisco 8500 Series Secure Routers share a common MAC address. The first bridge domain interface on a bridge domain is allocated a MAC address. Thereafter, the same MAC address is assigned to all the bridge domain interfaces that are created in that bridge domain.

Note You can configure a static MAC address on a bridge domain interface using the **mac-address** command.

Support for IP Protocols

Brigde domain interfaces enable the Cisco 8500 Series Secure Routers to act as a Layer 3 endpoint on the Layer 2 bridge domain for the following IP-related protocols:

- ARP
- DHCP
- HTTP
- ICMP
- NTP
- RARP
- SNMP
- TCP
- Telnet
- TFTP
- UDP

Support for IP Forwarding

Bridge domain interface supports the following IP forwarding features:

- IPv4 input and output access control lists (ACL)
- IPv4 input and output QoS policies. The operations supported for the input and output service policies on a bridge domain interface are:
  - Classification
  - Marking

- Policing
- IPv4 L3 VRFs

Packet forwarding

A bridge domain interface provides bridging and forwarding services between the Layer 2 and Layer 3 network infrastructure.

Layer 2 to Layer 3

During a packet flow from a Layer 2 network to a Layer 3 network, if the destination MAC address of the incoming packet matches the bridge domain interface MAC address, or if the destination MAC address is a multicast address, the packet or a copy of the packet is forwarded to the bridge domain interface.

Note MAC address learning cannot not be performed on the bridge domain interface.

Layer 3 to Layer 2

When a packet arrives at a Layer 3 physical interface of a router, a route lookup action is performed. If route lookup points to a bridge domain interface, then the bridge domain interface adds the layer 2 encapsulation and forwards the frame to the corresponding bridge domain. The byte counters are updated.

During a Layer 2 lookup on a bridge domain to which the bridge domain interface belongs, the bridge domain forwards the packets to the correct service instance based on the destination MAC address.

Link states of a bridge domain and a bridge domain interface

Bridge domain interface acts as a routable IOS interface on Layer 3 and as a port on a bridge domain. Both bridge domain interfaces and bridge domains operate with individual administrative states.

Shutting down a bridge domain interface stops the Layer 3 data service, but does not override or impact the state of the associated bridge domain.

Shutting down a bridge domain stops Layer 2 forwarding across all the associated members including service instances and bridge domain interfaces. The associated service instances influence operational state of a bridge domain. Bridge domain interface cannot be operational unless one of the associated service instances is up.

Note Because a bridge domain interface is an internal interface, the operational state of bridge domain interface does not affect the bridge domain operational state.

BDI initial state

The initial administrative state of a BDI depends on how the BDI is created. When you create a BDI at boot time in the startup configuration, the default administrative state for the BDI is up. It will remain in this state unless the startup configuration includes the shutdown command. This behavior is consistent with all the other interfaces. When you create a BDI dynamically at command prompt, the default administrative state is down.

BDI link state

A BDI maintains a link state that comprises of three states: administratively down, operationally down, and up. The link state of a BDI is derived from two independent inputs: the BDI administrative state set by the corresponding users and the fault indication state from the lower levels of the interface states. It defines a BDI link state based on the state of the two inputs.

Fault Indication State	BDI Admin{start straddle 2 columns}{end straddle 2 columns}	
{start emdash} {end emdash}	**Shutdown**	**No Shutdown**
No faults asserted	Admin-down	Up
At least one fault asserted	Admin-down	Operationally-Down

Bridge domain interface statistics

For virtual interfaces, such as the bridge domain interface, protocol counters are periodically queried from the QFP.

When packets flow from a Layer 2 bridge domain network to a Layer 3 routing network through the bridge domain interface, the packets are treated as bridge domain interface input packets and bytes. When packets arrive at a Layer 3 interface and are forwarded through the bridge domain interface to a Layer 2 bridge domain, the packets are treated as output packets and bytes, and the counters are updated accordingly.

A BDI maintains a standard set of Layer 3 packet counters as the case with all Cisco IOS interfaces. Use the show interface command to view the Layer 3 packet counters.

The convention of the counters is relative to the Layer 3 cloud. For example, input refers to the traffic entry to the Layer 3 cloud from the Layer 2 BD, while output refers to the traffic exit from the Layer 3 cloud to the Layer 2 BD.

Use the **show interfaces accounting** command to display the statistics for the BDI status. Use the **show interface** <*if-name*> command to display the overall count of the packets and bytes that are transmitted and received.

Creating or deleting a bridge domain interface

When you define an interface or subinterface for a Cisco IOS router, you name it and specify how it is assigned an IP address.You can create a bridge domain interface before adding a bridge domain to the system. This new bridge domain interface will be activated after the associated bridge domain is configured.

Note When a bridge domain interface is created, a bridge domain is automatically created.

When you create the bridge domain interface and the bridge domain, the system maintains the required associations for mapping the bridge domain-bridge domain interface pair.

The mapping of bridge domain and bridge domain interface is maintained in the system. The bridge domain interface uses the index of the associated bridge domain to show the association.

Bridge domain interface scalability

The following table lists the bridge domain interface scalability numbers, based on the type of Cisco 8500 Series Secure Routers Forwarding Processors.

Table 14: Bridge Domain Interface Scalability Numbers Based on the Type of Cisco 8500 Series Secure Routers Forwarding Processor

Description
Maximum bridge domain interfaces per router

Bridge-domain virtual IP interface

The Virtual IP Interface (VIF) feature helps to associate multiple BDI interfaces with a BD instance. The BD-VIF interface inherits all the existing L3 features of IOS logical IP interface.

Note You must configure every BD-VIF interface with a unique MAC address and it should belong to a different VRF.

The Virtual IP Interface (VIF) feature has the following limitations:

- BD-VIF interface does not support IP multicast.
- Number of BD-VIF interfaces with automatically generated MAC address varies on the basis of platforms.
- BD-VIF Interface does not support MPLS.
- The maximum number of BD-VIF interfaces per bridge-domain and the total number of BD-VIF interface for per system vary based on the type of platforms.

The maximum number of BD-VIF supported on Cisco 8500 Series Secure Routers are:

- C8570-G2 supports maximum 100 BD-VIF for a Bridge Domain
- C8550-G2 (support maximum 16 BD-VIF for a Bridge Domain

BD-VIF supports Flexible Netflow (FNF).

How to configure a bridge domain interface

To configure a bridge domain interface, perform the following steps:

Procedure

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface BDI** *{interface number}*

Example:

```
Router(config-if)# interface BDI3
```

Specifies a bridge domain interface on a Cisco 8500 Series Secure Routers.

Step 4 **encapsulation** *encapsulation dot1q <first-tag> [second-dot1q <second-tag>]*

Example:

```
Router(config-if)# encapsulation dot1Q 1 second-dot1q 2
```

Defines the encapsulation type.

The example shows how to define dot1q as the encapsulation type.

Step 5 Do one of the following:

Example:

ip address *ip-address mask*

Example:

Example:

ipv6 address *{X:X:X:X::X* **link-local**| *X:X:X:X::X/prefix [***anycast** | **eui-64***]* | **autoconfig** *[***default***]}*

Example:

```
Router(config-if)# ip address 2.2.2.1 255.255.255.0
```

Example:

Example:

```
Router(config-if)# ipv6 address AB01:CD1:123:C::/64 eui-64
```

Specifies either the IPv4 or IPv6 address for the bridge domain interface.

Step 6 **match security-group destination tag** *sgt-number*

Example:

```
Router(config-route-map)# match security-group destination tag 150
```

Configures the value for security-group destination security tag.

Step 7 **mac address** *{mac-address}*

Example:

```
Router(config-if)# mac-address 1.1.3
```

Specifies the MAC address for the bridge domain interface.

Step 8 **no shut**

Example:

```
Router(config-if)# no shut
```

Enables the bridge domain interface on the Cisco 8500 Series Secure Routers.

Step 9 **shut**

Example:

```
Router(config-if)# shut
```

Disables the bridge domain interface on the Cisco 8500 Series Secure Routers.

Example

The following example shows the configuration of a bridge domain interface at IP address 2.2.2.1 255.255.255.0:

```
Router# configure terminal
Router(config)# interface BDI3
Router(config-if)# encapsulation dot1Q 1 second-dot1q 2
Router(config-if)# ip address 2.2.2.1 255.255.255.0
Router(config-if)# mac-address 1.1.3
Router(config-if)# no shut
Router(config-if)# exit
```

Displaying and verifying bridge domain interface configuration

SUMMARY STEPS

1. **enable**
2. **show interfaces bdi**
3. **show platform software interface fp active name**
4. **show platform hardware qfp active interface if-name**
5. **debug platform hardware qfp feature**
6. **platform trace runtime process forwarding-manager module**
7. **platform trace boottime process forwarding-manager module interfaces**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router>enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**show interfaces bdi** **Example:** `Router# show interfaces BDI3`	Displays the configuration summary of the corresponding BDI.
Step 3	**show platform software interface fp active name** **Example:** `Router#show platform software interface fp active name BDI4`	Displays the bridge domain interface configuration in a Forwarding Processor.
Step 4	**show platform hardware qfp active interface if-name** **Example:** `Router#show platform hardware qfp active interface if-name BDI4`	Displays the bridge domain interface configuration in a data path.
Step 5	**debug platform hardware qfp feature** **Example:** `Router#debug platform hardware qfp active feature l2bd client all`	The selected CPP L2BD Client debugging is on.
Step 6	**platform trace runtime process forwarding-manager module** **Example:** `Router(config)#platform trace runtime slot F0 bay 0 process forwarding-manager module interfaces level info`	Enables the Forwarding Manager Route Processor and Embedded Service Processor trace messages for the Forwarding Manager process.
Step 7	**platform trace boottime process forwarding-manager module interfaces** **Example:** `Router(config)#platform trace boottime slot`	Enables the Forwarding Manager Route Processor and Embedded Service Processor trace messages for the Route Processor Forwarding Manager process during bootup.

