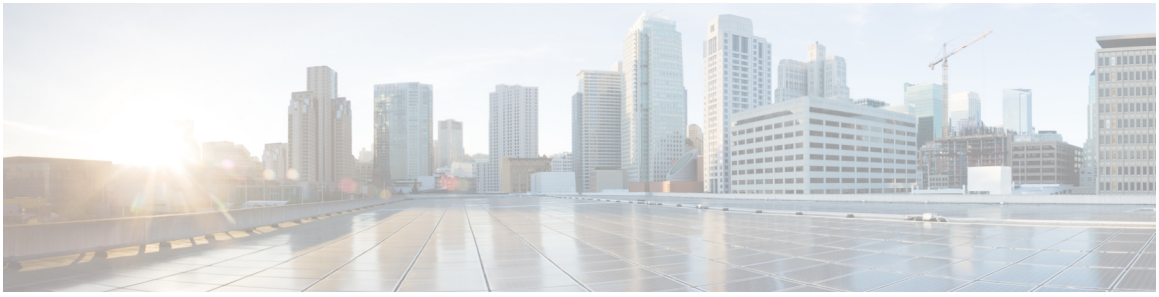
Cisco VG420 Voice Gateway Software Configuration Guide

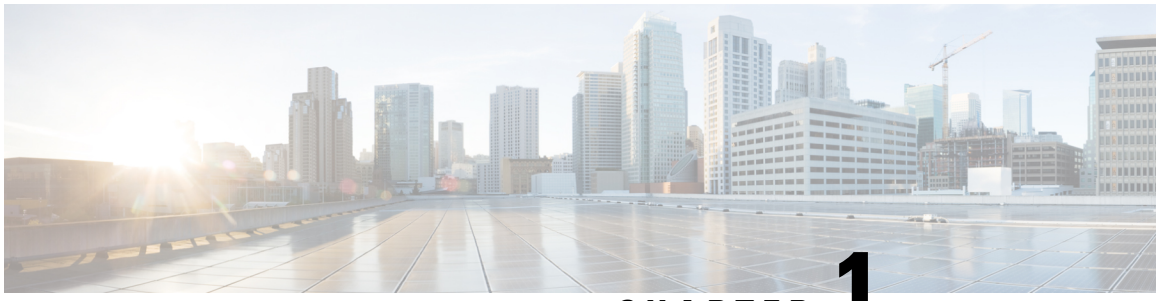
First Published: 2021-12-06

Last Modified: 2024-04-30

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

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Audience and Scope

This document is designed for the person who is responsible for configuring your Cisco Enterprise router. This document is intended primarily for the following audiences:

- Customers with technical networking background and experience.
- System administrators familiar with the fundamentals of router-based internetworking but who might not be familiar with Cisco IOS software.
- System administrators who are responsible for installing and configuring internetworking equipment, and who are familiar with Cisco IOS software.

Feature Compatibility

For more information about the Cisco IOS XE software, including features available on your device as described in the configuration guides, see the respective router documentation set.

To verify support for specific features, use the Cisco Feature Navigator tool. This tool enables you to determine the Cisco IOS XE software images that support a specific software release, feature set, or a platform.

Document Conventions

This documentation uses the following conventions:

Convention	Description
^ or **Ctrl**	The ^ and **Ctrl** symbols represent the Control key. For example, the key combination **^D** or **Ctrl-D** means hold down the **Control** key while you press the **D** key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

The command syntax descriptions use the following conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter exactly as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example, see the following table.

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`bold screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. Exclamation points are also displayed by the Cisco IOS XE software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials that may not be contained in this manual.

Communications, Services, and Additional Information

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

Cisco Bug Search Tool

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

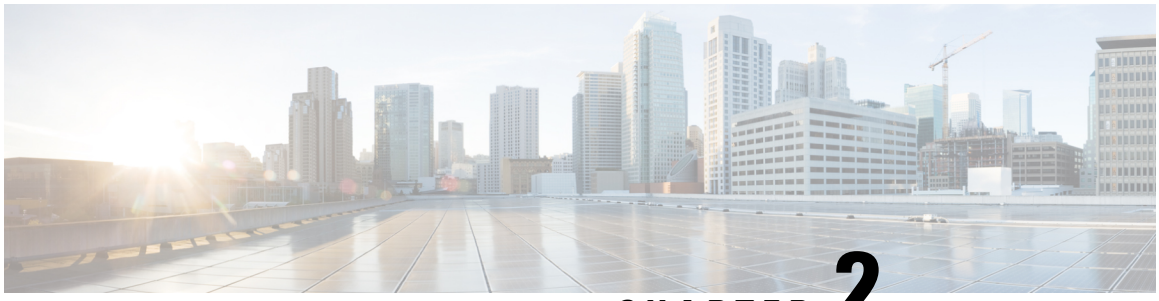
Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

Troubleshooting

For the most up-to-date, detailed troubleshooting information, see the Cisco TAC website at https://www.cisco.com/en/US/support/index.html.

Go to **Products by Category** and choose your product from the list, or enter the name of your product. Look under **Troubleshoot and Alerts** to find information for the issue that you are experiencing.



CHAPTER 2

Overview

This document is a summary of software functionality that is specific to the Cisco VG420 Voice Gateway.

Cisco VG420 Voice Gateway is a high-density analog voice gateway that provides enterprises, managed services providers, and service providers the ability to directly connect public-switched telephone networks (PSTNs) and existing telephony equipment to Cisco Enterprise Routers.

The fixed-port (FXS and FXO) modules in the voice gateway provide Dual-Tone Multifrequency (DTMF) detection, voice compression and decompression, call progress tone generation, Voice Activity Detection (VAD), echo cancellation, and adaptive jitter buffering. Cisco VG420 Voice Gateway is the intermediate path that enables TDM to IP transition.

The Cisco VG420 Voice Gateway supports the following interfaces:

- Gigabit Ethernet (GE)
- Micro USB Console Port
- RJ45 Console Port
- FXS Ports
- FXO Ports
- Network Interface Module (NIM)

To know how to install this voice gateway, see the *Cisco VG420 Voice Gateway Installation Guide*. After installing the voice-gateway, use this guide to complete a basic router configuration using the setup command facility.

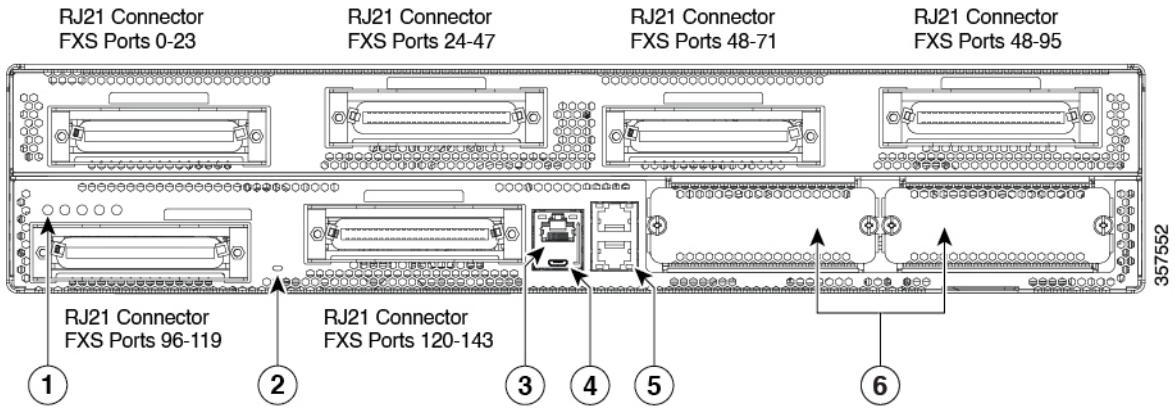
This guide also contains information on using the Cisco IOS software to perform other configuration tasks, such as configuring voice ports and other features.

- Identify the Device, on page 5

Identify the Device

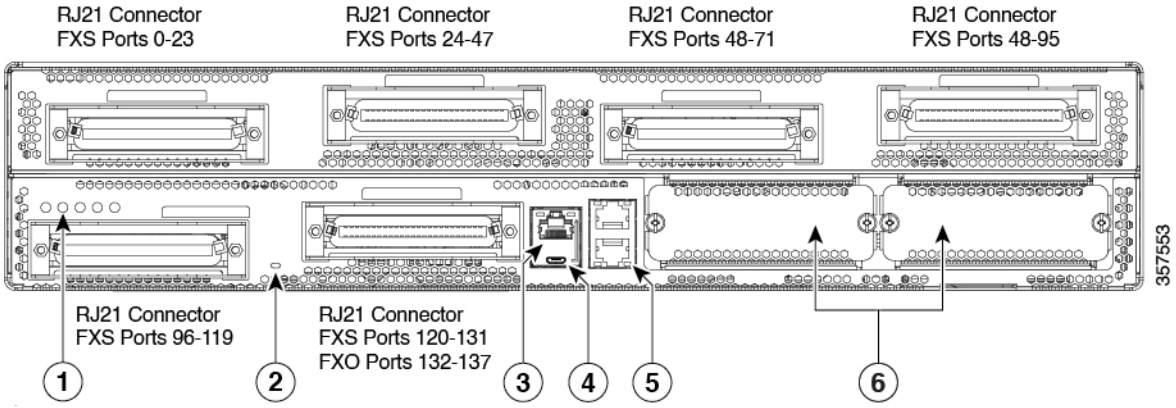
The following images show the I/O and side panel views of the Cisco VG420 Voice Gateway chassis that help you identify this device:

