

Trustworthy Systems Guide

First Published: 2025-08-14

Last Modified: 2026-07-17

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1 **Trustworthy Systems** 1

Feature history for trustworthy systems 1

Trustworthy Systems 1

Need for trustworthy systems 2

SUDI 2

Trust Anchor Module 2

CHAPTER 2 **BIOS Protection** 7

Feature history for BIOS protection 7

BIOS protection 7

Primary ROMMON and golden ROMMON 8

Upgrade on standalone, high availability, and SVL devices 8

Capsule upgrade 8

How Capsule Upgrade works 9

LMS-signed bootloader 9

CHAPTER 3 **Boot Integrity Visibility** 11

Feature history for boot integrity visibility 11

Boot Integrity Visibility 11

How boot integrity visibility works 12

Image signing 13

Verify software image and hardware 13

Verify platform identity 13

Verify software integrity 15

Verify image signing 16

CHAPTER 4 **Consent Token** 19

Feature history for consent token 19

Consent Token 19

How consent token authorization works 20

Consent token authorization process for system shell access 20

CHAPTER 5 **Factory Reset** 23

Feature history for factory reset 23

Factory Reset 23

When do you need to do factory reset 23

What happens during a factory reset 24

Secure Data Wipe 25

Guidelines to perform factory reset 25

Perform a Factory Reset 25

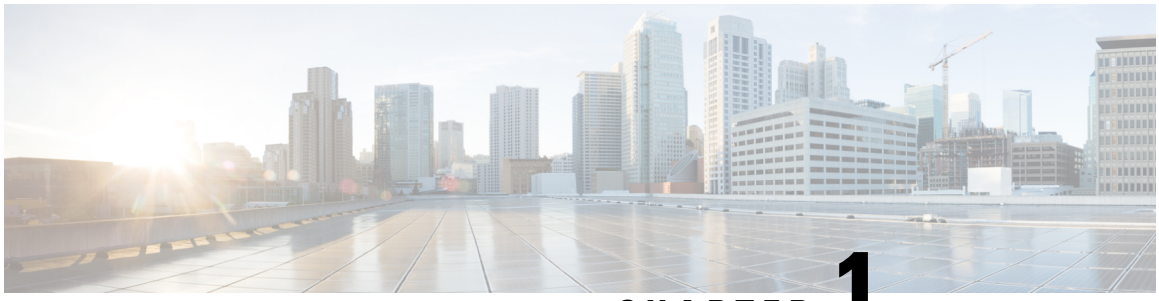
CHAPTER 6 **Secure Storage** 29

Feature history for secure storage 29

Secure Storage 29

Enable secure storage 29

Configuration example to verify the encryption status 30



CHAPTER 1

Trustworthy Systems

- Feature history for trustworthy systems, on page 1
- Trustworthy Systems, on page 1
- Need for trustworthy systems, on page 2
- SUDI, on page 2
- Trust Anchor Module, on page 2

Feature history for trustworthy systems

This table provides release and platform support information for the features explained in this module.

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

Release	Feature name and description	Supported platform
Cisco IOS XE 26.2.1ea	Trustworthy systems: Trustworthy systems feature support has been introduced.	Cisco C9550 Series Smart Switches
Cisco IOS XE 17.18.1	Trustworthy systems: This feature enhance the security and resilience of Cisco solutions.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Trustworthy Systems

Cisco's trustworthy technologies offer product assurance and foundational security capabilities. These features enhance the security and resilience of Cisco solutions.

To protect your devices from counterfeiting and malicious attacks on hardware and software and verify its authenticity and integrity, Cisco uses digitally signed software images, hardware-anchored secure boot, Secure Unique Device Identifier (SUDI), and other trustworthy technologies.

These security features

- protect your devices from counterfeiting and software modification,
- help you establish secure and encrypted communications, and

- ensure your Cisco network devices function as intended.

Trustworthy technologies run automated integrity checks on your device's hardware and software. Trust in your device starts at the hardware level and continues through the boot process, the operating system kernel, and into the operating system's runtime. If a device is compromised, they can shut down the boot process to protect your system.

Need for trustworthy systems

You rely on complex computing and communications networks to keep your services running without interruption. You need reliable data and IT infrastructure to keep your network secure and maintain user trust. As your personal data becomes accessible at any time and from any location, you expect consistent access and security across all networks.

Because adversaries have become more aggressive, you face a rapidly changing threat landscape. You must protect your network from attacks by malicious actors and from counterfeit or tampered products. Trustworthy systems help maintain network security, protect user trust, and defend against increasingly sophisticated threats.

SUDI

The SUDI provides your device with a tamperproof ID using an X.509 certificate. The private key is securely stored in the Trust Anchor Module (TAm) during manufacturing, ensuring your device’s identity is verified and protected against counterfeiting.

Your device is uniquely identified by SUDI, which combines its product ID and serial number. The X.509 version 3 certificate, which complies with IEEE 802.1AR and uses RSA or ECC encryption, helps prevent identity cloning or spoofing and protects your equipment from counterfeit risks.

Trust Anchor Module

The Cisco TAm is a tamper-resistant chip embedded in your Cisco devices. It includes non-volatile secure storage and serves as the foundation of the device's trust chain. The manufacturer's public key is securely stored within your device's TAm chip.

Your device includes Anti-Counterfeit Technology (ACT) by embedding specialized hardware chips called ACT and ACT2 within your devices.

The TAm provides these features:

Identity

During manufacturing, your device’s ACT2 chip receives the Cisco SUDI as an X.509v3 ECDSA or RSA certificate (or both), along with key pairs and certificate chains. The SUDI forms the basis for the hardware anti-counterfeit check and helps establish the device’s initial network identity.

Entropy

The ACT2 features a NIST SP 800-90B compliant entropy source, making it ideal for seeding host-based pseudo-random number generators.

Key Management

The ACT2 is capable of generating symmetric keys, as well as ECC and RSA asymmetric keypairs. Crucially, the symmetric keys and the private portions of these keypairs are never released from the chip. Access to these protected keys is facilitated solely through cryptographic APIs, and certificates can be enrolled for the keypairs generated by the ACT2.

Secure Storage

The ACT2 offers approximately 50 KB of host data storage, secured against physical tampering. This makes it an ideal repository for sensitive information like licenses and confidential data such as credentials. Crucially, important device keys and passwords are also safeguarded within this secure storage.

The TAm and SUDI in your device serve as unique hardware identifiers, much like a mobile device's IMEI number, and also offer added functions.

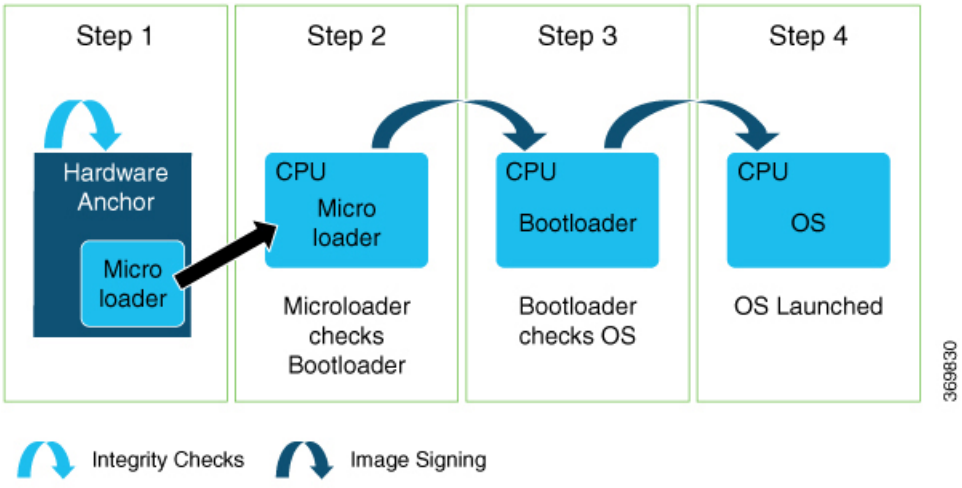
The TAm features secure nonvolatile storage for cryptographic keys, which applications access through TAm libraries. When your device starts, SUDI and boot measurements stored in the TAm help verify that your device is legitimate, making sure the boot process works as expected and helping detect tampering.