	Command or Action	Purpose
	`R0 bay 1 process forwarding-manager forwarding-manager level max`	

What to do next

For additional information on the commands and the options available with each command, see the Cisco IOS Configuration Fundamentals Command Reference Guide located at:

{start hypertext}http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html{end hypertext}

Configuring bridge-domain virtual IP interface

```
enable
configure terminal
[no] interface BD-VIF interface-number
 [[no] vrf forwarding vrf-name]
 [[no] mac address mac-address]
 [[no] ip address ip-address mask]
 [[no] ipv6 address {X:X:X:X::X link-local| X:X:X:X::X/prefix [anycast | eui-64] | autoconfig
 [default]}]

exit
```

To delete BD-VIF interface, use the 'no' form of the command.

Associating VIF interface with a bridge domain

```
enable
configure terminal
bridge-domain bridge-domain number
[no] member BD-VIF interface-number
exit
```

Verifying bridge-domain virtual IP interface

All existing show commands for interface and IP interface can be used for the BD-VIF interface.

show interface bd-vif *bd-vif-id*

show ip interface bd-vif *bd-vif-id*

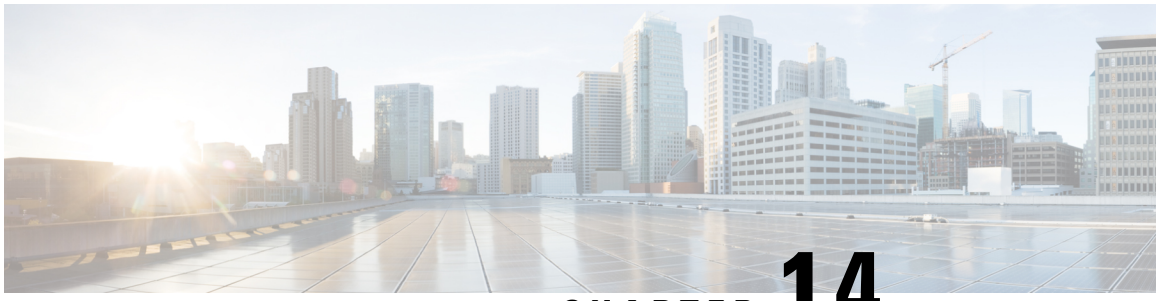
show bd-vif interfaces in fman-fp

show pla sof inter fp ac brief | i BD_VIF

Example configuration bridge-domain virtual IP interface

```
Detail sample:

interface Port-channel1
mtu 9000
no ip address
```

```
!Ethernet service endpoint one per neutron network
service instance 1756 ethernet
 description 4e8e5957-649f-477b-9e5b-f1f75b21c03c
 encapsulation dot1q 1756
 rewrite ingress tag pop 1 symmetric
 bridge-domain 1756
!
interface BD-VIF5001
no shutdown
vrf forwarding vrf5001
ip address 10.0.0.1 255.255.255.0
interface BD-VIF5002
no shutdown
vrf forwarding vrf5002
ip address 10.0.0.2 255.255.255.0

bridge-domain 1756
member Port-channel1 service-instance 1756
member bd-vif5001
member bd-vif5002
```



CHAPTER 14

Packet Trace

The Packet-Trace feature provides a detailed understanding of how data packets are processed by the Cisco IOS XE platform, and thus helps customers to diagnose issues and troubleshoot them more efficiently. This module provides information about how to use the Packet-Trace feature.

Information About Packet Trace

The Packet-Trace feature provides three levels of inspection for packets: accounting, summary, and path data. Each level provides a detailed view of packet processing at the cost of some packet processing capability. However, Packet Trace limits inspection to packets that match the debug platform condition statements, and is a viable option even under heavy-traffic situations in customer environments.

The following table explains the three levels of inspection provided by packet trace.

Table 15: Packet-Trace Level

Packet-Trace Level	Description
Accounting	Packet-Trace accounting provides a count of packets that enter and leave the network processor. Packet-Trace accounting is a lightweight performance activity, and runs continuously until it is disabled.
Summary	At the summary level of packet trace, data is collected for a finite number of packets. Packet-Trace summary tracks the input and output interfaces, the final packet state, and punt, drop, or inject packets, if any. Collecting summary data adds to additional performance compared to normal packet processing, and can help to isolate a troublesome interface.

Packet-Trace Level	Description
Path data	The packet-trace path data level provides the greatest level of detail in packet trace. Data is collected for a finite number of packets. Packet-Trace path data captures data, including a conditional debugging ID that is useful to correlate with feature debugs, a timestamp, and also feature-specific path-trace data. Path data also has two optional capabilities: packet copy and Feature Invocation Array (FIA) trace. The packet-copy option enables you to copy input and output packets at various layers of the packet (layer 2, layer 3 or layer 4). The FIA- trace option tracks every feature entry invoked during packet processing and helps you to know what is happening during packet processing. **Note** Collecting path data consumes more packet-processing resources, and the optional capabilities incrementally affect packet performance. Therefore, path-data level should be used in limited capacity or in situations where packet performance change is acceptable.

Usage Guidelines for Configuring Packet Trace

Consider the following best practices while configuring the Packet-Trace feature:

- Use of ingress conditions when using the Packet-Trace feature is recommended for a more comprehensive view of packets.
- Packet-trace configuration requires data-plane memory. On systems where data-plane memory is constrained, carefully consider how you will select the packet-trace values. A close approximation of the amount of memory consumed by packet trace is provided by the following equation:

memory required = (statistics overhead) + number of packets * (summary size + data size + packet copy size).

When the Packet-Trace feature is enabled, a small, fixed amount of memory is allocated for statistics. Similarly, when per-packet data is captured, a small, fixed amount of memory is required for each packet for summary data. However, as shown by the equation, you can significantly influence the amount of memory consumed by the number of packets you select to trace, and whether you collect path data and copies of packets.

Configuring Packet Trace

Perform the following steps to configure the Packet Trace feature.

Note The amount of memory consumed by the Packet-Trace feature is affected by the packet-trace configuration. You should carefully select the size of per-packet path data and copy buffers and the number of packets to be traced in order to avoid interrupting normal services. You can check the current data-plane DRAM memory consumption by using the **show platform hardware qfp active infrastructure exmem statistics** command.

SUMMARY STEPS

1. **enable**
2. **debug platform packet-trace packet** *pkt-num* **[fia-trace | summary-only] [circular] [data-size** *data-size***]**
3. **debug platform packet-trace {punt |inject|copy|drop|packet|statistics}**
4. **debug platform condition [ipv4 | ipv6] [interface** *interface***][access-list** *access-list -name* **|** *ipv4-address* **/** *subnet-mask* **|** *ipv6-address* **/** *subnet-mask***] [ingress | egress |both]**
5. **debug platform condition start**
6. **debug platform condition stop**
7. **show platform packet-trace {configuration | statistics | summary | packet {all |** *pkt-num***}}**
8. **clear platform condition all**
9. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables the privileged EXEC mode. Enter your password if prompted.
Step 2	**debug platform packet-trace packet** *pkt-num* **[fia-trace | summary-only] [circular] [data-size** *data-size***]** **Example:** `Router# debug platform packet-trace packets 2048 summary-only`	Collects summary data for a specified number of packets. Captures feature path data by default, and optionally performs FIA trace. *pkt-num*—Specifies the maximum number of packets maintained at a given time. **fia-trace**—Provides detailed level of data capture, including summary data, feature-specific data. Also displays each feature entry visited during packet processing. **summary-only**—Enables the capture of summary data with minimal details. **circular**—Saves the data of the most recently traced packets. *data-size*—Specifies the size of data buffers for storing feature and FIA trace data for each packet in bytes. When very heavy packet processing is performed on packets, users can increase the size of the data buffers if necessary. The default value is 2048.
Step 3	**debug platform packet-trace {punt |inject|copy|drop|packet|statistics}** **Example:** `Router# debug platform packet-trace punt`	Enables tracing of punted packets from data to control plane.

	Command or Action	Purpose
Step 4	**debug platform condition [ipv4 | ipv6] [interface** *interface*]**[access-list** *access-list -name* | *ipv4-address* / *subnet-mask* | *ipv6-address* / *subnet-mask*] **[ingress | egress |both]** **Example:** `Router# debug platform condition interface g0/0/0 ingress`	Specifies the matching criteria for tracing packets. Provides the ability to filter by protocol, IP address and subnet mask, access control list (ACL), interface, and direction.
Step 5	**debug platform condition start** **Example:** `Router# debug platform condition start`	Enables the specified matching criteria and starts packet tracing.
Step 6	**debug platform condition stop** **Example:** `Router# debug platform condition start`	Deactivates the condition and stops packet tracing.
Step 7	**show platform packet-trace {configuration | statistics | summary | packet {all |** *pkt-num***}}** **Example:** `Router# show platform packet-trace 14`	Displays packet-trace data according to the specified option. See {start cross reference} Table 21-1 {end cross reference} for detailed information about the **show** command options.
Step 8	**clear platform condition all** **Example:** `Router(config)# clear platform condition all`	Removes the configurations provided by the **debug platform condition** and **debug platform packet-trace** commands.
Step 9	**exit** **Example:** `Router# exit`	Exits the privileged EXEC mode.

Configuring Packet Tracer with UDF Offset

Perform the following steps to configure the Packet-Trace UDF with offset:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **udf** *udf name* **header** {**inner | outer**} {**13|14**} **offset** *offset-in-bytes* **length** *length-in-bytes*
4. **udf** *udf name* {**header | packet-start**} *offset-base offset length*
5. **ip access-list extended** {*acl-name* |*acl-num*}

6. **ip access-list extended { deny | permit } udf udf-name value mask**
7. **debug platform condition [ipv4 | ipv6] [interface** *interface*] **[access-list** *access-list -name* | *ipv4-address / subnet-mask* | *ipv6-address / subnet-mask*] **[ingress | egress |both]**
8. **debug platform condition start**
9. **debug platform packet-trace packet** *pkt-num* **[fia-trace | summary-only] [circular] [data-size** *data-size*]
10. **debug platform packet-trace {punt | inject|copy | drop |packet | statistics}**
11. **debug platform condition stop**
12. **exit**

DETAILED STEPS

Procedure

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

Step 3	**udf** *udf name* **header** {**inner	outer**} {**13|14**} **offset** *offset-in-bytes* **length** *length-in-bytes*
		Example:
		`Router(config)# udf TEST_UDF_NAME_1 header inner l3 64 1`
		`Router(config)# udf TEST_UDF_NAME_2 header inner l4 77 2`
		`Router(config)# udf TEST_UDF_NAME_3 header outer l3 65 1`
		`Router(config)# udf TEST_UDF_NAME_4 header outer l4 67 1`
			Configures individual UDF definitions. You can specify the name of the UDF, the networking header from which offset, and the length of data to be extracted.
			The **inner** or **outer** keywords indicate the start of the offset from the unencapsulated Layer 3 or Layer 4 headers, or if there is an encapsulated packet, they indicate the start of offset from the inner L3/L4.
			The **length** keyword specifies, in bytes, the length from the offset. The range is from 1 to 2.
			.