Figure 1: VG420-144FXS I/O Panel View



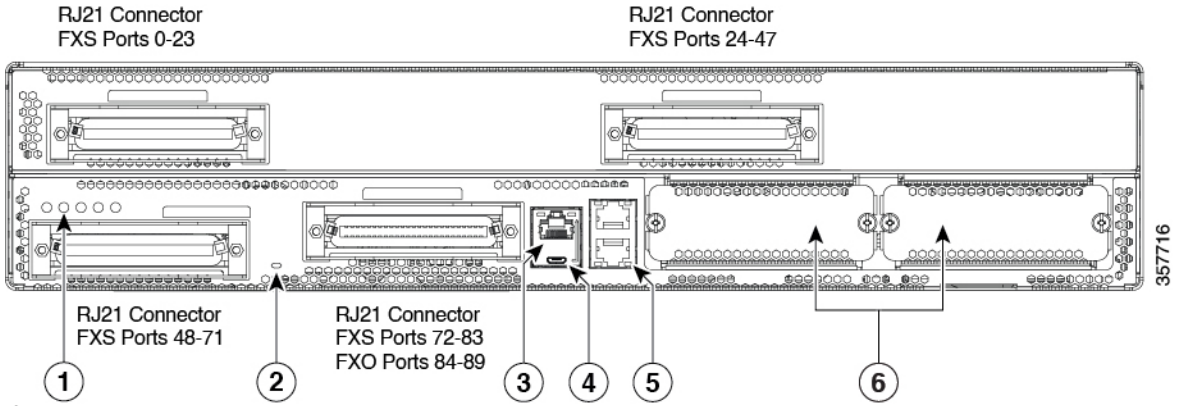
1	Status LEDs
2	FXS LED
3	Serial Console
4	Mini USB Console
5	Ethernet Ports
6	NIM Modules

Figure 2: VG420-132FXS/6FXO I/O Panel



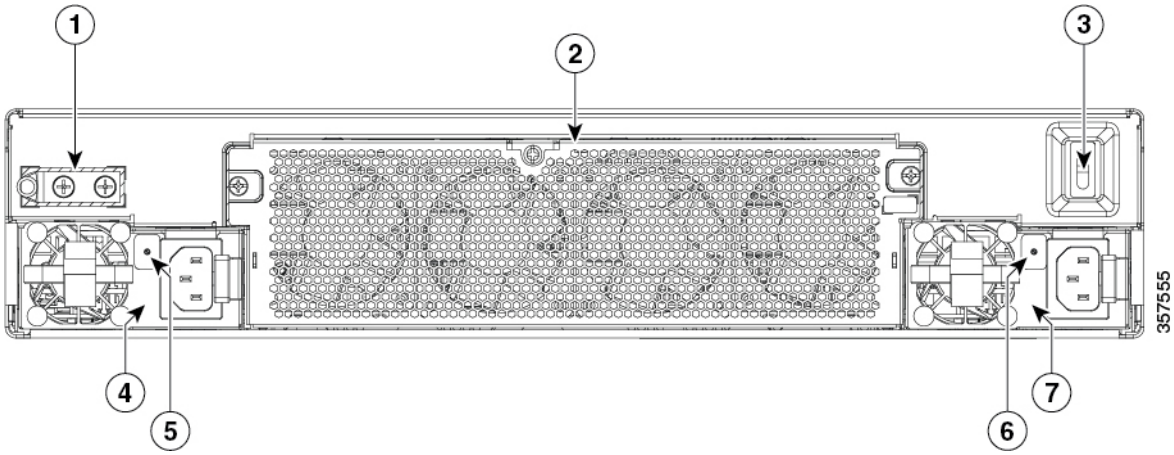
1	Status LEDs
2	FXS LED
3	Serial Console
4	Mini USB Console
5	Ethernet Ports
6	NIM Modules

Figure 3: VG420-84FXS/6FXO I/O Panel



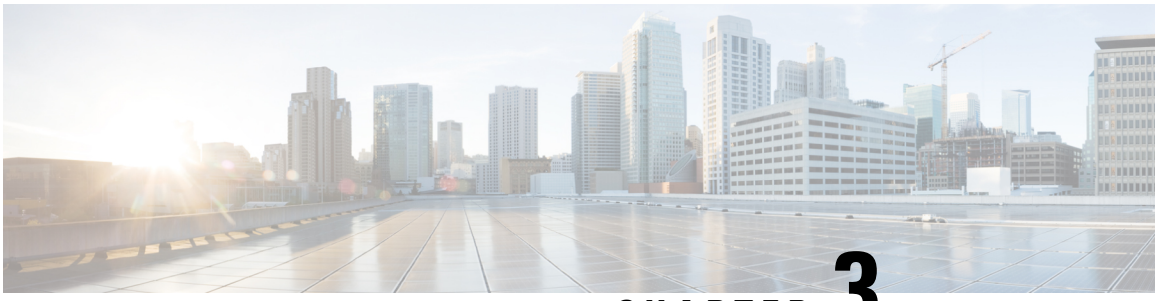
1	Status LEDs
2	FXS/FXO LED
3	Serial Console
4	Mini USB Console
5	Ethernet Ports
6	NIM Modules

Figure 4: Fan/Tray Side of the



1	Ground Lug
2	Removable Fan Tray
3	Power Switch
4	PSU1
5	PSU1 (Power LED)

6	PSU0 (Power LED)
7	PSU0



CHAPTER 3

Understanding Cisco IOS Software Basics

This section describes what you need to know about the Cisco IOS software before you configure the router using the CLI. Understanding these concepts will save time as you begin to use the commands. If you have never used Cisco IOS software or need a refresher, take a few minutes to read this chapter before you proceed to the next chapter.

If you are already familiar with Cisco IOS software, proceed to the *Configuring Host Name and Password* section in this guide.

Basics Before Using Commands

The following table specifies some basic rules and notes to configure the device by using the command line interface. Use the question mark (?) and arrow keys to help you enter commands:

Rule	Example
For a list of available commands, enter a question mark.	`Router> ?`
To complete a command, enter a few known characters followed by a question mark (with no space).	`Router> s?`
For a list of command variables, enter the command followed by a space and a question mark.	`Router> show ?`
To redisplay a command you previously entered, press the Up Arrow key. You can continue to press the Up Arrow key for more commands.	

Command Modes

The Cisco IOS user interface is divided into different modes. Each command mode permits you to configure different components on your router. The commands available at any given time depend on which mode you are currently in.

Entering a question mark (**?**) at the prompt displays a list of commands available for each command mode. The following table lists the most common command modes:

Table 1: Common Command Modes

Command Mode	Access Method	Router Prompt Displayed	Exit Method
User EXEC	Log in	Router>	Use the **logout** command.
Privileged EXEC	From user EXEC mode, enter the **enable** command.	Router#	To exit to user EXEC mode, use the **disable**, **exit**, or **logout** command.
Global configuration	From the privileged EXEC mode, enter the **configure terminal** command.	Router (config)#	To exit to privileged EXEC mode, use the **exit** or **end** command, or press **Ctrl-Z**.
Interface configuration	From the global configuration mode, enter the GigabitEthernet interface command such as, **gigabitethernet0/0**.	Router (config-if)#	To exit to global configuration mode, use the **exit** command. To exit directly to privileged EXEC mode, press **Ctrl-Z**.

Timesaver Each command mode restricts you to a subset of commands. If you are having trouble entering a command, check the prompt, and enter the question mark (?) for a list of available commands. You might be in the wrong command mode or be using the wrong syntax.

In the following example, notice how the prompt changes after each command, to indicate a new command mode for the voice gateway:

```
Router> enable
Password: <enable password>
Router# configure terminal
Router(config)# interface gigabitEthernet 0/0/0
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

The last message is normal and does not indicate an error. Press **Return** to get the Router# prompt.

Note Press **Ctrl-Z** in any mode to immediately return to enable mode (Router#) instead of entering **exit**, which returns you to the previous mode.

Undoing a Command or Feature

If you want to undo a command you entered or disable a feature, enter the keyword **no** before most commands.

For example, **no ip routing**.

Saving Configuration Changes

To prevent the loss of your device configuration, save the configuration changes to NVRAM.

Step 1 Router> enable

Example:

```
Password: password
```

Example:

```
Router#
```

Enables the privileged EXEC mode. Enter your password, if prompted.

Step 2 Router# copy running-config startup-config

Saves the configuration changes to NVRAM so that the changes are not lost during resets, power cycles, or power outages.

Step 3 Router(config-if)# Ctrl-z

Example:

```
Router#
```

Example:

