



## Boot Integrity Visibility

---

- [Information About Boot Integrity Visibility, on page 1](#)
- [Verifying the Software Image and Hardware, on page 2](#)
- [Verifying Platform Identity and Software Integrity, on page 3](#)
- [Verifying Image Signing, on page 6](#)
- [Additional References for Boot Integrity Visibility, on page 7](#)
- [Feature History for Boot Integrity Visibility, on page 7](#)

### Information About Boot Integrity Visibility

Boot Integrity Visibility allows Cisco's platform identity and software integrity information to be visible and actionable. Platform identity provides the platform's manufacturing installed identity. Software integrity exposes boot integrity measurements that can be used to assess whether the platform has booted trusted code.

During the boot process, the software creates a checksum record of each stage of the bootloader activities.

You can retrieve this record and compare it with a Cisco-certified record to verify if your software image is genuine. If the checksum values do not match, you may be running a software image that is either not certified by Cisco or has been altered by an unauthorized party.

### Image Signing and Bootup

The Cisco build servers generate the Cisco IOS XE images. Cisco IOS XE images use the Abraxas image signing system to sign these images securely with the Cisco private RSA keys.

When you copy the Cisco IOS XE image onto a Catalyst 9000 Series Switch, Cisco's ROMMON Boot ROM verifies the image using Cisco release keys. These keys are public keys that correspond to the Cisco release private key that is stored securely on the Abraxas servers. The release key is stored in the ROMMON.

Catalyst 9000 Series Switches support secure boot feature. Secure boot serves as a hardware trust anchor which validates the ROMMON software to ensure that the ROMMON software is not tampered with.

The Cisco IOS XE image is digitally signed during the build time. An SHA-512 hash is generated over the entire binary image file, and then the hash is encrypted with a Cisco RSA 2048-bit private key. The ROMMON verifies the signature using the Cisco public key. If the software is not generated by a Cisco build system, the signature verification fails. The device ROMMON rejects the image and stops booting. If the signature verification is successfully, the device boots the image to the Cisco IOS XE runtime environment.

The ROMMON follows these steps when it verifies a signed Cisco IOS XE image during the bootup:

1. Loads the Cisco IOS XE image into the CPU memory.
2. Examines the Cisco IOS XE package header.
3. Runs a non-secure integrity check on the image to ensure that there is no unintentional file corruption from the disk or TFTP. This is performed using a non-cryptographic SHA-1 hash.
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 2048-bit public release key.
6. Performs the Code Signing validation by calculating the SHA-512 hash of the Cisco IOS XE package and compares it with the Code Signing signature. The signed package is now validated.
7. Examines the Cisco IOS XE package header to validate the platform type and CPU architecture for compatibility.
8. Extracts the Cisco IOS XE software from the Cisco IOS XE package and boots it.



**Note** In above process, step 3 is a non-cryptographic check of the image which is intended to confirm the image against inadvertent corruption due to disk errors, file transfer errors, or copying errors. This is not part of the image code signing. This check is not intended to detect deliberate image tampering.

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

## Verifying the Software Image and Hardware

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



**Note** On executing the following commands, 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.

The messages **% Error retrieving SUDI certificate** and **% Error retrieving integrity data** signify a real CLI failure.

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>show platform sudi certificate</b> [ <b>sign</b> [ <b>nonce</b> ] ]  <b>Example:</b>	Displays checksum record for the specific SUDI.  • (Optional) <b>sign</b> - Show signature

	Command or Action	Purpose
	Device# <code>show platform sudi certificate sign nonce 123</code>	<ul style="list-style-type: none"> <li>• (Optional) <b>nonce</b> - Enter a nonce value</li> </ul>
<b>Step 2</b>	<p><code>show platform integrity [sign [nonce nonce ]]</code></p> <p><b>Example:</b></p> <p>Device# <code>show platform integrity sign nonce 123</code></p>	<p>Displays checksum record for boot stages.</p> <ul style="list-style-type: none"> <li>• (Optional) <b>sign</b> - Show signature</li> <li>• (Optional) <b>nonce</b> - Enter a nonce value</li> </ul>