Secure Boot

Cisco Secure Boot ensures authenticity and integrity of code executing on your device. Its hardware-anchored secure boot safeguards the micro loader—the very first piece of code that boots—within tamper-resistant hardware. This creates a root of trust, effectively preventing your Cisco devices from running tainted network software.

Cisco digitally signs all images using the manufacturer's private key. This happens in a secure and audited environment. The key pair is generated using RSA encryption with a 2048-bit modulus.

Your device, after booting, authenticates with the TAm through SUDI to ensure that its hardware is indeed genuine Cisco, rather than a counterfeit or compromised unit.



The public key, embedded directly into the hardware, is accessible via TAm libraries. Cisco IOS XE images are signed using private keys that are never removed from the build DevOps release environment. During the boot process, these keys are compared, confirming that both the hardware and software have been signed by Cisco.

The secure chain starts when the TAm verifies the microloader's integrity before loading it onto the CPU. Next, the microloader checks the BIOS or bootloader using the TAm's API to confirm the image is legitimate before loading. This verification also occurs when loading the Cisco IOS XE image.

After the CPU loads the image, the operating system utilizes the TAm API to verify software modules like KLM, RPM, and ASIC SDK prior to their activation.

Chip Guard (Chip Protection)

Your Cisco devices incorporate a chip protection feature that guarantees hardware integrity. This protection mechanism records each device's signature during manufacturing and then compares it during every device bootup, thereby ensuring that the peripherals of your device are not counterfeit.

Manufacturing Time Database

The manufacturing time database is the original copy of the unique IDs of Cisco ASICs, CPUs, SoCs, and other devices with their device types specific to a board. In most cases, the unique ID is a device serial number or other appropriate value of that device. The manufacturing database is a Known Good Values (KGV) database specific to a board. It is programmed onto the TAm device as part of the manufacturing process.

Collected Database

The collected database is collected by the firmware whenever the board is booted and extended to the TAm device. Measurements are collected either through firmware or through system drivers.

The BIOS boot process integrates the TAm library to populate the collection database. The BIOS detects various hardware components as part of initialization and uses the TAm library APIs to record the device type and unique IDs if the detected devices are part of the manufacturing time database. After all the device types and unique IDs are written to the collected database, the platform operating system invokes the TAm library API to validate the collected database against the manufacturing time database. If there is a mismatch, the platform holds the boot process.

Random Number Generation and Entropy Source

Strong random number generation (RNG) is at the core of encryption, while weak RNG can undermine the entire encryption system. Random number generators play a key role in creating cryptographic keys, establishing highly secure communications between users and websites, and in resetting passwords for email accounts. Without assured randomness, an attacker can predict what the system will generate and undermine the algorithm. Your Cisco devices uses the RNG from Linux. The RNG is seeded with a random value, typically obtained from a hardware random number generator (HRNG), which makes it impossible to guess. Hardware also contains a Trust Anchor module that is compliant with NIST specifications and capable of providing much more effective RNG that extracts entropy from a true random source within the Trust Anchor.

Multistage BIOS

The BIOS is split into multiple, smaller pieces so that they can be loaded, validated, and executed entirely in the RAM to protect the BIOS from external modification (such as a Time-of-Check to Time-of-Use attack [1]). The BIOS is composed of the following things:

- Pre-EFI Initialization (PEI)
- Firmware Dependency Module (FDM)
- Driver Execution environment (DXE)

Having a multistage BIOS in your device makes it very hard to bypass the Cisco BIOS. Any intervention with the BIOS will stop the bootloader from loading the OS image

Runtime Defenses (RTD)

Runtime defenses target injection attacks of malicious code into running software. Your Cisco devices runtime defenses include Address Space Layout Randomization (ASLR), Built-in Object Size Checking (BOSC), and X-space Runtime defenses. These defenses make it harder or impossible for attackers to exploit vulnerabilities in running software.

Address Space Layout Randomization (ASLR)

Address Space Layout Randomization (ASLR) is an important security hardening functionality that randomizes the locations of sections of all processes and the kernel for your Cisco devices making it more difficult for an attacker to exploit existing vulnerabilities. ASLR is a companion defense along with executable space protection, which prevents inadvertent execution of code from unauthorized areas and prohibits writing of code over executable areas.

ASLR functionality for processes can be categorized into Cisco binaries and 3rd party binaries, both of which need to support ASLR. For ASLR support, Cisco and 3rd party binaries and shared libraries need to be built with the correct flags. Cisco binaries, including 3rd party shared objects, must ensure the library is randomized so as not to compromise the randomization of the Cisco binary itself. 3rd party binaries and shared libraries might require vendor support to randomize them.

ASLR functionality for the Linux kernel brings support for address space randomization to running Linux kernel images by randomizing where the kernel code is placed at boot time. Kernel ASLR support is present in Cisco devices to prevent hackers from injecting malicious code.

Executable Space Protection (XSpace)

Executable space protection (X-space) is one of the most important security protections in your Cisco devices. This feature ensures that executable space protection is enabled for devices. This prevents the execution of code from unauthorized areas and prohibit writing code over executable areas.

X-space prevents hackers from penetrating into the devices and ensures the device is secure.

Object Size Checking (OSC)

Buffer overflow is probably the best-known form of software security vulnerability. A buffer overflow condition exists when a program attempts to put more data in a buffer than the buffer can hold, or when a

[1] A Time-of-Check to Time-of-Use (TOC/TOU) attack is a type of race condition vulnerability that occurs when a program checks a condition and then performs an action based on the result of that check. However, if another program modifies the condition between the time of the check and the time of the use, the action will be performed based on the modified condition, which could lead to unintended consequences.

program attempts to put data in a memory area past a buffer. In this case, a buffer is a sequential section of memory allocated to contain anything from a character string to an array of integers. Writing outside the bounds of a block of allocated memory can corrupt data, crash the program, or cause the execution of malicious code. Your Cisco devices have full protection to determine buffer overflows in C or C++ code by having object size checks before a write call.

SafeC Libraries

Cisco IOS XE software uses efficient library functions that promote safer, more secure C or C++ language programming and are based on the ISO/IEC 9989:2011 (C11) specification. Several standard C library functions are susceptible to vulnerabilities that can serve as launch points for more sophisticated attacks.

While providing "safe" replacements for standard functions in a consistent naming schema, SafeC aims to mitigate security exploits due to buffer overflows, provides bound checks that may not be present in the native library, and prevents string termination and truncation errors. This SafeC safeguards your Cisco device hardware from buffer overflow attacks.

Cisco Signed Kernel Modules

Signed Kernel Modules ensure that any kernel modules loaded into your device are authentic and unmodified. This prevents unapproved and untrusted executable code from being loaded into the kernel by conventional means.

If the modules are not signed by Cisco, then they cannot be used with the Cisco IOS XE software. This feature stops unauthorized software from running on your Cisco devices. These hardening features protect Cisco IOS XE software and its components from attacks.

