



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](#)

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 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.

Switches support boot integrity visibility feature. Boot integrity visibility 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-secure 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-secure 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
Step 1	show platform sudi certificate [sign [nonce]] Example:	Displays checksum record for the specific SUDI. • (Optional) sign - Show signature

	Command or Action	Purpose
	Device# show platform sudi certificate sign nonce 123	• (Optional) nonce - Enter a nonce value
Step 2	show platform integrity [sign [nonce nonce]] Example: Device# show platform integrity sign nonce 123	Displays checksum record for boot stages. • (Optional) sign - Show signature • (Optional) nonce - Enter a nonce value

Verifying Platform Identity and Software Integrity

Verifying Platform Identity

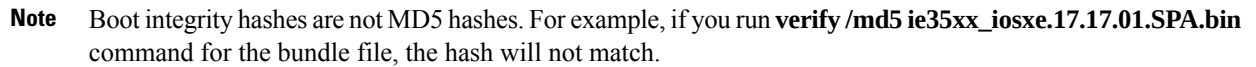
The following example displays the Secure Unique Device Identity (SUDI) chain in PEM format. Encoded into the SUDI is the Product ID and Serial Number of each individual device such that the device can be uniquely identified on a network of thousands of devices. The first certificate is the Cisco Root 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
MA0GCSqGSIb3DQEBAQUAA4IBDQAwggEIAoIBAQCwmrmrp68Kd6ficba0ZmKUEIhH
xmJVhEAYv8CrLqUccda8bnuoqrpu0hWISewdovyD0My5jOAmAHBKeN8hF570YQXJ
FcjPftolYYmUQ6iEqDGYeJu5Tm8sUxJsZr2tKyS7McQr/4NEb7Y9JHcJ6r8qqB9q
VvYgDxFU14FlpyXOWWqCZe+36ufijXWLBvLdT6ZeYpzPEApk0E5tzivMM/VggsdH
jWn0f84bcN5WgyDwbs2mAag8EtKpP6BrXruOIIt6keO1aO6g58QBdKhTCytKmg9l
Eg6CTY5j/e/rmxrbU6YTYK/CfdfHbBcl1HP7R2RQgYCUTOG/rksc35LtLgXfAgED
o1EwTzALBgNVHQ8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJ/PI
FR5umgIJFq0roIlGx9p7L6owEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQEF
BQADggEBAJ2dhISjQa18dwy3U8pORFbi71R803UXHOjgxkhLtv5M0hmBVrBW7hmW
Yqpao2TB9k5UM8Z3/sUcuuVdJcr18JOagxEu5sv4dEX+5wW4q+ffY0vhN4TauYuX
cB7w4ovXsNgOnbFpliqRe61JT37mjpXYgyC81WhJDtSd9i7rp77rMKSsH0T8lasz
Bvt9YAretIpjsJyp8qS5UwGH0GikJ3+r/+n6yUA4iGe0OcaEb1fJU9u6ju7AQ7L4
CYNu/2bPPu8XslgYJQk0XuPL1hS27PKSb3TkL4Eq1ZKR4OCXPDJoBYVL0fdX41Id
kxpUnwVwEpxYB5DC2Ae/qPOgRnhCzU=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEPDCCAYsGAWIBAgIKYQlufQAAAAAADANBgkqhkiG9w0BAQUFADA1MRYwFAYD
VQQKEw1DaXNjbyBTeXN0ZW1zMRSwGQYDVQQDExJDAXNjbyBSb290IENBIDIwNDgw
HhcnMTEwNjMwMTc1NjU3WhcNMjkwNTE0MjAyNTQyWjA1MRYwFAYDVQQKEw1DaXNj
bzEVMBMGA1UEAxMMQUNUMiBTvURJENBMTIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEAOm5l3THiXA9tN/hS5qR/6UZRpdd+9aE2JbFkNjht6gfHKd477AKS
5XAtUs5oxDYvt/zEbslZq3+LR6qrqKKQVu6JYvH05UYLBqCj38s76NLk53905Wzp
9pRcmRCPuX+a6tHF/qRuOiJ44mdeDYzo3qPCpxzprWJDpC1M4iYKHuMMQmqmgmg+
xghHiooWS80BOcdiyNEbeP5rZ7qRueWKmpl1TiI3WdBNjZjnpfjg66F+P4SaDkGj
BXDgJ13oVeF+EyFWLrFjj97fL2+8oauV43Qrvnf3d/GfqXj7ew+z/sX1XtEOjSXJ
URsyMEj53Rdd9tJwHky8neapszS+r+kdVQIDAQABo4IBWjCCAVYwCwYDVR0PBAQD
```

```
Signature version: 1
Signature:
```

```
[linux-host:~]openssl x509 -in sudicert.pem -subject -noout
subject= /serialNumber=PID:IE35xx-24T-4G SN:FDO1946BG05/O=Cisco/OU=ACT-2 Lite
SUDI/CN=IE35xx-24T-4G
```

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.