Step 4	**udf** *udf name* {**header	packet-start**} *offset-base offset length* **Example:** `Router(config)# udf TEST_UDF_NAME_5 packet-start 120 1`	• header—Specifies the offset base configuration. • packet-start—Specifies the offset base from packet-start. packet-start" can vary depending on if packet-trace is for an inbound packet or outbound packet. If the packet-trace is for an inbound packet then the packet-start will be layer2. For outbound, he packet-start will be layer3.

	Command or Action	Purpose
		• offset—Specifies the number of bytes offset from the offset base. To match the first byte from the offset base (Layer 3/Layer 4 header), configure the offset as 0. • length—Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs.
Step 5	**ip access-list extended** {*acl-name* |*acl-num*} **Example:** `Router(config)# ip access-list extended acl2`	Enables extended ACL configuration mode. The CLI enters the extended ACL configuration mode in which all subsequent commands apply to the current extended access list. Extended ACLs control traffic by the comparison of the source and destination addresses of the IP packets to the addresses configured in the ACL.
Step 6	**ip access-list extended { deny | permit } udf udf-name value mask** **Example:** `Router(config-acl)# permit ip any any udf TEST_UDF_NAME_5 0xD3 0xFF`	Configures the ACL to match on UDFs along with the current access control entries (ACEs) . The bytes defined in ACL is 0xD3. Masks are used with IP addresses in IP ACLs to specify what should be permitted and denied.
Step 7	**debug platform condition [ipv4 | ipv6] [interface** *interface***] [access-list** *access-list -name* **|** *ipv4-address* **/** *subnet-mask* **|** *ipv6-address* **/** *subnet-mask***] [ingress | egress |both]** **Example:** `Router# debug platform condition interface gi0/0/0 ipv4 access-list acl2 both`	Specifies the matching criteria for tracing packets. Provides the ability to filter by protocol, IP address and subnet mask, access control list (ACL), interface, and direction.
Step 8	**debug platform condition start** **Example:** `Router# debug platform condition start`	Enables the specified matching criteria and starts packet tracing.
Step 9	**debug platform packet-trace packet** *pkt-num* **[fia-trace | summary-only] [circular] [data-size** *data-size***]** **Example:** `Router# debug platform packet-trace packet 1024 fia-trace data-size 2048`	Collects summary data for a specified number of packets. Captures feature path data by default, and optionally performs FIA trace. *pkt-num*—Specifies the maximum number of packets maintained at a given time. **fia-trace**—Provides detailed level of data capture, including summary data, feature-specific data. Also displays each feature entry visited during packet processing. **summary-only**—Enables the capture of summary data with minimal details.

	Command or Action	Purpose
		circular—Saves the data of the most recently traced packets. *data-size*—Specifies the size of data buffers for storing feature and FIA trace data for each packet in bytes. When very heavy packet processing is performed on packets, users can increase the size of the data buffers if necessary. The default value is 2048.
Step 10	**debug platform packet-trace {punt | inject|copy | drop |packet | statistics}** **Example:** `Router# debug platform packet-trace punt`	Enables tracing of punted packets from data to control plane.
Step 11	**debug platform condition stop** **Example:** `Router# debug platform condition start`	Deactivates the condition and stops packet tracing.
Step 12	**exit** **Example:** `Router# exit`	Exits the privileged EXEC mode.

Displaying Packet-Trace Information

Use these **show** commands to display packet-trace information.

Table 16: show Commands

Command	Description
show platform packet-trace configuration	Displays packet trace configuration, including any defaults.
show platform packet-trace statistics	Displays accounting data for all the traced packets.
show platform packet-trace summary	Displays summary data for the number of packets specified.
show platform packet-trace {all | *pkt-num***} [decode]**	Displays the path data for all the packets or the packet specified. The **decode** option attempts to decode the binary packet into a more human- readable form.

Removing Packet Trace Data

Use these commands to clear packet-trace data.

Table 17: clear Commands

Command	Description
clear platform packet-trace statistics	Clears the collected packet-trace data and statistics.
clear platform packet-trace configuration	Clears the packet-trace configuration and the statistics.

Configuration Examples for Packet Trace

This section provides the following configuration examples:

Example: Configuring Packet Trace

This example describes how to configure packet trace and display the results. In this example, incoming packets to Gigabit Ethernet interface 0/0/1 are traced, and FIA-trace data is captured for the first 128 packets. Also, the input packets are copied. The **show platform packet-trace packet 0** command displays the summary data and each feature entry visited during packet processing for packet 0.

```
Router>
enable
Router# debug platform packet-trace packet 128 fia-trace
Router# debug platform packet-trace punt
Router# debug platform condition interface g0/0/1 ingress
Router# debug platform condition start
Router#! ping to UUT
Router# debug platform condition stop
Router# show platform packet-trace packet 0
Packet: 0           CBUG ID: 9
Summary
  Input     : GigabitEthernet0/0/1
  Output    : GigabitEthernet0/0/0
  State     : FWD
  Timestamp
    Start   : 1819281992118 ns (05/17/2014 06:42:01.207240 UTC)
    Stop    : 1819282095121 ns (05/17/2014 06:42:01.207343 UTC)
Path Trace
  Feature: IPV4
    Source      : 198.51.100.2
    Destination : 198.51.100.2
    Protocol    : 1 (ICMP)
  Feature: FIA_TRACE
    Entry     : 0x8059dbe8 - DEBUG_COND_INPUT_PKT
    Timestamp : 3685243309297
  Feature: FIA_TRACE
    Entry     : 0x82011a00 - IPV4_INPUT_DST_LOOKUP_CONSUME
    Timestamp : 3685243311450
  Feature: FIA_TRACE
    Entry     : 0x82000170 - IPV4_INPUT_FOR_US_MARTIAN
    Timestamp : 3685243312427
  Feature: FIA_TRACE
    Entry     : 0x82004b68 - IPV4_OUTPUT_LOOKUP_PROCESS
    Timestamp : 3685243313230
  Feature: FIA_TRACE
    Entry     : 0x8034f210 - IPV4_INPUT_IPOPTIONS_PROCESS
    Timestamp : 3685243315033
```

```
  Feature: FIA_TRACE
    Entry     : 0x82013200 - IPV4_OUTPUT_GOTO_OUTPUT_FEATURE
    Timestamp : 3685243315787
  Feature: FIA_TRACE
    Entry     : 0x80321450 - IPV4_VFR_REFRAG
    Timestamp : 3685243316980
  Feature: FIA_TRACE
    Entry     : 0x82014700 - IPV6_INPUT_L2_REWRITE
    Timestamp : 3685243317713
  Feature: FIA_TRACE
    Entry     : 0x82000080 - IPV4_OUTPUT_FRAG
    Timestamp : 3685243319223
  Feature: FIA_TRACE
    Entry     : 0x8200e500 - IPV4_OUTPUT_DROP_POLICY
    Timestamp : 3685243319950
  Feature: FIA_TRACE
    Entry     : 0x8059aff4 - PACTRAC_OUTPUT_STATS
    Timestamp : 3685243323603
  Feature: FIA_TRACE
    Entry     : 0x82016100 - MARMOT_SPA_D_TRANSMIT_PKT
    Timestamp : 3685243326183

Router# clear platform condition all
Router# exit
```

Linux Forwarding Transport Service (LFTS) is a transport mechanism to forward packets punted from the CPP into applications other than IOSd. This example displays the LFTS-based intercepted packet destined for binos application.

```
Router# show platform packet-trace packet 10
Packet: 10     CBUG ID: 52
Summary
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  State  : PUNT 55 (For-us control)
  Timestamp
    Start : 597718358383 ns (06/06/2016 09:00:13.643341 UTC)
    Stop : 597718409650 ns (06/06/2016 09:00:13.643392 UTC)
Path Trace
  Feature: IPV4
    Input  : GigabitEthernet0/0/0
    Output : <unknown>
    Source : 10.64.68.2
    Destination : 224.0.0.102
    Protocol : 17 (UDP)
      SrcPort : 1985
      DstPort : 1985
  Feature: FIA_TRACE
    Input  : GigabitEthernet0/0/0
    Output : <unknown>
    Entry  : 0x8a0177bc - DEBUG_COND_INPUT_PKT
    Lapsed time : 426 ns
  Feature: FIA_TRACE
    Input  : GigabitEthernet0/0/0
    Output : <unknown>
    Entry  : 0x8a017788 - IPV4_INPUT_DST_LOOKUP_CONSUME
    Lapsed time : 386 ns
  Feature: FIA_TRACE
    Input  : GigabitEthernet0/0/0
    Output : <unknown>
    Entry  : 0x8a01778c - IPV4_INPUT_FOR_US_MARTIAN
    Lapsed time : 13653 ns
  Feature: FIA_TRACE
    Input  : GigabitEthernet0/0/0
```

```
  Output : internal0/0/rp:1
  Entry  : 0x8a017730 - IPV4_INPUT_LOOKUP_PROCESS_EXT
  Lapsed time : 2360 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017be0 - IPV4_INPUT_IPOPTIONS_PROCESS_EXT
  Lapsed time : 66 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017bfc - IPV4_INPUT_GOTO_OUTPUT_FEATURE_EXT
  Lapsed time : 680 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017d60 - IPV4_INTERNAL_ARL_SANITY_EXT
  Lapsed time : 320 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017a40 - IPV4_VFR_REFRAG_EXT
  Lapsed time : 106 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017d2c - IPV4_OUTPUT_DROP_POLICY_EXT
  Lapsed time : 1173 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017940 - INTERNAL_TRANSMIT_PKT_EXT
  Lapsed time : 20173 ns
LFTS Path Flow: Packet: 10    CBUG ID: 52
  Feature: LFTS
  Pkt Direction: IN
  Punt Cause  : 55
       subCause : 0
```

Example: Using Packet Trace

This example provides a scenario in which packet trace is used to troubleshoot packet drops for a NAT configuration. This example shows how you can effectively utilize the level of detail provided by the Packet-Trace feature to gather information about an issue, isolate the issue, and then find a solution.