```
%SYS-5-CONFIG_I: Configured from console by console
```

Returns to the user EXEC mode.

Upgrading to a New Cisco IOS Release

To install or upgrade to a new Cisco IOS release, see How to Update or Upgrade Cisco IOS Software .

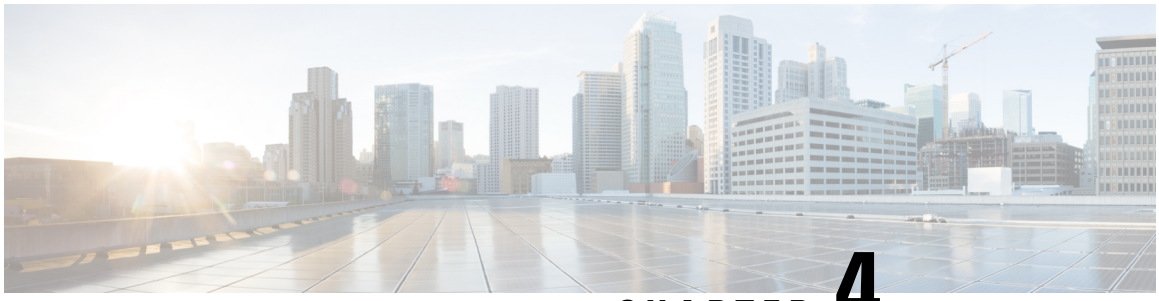
Note For Cisco VG410 Voice Gateway, the DSP container will be automatically upgraded when you upgrade the Cisco IOS-XE image.

Where to Go Next

Now that you have learned some Cisco IOS software basics, you can begin to configure the router using the CLI. However, before you begin, here are a few useful tips.

- Use the question mark (?) and arrow keys to help you enter commands.
- Each command mode restricts you to a set of commands. If you have difficulty entering a command, check the prompt and then enter the question mark (**?**) for a list of available commands. You might be in the wrong command mode or you might be using the wrong syntax.
- To disable a feature, enter the keyword **no** before the command. For example, **no ip routing**.
- Save your configuration changes to NVRAM so that the changes are not lost if there is a system reload or power outage.

To begin the configuration of the router, proceed to the *Configuring the Host Name and Password* section in this guide.



CHAPTER 4

Installing the Software Using install Commands

From Cisco IOS XE Cupertino 17.9.1a, Cisco Voice Gateways VG400, VG420, and VG450 are shipped in install mode by default. From Cisco IOS XE 17.12.1a, Cisco Voice Gateway VG410 is also shipped in the install mode. You can boot the platform, and upgrade or downgrade to Cisco IOS XE software versions using a set of **install** commands that are detailed in the following sections.

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

From Cisco IOS XE Cupertino 17.9.1a release, for devices shipped in install mode, a set of **install** commands can be used for starting, upgrading and downgrading of platforms in install mode. This update is applicable to the Cisco Voice Gateway 400 Series.

The following table describes the differences between Bundle mode and Install mode:

Table 2: Bundle Mode vs Install Mode

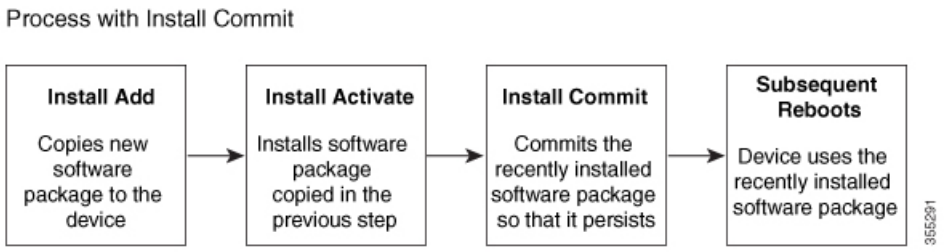
Bundle Mode	Install Mode
This mode provides a consolidated boot process, using local (hard disk, flash) or remote (TFTP) .bin image.	This mode uses the local (bootflash) packages.conf file for the boot process.
This mode uses a single .bin file.	.bin file is replaced with expanded .pkg files in this mode.

Bundle Mode	Install 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.

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 3: List of install Commands

Command	Syntax	Purpose
install add	**install add file** *location:filename.bin*	Copies the contents of the image and the package 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.

The following show commands are also available:

Table 4: 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.
show install active	**show install active**	Provides information about the active packages.
show install inactive	**show install inactive**	Provides information about the inactive packages, if any.
show install committed	**show install committed**	Provides information about the committed packages.
show install uncommitted	**show install uncommitted**	Provides information about uncommitted packages, if any.
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.
- 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

	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:
	See the following examples:
	• **VG400**:
	`Device#install add file bootflash:vg400-universalk9.BLD_V179_THROTTLE_LATEST_20220428_010838_V17_9_0_23.SPA.bin activate commit`
	• **VG410**:
	`Device# install add file bootflash:vg4x0-universalk9.17.12.01a.SPA.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:**	Exits privileged EXEC mode and returns to user EXEC mode.

	Command or Action	Purpose
	`Device# exit`	

Three-Step Installation

Note

- All the CLI actions (for example, add, activate, and so on) are executed.
- 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

	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:** See the following examples: • **VG400**: `Device#install add file bootflash:vg400-universalk9.BLD_V179_THROTTLE_LATEST_20220428_010838_V17_9_0_23.SSA.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.

	Command or Action	Purpose
	• **VG410**: `Device#install add file bootflash:vg410-universalk9.BLD_V172_THROTTLE_LATEST_20230828_043130_V17_12_1_1.SSA.bin`	
Step 3	**show install summary** **Example:** `Device#show install summary`	(Optional) Provides an overview of the image versions and their corresponding install state.
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 the 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.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Device#exit`	Exits privileged EXEC mode and returns to the 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:

```
install-vg400# install add file
bootflash:vg400-universalk9.BLD_V179_THROTTLE_LATEST_20220428_010838_V17_9_0_23.SSA.bin
activate commit
```

```
*May 11 23:45:54.588: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install
add_activate_commit
bootflash:vg400-universalk9.BLD_V179_THROTTLE_LATEST_20220428_010838_V17_9_0_23.SSA.bininstall_add_activate_commit:
 START Wed May 11 23:45:54 UTC 2022
install_add: Adding IMG
--- Starting initial file syncing ---
Copying
bootflash:vg400-universalk9.BLD_V179_THROTTLE_LATEST_20220428_010838_V17_9_0_23.SSA.bin
from  R0 to  R0
Info: Finished copying to the selected
Finished initial file syncing

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

Image added. Version: 17.09.01.0.5

install_activate: Activating IMG
Following packages shall be activated:
/bootflash/vg400-firmware_sm_dsp_sp2700.BLD_V179_THROTTLE_LATEST_20220428_010838_V17_9_0_23.SSA.pkg
/bootflash/vg400-mono-universalk9.BLD_V179_THROTTLE_LATEST_20220428_010838_V17_9_0_23.SSA.pkg
/bootflash/vg400-rpboot.BLD_V179_THROTTLE_LATEST_20220428_010838_V17_9_0_23.SSA.pkg

This operation may require a reload of the system. Do you want to proceed? [y/n]y


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

*May 11 23:47:07.393: %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

--- 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 Wed May 11 23:47:53 UTC 2022

install-vg400#
*May 11 23:47:53.019: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_mgr: Completed install
 add_activate_commitMay 11 23:4350: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is
exiting: reload action requested

Initializing Hardware ...

   :
Press RETURN to get started!
```

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

```
install-vg400# install add
bootflash:vg400-universalk9_npe.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.bin

*May 12 00:11:54.785: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install add

bootflash:vg400-universalk9_npe.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.bininstall_add:
 START Thu May 12 00:11:54 UTC 2022
install_add: Adding IMG
--- Starting initial file syncing ---
Copying
bootflash:vg400-universalk9_npe.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.bin
from  R0 to  R0
Info: Finished copying to the selected
Finished initial file syncing

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

Image added. Version: 17.09.01.0.158205

SUCCESS: install_add
/bootflash/vg400-universalk9_npe.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.bin
Thu May 12 00:12:26 UTC 2022

install-vg400#
*May 12 00:12:26.874: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_mgr: Completed install
 add bootflash:/vg400-universalk9_npe.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.bin
install-vg400#


install-vg400# install activate

*May 12 00:14:37.594: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install
activate NONEinstall_activate: START Thu May 12 00:14:37 UTC 2022
install_activate: Activating IMG
Following packages shall be activated:
/bootflash/vg400-firmware_sm_dsp_sp2700.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.pkg
/bootflash/vg400-mono-universalk9_npe.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.pkg
/bootflash/vg400-rpboot.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.pkg

This operation may require a reload of the system. Do you want to proceed? [y/n]y

--- Starting Activate ---
Performing Activate on all members

*May 12 00:18:06.168: %INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: R0/0: rollback_timer:
Install auto abort timer will expire in 7200 seconds [1] Activate package(s) on  R0
 [1] Finished Activate on  R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate

SUCCESS: install_activate Thu May 12 00:18:27 UTC 2022

install-vg400#
*May 12 00:18:27.511: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_mgr: Completed install
 activateMay 12 00:18:36.881: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting:
reload action requested
```

```
Initializing Hardware ...
     :
     :

Press RETURN to get started!

install-vg400>



install-vg400# install commit

*May 12 01:20:23.889: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install
commitinstall_commit: START Thu May 12 01:20:23 UTC 2022
--- Starting Commit ---
Performing Commit on all members
 [1] Commit packages(s) on  R0
 [1] Finished Commit packages(s) on  R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit operation

SUCCESS: install_commit Thu May 12 01:20:31 UTC 2022

install-vg400#
*May 12 01:20:31.351: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_mgr: Completed install
 commit
```

The following is an example of downgrading in install mode:

```
install-vg400# install add file bootflash:vg400-universalk9.17.08.01a.SPA.bin activate
commit

*May 12 02:13:24.633: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install
add_activate_commit bootflash:vg400-universalk9.17.08.01a.SPA.bininstall_add_activate_commit:
 START Thu May 12 02:13:24 UTC 2022
install_add: Adding IMG
--- Starting initial file syncing ---
Copying bootflash:vg400-universalk9.17.08.01a.SPA.bin from  R0 to  R0
Info: Finished copying to the selected
Finished initial file syncing

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

Image added. Version: 17.08.01.0.1526

install_activate: Activating IMG
Following packages shall be activated:
/bootflash/vg400-firmware_sm_dsp_sp2700.17.08.01a.SPA.pkg
/bootflash/vg400-mono-universalk9.17.08.01a.SPA.pkg
/bootflash/vg400-rpboot.17.08.01a.SPA.pkg

This operation may require a reload of the system. Do you want to proceed? [y/n]y

--- Starting Activate ---
Performing Activate on all members
```

```
 [1] Activate package(s) on  R0

*May 12 02:17:10.699: %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

--- 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 Thu May 12 02:17:55 UTC 2022

install-vg400#
*May 12 02:17:55.312: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_mgr: Completed install
 add_activate_commitMay 12 02:18:08.796: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is
 exiting: reload action requested



Initializing Hardware ...
     :
     :
Press RETURN to get started!


install-vg400# show version
Cisco IOS XE Software, Version 17.08.01a
Cisco IOS Software [Cupertino], ISR Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version
17.8.1a, RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2022 by Cisco Systems, Inc.
Compiled Wed 20-Apr-22 13:16 by mcpre


Cisco IOS-XE software, Copyright (c) 2005-2022 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: 16.12(2r)

install-vg400 uptime is 1 minute
Uptime for this control processor is 4 minutes
System returned to ROM by Install
System image file is "bootflash:packages.conf"
Last reload reason: Install



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.