```

Device#show platform integrity sign nonce 123
Platform: IE35xx-24T-4G
Boot 0 Version: SBOOT0.v27
Boot 0 Hash:
EE98DCD0D6AEA85C8891039F649664FCC3CF709CCFC7A6F248C9D5BA8463528F
Boot Loader Version: System Bootstrap, Version 10.2, DEVELOPMENT SOFTWARE
Boot Loader Hash:
9220B7EA15A79EB9E3731A1FF2313C996E21032F8A1EF49353E742765E40EE53E7B3C50E4121C00B2D5567864FE15300FF67F63F1A6B
OS Version: 17.17.01
OS Hashes:
ie35xx_lite-rpbase.17.17.01.SPA.pkg :
DD155C1DFB03EB0C6405AD6A9673E2114FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB0C6405AD6A9673E2114FA7CCAA7ED0AE935CB0B84E0
ie35xx_lite-rpboot.17.17.01.SPA.pkg :
AD6A9673E2114FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB0C6405AD6A9673E2114FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB0C6405
ie35xx_lite-srdriver.17.17.01.SPA.pkg :
4FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB0C6405AD6A9673E2114FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB0C6405AD6A9673E211
ie35xx_lite-webui.17.17.01.SPA.pkg :
CCAA7ED0AE935CB0B84E0DD155C1DFB03EB0C6405AD6A9673E2114FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB0C6405AD6A9673E2114FA7
ie35xx_wlc.17.17.01.SPA.pkg :
AA7ED0AE935CB0B84E0DD155C1DFB03EB0C6405AD6A9673E2114FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB0C6405AD6A9673E2114FA7CCA
PCR0: 750E5D2EDAE6E3A68050638E0BFD8619BE4EA13066025D39DF79408719F5177E
PCR8: EB6E739A63F53E703B6CDAF3F6188833CEF6D32E2F726006B9AA34E1E73048C4
Signature version: 1
Signature:

```

```
Device# show platform integrity sign nonce 123
Platform: IE35xx-24T-4G
Boot 0 Version: SBOOT0.v27
Boot 0 Hash:
EE98DCD0D6AEA85C8891039F649664FCC3CF709CCFC7A6F248C9D5BA8463528F
Boot Loader Version: System Bootstrap, Version 10.2, DEVELOPMENT SOFTWARE
Boot Loader Hash:
922087EA15A79FE9E37311A1FD231C999F21032F8A1E7E4935DE742765E40FE537E7B350F84121C00B2D5567864FE15530AFF67F63F1A6B
OS Version: 17.17.01
OS Hashes:
ie35xx_lite_iosxe.17.17.01.SPA.bin :
F42D08E1EF41C42E3D8540829F08F3FA9363FE45669D408B15AD15365B22AC84D005B63E2806A1EDF839D908D7E366A9FD68C11340
ie35xx_lite-rpbase.17.17.01.SPA.pkg :
DD155C1DEFB03EB0C6405AD6A9673E2114FA7CCAA7ED0FE95CB0B84EDDD155C1DEFB03EB0C6405AD6A9673E2114FA7CCAA7ED0FE95CB0B84ED
ie35xx_lite-rpboot.17.17.01.SPA.pkg :
```

```

AD6A9673E2114FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB064057AD6A9673E2114FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB064057
ie35xx_lite-srdriver.17.17.01.SPA.pkg :
4FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB064057AD6A9673E2114FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB064057AD6A9673E211
ie35xx_lite-webui.17.17.01.SPA.pkg :
CCAA7ED0AE935CB0B84E0DD155C1DFB03EB064057AD6A9673E2114FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB064057AD6A9673E2114FA7
ie35xx-wlc.17.17.01.SPA.pkg :
AA7ED0AE935CB0B84E0DD155C1DFB03EB064057AD6A9673E2114FA7CCAA7ED0AE935CB0B84E0DD155C1DFB03EB064057AD6A9673E2114FA7CCA
PCR0: 750E5D2EDAE6E3A68050638E0BFD8619BE4EA13066025D39DF79408719F5177E
PCR8: EB6E739A63F53E703B6CDAF3F6188833CEF6D32E2F726006B9AA34E1E73048C4
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: ie35xx-rpboot.17.17.1.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: 494C495459000000000000090000000B - ILITY
030: 4652555F52505F545950450000000009 - FRU_RP_TYPE
040: 000000184B45595F544C565F5041434B - KEY_TLV_PACK
050: 4147455F424F4F544152434800000009 - AGE_BOOTARCH
060: 0000000E415243485F693638365F5459 - ARCH_i686_TY
070: 5045000000000009000000144B45595F - PE KEY_
080: 544C565F424F4152445F434F4D504154 - TLV_BOARD_COMPAT
090: 00000009000000010424F4152445F6361 - BOARD_ca
0A0: 74396B5F545950450000000900000018 - t9k_TYPE
0B0: 4B45595F544C565F43525950544F5F4B - KEY_TLV_CRYPTOK
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_ie35xx_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=$$
TLV: T=9, L=17, V=CW_FAMILY=$ie35xx$
TLV: T=9, L=74, V=CW_IMAGE=$ie35xx-rpboot.17.17.1.SSA.pkg$
TLV: T=9, L=20, V=CW_VERSION=$X.X.X$
IOS version is X.X.X
TLV: T=9, L=53, V=CW_FULL_VERSION=$17.17.1$
TLV: T=9, L=52, V=CW_DESCRIPTION=$Cisco IOS Software, IOS-XE Software$
TLV: T=9, L=9, V=CW_END=$$
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.
```