In this scenario, you can detect that there are issues, but are not sure where to start troubleshooting. You should, therefore, consider accessing the Packet-Trace summary for a number of incoming packets.

```
Router# debug platform condition ingress
Router# debug platform packet-trace packet 2048 summary-only
Router# debug platform condition start
Router# debug platform condition stop
Router# show platform packet-trace summary
Pkt   Input             Output            State  Reason
0     Gi0/0/0           Gi0/0/0           DROP   402 (NoStatsUpdate)
1     internal0/0/rp:0  internal0/0/rp:0  PUNT   21  (RP<->QFP keepalive)
2     internal0/0/recycle:0  Gi0/0/0      FWD
```

The output shows that packets are dropped due to NAT configuration on Gigabit Ethernet interface 0/0/0, which enables you to understand that an issue is occurring on a specific interface. Using this information, you

can limit which packets to trace, reduce the number of packets for data capture, and increase the level of inspection.

```
Router# debug platform packet-trace packet 256
Router# debug platform packet-trace punt
Router# debug platform condition interface Gi0/0/0
Router# debug platform condition start
Router# debug platform condition stop
Router# show platform packet-trace summary
Router# show platform packet-trace 15
Packet: 15          CBUG ID: 238
Summary
  Input     : GigabitEthernet0/0/0
  Output    : internal0/0/rp:1
  State     : PUNT 55  (For-us control)
  Timestamp
    Start   : 1166288346725 ns (06/06/2016 09:09:42.202734 UTC)
    Stop    : 1166288383210 ns (06/06/2016 09:09:42.202770 UTC)
Path Trace
  Feature: IPV4
    Input       : GigabitEthernet0/0/0
    Output      : <unknown>
    Source      : 10.64.68.3
    Destination : 224.0.0.102
    Protocol    : 17 (UDP)
      SrcPort   : 1985
      DstPort   : 1985
IOSd Path Flow: Packet: 15    CBUG ID: 238
  Feature: INFRA
    Pkt Direction: IN
    Packet Rcvd From CPP
  Feature: IP
    Pkt Direction: IN
    Source      : 10.64.68.122
    Destination : 10.64.68.255
  Feature: IP
    Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.64.68.122
    Destination : 10.64.68.255
    Interface   : GigabitEthernet0/0/0
  Feature: UDP
    Pkt Direction: IN
    src         : 10.64.68.122(1053)
    dst         : 10.64.68.255(1947)
    length      : 48

Router#show platform packet-trace packet 10
Packet: 10          CBUG ID: 10
Summary
  Input     : GigabitEthernet0/0/0
  Output    : internal0/0/rp:0
  State     : PUNT 55  (For-us control)
  Timestamp
    Start   : 274777907351 ns (01/10/2020 10:56:47.918494 UTC)
    Stop    : 274777922664 ns (01/10/2020 10:56:47.918509 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet0/0/0
    Output      : <unknown>
    Source      : 10.78.106.2
    Destination : 224.0.0.102
    Protocol    : 17 (UDP)
```

```
      SrcPort   : 1985
      DstPort   : 1985

IOSd Path Flow: Packet: 10    CBUG ID: 10
  Feature: INFRA
    Pkt Direction: IN
Packet Rcvd From DATAPLANE
 Feature: IP
    Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.78.106.2
    Destination : 224.0.0.102
    Interface   : GigabitEthernet0/0/0

  Feature: UDP
    Pkt Direction: IN DROP
    Pkt : DROPPED
    UDP: Discarding silently
    src        : 881 10.78.106.2(1985)
    dst         : 224.0.0.102(1985)
    length      : 60

Router#show platform packet-trace packet  12
Packet: 12          CBUG ID: 767
Summary
  Input     : GigabitEthernet3
  Output    : internal0/0/rp:0
  State     : PUNT 11  (For-us data)
  Timestamp
    Start   : 16120990774814 ns (01/20/2020 12:38:02.816435 UTC)
    Stop    : 16120990801840 ns (01/20/2020 12:38:02.816462 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet3
    Output      : <unknown>
    Source      : 12.1.1.1
    Destination : 12.1.1.2
    Protocol    : 6 (TCP)
      SrcPort   : 46593
      DstPort   : 23
IOSd Path Flow: Packet: 12    CBUG ID: 767
  Feature: INFRA
    Pkt Direction: IN
    Packet Rcvd From DATAPLANE

  Feature: IP
    Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 12.1.1.1
    Destination : 12.1.1.2
    Interface   : GigabitEthernet3

  Feature: IP
    Pkt Direction: IN
    FORWARDEDTo transport layer
    Source        : 12.1.1.1
    Destination   : 12.1.1.2
    Interface     : GigabitEthernet3

  Feature: TCP
    Pkt Direction: IN
    tcp0: I NoTCB 12.1.1.1:46593 12.1.1.2:23 seq 1925377975 OPTS 4 SYN  WIN 4128

Router# show platform packet-trace summary
Pkt   Input                     Output                    State  Reason
```

```
0     INJ.2                    Gi1                      FWD
1     Gi1                      internal0/0/rp:0         PUNT   11  (For-us data)
2     INJ.2                    Gi1                      FWD
3     Gi1                      internal0/0/rp:0         PUNT   11  (For-us data)
4     INJ.2                    Gi1                      FWD
5     INJ.2                    Gi1                      FWD
6     Gi1                      internal0/0/rp:0         PUNT   11  (For-us data)
7     Gi1                      internal0/0/rp:0         PUNT   11  (For-us data)
8     Gi1                      internal0/0/rp:0         PUNT   11  (For-us data)
9     Gi1                      internal0/0/rp:0         PUNT   11  (For-us data)
10    INJ.2                    Gi1                      FWD
11    INJ.2                    Gi1                      FWD
12    INJ.2                    Gi1                      FWD
13    Gi1                      internal0/0/rp:0         PUNT   11  (For-us data)
14    Gi1                      internal0/0/rp:0         PUNT   11  (For-us data)
15    Gi1                      internal0/0/rp:0         PUNT   11  (For-us data)
16    INJ.2                    Gi1                      FWD
```

The following example displays the packet trace data statistics.

```
Router#show platform packet-trace statistics
Packets Summary
  Matched  3
  Traced   3
Packets Received
  Ingress  0
  Inject   0
Packets Processed
  Forward  0
  Punt     3
    Count       Code  Cause
    3           56    RP injected for-us control
  Drop     0
  Consume  0

          PKT_DIR_IN
             Dropped       Consumed       Forwarded
INFRA            0             0             0
TCP              0             0             0
UDP              0             0             0
IP               0             0             0
IPV6             0             0             0
ARP              0             0             0

         PKT_DIR_OUT
             Dropped       Consumed       Forwarded
INFRA            0             0             0
TCP              0             0             0
UDP              0             0             0
IP               0             0             0
IPV6             0             0             0
ARP              0             0             0
```

The following example displays packets that are injected and punted to the forwarding processor from the control plane.

```
Router#debug platform condition ipv4 10.118.74.53/32 both
 Router#Router#debug platform condition start
Router#debug platform packet-trace packet 200
Packet count rounded up from 200 to 256

Router#show platform packet-tracer packet 0
show plat pack pa 0
Packet: 0           CBUG ID: 674
Summary
```

```
  Input     : GigabitEthernet1
  Output    : internal0/0/rp:0
  State     : PUNT 11  (For-us data)
  Timestamp
    Start   : 17756544435656 ns (06/29/2020 18:19:17.326313 UTC)
    Stop    : 17756544469451 ns (06/29/2020 18:19:17.326346 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet1
    Output      : <unknown>
    Source      : 10.118.74.53
    Destination : 198.51.100.38
    Protocol    : 17 (UDP)
      SrcPort   : 2640
      DstPort   : 500

IOSd Path Flow: Packet: 0    CBUG ID: 674
  Feature: INFRA
  Pkt Direction: IN
    Packet Rcvd From DATAPLANE

  Feature: IP
  Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.118.74.53
    Destination : 198.51.100.38
    Interface   : GigabitEthernet1

  Feature: IP
  Pkt Direction: IN
  FORWARDED To transport layer
    Source        : 10.118.74.53
    Destination   : 198.51.100.38
    Interface     : GigabitEthernet1

  Feature: UDP
  Pkt Direction: IN
  DROPPED
 UDP: Checksum error: dropping
 Source      : 10.118.74.53(2640)
 Destination : 198.51.100.38(500)

Router#show platform packet-tracer packet 2
Packet: 2           CBUG ID: 2

IOSd Path Flow:
  Feature: TCP
  Pkt Direction: OUTtcp0: O SYNRCVD 198.51.100.38:22 198.51.100.55:52774 seq 3052140910
OPTS 4 ACK 2346709419 SYN  WIN 4128

  Feature: TCP
  Pkt Direction: OUT
  FORWARDED
 TCP: Connection is in SYNRCVD state
 ACK         : 2346709419
 SEQ         : 3052140910
 Source      : 198.51.100.38(22)
 Destination : 198.51.100.55(52774)


  Feature: IP
  Pkt Direction: OUTRoute out the generated packet.srcaddr: 198.51.100.38, dstaddr:
198.51.100.55
```