Secure JTAG (sJTAG)

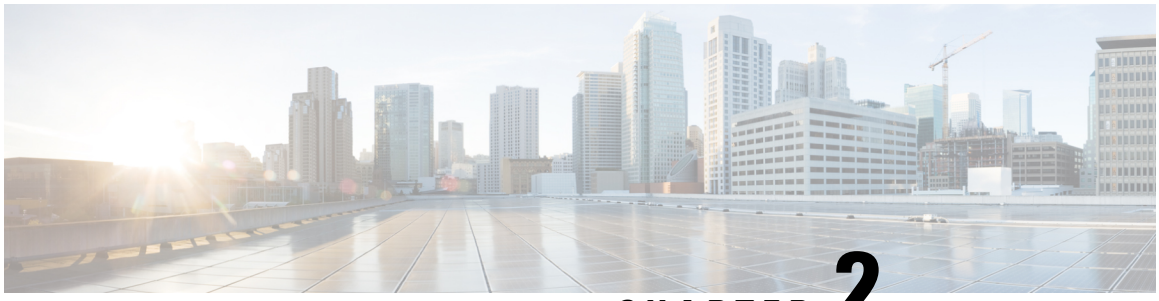
JTAG was also adopted to program FPGAs and provide a CPU debug access port. A laptop and a JTAG debugger are often all that is required to provide access to an embedded CPU allowing for retrieval of firmware images, dumping memory, and monitoring software execution. A small size interface coupled with a sophisticated toolset gives attackers a portable yet powerful means to exploit a system.

By having a secure JTAG on your Cisco device, we can mitigate intellectual property (IP) theft and avoid the stealing of passwords or keys from the memory.

Secure Erase

The secure erase feature erases all customer information within your Cisco device. Secure erase is an operation that removes all the identifiable customer information in the device for purposes of product removal due to Return Material Authorization (RMA), upgrade or replacement, or system end-of-life.

- RMA for a device: If you must return a device to Cisco for RMA, remove all customer-specific data before obtaining an RMA certificate for the device.
- Recovering a compromised device: If the key material or credentials that are stored on a device are compromised, reset the device to the factory configuration, and then reconfigure the device.



CHAPTER 2

BIOS Protection

- Feature history for BIOS protection, on page 7
- BIOS protection, on page 7
- Primary ROMMON and golden ROMMON, on page 8
- Upgrade on standalone, high availability, and SVL devices, on page 8
- Capsule upgrade, on page 8
- LMS-signed bootloader, on page 9

Feature history for BIOS protection

This table provides release and platform support information for the features explained in this module.

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

Release	Feature name and description	Supported platform
Cisco IOS XE 26.2.1ea	BIOS protection: BIOS protection feature support has been introduced.	Cisco C9550 Series Smart Switches
Cisco IOS XE 26.1.1a	LMS-signed bootloader: The device supports secure boot process based on LMS PQC algorithm.	Cisco C9610 Series Smart Switches
Cisco IOS XE 17.18.1	BIOS protection: This feature provides write-protection for the golden ROMMON image and ensures that any upgrades to this image are performed securely.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

BIOS protection

The BIOS protection feature provides write-protection for the golden ROMMON image and ensures that any upgrades to this image are performed securely.

Without the BIOS protection feature, the golden ROMMON image is vulnerable to corruption by malicious code during software upgrades.

Primary ROMMON and golden ROMMON

ROMMON images are stored on the SPI flash device as both primary ROMMON and golden ROMMON.

The primary ROMMON is used to boot the device each time it is powered on or restarted. If the primary ROMMON becomes corrupted, the device automatically uses the golden ROMMON to boot the IOS XE software image.

When the device boots from the primary ROMMON, the golden ROMMON remains locked.

With the BIOS protection feature enabled, the golden ROMMON is write-protected and cannot be upgraded using the standard flash utility upgrade mechanism. Access policies for the golden ROMMON are enforced by the FPGA firmware, which blocks unauthorized operations such as write and erase commands on the golden ROMMON SPI flash device.



Note Golden ROMMON upgrade becomes available only after a secure boot FPGA upgrade.

Upgrade on standalone, high availability, and SVL devices

The upgrade process varies between standalone, high availability, and SVL devices. This table explains how the upgrade process works.

Table 1: Upgrade on standalone, high availability, and SVL devices

Device configuration	Upgrade process
Standalone device	When you upgrade in install mode, the primary ROMMON upgrades automatically when the device boots. To upgrade the golden ROMMON, use the capsule upgrade process.
High availability and StackWise Virtual devices	Perform an In-Service Software Upgrade (ISSU) for devices in a high availability setup. This ISSU process includes FPGA upgrades. If you upgrade in install mode with reload, reload one supervisor at a time. With the standby supervisor in the ROMMON state, boot the active supervisor first. After the ROMMON upgrade completes on the active supervisor, upgrade the FPGA and software image as well.

Capsule upgrade

The primary ROMMON, primary FPGA, and golden FPGA (secure-boot FPGA) are automatically upgraded when the device boots. In contrast, the golden ROMMON can only be upgraded using the capsule upgrade process, ensuring an additional layer of security for critical boot components.

In a capsule upgrade, a secure update capsule is created and digitally signed, and is used by the primary ROMMON to upgrade the golden ROMMON after authentication. This process requires a secure flash certificate, which is generated using the product key and included in the primary ROMMON image to verify the authenticity of the update capsule. The capsule itself is created using the secure flash certificate along with a secure boot 32 MB flash image on the Cisco C9350 Series Smart Switches and 64 MB flash image on the Cisco C9610 Series Smart Switches, and is then signed to ensure the integrity and authenticity of the upgrade.

When the device boots, the primary ROMMON initiates the capsule upgrade process for the golden ROMMON. To manually perform a capsule upgrade for the golden ROMMON, use the **upgrade rom-monitor capsule golden** command on a switch or the **upgrade rom-monitor capsule golden switch** command on a switch stack in privileged EXEC mode.

How Capsule Upgrade works

Workflow

This section explains what happens during a capsule upgrade.

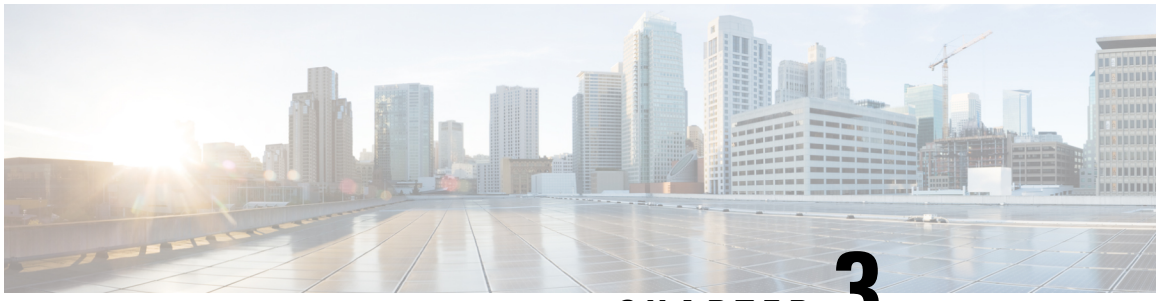
1. The device checks if secure-boot FPGA upgrade is enabled. If secure-boot FPGA upgrade is not enabled, the upgrade stops.
2. The device checks if bootloader protection is enabled. If bootloader protection is not enabled, a one-time upgrade of primary ROMMON, golden ROMMON, and primary FPGA is initiated.
3. If bootloader protection is already active, Cisco IOS XE copies the secure update capsule to bootflash and the device reboots.
4. When the device reboots, the system selects the secure update capsule to perform the upgrade.

LMS-signed bootloader

The device supports secure boot process based on Leighton-Micali Signature (LMS) Post-Quantum Cryptography (PQC) algorithm which is approved by National Institute of Standards and Technology (NIST). When the device boots up, the microloader residing in secure boot FPGA initiates the bootloader or ROMMON after verifying its authenticity. PQC provides encryption that protects critical data.