Suite License Information for Module:'esg'

--------------------------------------------------------------------------------
Suite                 Suite Current         Type           Suite Next reboot
--------------------------------------------------------------------------------

Technology Package License Information:

------------------------------------------------------------------
Technology    Technology-package           Technology-package
              Current       Type           Next reboot
------------------------------------------------------------------
uck9             uck9             Smart License    uck9
securityk9       None             Smart License    None
ipbase           ipbasek9         Smart License    ipbasek9

The current throughput level is 35000 kbps


Smart Licensing Status: Smart Licensing Using Policy

cisco VG400-8FXS (1RU) processor with 1654554K/3071K bytes of memory.
Processor board ID FGL2517L2XS
Router operating mode: Autonomous
2 Gigabit Ethernet interfaces
8 Voice FXS interfaces
32768K bytes of non-volatile configuration memory.
4194304K bytes of physical memory.
6598655K bytes of flash memory at bootflash:.

Configuration register is 0x2102

install-vg400#
```

The following is an example of terminating a software installation:

```
install-vg400# install abort
install_abort: START Tue May 03 18:31:20 UTC 2022

This operation may require a reload of the system. Do you want to proceed? [y/n]y

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

```
SUCCESS: install_abort Tue May 03 18:32:43 UTC 2022
install-vg400#May  3 18:32:48.735: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting:
 reload action requested

Initializing Hardware ...
  :
  :
  Press RETURN to get started!

install-vg400>
```

The following are sample outputs for show commands:

show install log

```
install-vg400# show install log
[0|install_op_boot]: START Thu May 12 06:22:15 Universal 2022
[0|install_op_boot]: END SUCCESS  Thu May 12 06:22:17 Universal 2022
```

show install summary

```
install-vg400# 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    17.09.01.0.5

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

show install package *filesystem: filename*

```
install-vg400# show install package
bootflash:vg400-universalk9.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.bin
  Package: vg400-universalk9.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.bin
    Size: 648938943
    Timestamp:
  Canonical path:
/bootflash/vg400-universalk9.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.bin

    Raw disk-file SHA1sum:
      80700b261910c44785f46cac327b3aa81ed42edb
  Header size:     1152 bytes
  Package type:    30000
  Package flags:   0
  Header version:  3

  Internal package information:
    Name: rp_super
    BuildTime: 2022-04-26_20.04
    ReleaseDate: 2022-04-27_02.02
    BootArchitecture: i686
    RouteProcessor: goldbeach
    Platform: VG400
    User: mcpre
    PackageName: universalk9
    Build: BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6
    CardTypes:

  Package is bootable from media and tftp.
```

```
  Package contents:

  Package: vg400-mono-universalk9.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.pkg
    Size: 606901316
    Timestamp:

    Raw disk-file SHA1sum:
      53642fa806fa46a262aa247118272e49b48f14c0
    Header size:     1092 bytes
    Package type:    30000
    Package flags:   0
    Header version:  3

    Internal package information:
      Name: mono
      BuildTime: 2022-04-26_20.04
      ReleaseDate: 2022-04-27_02.02
      BootArchitecture: i686
      RouteProcessor: goldbeach
      Platform: VG400
      User: mcpre
      PackageName: mono-universalk9
      Build: BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6
      CardTypes:

    Package is bootable from media and tftp.
    Package contents:

  Package:
vg400-firmware_sm_dsp_sp2700.BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6.SSA.pkg
    Size: 2094140
    Timestamp:

    Raw disk-file SHA1sum:
      3cc7413e84187ee831a8b92fde7516ccff8f68b2
    Header size:     1084 bytes
    Package type:    40000
    Package flags:   0
    Header version:  3

    Internal package information:
      Name: firmware_sm_dsp_sp2700
      BuildTime: 2022-04-26_20.04
      ReleaseDate: 2022-04-27_02.02
      BootArchitecture: none
      RouteProcessor: goldbeach
      Platform: VG400
      User: mcpre
      PackageName: firmware_sm_dsp_sp2700
      Build: BLD_POLARIS_DEV_LATEST_20220427_001035_V17_9_0_6
      CardTypes:

    Package is not bootable.
```

show install active

```
install-vg400# 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    17.09.01.0.5
```

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

show install inactive

```
install-vg400# 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

```
install-vg400# 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    17.09.01.0.5

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

show install uncommitted

```
install-vg400# 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.

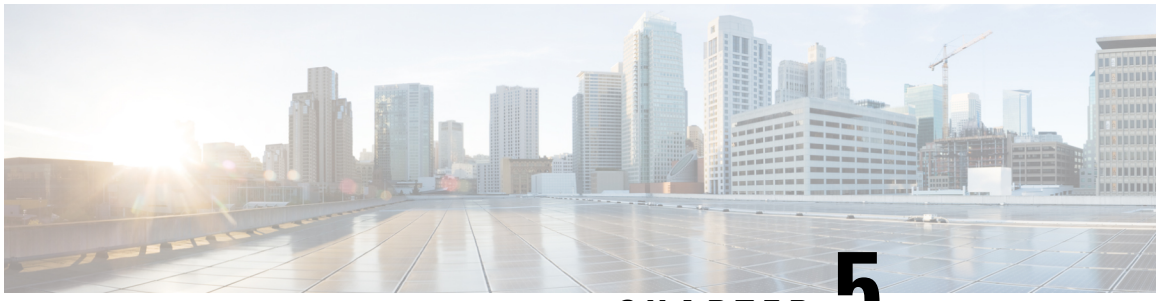
- **show install summary**
- **show install log**
- **show version**
- **show version running**

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 5

Configuring the Cisco VG420 Voice Gateway

This chapter describes how to use the Cisco IOS software CLI to configure basic analog functionalities. Follow the procedures in this chapter to configure the Cisco VG420 Voice Gateway, or if you want to change the configuration after you have run the setup command facility.

This chapter does not describe every configuration possible—only a small portion of the most commonly used configuration procedures. For advanced configuration topics, refer to the respective technology configuration guides.

One of the first configuration tasks you might want to do is to configure the host name and set an encrypted password. Configuring a host name allows you to distinguish a router from another. Setting an encrypted password allows you to prevent unauthorized configuration changes.

Configuring Host Name and Password

Procedure

	Command or Action	Purpose

Step 1	Router> enable
	Example:
	`Password: password`
	Example:
	`Router#`
		Enables privileged EXEC mode. Enter your password, if prompted.

Step 2	Router# configure terminal **Example:** `Router(config)#`	Enters the global configuration mode. Enter configuration commands, one per line. End with CNTL/Z.

	Command or Action	Purpose
Step 3	Router(config)# hostname 420	Changes the name of the Cisco VG420 Voice Gateway to a meaningful name. Substitutes the host name to Router.
Step 4	Router(config)# enable secret guessme	Enters an enable secret password. This password provides access to privileged EXEC mode. When you enter enable at the user EXEC prompt (Router>), you must enter the enable secret password to gain access to configuration mode. Substitute your enable secret password for guessme.
Step 5	Router(config)# line con 0	Enters line configuration mode to configure the console port.
Step 6	Router(config-line)# exec-timeout 0 0	Prevents the Cisco VG420 Voice Gateway, EXEC mode from timing out when you do not enter any information on the console screen for an extended period.
Step 7	Router(config-line)# exit	Exits from the config-line mode and enters into the global configuration mode.

Verifying the Host Name and Password

To verify that you configured the correct host name and password, perform the following steps:

Step 1 Enter the **show config** command.

Example:

```
Router# show config
Using 2745 out of 262136 bytes
!
version XX.X
.
.
.
!
hostname 420
!
enable secret 5 $1$60L4$X2JYOwoDc0.kqa1loO/w8/
.
.
.
```

Check the host name and encrypted password displayed near the top of the command output.

Step 2 Exit the Global Configuration mode and attempt to re-enter it using the new enable password:

Example:

```
Router# exit
.
.
.
Router con0 is now available
```

```
Press RETURN
 to get started.
Router> enable
Password: guessme
Router#
```

If you face any issues, check whether:

- The caps lock is off.
- You entered the correct password. Passwords are case sensitive.

TLS 1.2 support on SCCP Gateways

The TLS 1.2 support on SCCP Gateways feature details the configuration of TLS 1.2 on SCCP protocol for digital signal processor (DSP) farm including Unicast conference bridge

(CFB), Media Termination Point (MTP), and SCCP telephony control (STC) application (STCAPP).

DSP on gateways can be used as media resources for transrating or transcoding. Each media resource uses Secure Skinny Client Control Protocol (SCCP) to communicate with Cisco Unified Communications Manager. Currently SSL 3.1, which is equivalent to TLS1.0, is used for sending secure signals. This feature enhances the support to TLS 1.2. From Cisco IOS XE Cupertino 17.7.1a, TLS 1.2 is enhanced to support the Next-Generation Encryption (NGE) cipher suites.

Note Cisco Unified Communications Manager (CUCM) Version 14SU2 has been enhanced to support Secured SCCP gateways with the Subject Name field (CN Name) with or without colons, for example, AA:22:BB:44:55 or AA22BB4455.

CUCM checks the CN field of the incoming certificate from the SCCP Gateway and verifies it against the DeviceName configured in CUCM for this gateway. DeviceName contains MAC address of the gateway. CUCM converts the MAC address in the DeviceName to MAC address with colons (for example: AA:22:BB:44:55) and validates with the CN name in the Gateway's certificate. Therefore, CUCM mandates Gateway to use MAC address with colons for the CN field in the certificate, that is, subject name.

Due to new guidelines from Defense Information Systems Agency (DISA), it is a requirement not to use colons for the subject name field CN. For example, AA22BB4455.

SCCP TLS connection

CiscoSSL is based on OpenSSL. SCCP uses CiscoSSL to secure the communication signals.

If a resource is configured in the secure mode, the SCCP application initiates a process to complete Transport Layer Security (TLS) handshaking. During the handshake, the server sends information to CiscoSSL about the TLS version and cipher suites supported. Previously, only SSL3.1 was supported for SCCP secure signalling. SSL3.1 is equivalent to TLS 1.0. The TLS 1.2 Support feature introduces TLS1.2 support to SCCP secure signalling.