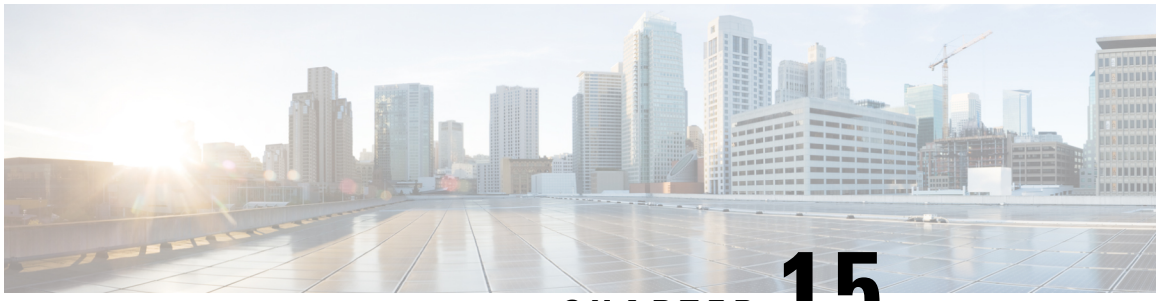
```
  Feature: IP
  Pkt Direction: OUTInject and forward successful srcaddr: 198.51.100.38, dstaddr:
198.51.100.55

  Feature: TCP
  Pkt Direction: OUTtcp0: O SYNRCVD 198.51.100.38:22 198.51.100.55:52774 seq 3052140910
OPTS 4 ACK 2346709419 SYN  WIN 4128
Summary
  Input     : INJ.2
  Output    : GigabitEthernet1
  State     : FWD
  Timestamp
    Start   : 490928006866 ns (06/29/2020 13:31:30.807879 UTC)
    Stop    : 490928038567 ns (06/29/2020 13:31:30.807911 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : internal0/0/rp:0
    Output      : <unknown>
    Source      : 172.18.124.38
    Destination : 172.18.124.55
    Protocol    : 6 (TCP)
      SrcPort   : 22
      DstPort   : 52774
  Feature: IPSec
    Result    : IPSEC_RESULT_DENY
    Action    : SEND_CLEAR
    SA Handle : 0
    Peer Addr : 55.124.18.172
    Local Addr: 38.124.18.172


Router#
```

Feature Information for Packet Trace

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to{start hypertext} http://www.cisco.com/go/cfn{end hypertext}. An account on Cisco.com is not required.

Table 18: Feature Information for Packet Trace

Feature Name	Releases	Feature Information
Packet Trace	Cisco IOS XE	The Packet Trace feature was introduced.



CHAPTER 15

Packet Drops

This document provides information about packet drops on the Cisco 8500 Series Secure Router.

Information about packet drops

Viewing packet drops

You can run the show drops command to troubleshoot the root cause of packet drops.

With the **show drops** command, you can identify the following:

- The root cause of the drop based on the feature or the protocol.
- The history of the QFP Drops.

Viewing packet drop information

Perform the following steps to view and filter the packet drop information for your instance based on the interface, protocol, or feature:

SUMMARY STEPS

1. **enable**
2. **show drops**
3. **show drops** { **bqs** | **crypto**| **firewall**| **interface**| **ip-all**| **nat**| **punt**| **qfp**| **qos**|**history**}

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**show drops** **Example:** `Router# show drops`	Displays the drop statistics.
Step 3	**show drops** { **bqs** | **crypto**| **firewall**| **interface**| **ip-all**| **nat**| **punt**| **qfp**| **qos**|**history**} **Example:** `Router# show drops qfp`	Displays the drop statistics and the summary for the interface or the protocol that you choose. **Note** From Cisco IOS XE 17.13.1a, a new keyword option history is added to the **show drops** command. The **show drops history qfp** command will allow the user to view the history of the QFP drops.

Example

Example for Viewing Packet Drop Information: Sample Output

The following is a sample output of the show drops command. This sample output displays the **packet drops** information related to the Quantum Flow Processor (QFP).

```
Router#show drops
bqs BQS related drops
crypto IPSEC related drops
firewall Firewall related drops
history History of drops
interface Interface drop statistics
ip-all IP related drops
nat NAT related drops
punt Punt path related drops
qfp QFP drop statistics
qos QoS related drops
| Output modifiers
<cr> <cr>
```

```
Router#show drops qfp
------------------ show platform hardware qfp active statistics drop detail
Last clearing of QFP drops statistics : Fri Feb 18 08:02:37 2022
(6d 23h 54m 29s ago)
-------------------------------------------------------------------------------
ID Global Drop Stats Packets
Octets
-------------------------------------------------------------------------------
319 BFDoffload 9
1350
61 Icmp 84
3780
```

```
53 IpFragErr 32136
48718168
244 IpLispHashLkupFailed 3
213
56 IpsecInput 18
4654
23 TailDrop 26713208
10952799454
216 UnconfiguredIpv6Fia 241788
26596680
------------------ show platform hardware qfp active interface all
statistics drop_summary
-----------------------------------------------------------------
Drop Stats Summary:
note: 1) these drop stats are only updated when PAL
reads the interface stats.
2) the interface stats include the subinterface
Interface Rx Pkts Tx Pkts
-------------------------------------------------------------------------------
GigabitEthernet1 60547 0
GigabitEthernet2 60782 27769658
GigabitEthernet3 60581 0
GigabitEthernet4 60502 1323990
Tunnel14095001 0 1990214
Tunnel14095002 0 3883238
Tunnel14095003 0 3879243
Tunnel14095004 0 2018866
Tunnel14095005 0 3875972
Tunnel14095006 0 3991497
Tunnel14095007 0 4107743
Tunnel14095008 0 3990601
```

Verifying packet information

This section shows examples of command output to verify packet information.

In order to display statistics of drops for all interfaces in Packet Processor Engine (PPE), use the command **show drops qfp**.

Note The wrapper command **show drops qfp** is the shorthand notation for the original **show platform hardware qfp active statistics drop** command.

```
Router#show drops qfp
-----------------------------------------------------------------
Global Drop Stats Octets
Packets
-----------------------------------------------------------------
AttnInvalidSpid 0 0
BadDistFifo 0 0
BadIpChecksum 0 0
```

In order to display the history of QFP drops for all interfaces in Packet Processor Engine (PPE), use the command **show drops history qfp**. This command can also track the number of packet drops in the last 1-min, 5-min and 30-min time period.

Note The wrapper command **show drops history qfp** is the shorthand notation for the original **show platform hardware qfp active statistics drop history** command.

```
Router#show drops history qfp
Last clearing of QFP drops statistics : Mon Jun 26 07:29:14
2023
(21s ago)
---------------------------------------------------------------
Global Drop Stats 1-Min
5-Min 30-Min All
---------------------------------------------------------------
Ipv4NoAdj 0
0 0 99818
Ipv4NoRoute 0
0 0 99853
```

Packet drops warnings

You can configure the warning thresholds for per drop cause and/or total QFP drop in packets per second. If the configured thresholds are exceeded, then a rate-limited syslog warning is generated. One warning is generated for total threshold exceeded and one warning per drop cause will be generated.

The warning is generated a maximum of once per minute for each drop cause. The drops over the previous minute are checked against the threshold (packets per second) x 60, and if the drops exceed this value, a warning is generated.

The following are the sample warnings for total and per drop cause respectively.

```
%QFP-5-DROP_OVERALL_RATE: Exceeded the overall drop threshold 10000 pps during the last
60-second measurement period, packets dropped in last 1 minute: 641220, last 5 minutes:
1243420, last 30 minutes: 124342200
```

```
%QFP-5-DROP_CAUSE_RATE: Exceeded the drop threshold 1000 pps for QosPolicing (drop code:
20) during the last 60-second measurement period, packets dropped due to QosPolicing in
last 1 minute: 61220, last 5 minutes: 43420, last 30 minutes: 4611200
```

Configuring packet drops warning thresholds

Perform the following steps to configure the warning thresholds for per drop cause and/or total QFP drop in packets per second.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **platform qfp drops threshold** {**per-cause** *drop_id threshold* | **total** *threshold*}

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**platform qfp drops threshold** {**per-cause** *drop_id threshold* | **total** *threshold*} **Example:** `Router# platform qfp drops threshold per-cause 206 10`	Specifies the per drop cause or total threshold value for the drop. **Note** Use the **show platform hardware qfp active statistics drop detail** command to view the drop cause ID.

Example

The following examples show how to configure the warning thresholds for per drop cause and total QFP drops.

Example for configuring warning threshold for per drop cause QFP drops

The following example shows how to configure the warning threshold of 15 pps for drop cause ID 24.

```
Router> enable
Router# configure terminal
Router(config)#platform qfp drops threshold ?
per-cause Set warning threshold for per cause QFP drops
total Set warning threshold for total QFP drops
Router(config)#platform qfp drops threshold per-cause ?
<0-1024> QFP drop cause ID
Router(config)#platform qfp drops threshold per-cause 24 ?
<0-2147483647> Drop threshold in packets per second (pps)
Router(config)#platform qfp drops threshold per-cause 24 15
```

Example for configuring warning threshold for total QFP drops

The following example shows how to configure the warning threshold of 100 pps for total QFP drops.

```
Router> enable
Router# configure terminal
Router(config)#platform qfp drops threshold ?
per-cause Set warning threshold for per cause QFP drops
total Set warning threshold for total QFP drops
Router(config)#platform qfp drops threshold total ?
<0-2147483647> Drop threshold in packets per second (pps)
Router(config)#platform qfp drops threshold total 100
```

Viewing packet drops warning thresholds

Perform the following steps to view the configured warning thresholds for per drop cause and total QFP drops.

SUMMARY STEPS

1. **enable**
2. **show platform hardware qfp active statistics drop threshold**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**show platform hardware qfp active statistics drop threshold** **Example:** `Router# show platform hardware qfp active statistics drop thresholds`	Displays the configured warning thresholds for per drop cause and total QFP drops. **Note** • The wrapper command **show drops thresholds** is the shorthand notation of the **show platform hardware qfp active statistics drop threshold**command.