Note LMS is only supported on the Cisco C9610 Series Smart Switches running hardware version 2.0. Use the **show module** command to view the hardware version.



CHAPTER 3

Boot Integrity Visibility

- Feature history for boot integrity visibility, on page 11
- Boot Integrity Visibility, on page 11
- How boot integrity visibility works, on page 12
- Image signing, on page 13
- Verify software image and hardware, on page 13
- Verify platform identity, on page 13
- Verify software integrity, on page 15
- Verify image signing, on page 16

Feature history for boot integrity visibility

This table provides release and platform support information for the features explained in this module.

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

Release	Feature name and description	Supported platform
Cisco IOS XE 26.2.1ea	Boot integrity visibility: Boot integrity visibility feature support has been introduced.	Cisco C9550 Series Smart Switches
Cisco IOS XE 17.18.1	Boot integrity visibility: This feature acts as a hardware trust anchor by validating the ROMMON software to ensure its integrity.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Boot Integrity Visibility

Boot integrity visibility acts as a hardware trust anchor by validating the ROMMON software to ensure its integrity.

Boot integrity visibility enables Cisco platforms to make both platform identity and software integrity information visible and actionable. Platform identity refers to the unique identity assigned during manufacturing,

ensuring each device can be reliably identified. Software integrity involves capturing boot integrity measurements, which help determine whether the platform has started up using trusted code.

Your Cisco device supports the boot integrity visibility feature.

How boot integrity visibility works

Summary

When you copy a Cisco IOS XE software image onto your Cisco device, the ROMMON Boot ROM verifies the image using Cisco release keys. These release keys are public keys that correspond to the private release key securely stored on Cisco build servers. The public release key is embedded within the ROMMON, enabling your device to validate the authenticity and integrity of the Cisco IOS XE software image before booting.

Workflow

The ROMMON follows these steps to verify a signed Cisco IOS XE software image when you boot the device:

1. Loads the Cisco IOS XE software image into the CPU memory.
2. Examines the Cisco IOS XE software package header.
3. Runs a nonsecure integrity check on the image to ensure there is no file corruption from disk or TFTP. The check uses a nonsecure SHA-1 hash.



Note This step checks for inadvertent corruption caused by disk, file transfer, or copying errors. It is not part of image code signing and does not detect deliberate image tampering.

4. Copies the Cisco's RSA 2048-bit public release key from the ROMMON storage and validates that the Cisco's RSA 2048-bit public release key is not tampered.
5. Extracts the Code Signing signature (SHA-512 hash) from the package header and verifies it using Cisco's RSA public release key.
6. Performs the Code Signing validation by calculating the SHA-512 hash of the Cisco IOS XE software package and compares it with the Code Signing signature. The signed package is now validated.
7. Examines the Cisco IOS XE software package header to validate the platform type and CPU architecture for compatibility.
8. Extracts the Cisco IOS XE software from the software package and boots the software.

Result



Note Image Code Signing validation occurs in steps 4, 5, and 6. This is a secure code signing check of the image using an SHA-512 hash, which is encrypted with a 2048-bit RSA key. This check is intended to detect deliberate image tampering.

If the software is not generated by a Cisco build server, signature verification fails. Your device's ROMMON rejects the image and stops booting.

If the signature verification is successful, the device boots the image to the Cisco IOS XE software runtime environment.

Image signing

Cisco build servers generate and digitally sign Cisco IOS XE software images at build time. An SHA-512 digest is calculated over the entire binary image, then signed using a Cisco-controlled RSA-2048 or ML-DSA private key. The device verifies the signature with the corresponding trusted public key, confirming the authenticity and integrity of the software image.

Verify software image and hardware

This section describes how to retrieve the checksum record that was created during a switch bootup. Enter the following commands in privileged EXEC mode.

Table 2: Command to retrieve the checksum records

Command	Description
show platform sudi certificate [**sign** [**nonce** *nonce*]]	Displays checksum record for the specific SUDI. • (Optional) **sign**: Show signature • (Optional) **nonce**: Enter a nonce value
show platform integrity [**sign** [**nonce** *nonce*]]	Displays checksum record for boot stages. • (Optional) **sign**: Show signature • (Optional) **nonce**: Enter a nonce value

Verify platform identity

The following example displays the Secure Unique Device Identity (SUDI) chain in PEM format.

Encoded into the SUDI is the Product ID and Serial Number of each individual device such that the device can be uniquely identified on a network of thousands of devices.

- The first certificate is the Cisco Root CA 2048
- The second is the Cisco subordinate CA (ACT2 SUDI CA)

  Both certificates can be verified to match those published on https://www.cisco.com/security/pki/.
- The third is the SUDI certificate.