# Verifying Platform Identity and Software Integrity

## Verifying 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 and 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.

```
Device# show platform sudi certificate sign nonce 123
-----BEGIN CERTIFICATE-----
MIIDQzCCAiugAwIBAgIQX/h7KctU3I1CoxW1aMmt/zANBgkqhkiG9w0BAQUFADA1
MRYwFAYDVQQKEw1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDAXNjbyBSb290IENB
IDIwNDgwHhcNMDQwNTE0MjAxNzEyWhcNMjkwNTE0MjAyNTQyWjA1MRYwFAYDVQQK
Ew1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDAXNjbyBSb290IENBIDIwNDgwggEg
MA0GCSCqGSIB3DQEBAQUAA4IBDQAwggEIAoIBAQCwrmrmp68Kd6fichba0ZmKUEIhH
xmJVhEAYv8CrLqUccda8bnuoqrpu0hWIESEWdovyD0My5jOAmAHBKeN8hF570YQXJ
FcjPftolYYmUQ6iEqDGYeJu5Tm8sUxJsZr2tKyS7McQr/4NEb7Y9JHcJ6r8qqB9q
VvYgDxFU14F1pyXOWWqCZe+36ufijXWLBvLdT6ZeYpzPEApk0E5tzivMM/VgpdSdH
jWn0f84bcN5WgyDWbs2mAag8EtKpP6BrXruOIIt6ke01a06g58QBdKhTCytKmg9l
Eg6CTY5j/e/rmxrbU6YTYK/CfdfHbBcl1HP7R2RQgYCUTOG/rksc35LTLgXfAgED
o1EwTzALBgNVHQ8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUUJ/PI
FR5umgIJFq0roIlgX9p7L6owEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQEF
BQADggEBAJ2dhISjQa18dwy3U8pORFbi71R803UXHOjgXkHlTv5M0hmBvrbW7hmW
Yqpao2TB9k5UM8Z3/sUcuuVdJcr18JOagxEu5sv4dEX+5wW4q+ffY0vhn4TauYuX
cB7w4ovXsNgOnbFpliqr61JT37mjpXYgy81WhJdtSd9i7rp77rMKSsH0T8lasz
Bvt9YAretIpjsJyp8qS5UwGH0GikJ3+r/+n6yUA4iGe0OcaEb1fJU9u6ju7AQ7L4
CYNu/2bPPu8Xs1gYJQk0XuPL1hS27PKSb3TkL4Eq1ZKR4OCXPDJoBYVL0fdX41Id
kxpUnwVwwEpxYB5DC2Ae/qPOgRnhCzU=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEPDCCAySgAwIBAgIKYQ1ufQAAAAAADANBgkqhkiG9w0BAQUFADA1MRYwFAYD
VQQKEw1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDAXNjbyBSb290IENBIDIwNDgw
HhcNMTUwNjMwMjE0MjAxNzEyWhcNMjkwNTE0MjAyNTQyWjA1MRYwFAYDVQQK
Ew1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDAXNjbyBSb290IENBIDIwNDgwggEg
MA0GCSCqKCAQEAOM5l3THIx9tN/hS5qR/6UZRpdd+9aE2JbFknjht6gfHKd477AKS
5XAtUs5oxDYVt/zEbslZq3+LR6qrqKKQV6JYvH05UYLBqCj38s76NLk53905Wzp
9pRcmRCPuX+a6tHF/qRuOiJ44mdeDYzo3qPCpxzprWJDPc1M4iYKHuMMQmqmgm+
xghHIooWS80BOccdiynEbeP5rZ7qRueWKmpl1TiI3WdBNjZjnpfjg66F+P4SaDkGb
BXDgJ13oVeF+EyFWLrFjj97fL2+8oauV43Qrvnf3d/GfqXj7ew+z/sX1XtEOjSXJ
URsyMEj53Rdd9tJwHky8neapszS+r+kdVQIDAQABo4IBWjCCAVYwCwYDVR0PBAQD
```





**Note** Boot integrity hashes are not MD5 hashes. For example, if you run `verify /md5 cat9k_iosxe.16.10.01.SPA.bin` command for the bundle file, the hash will not match.