Example

Example for Viewing Packet Drop Warning Thresholds

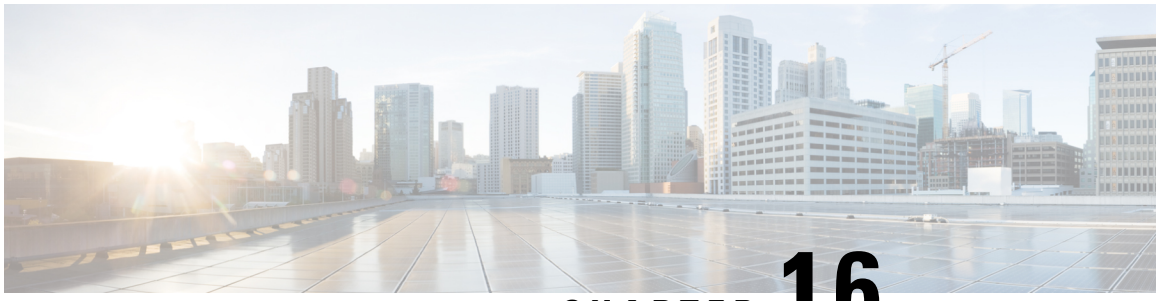
The following is a sample output of the **show platform hardware qfp active statistics drop threshold** command.

```
Router#show platform hardware qfp active statistics drop thresholds
-----------------------------------------------------
Drop ID         Drop Cause Name             Threshold
-----------------------------------------------------
10              BadIpChecksum                100
206             PuntPerCausePolicerDrops     10
20              QosPolicing                  200
                Total                        30
```

The following is a sample output of the **show drops thresholds** wrapper command.

```
Router#show platform hardware qfp active statistics drop thresholds
-----------------------------------------------------
Drop ID         Drop Cause Name             Threshold
-----------------------------------------------------
10              BadIpChecksum                100
206             PuntPerCausePolicerDrops     10
```

```
20              QosPolicing                 200
                Total                       30
```



CHAPTER 16

EVPN VPWS over SR-TE Preferred Path

The Ethernet VPN Virtual Private Wire Service (EVPN VPWS) functionality implements the signaling and encapsulation techniques for establishing an EVPN instance between a pair of PEs. This enhancement extends EVPN VPWS to support the specification of an SR-TE policy using the **preferred path** feature.

Feature information for EVPN VPWS over SR-TE preferred path

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

Table 19: Feature Information for EVPN VPWS over SR-TE Preferred Path

Feature Name	Releases	Feature Information
EVPN VPWS over SR-TE Preferred Path	Cisco IOS XE Cupertino 17.15.4a	This feature was introduced.

Restrictions for EVPN VPWS over SR-TE preferred path

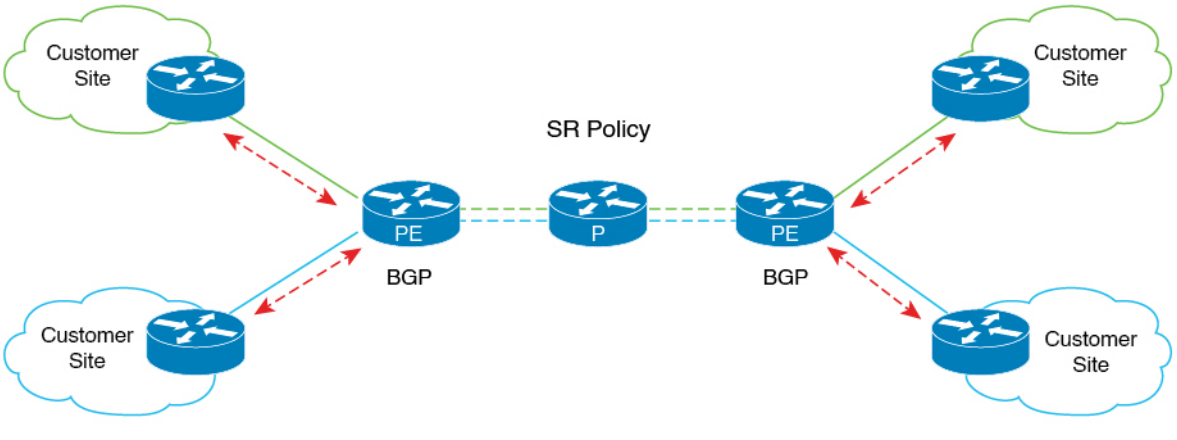
- SR On-Demand Next Hop (ODN) policy is not supported; only SR static policy is supported.
- SR Per-Flow Policy (PFP) is not supported; only SR Per-Destination Policy (PDP) is supported.
- Interior Gateway Protocol (IGP) is Intermediate System-to-Intermediate system (IS-IS).

Information about EVPN VPWS over SR-TE preferred path

The EVPN VPWS functionality implements the signaling and encapsulation techniques for establishing an EVPN instance between a pair of PEs. This enhancement enables EVPN VPWS to support the specification of an SR-TE policy using the **preferred path** feature. This feature includes the **fallback disable** option, which disables the default behavior of falling back on an alternate path if the preferred path is down.

The following figure illustrates the architecture:

Figure 1: EVPN VPWS over SR-TE Architecture



How to Configure EVPN VPWS over SR-TE preferred path

The following sections provide information about the tasks involved in configuring EVPN VPWS over the SR-TE preferred path.

Configuring EVPN VPWS over SR-TE preferred path

The following example shows how to enable EVPN VPWS over the configured SR-TE preferred path:

```
l2vpn evpn instance 100 point-to-point
rd 100:100
route-target export 100:100
route-target import 100:100
!
vpws context vc100
 preferred-path segment-routing traffic-eng policy p-100
 service target 100 source 100
interface GigabitEthernet0/0/3
service instance 100 ethernet
encapsulation dot1q 100
```

Configuring EVPN VPWS over SR-TE preferred path with fallback disable

The **fallback disable** command prevents a device from using the default path if the preferred path SR policy goes down.

```
l2vpn evpn instance 100 point-to-point
rd 100:100
route-target export 100:100
route-target import 100:100
vpws context vc100
service target 100 source 100
member GigabitEthernet0/0/3 service-instance 100
preferred-path segment-routing traffic-eng policy p-100 disable-fallback
```

Removing fallback disable from EVPN VPWS over SR-TE preferred path

The following example shows how to remove the fallback disable option in EVPN VPWS over SR-TE preferred path:

```
l2vpn evpn instance 100 point-to-point
vpws context vc100
 preferred-path segment-routing traffic-eng policy p-100
```

Disabling EVPN VPWS over SR-TE preferred path configuration

The following example shows how to disable the EVPN VPWS over SR-TE preferred path configuration:

```
l2vpn evpn instance 100 point-to-point
vpws context vc100
no  preferred-path segment-routing traffic-eng policy p-100 disable-fallback
```

Verifying EVPN VPWS over SR-TE preferred path

The following sample outputs show how to to verify the EVPN VPWS over SR-TE preferred path and fallback disable configurations.

- The following is a sample output showing the EVPN VPWS configuration over an SR-TE preferred path:

```
device# show l2vpn evpn vpws VC ID 100 detail
EVPN name: vc100, state: up, type: point-to-point
 EVPN ID: 100
 VPWS Service Instance ID: Source 1, Target 2
 Labels: Local 17, Remote 17
 Next Hop Address: 6.6.6.6
 Associated member interface Gi0/0/3 up, Gi0/0/3:3000 status is up
 Output interface: Tu65536, imposed label stack {16016 17}
 Preferred path:   active
 Default path: ready

device# show l2vpn evpn vpws vc preferred-path
Tunnel            EVPN ID     Source       Target       Name         Status
```

```
-------------       ------      --------     --------     --------     ----------
Tunnel65536         100         1            2            vc100          up
```

- The following is a sample output showing the EVPN VPWS configuration over an SR-TE preferred path, with fallback disabled:

```
device# show l2vpn evpn vpws VC ID 100 detail
EVPN name: vc100, state: up, type: point-to-point
 EVPN ID: 100
 VPWS Service Instance ID: Source 1, Target 2
 Labels: Local 17, Remote 17
 Next Hop Address: 6.6.6.6
 Associated member interface Gi0/0/3 up, Gi0/0/3:3000 status is up
 Output interface: Tu65536, imposed label stack {16016 17}
 Preferred path:   active
 Default path:     disabled
Dataplane:
SSM segment/switch IDs: 25037/12290 (used), PWID: 1
Rx Counters
1241 input transit packets, 463266 bytes
0 drops
Tx Counters
828 output transit packets, 402840 bytes
0 drops
24 VC FSM state transitions, Last 10 shown
DpUp: Act -> Est, Mon Sep 06 23:32:43.809 (2w2d ago)
RemDn: Est -> RemWait, Mon Sep 06 23:32:43.809 (2w2d ago)
RemUp: RemWait -> Act, Mon Sep 06 23:32:43.816 (2w2d ago)
DpUp: Act -> Est, Mon Sep 06 23:32:43.816 (2w2d ago)
DpDn: Est -> Act, Mon Sep 06 23:35:57.944 (2w2d ago)
DpUp: Act -> Est, Mon Sep 06 23:43:50.071 (2w2d ago)
DpDn: Est -> Act, Mon Sep 06 23:46:15.361 (2w2d ago)
DpUp: Act -> Est, Mon Sep 06 23:54:11.508 (2w2d ago)
DpDn: Est -> Act, Tue Sep 07 00:00:11.248 (2w2d ago)
DpUp: Act -> Est, Tue Sep 07 00:06:27.355 (2w2d ago)
```

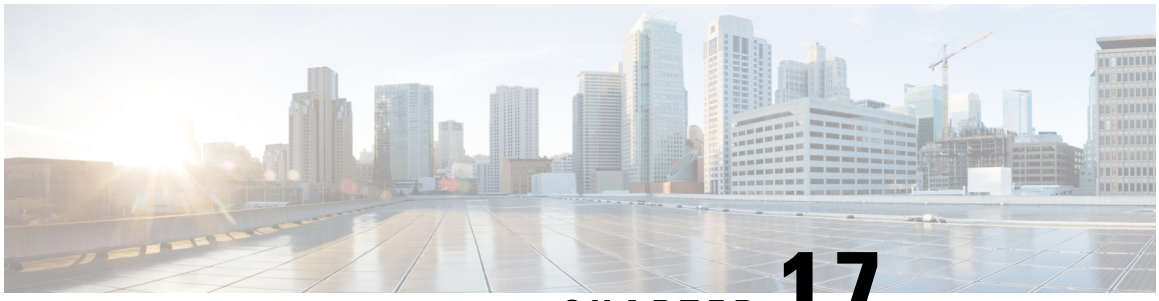
- The following is a sample output showing the EVPN VPWS configuration over an SR-TE preferred path, with fallback disable option removed:

```
device# show l2vpn evpn vpws VC ID 100 detail
EVPN name: vc100, state: up, type: point-to-point
 EVPN ID: 100
 VPWS Service Instance ID: Source 1, Target 2
 Labels: Local 17, Remote 17
 Next Hop Address: 6.6.6.6
 Associated member interface Gi0/0/3 up, Gi0/0/3:3000 status is up
 Output interface: Tu65536, imposed label stack {16016 17}
 Preferred path:   active
 Default path:     ready
```