Note On executing the following command, you might see the message **% Please Try After Few Seconds** displayed on the CLI. This does not indicate a CLI failure, but indicates setting up of underlying infrastructure required to get the required output. We recommend waiting for a few minutes and then try the command again

```
Device# show platform sudi certificate sign

-----BEGIN CERTIFICATE-----
MIIDITCCAgmgAwIBAgIJAZozWHjOFsHBMA0GCSqGSIb3DQEBCwUAMC0xDjAMBgNV
BAoTBUNpc2NvMRswGQYDVQQDExJDaXNjbyBSb290IENBIDIwOTkwIBcNMTYwODA5
MjA1ODI4WhgPMjA5OTA4MDkyMDU4MjhaMC0xDjAMBgNVBAoTBUNpc2NvMRswGQYD
VQQDExJDaXNjbyBSb290IENBIDIwOTkwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQDTtuM1fg0+9Gflik4axlCK1I2fb3ESCL8+tk8kOXlhfrJ/zlfRbe60
xRP0iUGMKWKBj0IvvWFf4AW/nyzCR8ujTt4a11Eb55SAKXbXYQ7L4YMg+lmZmg/I
v3GJEc3HCYU0BsY8g9LuLMvqwiNmAwM2jWzNq0EPArt/F6RiQKq6Ta3e7VIfDZ7J
65OA2xASA2FrSe9Vj97KpQReDcm6G7cqFH5f+CrdQ4qwAa4zWNyM3kOpUb637DNd
9m+n6WECyc/IUD+2e+yp21kBZIKH7JvDpu2U7NBPfr52mFX8AfCZgkXV69bp+iYf
saH1DvXIfPpNp93zGKUSXxEj4w881t2zAgMBAAGjQjBAMA4GA1UdDwEB/wQEAwIB
BjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBQ4lVcPNCNO86EmILoUkcdBiB2j
WzANBgkqhkiG9w0BAQsFAAOCAQEAjeKZo+4xd05TFtq99nKnWA0J+DmydBOnPMwY
lDrKfBKe2wVu5AJMvRjgJIoY/CHVPaCOWH58UTqfji95eUaryQ/s36RKrBgMMlwr
WNItxE625PHuaN6EjD1WdWiRMZ2hy8F4FCKz5hgUEvN+PUNZwsPnpU6q3Ay0+11T
4TriwCV8kJx3cWu0NvTypYCCXMscSfLFQR13bo+1z6XNm30SecmrxkmQBVMqjCZM
VvAxhxW1iGnYdPRQuNqt0xITzCSERqg3QVVqYnFJUkNVN6j0dmmMVKZh17HgqLnF
PKkmBlNQ9hQcNM3CSzVvEAK0CCEo/NJ/xzZ6WX1/f8Df1eXbFg==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEZzCCA0+gAwIBAgIJCmR1UkzYYXxiMA0GCSqGSIb3DQEBCwUAMC0xDjAMBgNV
BAoTBUNpc2NvMRswGQYDVQQDExJDaXNjbyBSb290IENBIDIwOTkwIBcNMTYwODEx
MjAyODA4WhgPMjA5OTA4MDkyMDU4MjdaMDExHzAdBgNVBAMTFkhpZ2ggQXNzdXJh
bmNlIFNVREkgQ0ExDjAMBgNVBAoTBUNpc2NvMIIBIjANBgkqhkiG9w0BAQEFAAOC
AQ8AMIIBCgKCAQEAvdzeSWdDI61RZDYRvA6JqaRvQyy6Dx1WaqI82UeKR4ZRn0ef
xMGvp4c88/VMS8WSjQO1qolMfMxqHkcSiFBOULx6Trquw4TrEf9sIuzvgJvDaEa8
IllXPwtPtNqZEIWi8jlinz2uGam93KuGPcioHfruzbDKWHL/HWFGYMgz+OKwhD3J
4NRySknQvUovfV8eWLeVOqW8rbnG3TZxv5VexOiK4jL3ObvsQPuAWUwUoo7nuFlE
GTG/VCeyCe/H8+afIScbZOkI9xejtckflnBYFVCyFxzm2H3YZatb6ohbyRXLtOPj
T3SJ+OOoYMlSLd28z727LpRbFFLGYhyWxEXDuQIDAQABo4IBgjCCAX4wDgYDVR0P
AQH/BAQDAgEGMBIGA1UdEwEB/wQIMAYBAf8CAQAwfwYIKwYBBQUHAQEEczBxMEEG
CCsGAQUFBzAChjVodHRwczovL3d3dy5jaXNjby5jb20vc2VjdXJpdHkvcGtpL2Nl
cnRzL2NyY2EyMDk5LmNlcjAsBggrBgEFBQcwAYYgaHR0cDovL3BraWN2cy5jaXNj
by5jb20vcGtpL29jc3AwHwYDVR0jBBgwFoAUOJVXDzQjTvOhJiC6FJHHQYgdo1sw
UgYDVR0gBEswSTBHBgorBgEEAQkVAR4AMDkwNwYIKwYBBQUHAgEWK2h0dHA6Ly93
d3cuY2lzY28uY29tL3NlY3VyaXR5L3BraS9wb2xpY2llcy8wQwYDVR0fBDwwOjA4
oDagNIYyaHR0cDovL3d3dy5jaXNjby5jb20vc2VjdXJpdHkvcGtpL2NybC9jcmNh
MjA5OS5jcmwwHQYDVR0OBBYEFOpro7nBE5d+G/s6jWhgBzlfh0j6MA0GCSqGSIb3
DQEBCwUAA4IBAQBcqYEOgAHhGWKndwM901XX2Enh4hjXR5avDg7G/f6Tb9H509dt
QW+AeZGEghhwUrw1EeG79tHkncAe+m+64xMC1ttyI1RSyn8rBqQYkXnnCRbtF/Nw
pQe5fjvdeIFWJhUI16TOt/ZlkNnWnLsUU1alZmN+J/FhSr8VTJWGRM9gY8hefH8f
5U7LMiDXxsFVHB7R6KGNjvtawrl6W6RKp2dceGxEIIvMahgMWWHHiWOQAOtVrHuE
NEjYR/7klLLwdgQF/NNCA2z47pSfMFnBcr8779GqVIbBTpOP2E6+1pBrE2jBNNoc
uBG1fgvh1qtJUdBbTziAKNoCo4sted6PW2/U
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIE8jCCA9qgAwIBAgIKB2EQhTQQhWJhIDANBgkqhkiG9w0BAQsFADAxMR8wHQYD
VQQDExZIaWdoIEFzc3VyYW5jZSBTVURJIENBMQ4wDAYDVQQKEwVDaXNjbzAgFw0y
NjA0MTYwODI5MzVaGA8yMDk5MDgwOTIwNTgyNlowbDEnMCUGA1UEBRMeUElEOkM5
NTUwLTk2TDREIFNOOkZETzI5NDEwTkZLMQ4wDAYDVQQKEwVDaXNjbzEYMBYGA1UE
CxMPQUNULTIgTGl0ZSBTVURJMRcwFQYDVQQDEw5RNUdHLVlSUEYtNE5OVTCCASIw
DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMPgMraZ0Jd37GVAcAwU6BgRU53E
l88DM1SKv31JkQ5rt5RpnpdzDIBFfeAp3He9Q42ITWaXHoVCFQNhZnBkI4NOeuRo
UPWiRmgZdZ0cLvrpQJ8HkrDvXTBavLLLSNP72clvKjDJvf/O/kDuoo1C4cpwKh0E
```

```
a3PxFdGFveHstE8uda1G29KpXpSLTUNHIiDNZWdy8stFGNqUopjKA/J1MULyOlTI
iRQlwrlegyqAc1FKWux28DnOEBiMkbdcQTVeb/oO6FI1X+c1KZTOBNJoUl+goGDk
VuijxKqSEJVSbX4/IOebPHYRaW4YoxmebBGjGbk9D9Dl99VQVBgGXmMyrY0CAwEA
AaOCAc0wggHJMA4GA1UdDwEB/wQEAwIF4DAMBgNVHRMBAf8EAjAAMB8GA1UdIwQY
MBaAFOpro7nBE5d+G/s6jWhgBzlfh0j6MIIBZwYDVR0RBIIBXjCCAVqgUAYKKwYB
BAEJFQMEAqBCE0A5NzNGQTQ5ODE5RTE0NTY1NjYwNzFBQzVFMTU0RjI1ODI2RTM2
RjFGOTkzM0M3MEM5NjAwNDVGRjEyOTRGQUU4oHAGCisGAQQBCRUDBASgYhNgMUI2
Mjk3NEY0MDExNzA0MDgyMzQ3NEZBNkEyRjg3REZGMTNGQjM1MjRDQ0EwNDkxQzQy
NDY4NjkwNjVEQzIyQjA5M0E0MkUzMzY5MkQwRTRFMEE5NUNGMjNCQzE1QUFEoFAG
CisGAQQBCRUDBAOgQhNAMDAwMDEwMDAwMDA0MTQyMDgyMUREMTlEMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDA0QjUyNTk1MKBCBgkrBgEEAQkVAgOgNRMz
Q2hpcElEPUx2VzZpVlJaam9zRGs3L0xSOVBmdmdBQUFBQUFBQUFBQUFBQUFB
QUE9MB0GA1UdDgQWBBTr19uzsiktLCPintxH66NV1TbxVjANBgkqhkiG9w0BAQsF
AAOCAQEArX2KKkI4E7Z7unNFFkiUNcyDegqNbZTqXgmcEpPkz3ZdFphhRfN4vnmN
KKGkU+WY6/y53FrjtueXuVIdU5MChKeTkuJsMOr+PbXeS3cnhapyvEzzaL4b/oi5
DkSXw5MyzcLh5T7ZVQ5sHO54F5MXIZW0/EMREf9ir5uTZs/iTGoe6pYW46ZmlHtq
H2t2SbOXHYpX1+aZOKuVU37dduLoMUKJNFVAeEDU3HgEUZxQuc94M6vVhUj7K+0X
jPbCAuG0v6DUOQVv6nMPEHdofPMRcMyoexzLk7jAL/8GwZlRJftmcyH4x329c0Yl
Z0ahgllmAN9Q8V8UY25KOeUzDkAyWA==
-----END CERTIFICATE-----
Signature version: 1
Signature:
```

The optional RSA 2048 signature is across the three certificates, the signature version and the user-provided nonce.

```
RSA PKCS#1v1.5 Sign {<Nonce (UINT64)> || <Signature Version (UINT32)> || <Cisco Root CA
2048 cert (DER)> ||
<Cisco subordinate CA (DER)> || <SUDI certificate (DER)> }
```

Cisco management solutions are equipped with the ability to interpret the above output. However, a simple script using OpenSSL commands can also be used to display the identity of the platform and to verify the signature, thereby ensuring its Cisco unique device identity.

```
linux-host:~]openssl x509 -in sudicert.pem -subject -noout
subject= /serialNumber=PID:C9300-24P SN:FDO1946BG05/O=Cisco/OU=ACT-2 Lite SUDI/CN=C9300-24P
```

Verify software integrity

The following example displays the checksum record for the boot stages. The hash measurements are displayed for each of the three stages of software successively booted. These hashes can be compared against Cisco-provided reference values. An option to sign the output gives a verifier the ability to ensure the output is genuine and is not altered. A nonce can be provided to protect against replay attacks.