After TLS handshaking is complete, SCCP is notified and SCCP kills the process.

If the handshaking is completed successfully, a REGISTER message is sent to Cisco Unified Communications Manager through the secure tunnel. If handshaking fails and a retry is needed, a new process is initiated.

Note For SCCP-based signalling, only TLS_RSA_WITH_AES_128_CBC_SHA cipher suite is supported.

Cipher Suites

For SCCP-based signaling, TLS_RSA_WITH_AES_128_CBC_SHA cipher suite is supported.

From Cisco IOS XE Cupertino 17.7.1a, the following NGE cipher suites are also supported:

- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-RSA-AES256-GCM-SHA384

These cipher suites enable secure voice signaling for both STCAPP analog phone and SCCP DSPFarm conferencing service. The cipher suite selection is negotiated between GW and CUCM.

The following prerequisites are applicable for using NGE cipher suites:

- Configure TLS 1.2. For more information, see *Configuring TLS*.
- Use the CUCM Release 14.1 SU1 or later, and Voice Gateways or platforms that support TLS 1.2.
- From CUCM Web UI, navigate to Cipher Management and set the CIPHER switch as NGE. For more information, Cipher Management.

For more information about verifying these cipher suites, see *Verifying TLS version and Cipher Suites*.

For the SRTP encrypted media, you can use higher-grade cipher suites: AEAD-AES-128-GCM or AEAD-AES-256-GCM. These cipher suites selection is automatically negotiated between GW and CUCM for both secure analog voice and hardware conference bridge voice media. Authenticated Encryption with Associated Data (AEAD) ciphers simultaneously provide confidentiality, integrity, and authenticity, without built-in SHA algorithms to validate message integrity.

Supported Platforms

The TLS 1.2 support on SCCP Gateways feature is supported on the following platforms:

- Cisco VG400, VG420, and VG450 Analog Voice Gateways

Configuring TLS version for STC application

Perform the following task to configure a TLS version for the STC application:

```
enable
configure terminal
stcapp security tls-version v1.2
exit
```

Note The stcapp security tls command sets the TLS version to v.1.0, v1.1, or v1.2 only. If not configured explicitly, TLS v1.0 is selected by default.

Configuring TLS version in Secure Mode for DSP Farm Profile

Perform the following task to configure the TLS version in secure mode for DSP farm profile:

```
enable
configure terminal
dspfarm profile 7 conference security
  tls-version v1.2
  exit
```

Note Note: The **tls** command can be configured only in security mode.

Verifying TLS version and Cipher Suites

Perform the following task to verify the TLS version and cipher suite:

```
# show dspfarm profile 100
Dspfarm Profile Configuration

 Profile ID = 100, Service = CONFERENCING, Resource ID = 2
 Profile Service Mode : secure
 Trustpoint : Overlord_DSPFarm_GW
 TLS Version  : v1.2
 TLS Cipher   : ECDHE-RSA-AES256-GCM-SHA384
 Profile Admin State : UP
 Profile Operation State : ACTIVE
 Application : SCCP   Status : ASSOCIATED
 Resource Provider : FLEX_DSPRM   Status : UP
 Total Number of Resources Configured : 10
 Total Number of Resources Available : 10
 Total Number of Resources Out of Service : 0
 Total Number of Resources Active : 0
 Maximum conference participants : 8
 Codec Configuration: num_of_codecs:6
 Codec : g711ulaw, Maximum Packetization Period : 30 , Transcoder: Not Required
 Codec : g711alaw, Maximum Packetization Period : 30 , Transcoder: Not Required
 Codec : g729ar8, Maximum Packetization Period : 60 , Transcoder: Not Required
 Codec : g729abr8, Maximum Packetization Period : 60 , Transcoder: Not Required
 Codec : g729r8, Maximum Packetization Period : 60 , Transcoder: Not Required
 Codec : g729br8, Maximum Packetization Period : 60 , Transcoder: Not Required
```

Verifying STCAPP Application TLS version

Perform the following tasks to verify TLS version of the STCAPP application:

```
Device# show call application voice stcapp
App Status: Active
CCM Status: UP
CCM Group: 120
Registration Mode: CCM
Total Devices: 0
Total Calls in Progress: 0
Total Call Legs in Use: 0
ROH Timeout: 45
TLS Version: v1.2

# show stcapp dev voice 0/1/0
Port Identifier:  0/1/0
Device Type:      ALG
Device Id:        585
```

```
Device Name:      ANB3176C85F0080
Device Security Mode : Encrypted
  TLS version        : TLS version 1.2
  TLS cipher         : ECDHE-RSA-AES256-GCM-SHA384
Modem Capability: None
Device State:     IS
Diagnostic:       None
Directory Number: 80010
Dial Peer(s):     100
Dialtone after remote onhook feature: activated
Busytone after remote onhook feature: not activated
Last Event:       STCAPP_CC_EV_CALL_MODIFY_DONE
Line State:       ACTIVE
Line Mode:        CALL_CONF
Hook State:       OFFHOOK
mwi:              DISABLE
vmwi:             OFF
mwi config:       Both
Privacy:          Not configured
HG Status:        Unknown
PLAR:             DISABLE
Callback State:   DISABLED
CWT Repetition Interval: 0 second(s) (no repetition)
Number of CCBs:   1
Global call info:
    Total CCB count      = 3
    Total call leg count = 6

Call State for Connection 2 (ACTIVE): TsConnected
Connected Call Info:
   Call Reference: 33535871
   Call ID (DSP):  187
   Local IP Addr:  172.19.155.8
   Local IP Port:  8234
   Remote IP Addr: 172.19.155.61
   Remote IP Port: 8154
   Calling Number: 80010
   Called Number:
   Codec:          g711ulaw
   SRTP:           on
   RX Cipher:      AEAD_AES_256_GCM
   TX Cipher:      AEAD_AES_256_GCM
```

Perform the following task to verify the sRTP cipher suite for the DSPfarm connection.

```
# show sccp connection detail

bridge-info(bid, cid) - Normal bridge information(Bridge id, Calleg id)
mmbridge-info(bid, cid) - Mixed mode bridge information(Bridge id, Calleg id)

sess_id    conn_id    call-id    codec   pkt-period dtmf_method    type
bridge-info(bid, cid)   mmbridge-info(bid, cid) srtp_cryptosuite          dscp
                        call_ref   spid       conn_id_tx

16778224   -          125        N/A     N/A        rfc2833_pthru     confmsp   All RTPSPI
 Callegs      All MM-MSP Callegs      N/A                        N/A
                        -          -          -

16778224   16777232   126        g711u   20         rfc2833_pthru  s- rtpspi    (101,125)
               N/A                       AEAD_AES_256_GCM          184
                        30751576   16777219   -

16778224   16777231   124        g711u   20         rfc2833_pthru  s- rtpspi    (100,125)
               N/A                       AEAD_AES_256_GCM          184
                        30751576   16777219   -
```

```
Total number of active session(s) 1, connection(s) 2, and callegs 3
```

Verifying Call Information

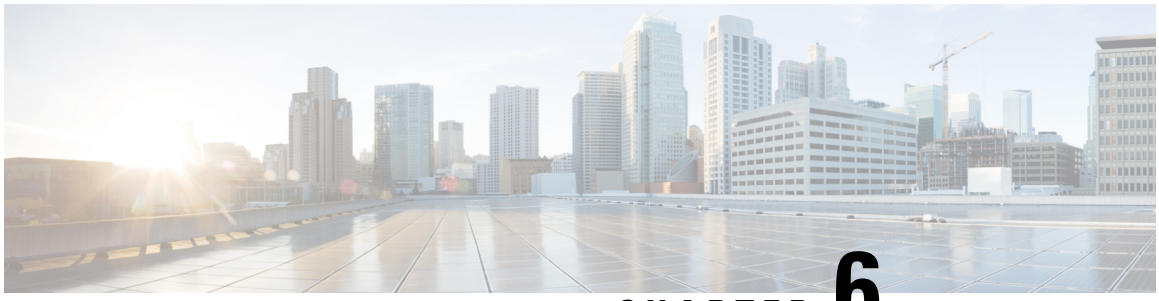
To display call information for TDM and IVR calls stored in the Forwarding Plane Interface (FPI), use the **showvoipfpi calls** command. You can select a call ID and verify the cipher suite using the command **show voip fpi calls confID** *call_id_number*. In this example, cipher suite 6 is AES_256_GCM.

```
#show voip fpi calls
Number of Calls : 2
---------- ---------- ---------- ---------- --------------- ---------------
    confID correlator    AcallID     BcallID           state           event
---------- ---------- ---------- ---------- --------------- ---------------
         1          1         87         88       ALLOCATED DETAIL_STAT_RSP
        21         21         89         90       ALLOCATED DETAIL_STAT_RSP

#show voip fpi calls confID 1
--------------------------------------------------------------------------------
VoIP-FPI call entry details:
--------------------------------------------------------------------------------
Call Type        :           TDM_IP     confID          :                1
correlator       :                1     call_state      :        ALLOCATED
last_event       :  DETAIL_STAT_RSP     alloc_start_time :      1796860810
modify_start_time:                0     delete_start_time:               0
Media Type(SideA):             SRTP     cipher suite    :               6
--------------------------------------------------------------------------------
FPI State Machine Stats:
------------------------
create_req_call_entry_inserted             :         1
.........
```

Table 5: Feature Information for TLS 1.2 support on SCCP Gateways

Feature Name	Releases	Feature Information
Support for NGE Cipher Suites	Cisco IOS XE Cupertino 17.7.1a	This feature supports NGE cipher suites for secure voice signaling and secure media. These cipher suites are applicable for both STCAPP analog phone and SCCP DSPFarm conferencing service.