- The following is a sample output showing the EVPN VPWS configuration over an SR-TE preferred path disabled:

```
device# show l2vpn evpn vpws VC ID 100 detail
EVPN name: vc100, state: up, type: point-to-point
 EVPN ID: 100
 VPWS Service Instance ID: Source 1, Target 2
 Labels: Local 17, Remote 17
 Next Hop Address: 6.6.6.6
```

```
Associated member interface Gi0/0/3 up, Gi0/0/3:3000  status is up
Output interface: Gi0/0/0, imposed label stack {16 16}
Preferred path: not configured
Default path:     active
```



CHAPTER 17

Configuring SFP

Configuring SFP+

Note Several Cisco platforms, NIMs, and SM cards support configuring multiple-rate SFPs on same interface, e.g., 1G SFP or 10G SFP+ on a 10G port.

In a port-channel bundle, all member interfaces should be of same speed, and duplex. It is recommended to use duplex interfaces of the same speed as member interfaces for configuring a port-channel.

For more information about interfaces that support multiple-rate SFPs, see the corresponding datasheets.

SUMMARY STEPS

1. **enable** *source-interface gigabitethernet slot/port*
2. **configure terminal**
3. **interface tengigabitethernet** *slot/port*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** *source-interface gigabitethernet slot/port* **Example:** `Router# enable`	Enables the privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:**	Enters the global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface tengigabitethernet** *slot/port* **Example:** `Router(config)# interface tengigabitethernet 4/11`	Specifies the 10-Gigabit Ethernet interface to be configured. Here: slot/port—Specifies the location of the interface.

Configuring FEC

Forward Error Correction (FEC) checks and recovers potential errors during long-range data transmission. The Cisco 8500 Series Secure Router have long range SFP, therefore FEC must be configured.

SUMMARY STEPS

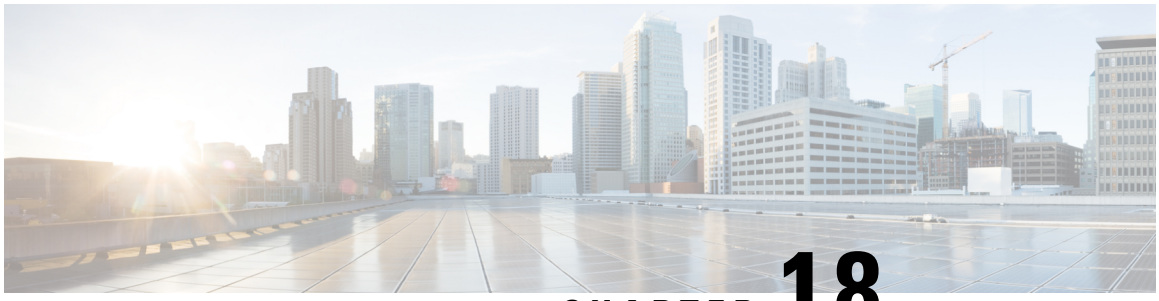
1. **enable** *source-interface gigabitethernet slot/port*
2. **configure terminal**
3. **interface twentyfivegigabitethernet** *slot/port*
4. **fec** { **auto** | **cl108** | **cl74** | **off**}

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** *source-interface gigabitethernet slot/port* **Example:** `Router# enable`	Enables the privileged EXEC mode. If prompted, enter your password.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters the global configuration mode.
Step 3	**interface twentyfivegigabitethernet** *slot/port* **Example:** `Router(config)# interface twentyfivegigabitethernet 0/0/16 4/11`	Specifies the 10-Gigabit Ethernet interface to be configured. Here: slot/port—Specifies the location of the interface.
Step 4	**fec** { **auto**	**cl108**	**cl74**	**off**} **Example:** `Router(config)# interface twentyfivegigabitethernet 0/0/16 4/11`	Configures FEC. Following are the modes of the fec command: • auto— Enables FEC based on SFP type • cl108— Enables clause108 <= RS-FEC(528,514)

	Command or Action	Purpose
		• cl74— Enables clause74 <= FC-FEC • disable— Disables FEC on interface **Note** • The fec command is only applicable to 25G links. • For 10/25G dual-rate SFP, if the speed is changed from 25G to 10G, fec configuration should be removed first before speed change.



CHAPTER 18

Cisco thousand eyes enterprise agent application hosting

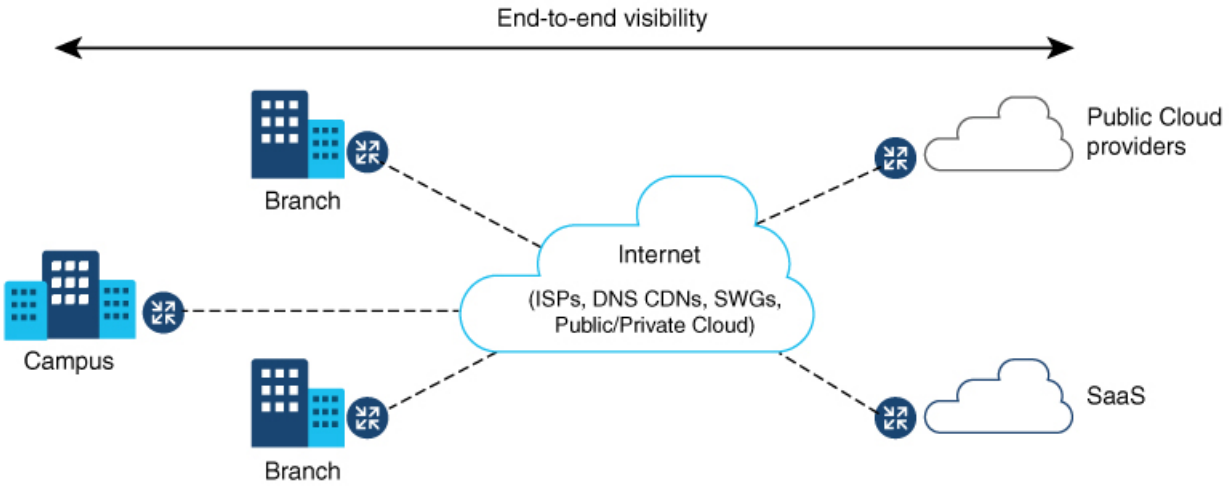
This chapter provides information on Cisco Thousand Eyes Enterprise Agent Application Hosting. The following sections are included in this chapter:

Cisco ThousandEyes enterprise agent application hosting

Cisco ThousandEyes is a network intelligence platform that allows you to use its agents to run a variety of tests from its agents to monitor the network and application performance. This application enables you to view end-to-end paths across networks and services that impact your business. Cisco ThousandEyes application actively monitors the network traffic paths across internal, external, and internet networks in real time, and helps to analyse the network performance. Also, isco ThousandEyes application provides application availability insights that are enriched with routing and device data for a multidimensional view of digital experience.

You can use application-hosting capabilities to deploy the Cisco ThousandEyes Enterprise Agent as a container application on Cisco 8500 Series Secure Routers. This agent application runs as a docker image using Cisco IOx docker-type option. For more information on how to configure Cisco ThousandEyes in controller mode, see Cisco SD-WAN Systems and Interfaces Configuration Guide.

Figure 2: Network View through ThousandEyes Application



Feature information for Cisco ThousandEyes enterprise agent application hosting

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 20: Feature Information for ThousandEyes Enterprise Agent Application Hosting

Feature Name	Releases	Feature Information
Cisco ThousandEyes Enterprise Agent Application Hosting	Cisco IOS XE 17.15.4a	With the integration of ThousandEyes Agent Application running on routing platforms using the app-hosting capabilities as container, you can have visibility into application experience with deep insights into the Internet, cloud providers, and enterprise networks.

Supported platforms and system requirements

The following table lists the supported platforms and system requirements.

Platform	Bootflash	DRAM
C8570-G2	480 GB NVMe SSD	32 GB default (two DIMMS) can be upgraded to 64 GB total
C8550-G2	480 GB NVMe SSD	32 GB default (two DIMMS) can be upgraded to 64 GB total

Workflow to install and run the Cisco ThousandEyes application

To install and run the Cisco ThousandEyes image on a device, perform these steps:

Procedure

Step 1 Create a new account on the Cisco ThousandEyes portal.

Step 2 Download the Cisco ThousandEyes application package from the software downloads page and ensure that you use the agent version 5.1

Step 3 Copy the image on the device.

Step 4 Install and launch the image.

Step 5 Connect the agent to the controller.

Workflow to host the Cisco ThousandEyes application

To install and launch the application, perform these steps:

Before you begin

Create a new account on the Cisco ThousandEyes portal and generate the token. The Cisco ThousandEyes agent application uses this token to authenticate and check into the correct Cisco ThousandEyes account. you see a message stating that your token is invalid and you want to troubleshoot the issue, see Troubleshooting the Cisco ThousandEyes application, on page 141.

Note If you configure the correct token and Domain Name Server (DNS) information, the device is discovered automatically.

Procedure

Step 1 Enable Cisco IOX application environment on the device.