Note Boot integrity hashes are not MD5 hashes. If you run **verify /md5 cisco9k_iosxe.17.18.01.SPA.bin** command for the bundle file, the hash will not match.

The following is a sample output of the **show platform integrity sign** command in install mode.

This output includes measurements of each installed package file.

```
Device# show platform integrity sign

Platform: C9550-96L4D
Boot 0 Version: MP2002R07.1120260302.3ea7c865
Boot 0 Hash: 8ADA8820CF37AAF1ECFEAB5BD75415D87589D72AE4A544880E0531A000EA980E
Boot Loader Version: 26.1.1.A%26.1.1.A
```

```
Boot Loader Hash:
608EEFD92E95C13C48C9A4742F4E31D636D08E3E4F1D8A3B45A8FD57EBA28FD8B33508522D88370761A9A5063C6A3962E3239323EAC18C8AF162B55165FFA057
OS Version: BLD_V262_THROTTLE_LATEST_20260709_003125
OS Hashes:
cisco9k_iosxe.BLD_V262_THROTTLE_LATEST_20260709_003125.SSA.bin:
9EC3AEB72E8A8DC5609368E4CFFACD35F55F02A42045AAC36C3E2A590CBC0F6E9B6904355C5A90E398E72DC2E3F9E8F665515C0EEBC04E8D54FA459993F64949
cisco9k-rpbase.BLD_V262_THROTTLE_LATEST_20260709_003125.SSA.pkg:
0B0072BA547C21C15C33E3D469EB32CDA9F9A86667E0F62E9BE0341DEEB1B432F48A4F62C30C82DDA9658C966E0A8D61AD155D67E69A60B4A6142D35C08DE741
cisco9k-wlc.BLD_V262_THROTTLE_LATEST_20260709_003125.SSA.pkg:
AA01F3AC8A6C7E4DEFC1A29C733D609BC9AC51F8E1B8B60880CB717A9EB712243E83B56C9E821F403AB4C9F731FB332DFF8129C587E16876FE5134F2081780D8
cisco9k-srdriver.BLD_V262_THROTTLE_LATEST_20260709_003125.SSA.pkg:
893283C5B038B79E17C42D99092C6A73F4C231B1463FF8843EA935A5ED2FA9056F9A101A8B0C68D640122833D4D011AB22B566158780111755446048569D9D62
cisco9k-webui.BLD_V262_THROTTLE_LATEST_20260709_003125.SSA.pkg:
46EE0653035B37F98FD1FFE639D56C165C70AC376802FA1AA72D6645362F0E7167A54A59AAA560294D5E52EDAF81BB7CFCCA40348E6910A0EBD23E71FDB8E023
cisco9k-guestshell.BLD_V262_THROTTLE_LATEST_20260709_003125.SSA.pkg:
E592AE468794F0C22FE28337462FE22C423733A13F672216C8C682B542706B5F9318FAF20066771E7BFEA89C9683A84D4BCB622454C7E51CCABB430BC63F716A
cisco9k-lni.BLD_V262_THROTTLE_LATEST_20260709_003125.SSA.pkg:
D3247E6820E5DF343FDCD673850706FA27CBE825393C31CFDFACEB0F3CD523764BF09F306B99180208DAFFC4E2F596E23EF894D63F4D18B0646B0275E1209620
cisco9k-cc_srdriver.BLD_V262_THROTTLE_LATEST_20260709_003125.SSA.pkg:
9851AE6F435841B7DF5586FE8DA34929FF4C3122150FE7992EA1030AC3D2765ED5F51F87419BCD86090907557C3030D89C4FA271EC0237501CA9A8940C1D134B
PCR0: 7244B76E2DF81C6585BC9FE5303A3B0734C0CB9752D097FC27546C935FBEFA5D
PCR8: 36E69ED3949CA95A9B77D41A02B97ECAB9AEFCD87963160EFDEAD4C8B738FA67
Signature version: 1
Signature:
[redacted]
```

Verify image signing

The following example displays the secure code signing check of the image during bootup using an SHA-512 hash

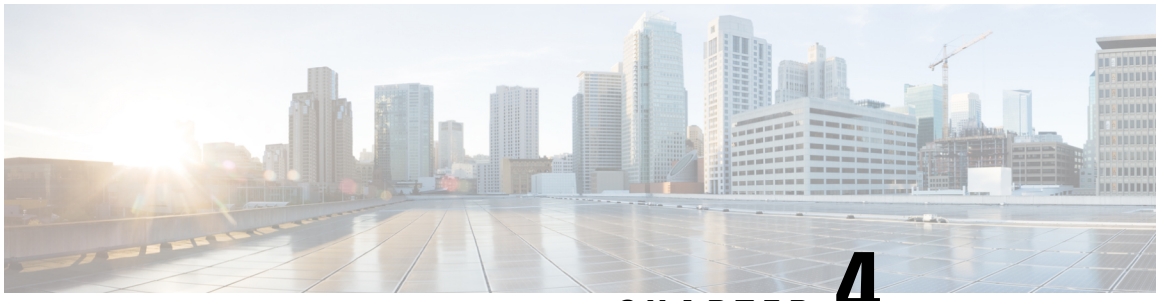
```
switch:boot
system has been configured to boot from device configuration.
Preparing to autoboot. [Press Ctrl-C to interrupt]  0
boot: attempting to boot from
[tftp://10.106.16.20/auto/tftp-blr-users1/kbalucha/cisco9k_iosxe.26.02.01.SPA.bin]
tftp:
 blocksize : 8192
 server    : 10.106.16.20
 path      : auto/tftp-blr-users1/kbalucha/cisco9k_iosxe.26.02.01.SPA.bin
 (ascii)   : auto/tftp-blr-users1/kbalucha/cisco9k_iosxe.26.02.01.SPA.bin
 url      : tftp://10.106.16.20/auto/tftp-blr-users1/kbalucha/cisco9k_iosxe.26.02.01.SPA.bin
tftp: preparing TFTPv4 GUID vector
network: enabling IPv4 static configuration
network: enabling IPv4 static configuration

h/w (environment):
 interface : eth0
 mac       : 14:BC:68:D3:7C:00
n/w (environment):
 ip        : 10.106.21.16
 mask      : 255.255.255.0
 gateway   : 10.106.21.1

h/w:
 interface : eth0 (Ethernet)
 mac       : 14:BC:68:D3:7C:00
n/w (ip v4):
 ip        : 10.106.21.16
 mask      : 255.255.255.0
 route(s)  : 0.0.0.0 -> 10.106.21.0/255.255.255.0
           : 10.106.21.1 -> 0.0.0.0/0.0.0.0
```

```
n/w (ip v6):
 ip(s)     : FE80::16BC:68FF:FED3:7C00/64
 route(s)  : :: -> FE80::/64

tftp v4:
 server    : 10.106.16.20
 file      : auto/tftp-blr-users1/kbalucha/cisco9k_iosxe.26.02.01.SPA.bin
 blocksize : 8192
tftp: preparing TFTPv4 function vector
tftp v4:
 station   : 10.106.21.16
 netmask   : 255.255.255.0
 gateway   : 10.106.21.1
 server    : 10.106.16.20
tftp: configured TFTPv4 protocol
tftp: value(s): 0
tftp: value(s): 0 8192
Option#1  : tsize
Value     : 0
Option#2  : blksize
Value     : 8192
tftp: requesting 2 option(s)
tftp: received 2 option(s)
Option#1  : tsize
Value     : 1258756397
Option#2  : blksize
Value     : 8192
tftp: TFTPv4 option 'blksize' value '8192'
!
!
!
<<output truncated>>
```



CHAPTER 4

Consent Token

- Feature history for consent token, on page 19
- Consent Token, on page 19
- How consent token authorization works, on page 20
- Consent token authorization process for system shell access, on page 20

Feature history for consent token

This table provides release and platform support information for the features explained in this module.