CHAPTER 6

Configuring the Voice Ports

Voice Ports in Cisco VG420 Voice Gateway

The Cisco VG420 Voice Gateway supports the following SKUs:

- VG420-144FXS: This has144 analog FXS ports and no FXO port
- VG420-132FXS/6FXO: This has 132 analog FXS ports and 6 FXO ports
- VG420-84FXS/6FXO: This has 84 analog FXS ports and 6 FXO ports

Signaling Types for the Analog Ports

- FXS Ports: This voice port supports loop start, ground start, and DID signaling types.
- FXO Ports: This voice port supports loop start and ground start signaling types.

SKU Information

See the following table for information on the voice ports that are supported on these SKUs

SKUs	VG420-144FXS	VG420-132FXS/6FXO	VG420-84FXS/6FXO
FXS Ports	144	132	84
FXO Ports	0	6	6
Number of Failed Over Ports	N/A	6	6
DID and long loop ports	108, 1/0/0 to 1/0/107	108, 1/0/0/ to 1/0/107	84, 1/0/0 to 1/0/83
Maximum REN	80	80	80
RJ21 Connectors	6	6	4

Fail Over Port Mapping

To view the fail over port mapping for Cisco VG420 Voice Gateway, see the following sample output of the show voice port summary:

```
VG420-132FXS/6FXO: provide 6 power fail-over ports:
PWR FAILOVER PORT        PSTN FAILOVER PORT
==================       ==================
1/0/126                                      FXO BYPASS 1/0/132
1/0/127                                      FXO BYPASS 1/0/133
1/0/128                                      FXO BYPASS 1/0/134
1/0/129                                      FXO BYPASS 1/0/135
1/0/130                                      FXO BYPASS 1/0/136
1/0/131                                      FXO BYPASS 1/0/137

VG420-84FXS/6FXO: provide 6 power fail-over ports:
PWR FAILOVER PORT        PSTN FAILOVER PORT
==================       ==================
1/0/78                                     FXO BYPASS 1/0/84
1/0/79                                     FXO BYPASS 1/0/85
1/0/80                                     FXO BYPASS 0/1/86
1/0/81                                     FXO BYPASS 1/0/87
1/0/82                                     FXO BYPASS 0/1/88
1/0/83                                     FXO BYPASS 1/0/89
```

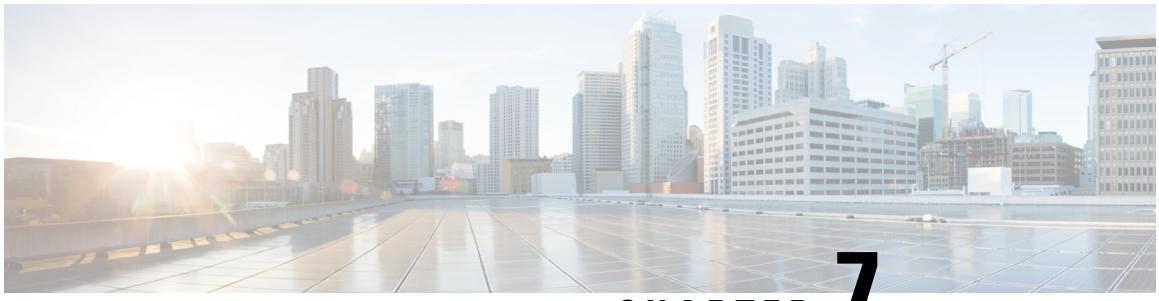
Configuring the Voice Ports

Cisco VG420 Voice Gateway supports ds0-group and E&M voice ports on existing voice NIMs (NIM slot 0). To know how to configure these digital ds0-group port, E&M, FXS and FXO voice ports, see the Voice Port Configuration Guide.

To view detailed information about voice port configuration, see the Cisco IOS Voice Configuration Library.

Note It is recommended that you configure the interdigit timeout value for the voice ports. To configure this value for a specified voice port, use the **timeouts interdigit <seconds>** command in the voice-port configuration mode. If you do not configure this value, by default, the value is 10 seconds.

To learn more about this command, see the Cisco IOS Voice Command Reference Guide.



CHAPTER 7

Configuring Support for 4096 Key Pair

RSA-4096 is an encryption system that offers enhanced security to protect your data during transmission. From Cisco IOS XE 17.14.1a, Cisco VG400 Voice Gateway and Cisco VG420 Voice Gateway support 4096 key pair with SHA256 hash function during a TLS handshake process.

Cisco VG420 Voice Gateway and Cisco VG400 Voice Gateways, via Signaling Connection Control Protocol (SCCP), use TLS to secure the signaling channel to CUCM and SRST. As a default, Cisco IOS XE devices support 2048 RSA key encryption for TLS handshake process. For enhanced security and protection during data transmission, you can enable 4096 key pair with SHA256 hash function for these two voice gateways.

Note Currently, Cisco VG400 Voice Gateway supports up to 8 FXS ports and Cisco VG420 Voice Gateway supports up to 144 ports for the TLS handshake process.

Configuring 4096 Key Pair Support

During a TLS session, certificate authentication and key exchange are critical. During certificate authentication, the client verifies the server's digital certificate to ensure it is valid and whether it is issued by a trusted Certificate Authority (CA). This step confirms the server's identity. Key exchange is then established, where the client and server negotiate and agree upon keys that will be used for encryption and decryption of data during the TLS session.

During a TLS session, all the STCAPP-based FXS ports of the voice gateways are enabled for a short period of time, for example, during a shut/no shut or boot up period. When you configure the 4096 key pair, these FXS ports securely interface with the CUCM.

To configure 4096 key pair, perform the following steps:

1. Create a new key pair.
2. Associate a trustpoint with this keypair.
3. Authenticate the certificate by a CA Server.

For detailed information on each of these steps, see Configuring and Managing a Certificate Server.

Create a New Key Pair

The following is the sample configuration to create a new RSA-4096 key pair.

```
vg400# crypto key generate rsa exportable general-keys label 4k_keypair modulus 4096
The name for the keys will be: 4k_keypair
% The key modulus size is 4096 bits
% Generating crypto RSA keys in background ...
```

Create a Trustpoint and Associate with the Key Pair

The following is the sample configuration to create a trustpoint and associate it with the keypair you've already created. The enrollment URL is the CA server whose keypair was created in the above-mentioned sample configuration.

```
vg400(config)# crypto pki trustpoint 4k_keypair
vg400(ca-trustpoint)# enrollment url http://10.75.167.250:80
vg400(ca-trustpoint)# serial-number none
vg400(ca-trustpoint)# fqdn none
vg400(ca-trustpoint)# ip-address none
vg400(ca-trustpoint)# subject-name cn=6c:03:09:ac:f9:80
vg400(ca-trustpoint)# revocation-check none
vg400(ca-trustpoint)# hash sha256
vg400(ca-trustpoint)# rsakeypair 4k_keypair
vg400(ca-trustpoint)# end
```

Authenticate the Certificate by a CA Server

The following is a sample configuration to authenticate the CA certificate.

```
vg400(config)# crypto pki authenticate 4k_keypair
Certificate has the following attributes:
      Fingerprint MD5: 7BD20233 A5078333 8CC8C7C5 DAE614EC
      Fingerprint SHA1: B452068C CD39D071 046EEED6 5B0424F6 2439D5BE
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
vg400(config)#crypto pki enroll 4k_keypair
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
 For security reasons your password will not be saved in the configuration.
 Please make a note of it.
Password:
Re-enter password:
% The subject name in the certificate will include: cn=6c:03:09:ac:f9:80
% The fully-qualified domain name will not be included in the certificate
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate verbose 4k_keypair' command will show the fingerprint.
```

Verifying Support for 4096 Key Pair

After configuring the 4096 key pair, use the following show commands to verify the configuration.

1. **show crypto key**

```
vg400# show crypto key mypubkey rsa 4k_keypair
% Key pair was generated at: 22:20:02 UTC Feb 19 2024
Key name: 4k_keypair
Key type: RSA KEYS      4096 bits
 Storage Device: private-config
 Usage: General Purpose Key
 Key is exportable. Redundancy enabled.
 Key Data:
  30820222 300D0609 2A864886 F70D0101 01050003 82020F00 3082020A 02820201
  00CF7270 D5B4A356 69B55B18 0CEA4927 6CEC9A48 0B191804 72D316CE 97A97BA8
  37F5690B B3A169B9 6E2A12F8 5595435A 8FA2AE7F BD7996ED 3C406EA4 266C9542
  A3ACE221 9943AB27 9B16397E FA438E7C D9D95689 B9F8F508 CC38CDD1 C7F56489
  0FB3A18C F90066A5 42AE9F36 A1FA29BD 7A70A7A9 A0C96C8B 543A6217 C1B9D751
  682BEF3A 84B82045 9C2A9C22 0D6005E8 3F30DA9B 67F1BE1F F366813F 65B8F2DC
  AC48E6B4 65DFCFCA F19C6522 F2C25089 D59706CB E03D0C87 5912A26E B4DCE624
  4FCE6D8A FA4333BE F7C40A0D F2CB0D5C 73A66587 664BF192 FEAB5EE1 86081679
  390CF73D 5A11E3C9 B43FCC50 B5478885 B938EBD9 09FDB453 779F8F3E 5E168BCA
  6F7F896B 6A0D941C E087E667 C9CE41CB 029FD40E 31099EBE F2BD5706 8C9E40BF
  1CD432AA 71A91FB4 12388ED9 4C8552F3 B5DFF37C D6EDA7E6 59DDC0EB FD24496D
  F3D1ED6B 6CD17100 502BFEE2 2AFC8707 E32329EF 41E6FBDB 0094D82D 044C78F0
  6F03583E 04D979B6 317B19EB FB1155A9 6F2D9F64 4C99D779 45FCD19F C767177F
  886C4D7F 4E9B39BB 16027A40 D99E1575 AA160CB4 E039BB6D 6C60DAA2 228C0C2D
  492D0BF9 5EF083DC F2ADE958 78F5361D 0502B89A C50D5FFF A4E57865 B8E872C6
  BE6D2630 78C94D06 3D23D46A 5B6E5AEA E9355ADB 7206CD5D A405B996 3834F5D6
  65CE5BEB FE4B6345 7984C4A9 2E302E37 055DC42A 325C6B80 C12820CC BE6233D5
  31020301 0001
```

The highlighted line in the above output displays the 4096 key type, implying a successful configuration.

