



## Boot Integrity Visibility

---

- [Finding Feature Information, on page 1](#)
- [Information About Boot Integrity Visibility, on page 1](#)
- [Verifying the software image and hardware, on page 1](#)
- [Verifying Platform Identity and Software Integrity, on page 2](#)

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

### 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, and 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 boot loader 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.

### Verifying the software image and hardware

This task describes how to retrieve the checksum record that was created during 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. It is recommended to wait for few minutes and then try the command again.

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

### SUMMARY STEPS

1. `show platform sudi certificate [sign [nonce nonce]]`
2. `show platform integrity [sign [nonce nonce]]`

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<p><code>show platform sudi certificate [sign [nonce nonce]]</code></p> <p><b>Example:</b></p> <pre>Device# show platform sudi certificate sign nonce 123</pre>	<p>Displays checksum record for the specific SUDI.</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>
<b>Step 2</b>	<p><code>show platform integrity [sign [nonce nonce]]</code></p> <p><b>Example:</b></p> <pre>Device# show platform integrity sign nonce 123</pre>	<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. 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
IDIwNDgwHhcNMDQwNTE0MjAxNzEyWhcNMjkwNTE0MjAyNTQyWjAlMRYwFAYDVQQK
Ew1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDaXNjbyBSb290IENBIDIwNDgwggEg
MA0GCSqGSIb3DQEBAQUAA4IBDQAwggEIAoIBAQCwmrmrp68Kd6ficba0ZmKUeIhH
xmJVhEAYv8CrLqUccda8bnuoqrpu0hWISEWdovyD0My5jOAmahBKeN8hF570YQXJ
FcjPFto1YYmUQ6iEqDGYeJu5Tm8sUxJsZr2tKyS7McQr/4NEb7Y9JHcJ6r8qqB9q
VvYgDxFU14F1pyXOWWqCZe+36ufijXWlLvLdT6ZeYpzPEApk0E5tzivMW/VgpSdh
```



```
BFACD6BE9CF9C84C961FACE9FA0FEE64D85AE4FA0086969D0702C536ABDB8FBFDC47
C14C17D02FEBF4F7F5BB24D2932FA876F56B4C07816270AA0B4195C53D975C85AEAE
3A74F2DBF293F52423ECB7B8539667080A9C57DA3E4B08B2B2CA623B2CBAF7080A0A
EB09B222E5B756970A3AA27E0F1D17C8A243
```

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:WS-C3650-12X48UQ SN:FD01946BG05/O=Cisco/OU=ACT-2 Lite
SUDI/CN=WS-C3650-12X48UQ
```

### Verifying 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.

Device #**show platform integrity sign nonce 456**

```
Platform: WS-C3650-12X48UQ
Boot Loader Version: CAT3K_CAA Boot Loader (CAT3K_CAA-HBOOT-M) Version 4.16, engineering
software (D)
Boot Loader Hash: DB5A686E9F4CE358481DE3AF8B9C762F0A604E3B4764DF2A351F176E3D7
D3C60EB85C02906BD8CF28228C0DFC2AA8960CAFE6675D696E4ABA0CD687C0609E7E2
Boot 0 Version: F01062R15.0508d68fa2015-09-15
Boot 0 Hash: 6EF15CD54D3C66A8B644194A67B7ED57044C8C2E0EECB69736A7FFEC1F6D0EAD
OS Version: 2016-10-18_10.57_mundru
OS Hash: 4C85AEC88DAA49D940BBF65B1F17269F55C8D98DEFB4140F981923AA961140293E1
3B3E6E68CE3F8ED7F596CD858ACDD4BEF6538F59C1E243C351353026E6CD
PCR0: 90214167AAF35C06B2AC97292596E5669EAB72578FCDAD0B91746683BAA7B2B0
PCR8: FC2CE1BAC397F97008936DF372A2218BB16A798222B8FF55A7B6AEDA8018EDF5
Signature version: 1
Signature:
632A724F1AB6ADE134F6B0E8724D2052B3157F45B47E547763EE224A848E807CD737600587FF68
2526A8FE354A116CC9EDEBD9C659B9927336542EE4295084368327D01BD22AB4849BB3C007B6EB
B67708685FD6BC85DD045431E19A389FEB358894D4FBCF7C0FC960AC9133B61099DFD507F316C1
BF82F7F98687C7E7E8F99335DC1A95BD511B0B8DCB0CA909828F9EFBDF18847930392A8E3D072D
F3D90536880BAE9B7D7CF0E301D3F5AF16E7517FC2700E2F75911B836D6559A18E15B4CF452555
91656DF22DF73392F777AEB796BCF9AC046C581ADEF19CA48A98F620BB58A79B32DA8B3BFB1CF
8399468A096E2F0C54B8B3ECD15EE3FE2C5ABDB5A029
```

The optional RSA 2048 signature is produced with the SUDI private key and can be verified with the SUDI public key contained in the SUDI certificate. The signature across PCR values, the signature version and the user-provided nonce is displayed.

```
RSA PKCS# 1 v1.5 Sign { <Nonce (UINT64)> || <Signature Version (UINT32)> || <PCR0 (32 bytes)>
|| <PCR8 (32 bytes)> }
```

Cisco management solutions are equipped with the ability to interpret the above output, compare the results against published Cisco values, and to verify the signature.