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

Release	Feature name and description	Supported platform
Cisco IOS XE 26.2.1ea	Consent Token: Consent Token feature support has been introduced.	Cisco C9550 Series Smart Switches
Cisco IOS XE 17.18.1	Consent Token: This feature authenticates the network administrator of an organization to access the system shell with mutual consent from the network administrator and Cisco TAC.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Consent Token

Consent Token is a security feature that authenticates the network administrator of an organization to access the system shell with mutual consent from the network administrator and Cisco Technical Assistance Centre (Cisco TAC).

In some debugging scenarios, the Cisco TAC engineer may have to collect certain debug information or perform live debug on a production system. In such cases, the Cisco TAC engineer will ask you (the network administrator) to access system shell on your device. Consent Token is a lock, unlock, and re-lock mechanism that provides you with privileged, restricted, and secure access to the system shell.

How consent token authorization works

Summary

When you request access to system shell, you need to be authorized. You must first run the command to generate a challenge using the Consent Token feature on your device. The device outputs a unique challenge. You must then copy this challenge string and send it to a Cisco Authorized Personnel by email or instant message.

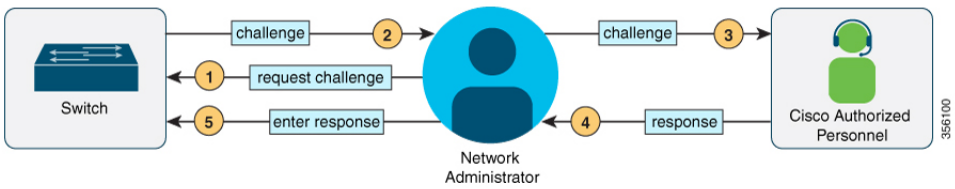
The Cisco Authorized Personnel processes the unique challenge string and generates a unique response. The Cisco Authorized Personnel copies this response string and sends it to you through email or Instant Message.

You must then input this response string into your device. If the challenge-response pair matches, you are authorized to access the system shell. If not, an error is displayed and you are required to repeat the authentication process.

Once you gain access to system shell, collect the debug information that Cisco TAC needs. After you finish accessing the system shell, end the session and continue debugging.

Workflow

Figure 1: Consent Token



1. Generate a challenge to request access to system shell for the specified time period.
2. Send the challenge string to a Cisco Authorized Personnel.
3. Input the response string onto your device.
4. Terminate the session.

Consent token authorization process for system shell access

This procedure details the Consent Token authorization process to access system shell.

Procedure

Step 1 Send a request for a challenge using the **request consent-token generate-challenge shell-access** *time-validity-slot* command.

Example:

```
Device# request consent-token generate-challenge shell-access auth-timeout 900
```
[illegible]
```
Device#
*Jan 18 02:47:06.733: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (challenge generation
attempt: Shell access 0).
```

time-validity-slot: The duration in minutes for which you are requesting access to system shell.

In this example, the session expires after 900 minutes.

The device generates a unique challenge in base-64 format.

Step 2 Send the challenge string generated by the device to a Cisco Authorized Personnel through email or instant message.

A Cisco Authorized Personnel processes the unique challenge string and generates a response. The response is also a base-64 string that is unique. The Cisco Authorized Personnel copies this response string and sends it to you through email or Instant Message.

Step 3 Input the response string sent to you by the Cisco Authorized Personnel using the **request consent-token accept-response shell-access** *response-string* command.

Example:

```
Device# request consent-token accept-response shell-access
```
[illegible]
```
% Consent token authorization success
*Jan 18 02:51:37.807: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (authentication success:
 Shell access 0).

Device# request platform software system shell
Activity within this shell can jeopardize the functioning of the system.
Are you sure you want to continue? [y/n] y
Device#
*Jan 18 02:56:59.714: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (authorization for Shell
 access 0 will expire in 10 min).
```

If the challenge-response pair match, you are authorized to access system shell. If the challenge-response pair does not match, an error is displayed. Repeat steps 1 to 3 to try again.

After you are authorized, you can access system shell for the requested time-slot.

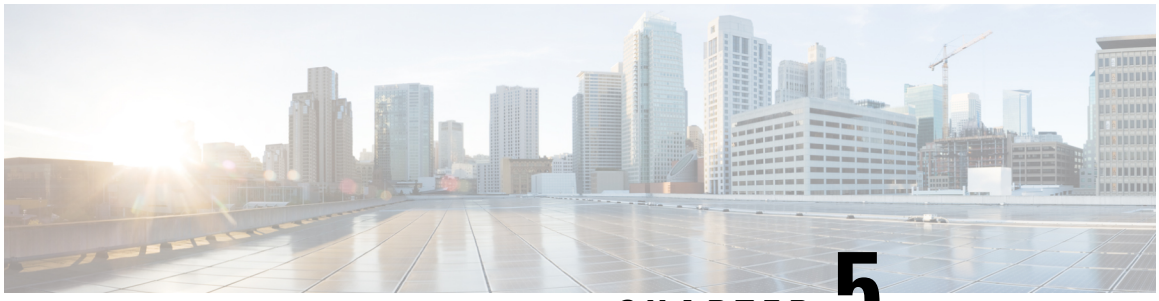
The device notifies you when ten minutes remain in the authorization session.

Step 4 When you finish accessing system shell, you can end the session using the **request consent-token terminate-auth** command. You can also force-terminate the session before the authorization timeout using this command. The session also gets terminated automatically when the requested time slot expires.

Example:

```
Device# request consent-token terminate-auth
% Consent token authorization termination success

Device#
*Jan 18 23:33:02.937: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (terminate authentication:
 Shell access 0).
Device#
```



CHAPTER 5

Factory Reset

- Feature history for factory reset, on page 23
- Factory Reset, on page 23
- Secure Data Wipe, on page 25
- Guidelines to perform factory reset, on page 25
- Perform a Factory Reset, on page 25

Feature history for factory reset

This table provides release and platform support information for the features explained in this module.

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

Release	Feature name and description	Supported platform
Cisco IOS XE 26.2.1ea	Factory reset: Factory reset feature support has been introduced.	Cisco C9550 Series Smart Switches
Cisco IOS XE 17.18.1	Factory reset: This feature deletes all data stored on a device and restores the device to its original configuration.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Factory Reset

A factory reset deletes all customer-specific data stored on a device and restores the device to its original configuration as it was at the time of shipping. This process erases configurations, log files, boot variables, core files, and credentials, including Federal Information Processing Standard (FIPS)-related keys. The data erasure performed during a factory reset aligns with the *clear* method defined in NIST Special Publication 800-88 Revision 1.

When do you need to do factory reset

You need to perform factory reset in the following scenarios

- Return Material Authorization (RMA) for a device

  When returning a device to Cisco for RMA, remove all customer-specific data before obtaining the RMA certificate.