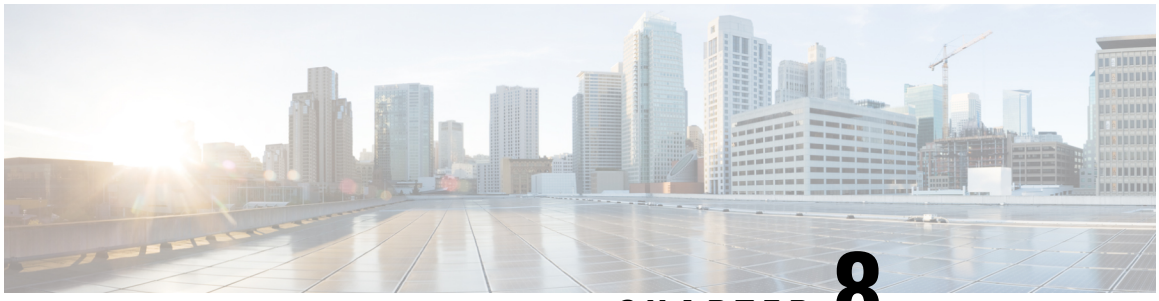
2. **show pki certificates**

```
vg400# show crypto pki certificates verbose 4k_keypair
Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 00DC
  Certificate Usage: General Purpose
  Issuer:
    cn=CA-4k.labtest.com
  Subject:
    Name: 6c:03:09:ac:f9:80
    cn=6c:03:09:ac:f9:80
  Validity Date:
    start date: 06:35:09 UTC Feb 20 2024
    end   date: 08:48:45 UTC Aug 12 2033
  Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (4096 bit)
  Signature Algorithm: SHA256 with RSA Encryption
  Fingerprint MD5: 5C2556ED 55BA2D9C 47CE2668 B1927DF7
  Fingerprint SHA1: 42CD404B 43BF7AD0 AD516F90 565087D4 04277F15
  X509v3 extensions:
    X509v3 Key Usage: A0000000
      Digital Signature
      Key Encipherment
    X509v3 Subject Key ID: 6B466F52 A518B5E7 902DAFA7 658D15FE 57B2E541
```

```
    X509v3 Authority Key ID: 4773C5C3 A4161090 7442B558 62713E6B 744857F6
    Authority Info Access:
  Cert install time: 22:35:39 UTC Feb 19 2024
  Associated Trustpoints: 4k_keypair
  Key Label: 4k_keypair
  Key storage device: private config

CA Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 01
  Certificate Usage: Signature
  Issuer:
    cn=CA-4k.labtest.com
  Subject:
    cn=CA-4k.labtest.com
  Validity Date:
    start date: 08:48:45 UTC Aug 15 2023
    end   date: 08:48:45 UTC Aug 12 2033
  Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (4096 bit)
  Signature Algorithm: SHA256 with RSA Encryption
  Fingerprint MD5: 7BD20233 A5078333 8CC8C7C5 DAE614EC
  Fingerprint SHA1: B452068C CD39D071 046EEED6 5B0424F6 2439D5BE
  X509v3 extensions:
    X509v3 Key Usage: 86000000
      Digital Signature
      Key Cert Sign
      CRL Signature
    X509v3 Subject Key ID: 4773C5C3 A4161090 7442B558 62713E6B 744857F6
    X509v3 Basic Constraints:
        CA: TRUE
    X509v3 Authority Key ID: 4773C5C3 A4161090 7442B558 62713E6B 744857F6
    Authority Info Access:
  Cert install time: 13:56:18 UTC Feb 6 2024
  Associated Trustpoints: 4k_keypair TP_IP6 4k
  Storage: nvram:CA-4klabtest#1CA.cer
```

The highlighted portions in the above-mentioned sample output indicate successful 4096 key pair configuration.



CHAPTER 8

Configuring the Supplementary Features

The following chapter explains how to configure voice gateway SIP Line Side features such as Directed Call Park, Call Pick Up, Call Transfer and so on. To provision these features, you must configure outbound VOIP Dial-peer, Pots Dial-peer, Voice Card, and SIP which is described in this chapter.

Configure FXS Ports for Supplementary Services

To handle supplementary services for Foreign Exchange Station (FXS) ports, the event handler handles the hookflash or onhook events. Additionally, the event handler also sends events to call control and triggers the supplementary service on SIP SPI. However, currently, FXS ports do not register on Cisco Unified Communications Manager (CUCM) as SIP endpoints. To ensure the FXS port are registered as a SIP endpoints, make sure that:

- Each configured FXS ports is registered to CUCM. The CUCM creates the database for proper call routing based on the registered endpoint.
- The SIP stack adds or modifies SIP headers content to a proper interface with CUCM and enables new features such as directed call retrieval, call pick-up, and so on.

The FXS ports for Supplementary Services supports CUCM verion 12.5 SU4 or later., and CUCM 14.0 SU1 or later.

Note You must use the **no local-bypass** command for all the media in this configuration.

Call Transfer

The call transfer status includes the following concepts:

- Hookflash: A hookflash is a brief interruption in the loop as the system places the active call on hold.
- On hook: This option completes the call transfer.

The following table describes the call transfer action.

Table 6: Supported Call Transfer Action

State	Action	Result	Response on FXS Line
Active call	Controller hookflash	Held call	Second dial tone
Held call and outgoing dialed, alerting, and active call	Controller on hook	Held call and active call transferred	Transfer

Three-Way Conference

A three-way conference call allows three people to participate in a single phone session. The following table describes the three-way conference action.

Table 7: Supported Three-Way Conference Action

State	Action	Result
Active Call	First party hookflash	Held call
First party held and second party active	Active call hookflash	First and second calls are bridged
Three-way conference	Controller on hook	Both call legs torn down
Three-way conference	First called party on hook	Call between controller and first called party terminated. Call between controller and second called party remains active.
Three-way conference	Second called party on hook	Call between controller and second called party terminated. Call between controller and first called party remains active.
Three-way conference	Controller hookflash	Call between controller and second called party terminated, call between controller and first called party remains.

Restrictions for Configuring the Supplementary Services

The following functionalities are not supported for this configuration:

- Only one line number per FXS port is supported, and shared line is not supported.
- The line side SIP endpoints are controlled by one CUCM only. Switch over and switch back are not supported.
- Non-DSAPP-controlled devices are not supported. You must configure 'service dsapp' before you configure pots dial-peer and voip dial-peer.

- You cannot combine IPv4 and IPv6 in this configuration.
- The SIP analog calls go through the CUCM, and hairpin calls are not supported.
- SIP analog ports signaling for failover between the CUCMs is not supported.
- 3-way conference only supports G711 codec.
- Media recording, overlap dialing, secure calls, failover, and fallback are not supported.
- In the CUCM web interface, you can add more than one line under **Phone Configuration**. However, only the first line will be associated with the phone, and the rest of the lines will not be applied

Configuring the Device Control Session Application

Procedure

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

Step 3	application global service default dsapp
	Example:
	`router(config)#application`
	`router(config-app)#global`
	`router(app-global)#service default dsapp`
		(Optional) Enables the new hookflash functionality globally. Device Control Session Application (DSAPP) application global service default drives these hookflash features and it must be configured for new bookflash functionality for an application framework module in IOS. DSAPP can be configured globally or on a dial-peer basis.
		Note This is a global configuration command. After you configure this command, all the calls are impacted. Even a FXO call will be controlled by DSAPP application which can lead to a failure. If the gateway is controlled by a DSAPP application, it is not recommended to make DSAPP as the default call controler.

Step 4	param dial-peer **`number`**
	Example:
	`router(config)#application`
	`router(config-app)#service dsapp`
	`router(app-global)#param dial-peer 100`
		If multiple dial-peer matches are made for the destination-pattern, dial-peer 100 command is used.
		Note When you configure DSAPP on a dial-peer basis,specify a VOIP dial-peer for any outbound call. If all outbound calls that use the hookflash functionality are on the same server, it is recommended to use the param dial-peer command.

	Command or Action	Purpose

Step 5	param callWaiting **string**
	Example:
	`router(config)#application`
	`router(config-app)#service dsapp`
	`router(app-global)#param dial-peer 100`
	`router(app-global)#param callWaiting TRUE`
		Enables the call waiting feature.

Step 6	param callConference **string**
	Example:
	`router(config)#application`
	`router(config-app)#service dsapp`
	`router(app-global)#param dial-peer 100`
	`router(app-global)#param callWaiting TRUE`
	`router(app-global)#param callConference TRUE`
		Enables the call conference feature.

Step 7	param callTransfer **string**
	Example:
	`router(config)#application`
	`router(config-app)#service dsapp`
	`router(app-global)#param dial-peer 100`
	`router(app-global)#param callWaiting TRUE`
	`router(app-global)#param callConference TRUE`
	`router(app-global)#param callTransfer TRUE`
		Enables the call transfer feature.

Configuring the Outbound Voip Dial-peer

Outbound dial-peer is configured like regular voip dial-peer for SIP. In addition to the parameters required, the following configurations are required:

- service dsapp: Specifies that the dial-peer is controlled by a DSAPP application
- session transport tcp: Specifies that only TCP signaling is supported
- voice-class sip extension gw-ana: Indicates that this parameter is used to interop with CUCM
- voice-class sip bind control source-interface GigabitEthernetx/y/z: Indicates that this interface's mac address is the base mac.
- dual tone multifrequency (DTMF): Specifies how a Session Initiation Protocol (SIP) gateway relays dual tone multifrequency (DTMF) tones between telephony interfaces and an IP network. This feature supports **rtp-nte** DTMF relay mechanisms for the SIP dial peers.