The following is a sample output of the `show platform integrity sign nonce 123` command in install mode. This output includes measurements of each installed package file.

```
Device# show platform integrity sign nonce 123
Platform: WS-XC7R
Boot 0 Version: MA1004R06.1604052017
Boot 0 Hash: A99EF9F31CE3F3F8533055407F1C88C62176E667E4E1DA0649EAA7A1282F205E0A
Boot Loader Version: System Bootstrap, Version 16.8.0.7, DEVELOPMENT SOFTWARE
Boot Loader Hash:
942C2511D0EB10C8F5EC8B3ED529A5F2D210C4154434C6A591BF5553B06CBBE2039DADD949C05722CABEB1429C41737CFC2C593A814FC87F6FBA0E9A0ADB09B
OS Version: 16.10.01
OS Hashes:
cat9k-cc_srdriver.16.10.01.SPA.pkg :
D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0
cat9k-espsbase.16.10.01.SPA.pkg :
3EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEF43
cat9k-guestshell.16.10.01.SPA.pkg :
B0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03E
cat9k-rpbase.16.10.01.SPA.pkg :
4057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C6
cat9k-rpboot.16.10.01.SPA.pkg :
AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057
cat9k-sipbase.16.10.01.SPA.pkg :
9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A
cat9k-sipspa.16.10.01.SPA.pkg :
E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673
cat9k-srdriver.16.10.01.SPA.pkg :
4FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E211
cat9k-webui.16.10.01.SPA.pkg :
CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7
cat9k-wlc.16.10.01.SPA.pkg :
AA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAA
PCR0: A32CFED4F960494BC1311F7A31B52D5DE90FF501932670CD43AE6DBAD8735052
PCR8: D2F8474CD82072464C11D7F7A3D5C37D078A8AA832D94B1B12E01BF400E0BBB4
Signature version: 1
Signature:
A99EF9F31CE3F3F8533055407F1C88C62176E667E4E1DA0649EAA7A1282F205E0A
```

The following is a sample output of the `show platform integrity sign nonce 123` command in bundle mode. This output includes measurements of the bundle file and each installed package.

```
Device# show platform integrity sign nonce 123
Platform: WS-XC7R
Boot 0 Version: MA1004R06.1604052017
Boot 0 Hash: A99EF9F31CE3F3F8533055407F1C88C62176E667E4E1DA0649EAA7A1282F205E0A
Boot Loader Version: System Bootstrap, Version 16.8.0.7, DEVELOPMENT SOFTWARE
Boot Loader Hash:
942C2511D0EB10C8F5EC8B3ED529A5F2D210C4154434C6A591BF5553B06CBBE2039DADD949C05722CABEB1429C41737CFC2C593A814FC87F6FBA0E9A0ADB09B
OS Version: 16.10.01
OS Hashes:
cat9k_iosxe.16.10.01.SPA.bin :
F4CAD08FE1EF841C3A2E3ED8540829F08F3CBA9336F38E45669D4D8B15AD15E365B922AC8B4DC0D5B63E2806D6A1BDAB7839DD9DC8D7E366A49ED648C113440
cat9k-cc_srdriver.16.10.01.SPA.pkg :
D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0
cat9k-espsbase.16.10.01.SPA.pkg :
3EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEF43
cat9k-guestshell.16.10.01.SPA.pkg :
```

```

B0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03E
cat9k-rpbase.16.10.01.SPA.pkg :
4057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C6
cat9k-rpboot.16.10.01.SPA.pkg :
AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057
cat9k-sipbase.16.10.01.SPA.pkg :
9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A
cat9k-sipspa.16.10.01.SPA.pkg :
E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673
cat9k-srdriver.16.10.01.SPA.pkg :
4FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E211
cat9k-webui.16.10.01.SPA.pkg :
CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7
cat9k-wlc.16.10.01.SPA.pkg :
AA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAAA7ED0AE935CB0BD84E0D0D155C1DEFB03EB0C64057AD6A9673E2114FA7CCAA
PCR0: A32CFED4F960494BC1311F7A31B52D5DE90FF501932670CD43AE6DBAD8735052
PCR8: D2F8474CD82072464C11D7F7A3D5C37D078A8AA832D94B1B12E01BF400E0BBB4
Signature version: 1
Signature:

```

## Verifying Image Signing

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

```

switch:boot flash:packages.conf
boot: attempting to boot from [flash:packages.conf]
boot: reading file packages.conf
#
Performing Integrity Check ...
boot: parsed image from conf file: cat9k-rpboot.17.02.01.SSA.pkg

```

Loading image in Verbose mode: 1

```

Image Base is: 0x100099000
Image Size is: 0x2C83487
Package header rev 3 structure detected
Package type:30001, flags:0x0
IsoSize = 0
Parsing package TLV info:
000: 0000000900000001D4B45595F544C565F - KEY_TLV_
010: 5041434B4147455F434F4D5041544942 - PACKAGE_COMPATIB
020: 494C49545900000000000000090000000B - ILITY
030: 4652555F52505F545950450000000009 - FRU_RP_TYPE
040: 000000184B45595F544C565F5041434B - KEY_TLV_PACK
050: 4147455F424F4F544152434800000009 - AGE_BOOTARCH
060: 000000E415243485F693638365F5459 - ARCH_i686_TY
070: 5045000000000000090000000144B45595F - PE KEY_
080: 544C565F424F4152445F434F4D504154 - TLV_BOARD_COMPAT
090: 0000009000000010424F4152445F6361 - BOARD_ca
0A0: 74396B5F545950450000000900000018 - t9k_TYPE
0B0: 4B45595F544C565F43525950544F5F4B - KEY_TLV_CRYPTO_K
0C0: 4559535452494E470000000900000004 - EYSTRING

```

TLV: T=9, L=29, V=KEY\_TLV\_PACKAGE\_COMPATIBILITY

```

TLV: T=9, L=11, V=FRU_RP_TYPE
TLV: T=9, L=24, V=KEY_TLV_PACKAGE_BOOTARCH
TLV: T=9, L=14, V=ARCH_i686_TYPE
TLV: T=9, L=20, V=KEY_TLV_BOARD_COMPAT
TLV: T=9, L=16, V=BOARD_cat9k_TYPE
TLV: T=9, L=24, V=KEY_TLV_CRYPTO_KEYSTRING
TLV: T=9, L=4, V=none
TLV: T=9, L=11, V=CW_BEGIN=$$
TLV: T=9, L=17, V=CW_FAMILY=$cat9k$
TLV: T=9, L=74, V=CW_IMAGE=$cat9k-rpboot.17.02.01.SSA.pkg$
TLV: T=9, L=20, V=CW_VERSION=$17.2.01$
IOS version is 17.2.1
TLV: T=9, L=53, V=CW_FULL_VERSION=$17.2.01.0.869.1580816579..Amsterdam$
TLV: T=9, L=52, V=CW_DESCRIPTION=$Cisco IOS Software, IOS-XE Software$
TLV: T=9, L=9, V=CW_END=$$
Found DIGISIGN TLV type 12 length = 392
RSA Self Test Passed

Expected hash:
DDAF35A193617ABACC417349AE204131
12E6FA4E89A97EA20A9EEEE64B55D39A
2192992A274FC1A836BA3C23A3FEEBBD
454D4423643CE80E2A9AC94FA54CA49F

Obtained hash:
DDAF35A193617ABACC417349AE204131
12E6FA4E89A97EA20A9EEEE64B55D39A
2192992A274FC1A836BA3C23A3FEEBBD
454D4423643CE80E2A9AC94FA54CA49F
Sha512 Self Test Passed
Found package arch type ARCH_i686_TYPE
Found package FRU type FRU_RP_TYPE
Performing Integrity Check ...

RSA Signed DEVELOPMENT Image Signature Verification Successful.
    
```

## Additional References for Boot Integrity Visibility

### Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	<i>Command Reference (Catalyst 9500 Series Switches)</i>

## Feature History for Boot Integrity Visibility

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

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

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.8.1a	Boot Integrity Visibility	<p>Boot Integrity Visibility allows Cisco's platform identity and software integrity information to be visible and actionable. Platform identity provides the platform's manufacturing installed identity.</p> <p>Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.</p>
Cisco IOS XE Fuji 16.9.1	Boot Integrity Visibility	<p>Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.</p>

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