- Recovering a compromised device

  If the key material or credentials stored on the device are compromised, reset the device to the factory configuration. Then, reconfigure the device.

What happens during a factory reset

1. During a factory reset, the device reloads and enters ROMMON mode.

   After the factory reset, the device removes all its environment variables, including the **MAC_ADDRESS** and the **SERIAL_NUMBER** variables, which are required to locate and load the software.

2. Perform a reset in ROMMON mode to automatically set the environment variables.

   The BAUD rate environment variable returns to its default value after a factory reset. Make sure that the BAUD rate and the console speed are the same at all times. Otherwise, the console becomes unresponsive.
3. After the system reset in ROMMON mode is complete, add Cisco IOS XE software either through an USB or TFTP.

The following table provides details about the data that is erased and retained during the factory reset process:

Table 3: Data erased and retained during a factory reset

Data erased	Data retained
All Cisco IOS XE software images, including the current boot image	Data from remote field-replaceable units (FRUs)
Crash information and logs	Value of the configuration register.
User data, startup and running configuration, and contents of removable storage devices, such as Serial Advanced Technology Attachment (SATA), Solid State Drive (SSD), or USB	—
Credentials such as FIPS-related keys	Credentials such as Secure Unique Device Identifier (SUDI) certificates, and public key infrastructure (PKI) keys.
Onboard Failure Logging (OBFL) logs	—
ROMMON variables added by a user.	—
Licenses	—

Secure Data Wipe

The device stores software images, device configurations, software logs, and operational history. Customer-specific data may be present in any of these areas, and this information can include details about the network architecture and design implemented by customers.

How to do secure data wipe

To perform secure data wipe, use the **all secure** option in the **factory-reset** command. This performs data sanitization and securely resets the device. The booted image of the device is retained.

When data sanitization is completed, the device reloads, and the device image is retained in flash if it was booted with an image from the flash.

Secure data wipe standards

The secure data wipe feature uses media sanitization guidelines as described in NIST SP 800-88 Rev. 1. NIST 800-88 is a standard that is published by the National Institute of Standards and Technology (NIST) that provides guidelines for media sanitization.

The PURGE standard within NIST 800-88 specifies methods to render data on storage media unrecoverable using laboratory techniques. When a device is sanitized using NIST 800-88 PURGE method, data cannot be recovered through simple non-invasive data recovery techniques or advanced laboratory techniques.

Guidelines to perform factory reset

- Back up all software images (including the current image), configurations, and personal data before you begin the factory reset process.
- Ensure that the power supply is uninterrupted during the factory reset process.
- Complete all In-Service Software Upgrade (ISSU) procedures before you begin the factory reset process.
- After you perform a factory reset, software patches you have installed will not be restored.
- If the **factory-reset** command is issued through a VTY session, the session is not restored after completion of the factory reset process.
- The **config** keyword of the **factory-reset** command is not supported when the switch is in stacking or Stackwise Virtual Link (SVL) mode.
- For modular chassis devices configured in high-availability mode, you must apply a factory reset to each supervisor module.

Perform a Factory Reset

Perform this procedure for a factory reset.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 Enter one of these commands based on the switch configuration:

- Switch

  factory-reset {**all** [**secure**] [**3-pass**] | **config** | **boot-vars**}
- Switch stack

  factory-reset {**all** [**secure 3-pass**] | **config** | **boot-vars** | **switch** {*switch-number* | **all** {**all** [**secure 3-pass**] | **config** | **boot-vars**}}

Example:

```
Device# factory-reset all
OR
Device# factory-reset all secure
```

Resets the device to its configuration at the time of its shipping.

Note
No system configuration is required to use the **factory reset** command.

- **all**: Erases all the content from the NVRAM, all the images, including the current boot image, boot variables, startup and running configuration data, and user data. We recommend that you use this option.
- **all secure**: Performs data sanitization and securely resets the device.

  Note
  - You can use the **all secure** option only on standalone devices.

    This option implements guidelines for media sanitization as described in NIST SP 800-88 Rev. 1.
  - The **factory-reset all secure** command initiates data sanitization. The booted image of the device is retained.

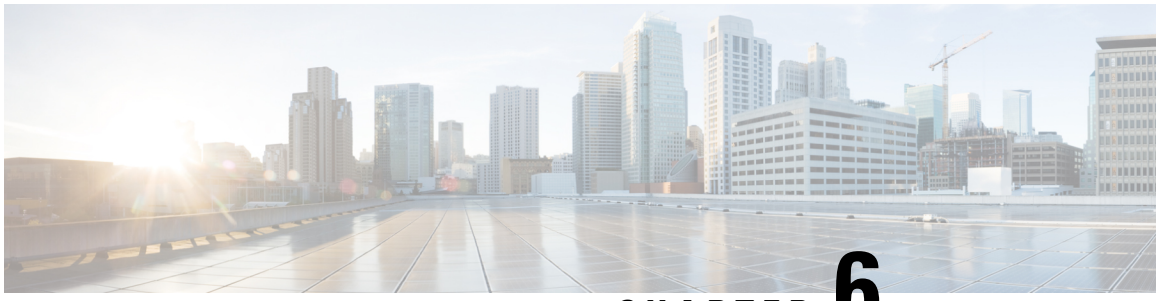
    When data sanitization is completed, the device reloads, and the device image is retained in flash if it was booted with an image from the flash.

- **secure 3-pass**: Erases all the content from the device with 3-pass overwrite.
  - Pass 1: Overwrites all addressable locations with binary zeroes.
  - Pass 2: Overwrites all addressable locations with binary ones.
  - Pass 3: Overwrites all addressable locations with a random bit pattern.

    Note
    This option takes approximately thrice the time taken to perform any other option.

- **config**: Resets the startup configurations.
- **boot-vars**: Resets the user-added boot variables.
- **switch** {*switch-number* | **all**}:
  - *switch-number*: Specifies the switch number.
  - **all**: Selects all the switches in the stack.

After the factory reset process is successfully completed, the device reboots and enters ROMMON mode.



CHAPTER 6

Secure Storage

- Feature history for secure storage, on page 29
- Secure Storage, on page 29
- Enable secure storage, on page 29
- Configuration example to verify the encryption status, on page 30

Feature history for secure storage

This table provides release and platform support information for the features explained in this module.

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

Release	Feature name and description	Supported platform
Cisco IOS XE 26.2.1ea	Secure Storage: Secure Storage feature support has been introduced.	Cisco C9550 Series Smart Switches
Cisco IOS XE 17.18.1	Secure Storage: This feature protects critical configuration information by encrypting it.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Secure Storage

The Secure Storage feature protects critical configuration information by encrypting it. This includes asymmetric key-pairs, pre-shared secrets, the type 6 password encryption key, and specific credentials. To ensure security, an instance-unique encryption key is stored within the hardware trust anchor, preventing unauthorized access or compromise.

Enable secure storage

Secure storage is disabled by default. Perform this task to enable secure storage.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 [**no**] **service private-config-encryption**

Example:

```
Device(config)# service private-config-encryption
```

Enables secure storage on the device.

Use the no form of this command to disable secure storage on the device

Note
When secure storage is disabled, all the user data is stored in plain text in the NVRAM.

Step 4 **end**

Example:

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

Returns to privileged EXEC mode.

Step 5 **write memory**

Example:

```
Device# write memory
```

Encrypts the private-config file and saves the file in an encrypted format.

Configuration example to verify the encryption status

The following is a sample output to verify if encryption status is enabled.

```
Device# show parser encrypt file status
Feature:            Enabled
File Format:        Cipher text
Encryption Version: ver1
```

In this example, the file format is displayed as *Cipher text*. This indicates that file is encrypted and secure storage is enabled.