Here is a sample outbound voip dial-peer configuration:

```
dial-peer voice 714281111 voip
service dsapp
destination-pattern .+
session protocol sipv2
session target ipv4:172.16.0.0
incoming called-number 7141116...
voice-class sip bind control source-interface GigabitEthernet0/0/0
codec g711ulaw
```

Note G711 is the only codec supported for conference calls. Hence it is recommended that you add this codec for conference calls.

The following is a sample configuration for DTMF relay:

```
dtmf-relay method1 [...[method6]]
dtmf-relay rtp-nte
```

Configuring POTS Dial-peer

Plain Old Telephone Service (POTS) dial peers retain the characteristics of a traditional telephony network connection. POTS dial peers map a dialed string to a specific voice port on the local router, normally the voice port connecting the router to the local PSTN, PBX, or telephone.

You can configure the POTS dial-peer feature by using the **dial-peer voice** command. In addition to the parameters required, you can also configure the following commands under POTS dial-peer to interpret hookflash (HF) and to interop with CUCM:

- service dsapp: Specifies this dial-peer is to be controlled by the DSAPP application
- voice-class sip extension gw-ana: Indicates that this parameter is used to interop with CUCM

See the following sample configuration of the POTS dial-peer feature here:

```
dial-peer voice 19993000 pots
service dsapp
destination-pattern 2124506300
voice-class sip extension gw-ana
port 3/0/0
```

Configuring Voice-card and SIP

When you configure the voice-card, all the traffic should go through the CUCM. Hairpin calls are not supported. You have to execute the **no local-bypass** command for the voice-card that have FXS SIP endpoints.

For FXS SIP endpoints to register, configure the **registrar IP address** command under the sip-ua mode and use the TCP as the transport type. Note that UDP protocal is not supported.

```
!
voice-card 3/0
no local-bypass
no watchdog
!
!
sip-ua
registrar ipv4:172.16.0.0 expires 3600 tcp
protocol mode dual-stack
!
```

Enabling Device Control Session Application Line features

To register to CUCM as a SIP endpoint, and to distinguish line feature from trunk, you should configure the **dsapp line** command.

Procedure

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

Step 3	dsapp line
	Example:
	`router(config)#`
	`router(config)#dsapp line`
	`router(config)#`
		Specifies the format of each call feature.
		Note If you do not configure the **dsapp line** command, the gateway acts like a SIP trunk and the analog phones might not register as SIP endpoints. Further, you cannot configure the Feature Access Code (FAC). You must run the **dsapp line** command to use the SIP line features.

Configuring Feature Access Code

The **dsapp line feature access-code** command invokes the feature to translate the Feature Access Code (FAC) to the format that the CUCM understands. If you do not configure this command, the whole FAC digits are sent to the CUCM and may not invoke features. You can also change the default FAC in the sub-mode.

Analog phones do not have soft keys. The required supplementary service features are invoked through FAC. By default, the FAC has '**' prefix which can be changed using the CLI command.

```
router(config)#dsapp line feature access-code
router(config-dsappline-fac)#prefix *#
router(config-dsappline-fac)#cancel-call-waiting **4
router(config-dsappline-fac)#exit
router# show dsapp line feature codes
dsapp line feature access-code
prefix *#
call forward all *#1
call forward cancel *#2
pickup local *#5
pickup group *#7
pickup direct *#6
cancel-call-waiting **4
last-redial *#3
```

If you don't configure the **dsapp line feature access-code**, the voice gateway does not translate the FAC to the format that the CUCM understands. The whole FAC digits is sent to the CUCM.

After the FAC is disabled and re-enabled, all the FAC and prefix are rolled back to the default values.

```
router(config)#no dsapp line feature access-code
Feature access-code disabled
router(config)#do show dsapp line feature codes
dsappline feature access-code disabled
router(config)#dsapp line feature access-code
```

```
router(config-dsappline-fac)#do show dsapp line feature codes
dsapp line feature access-code
prefix **
call forward all **1
call forward cancel **2
pickup local **5
pickup group **7
pickup direct **6
cancel-call-waiting **9
last-redial **3
router(config-dsappline-fac)#do show run | b dsapp line
dsapp line
!
dsapp line feature access-code
!
```

Autoconfiguration

Auto configuration of SIP line features allows you to automatically configure the dial peers to set the endpoint to a SIP line. The auto configuration procedure adds the dial peers for each of the endpoint that you have configured on CUCM.

For CUCM-controlled SIP analog endpoints, you must perform configurations on the CUCM as well as the voice gateway. You must first perfrom the configuration on the CUCM, and after this configuration is complete, the voice gateway allows you to perform the configurations on the voice gateway.

For the auto configuration, initiate the configuration from the voice gateway and download the resulting configuration file. The XML configuration file is pushed from the CUCM to the gateway. Subsequently, the gateway parses the XML file and configures the pots dial-peer as per the configuration specified in the file.

Enabling the Auto Configuration

For the auto configuration to work, you must first specify the CUCM to the SIP line. Doing so indicates that the CUCM is the configuration server to the SIP line. To perform this step and enable the SIP line auto-configuration feature, run the **ccm-manager sipana auto-config local** command.

Then, run the **ccm-manager config server** command. This command initiates a download request of the configuration file. After the file is downloaded from the CUCM server, the XML file is parsed to determine the number of ports that are configured on the CUCM and the corresponding port IDs. The auto configuration then processes all the port information before configuring the corresponding dial-peers to set the endpoint to a SIP line. The dial-peers are added for each of the endpoints that are configured on CUCM.

Note For DSAPP auto-configuration, only pots dial-peer is auto configured. You must manually configure the outbound dial-peer and the voice card.

```
!
ccm-manager sipana auto-config local GigabitEthernet x/y/z
!
ccm-manager config server x.x.x.x
```

Here, GigabitEthernet x/y/z is the interface that is used for the SIP signaling.

Sample Configuration

```
!
ccm-manager sipana auto-config local GigabitEthernet0/0/1
!
ccm-manager config server 172.19.156.84

!
```

Verifying the Device Control Session Application Configuration

Use the following commands to verify the the DSAPP configuration:

- show dsapp line device summary
- show dsapp line feature codes
- show ccm-manager config-download

The **show dsapp line device summary** command shows whether the FXS ports are successfully registered to the CUCM as SIP endpoints.

```
router#show dsapp line device summary
Total Devices: 3
Port Device Registration Dev Directory Last Number
Identifier Name State Type Number Dialed
---------- --------------- ------------- ------- ----------- -----------
3/0/0 ANDD309DD761600 REGISTERED   ALG 2124506300 Not Avail
3/0/1 ANDD309DD761601 REGISTERED   ALG 2124506301 Not Avail
3/0/2 ANDD309DD761602 UNREGISTERED ALG 2124506302 Not Avail
router#
```

The **show dsapp line feature codes** command shows whether FAC is enabled and displays the feature codes.

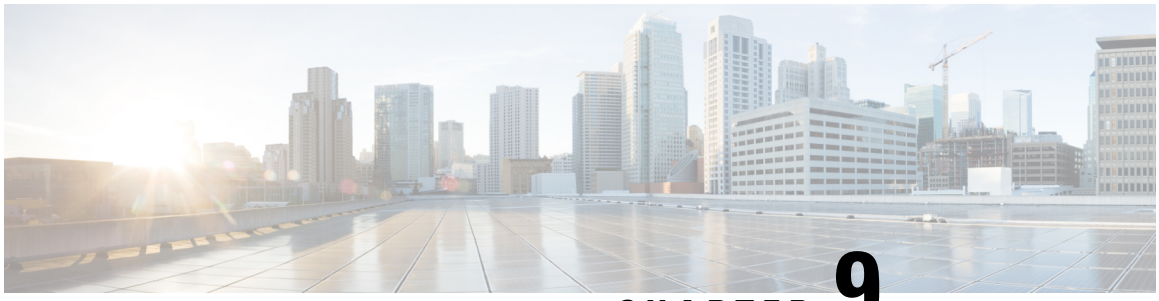
```
router#show dsapp line feature codes
dsapp line feature access-code
prefix **
call forward all **1
call forward cancel **2
pickup local **5
pickup group **7
pickup direct **6
cancel-call-waiting **9
last-redial **3
router#
```

The show ccm-manager config-download command provides download status and history of the auto-configuration.

```
Router#show ccm-manager config-download
SIP Line Side Analog auto-configuration status
============================================================
Registered with Call Manager: Yes
Local interface: GigabitEthernet0/0/0 (2c5a.0fc8.8b70)
Current version-id: 1541004382-f60b9ac2-ce5b-439e-92e5-02b62e26d15c
Current config applied at: 16:47:40 UTC Dec 18 2021
Gateway downloads succeeded: 2
Gateway download attempts: 2
Last gateway download attempt: 16:47:40 UTC Dec 18 2021
Last successful gateway download: 16:47:40 UTC Dec 18 2021
Current TFTP server: 172.19.156.84
Gateway resets: 1
Managed endpoints: 3
```

```
Endpoint downloads succeeded: 6
Endpoint download attempts: 6
Last endpoint download attempt: 16:47:40 UTC Dec 18 2021
Last successful endpoint download: 16:47:40 UTC Dec 18 2021

Endpoint resets: 0
Endpoint restarts: 0
Configuration Error History:
```



CHAPTER 9

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.

From Cisco IOS XE 17.13.1a, 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.

Supported Platforms

From Cisco IOS XE 17.13.1a, SELinux is enabled on the following platforms:

- Cisco 1000 Series Aggregation Services Routers
- Cisco 1000 Series Integrated Services Routers
- Cisco 4000 Series Integrated Services Routers
- Cisco Catalyst 8000v Edge Software
- Cisco Catalyst 8200 Series Edge Platforms
- Cisco Catalyst 8300 Series Edge Platforms
- Cisco Catalyst 8500 and 8500L Series Edge Platforms
- Cisco VG Series Gateways: VG400, VG410, VG420, and VG450
- Cisco 1100 Terminal Services Gateway

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:

```
set platform software selinux {default | enforcing | permissive}
platform security selinux {enforcing | permissive}
show platform software selinux
```

Note These new commands are implemented as **service internal** commands.

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