- Use the following commands for non-SD-WAN (autonomous mode) images:

```
config terminal
iox
end
write
```

- Use the following commands for SD-WAN (controller mode) images:

```
config-transaction
iox
commit
```

Step 2 If the IOx command is accepted, wait for a few seconds and check whether the IOx process is up and running by using the **show iox** command. The output must display that the show IOxman process is running.

```
Device #show iox

IOx Infrastructure Summary:
---------------------------
IOx service (CAF) 1.11.0.0     : Running
IOx service (HA)               : Not Supported
IOx service (IOxman)           : Running
IOx service (Sec storage)      : Not Supported
Libvirtd 1.3.4                 : Running
```

Step 3 Ensure that the ThousandEyes application LXC tarball is available in the device *bootflash:*.

Step 4 Create a virtual port group interface to enable the traffic path to the Cisco ThousandEyes application:

```
interface VirtualPortGroup 0
         ip address 192.168.35.1 255.255.255.0
       exit
```

Step 5 Configure the app-hosting application with the generated token:

```
app-hosting appid te
         app-vnic gateway1 virtualportgroup 0 guest-interface 0
         guest-ipaddress 192.168.35.2 netmask 255.255.255.0
         app-default-gateway 192.168.35.1 guest-interface 0
         app-resource docker
                  prepend-pkg-opts  Required to get the default run-time options from package.yaml
                  run-opts 1 "--hostname thousandeyes"
            run-opts 2 "-e TEAGENT_ACCOUNT_TOKEN=<ThousandEyes token>"
                    run-opts 3 "-e TEAGENT_PROXY_TYPE=STATIC -e
TEAGENT_PROXY_LOCATION=proxy.something.other:80"
         name-server0 75.75.75.75  ISP's DNS server
       end

app-hosting appid te
app-resource docker
 prepend-pkg-opts
 run-opts 2 "--hostname
```

Note
You can use the proxy configuration only if the Cisco ThousandEyes agent does not have an internet access without a proxy. Also, the hostname is optional. If you do not provide the hostname during the installation, the device hostname is used as the Cisco ThousandEyes agent hostname. The device hostname is displayed on the Cisco ThousandEyes portal. The DNS name server information is optional. If the Cisco ThousandEyes agent uses a private IP address, ensure that you establish a connection to the device through NAT.

Step 6 Configure the **start** command to run the application automatically when the application is installed on the device using the **install** command:

```
app-hosting appid te
        start
```

Step 7 Select a location to install the ThousandEyes application from these options:

```
Device# app-hosting install appid te package ?
          bootflash:  Package path □ if image is locally available in bootflash:
          harddisk:   Package path □ if image is locally available in M.2 USB
          https:      Package path □ Download over the internet if image is not locally present in
router. URL to ThousandEyes site hosting agent image to be provided here
```

Step 8 Check if the application is up and running:

```
Device#show app-hosting list
App id                                   State
---------------------------------------------------------
 te                                       RUNNING
```

Note
If any of these steps fail, use the **show logging** command and check the IOx error message. If the error message is about insufficient disk space, clean the storage media (bootflash or hard disk) to free up the space. Use the **show app-hosting resource** command to check the CPU and disk memory.

Downloading and copying the image to the device

To download and copy the image to bootflash, perform these steps:

Procedure

Step 1 Check if the Cisco ThousandEyes image is precopied to *bootflash:/<directory name>*.

Step 2 If the image is not available in the device directory, perform these steps:

a) If the device has a direct access to internet, use the *https:*. option in the **application install** command. This option downloads the image from the Cisco ThousandEyes software downloads page into *bootflash:/apps* and installs the application.

```
Device# app-hosting install appid <appid string> package [bootflash: | flash | http | https://
| ftp | ] URL to image location hosted on ThousandEyes portal

Device#app-hosting install appid te1000 package
https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-4.0.2.cisco.tar

Installing package
'https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-4.0.2.cisco.tar'
 for 'te1000'.

Use 'show app-hosting list' for progress.
*Jun 29 23:43:29.244: %IOSXE-6-PLATFORM: R0/0: IOx:  App verification successful
*Jun 29 23:45:00.449: %IM-6-INSTALL_MSG: R0/0: ioxman: app-hosting: Install succeeded: te1000
installed successfully Current state is DEPLOYED
*Jun 29 23:45:01.801: %IOSXE-6-PLATFORM: R0/0: IOx:  App verification successful
*Jun 29 23:45:51.054: %IM-6-START_MSG: R0/0: ioxman: app-hosting: Start succeeded: te1000 started
 successfully Current state is RUNNING

Device#show app-hosting detail appid te1000 ( Details of Application)
App id                 : te1000
Owner                  : iox
State                  : RUNNING
```

```
Application
 Type                 : docker
 Name                 : ThousandEyes Enterprise Agent
 Version              : 4.0
 Author               : ThousandEyes <support@thousandeyes.com>
 Path                 : bootflash:thousandeyes-enterprise-agent-4.0-22.cisco.tar
Resource reservation
 Memory               : 500 MB
 Disk                 : 1 MB
 CPU                  : 1500 units
 CPU-percent          : 70 %
```

b) If the device has a proxy server, copy the image manually to *bootflash:/apps*.
c) Download the Cisco ThousandEyes application package from the software downloads page and ensure that you use the agent version 4.0.2.
d) Create an application directory in the *bootflash:* to copy the image:

```
Device# mkdir bootflash:apps
Create directory filename [apps]?
Created dir bootflash:/apps
```

e) Copy the Cisco ThousandEyes image to the *bootflash:apps* directory.
f) Validate the image using the **verify** command:

```
verify /md5 bootflash:apps/<file name>
```

Connecting the Cisco ThousandEyes agent with the controller

Before you begin

Ensure that you have an Internet connection before you connect the agent with the controller.

Procedure

After the Cisco ThousandEyes application is up and running, the agent (ThousandEyes-agent) process connects to the controller that is running on the cloud environment.

Note
If you have issues related to connectivity, the application logs the relevant error messages in the application-specific logs (*/var/logs*).

Modifying the agent parameters

To modify the agent parameters, perform these actions:

Procedure

Step 1 Stop the application using the **app-hosting stop appid appid** command.
Step 2 Deactivate the application using the **app-hosting deactivate appid appid** command.
Step 3 Make the required changes to app-hosting configuration.
Step 4 Activate the application using the **app-hosting activate appid appid** command.
Step 5 Start the application using the **app-hosting start appid appid** command.

Uninstalling the application

To uninstall the application, perform these steps:

Procedure

Step 1 Stop the application using the **app-hosting stop appid te** command.
Step 2 Check if the application is in active state using the **show app-hosting list** command.
Step 3 Deactivate the application using the **app-hosting deactivate appid te** command.
Step 4 Ensure that the application is not in active state. Use the **show app-hosting list** command to check status of the application.
Step 5 Uninstall the application using the **app-hosting uninstall appid te** command.
Step 6 After the uninstallation process is complete, use the **show app-hosting list** command to check if the application is uninstalled successfully.

Troubleshooting the Cisco ThousandEyes application

To troubleshoot the Cisco ThousandEyes application, perform these steps:

1. Connect to Cisco ThousandEyes agent application using the **app-hosting connect appid appid session /bin/bash** command.
2. Verify the configuration applied to the application at the following path */etc/te-agent.cfg*.
3. View the logs at the following path */var/log/agent/te-agent.log*. You can use these logs to troubleshoot the configuration.

Checking the ThousandEyes Application Status

When the Cisco ThousandEyes application is in running state, it is registered on the ThousandEyes portal. If the application does not show up in a few minutes after the agent is in running state, check the following using the **app-hosting connect appid thousandeyes_enterprise_agent session** command:

```
Device#app-hosting connect appid thousandeyes_enterprise_agent session
Device#cat /var/log/agent/te-agent.log
2021-02-04 08:59:29.642 DEBUG [e4736a40] [te.agent.AptPackageInterface] {} Initialized APT
 package interface
2021-02-04 08:59:29.642 INFO  [e4736a40] [te.agent.main] {} Agent version 1.103.0 starting.
  Max core size is 0 and max open files is 1024
2021-02-04 08:59:29.642 DEBUG [e4736a40] [te.agent.db] {} Vacuuming database
2021-02-04 08:59:29.643 INFO  [e4736a40] [te.agent.db] {} Found version 0, expected version
 50
2021-02-04 08:59:29.672 INFO  [e4708700] [te.probe.ServerTaskExecutor] {} ProbeTaskExecutor
 started with 2 threads.
2021-02-04 08:59:29.673 INFO  [e2f05700] [te.probe.ProbeTaskExecutor.bandwidth] {}
ProbeTaskExecutor started with 1 threads.
2021-02-04 08:59:29.673 INFO  [e2704700] [te.probe.ProbeTaskExecutor.realtime] {}
ProbeTaskExecutor started with 1 threads.
2021-02-04 08:59:29.673 INFO  [e1f03700] [te.probe.ProbeTaskExecutor.throughput] {}
ProbeTaskExecutor started with 1 threads.
2021-02-04 08:59:29.674 DEBUG [e4736a40] [te.agent.DnssecTaskProceessor] {} Agent is not
running bind
2021-02-04 08:59:29.674 DEBUG [e4736a40] [te.snmp.RequestDispatcher] {} Initialised SNMP++
 session
2021-02-04 08:59:29.674 DEBUG [e4736a40] [te.snmp.RequestDispatcher] {} Initialised SNMP++
 session
2021-02-04 08:59:29.674 DEBUG [e4736a40] [te.snmp.RequestDispatcher] {} Initialised SNMP++
 session
2021-02-04 08:59:29.674 INFO  [e4736a40] [te.agent.main] {} Agent starting up
2021-02-04 08:59:29.675 INFO  [e4736a40] [te.agent.main] {} No agent id found, attempting
to obtain one
2021-02-04 08:59:29.675 INFO  [e4736a40] [te.agent.ClusterMasterAdapter] {} Attempting to
get agent id from sc1.thousandeyes.com
2021-02-04 08:59:29.679 ERROR [e4736a40] [te.agent.main] {} Error calling create_agent:
Curl error - Couldn't resolve host name
2021-02-04 08:59:29.680 INFO  [e4736a40] [te.agent.main] {} Sleeping for 30 seconds
Note :
```

Note Check the DNS server connection. If the Cisco ThousandEyes agent is assigned to a private IP address, check the NAT configuration.