IOS Self-Signed Certificate Expiration on January 1 2020

Contents

Introduction

This document describes errors caused by the expiration of the Self-Signed Certificates (SSC) on Cisco software systems and provides workarounds.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Self-Signed Certificates (SSC)
- Cisco IOS® version 12.x and later

Components Used

The components are the software systems affected by the expiration of the SSC.

All Cisco IOS and Cisco IOS XE systems that use a Self-Signed Certificate, that do not have the Cisco bug ID CSCvi48253 fix, or that did not have the Cisco bug ID CSCvi48253 fix when the SSC was generated. This includes:

- All Cisco IOS 12.x
- All Cisco IOS 15.x prior to 15.6(3)M7, 15.7(3)M5, 15.8(3)M3, 15.9(3)M
- All Cisco IOS XE prior to 16.9.1

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background

Note: This document contains the contents of FN40789 , along with additional context, examples, updates, and Q&As.

At 00:00 on 1 Jan 2020 UTC, all Self-Signed Certificates (SSC) generated on Cisco IOS and Cisco IOS XE systems were set to expire, unless the system ran a fixed version of Cisco IOS and Cisco IOS XE when the SSC was generated. After that time, unfixed Cisco IOS systems are unable to generate new SSCs. Any service that relies on these Self-Signed Certificates to establish or terminate a secure connection does not work after the certificate expires.

This issue affects only Self-Signed Certificates that were generated by the Cisco IOS or Cisco IOS XE device and applied to a service on the device. Certificates that were generated by a Certificate Authority (CA), which includes those certificates generated by the Cisco IOS CA feature, are not impacted by this issue.

Certain features in Cisco IOS and Cisco IOS XE software rely on digitally signed X.509 certificates for cryptographic identity validation. These certificates are either generated by an external third-party CA, or on the Cisco IOS or Cisco IOS XE device itself as a Self-Signed Certificate. Affected Cisco IOS and Cisco IOS XE software releases set the Self-Signed Certificate expiration date to 2020-01-01 00:00:00 UTC. After this date, the certificate expires and is invalid.

Services that can rely on a Self-Signed Certificate include:

General Features

- HTTP Server over TLS (HTTPS) - HTTPS produces an error in the browser which indicates that the certificate is expired.
- SSH Server - Users who use X.509 certificates to authenticate the SSH session can fail to authenticate. (The use of X.509 certificates is rare. Username/password authentication and public/private key authentication are not affected.)
- RESTCONF - RESTCONF connections can fail.

Collaboration Features

- Session Initiation Protocol (SIP) over TLS
- Cisco Unified Communications Manager Express (CME) with encrypted signaling enabled
- Cisco Unified Survivable Remote Site Telephony (SRST) with encrypted signaling enabled
- Cisco IOS dspfarm resources (Conference, Media Termination Point, or Transcoding) with encrypted signaling enabled
- Skinny Client Control Protocol (SCCP) Telephony Control Application (STCAPP) ports configured with encrypted signaling
- Media Gateway Control Protocol (MGCP) and H.323 call signaling over IP security (IPSec) without a pre-shared key
- Cisco Unified Communications Gateway Services API in Secure Mode (that use HTTPS)

Wireless Features

- LWAPP/CAPWAP connections between older Cisco IOS access points (manufactured in 2005 or earlier) and Wireless LAN Controller. See Cisco Field Notice FN63942 for more details.

Problem

An attempt to generate a Self-Signed Certificate on an affected Cisco IOS or Cisco IOS XE software release after 2020-01-01 00:00:00 UTC results in this error:

```
../cert-c/source/certobj.c(535) : E_VALIDITY : validity period start later than end
```

Any services that rely on the Self-Signed Certificate do not function. For example:

- SIP over TLS calls do not complete.
- Devices registered to Cisco Unified CME with encrypted signaling enabled no longer function.
- Cisco Unified SRST with encrypted signaling enabled does not allow devices to register.
- Cisco IOS dspfarm resources (Conference, Media Termination Point, or Transcoding) with encrypted signaling enabled no longer register.
- STCAPP ports configured with encrypted signaling no longer register.
- Calls through a gateway that MGCP or H.323 call signaling over IPSec without a pre-shared key can fail.
- API calls that use the Cisco Unified Communications Gateway Services API in Secure Mode (that use HTTPS) can fail.
- RESTCONF can fail.
- HTTPS sessions to manage the device display a browser warning, which indicates that the certificate has expired.
- AnyConnect SSL VPN sessions fails to establish or report an invalid certificate.
- IPSec connections can fail to establish.

How to Identify Affected Products

Note: To be impacted by this field notice, a device must have a Self-Signed Certificate defined *and* the Self-Signed Certificate must be applied to one or more features as outlined below. Presence of a Self-Signed Certificate alone does not impact the operation of the device when the certificate expires and does not require immediate action. **To be impacted, a device must meet the criteria in both Step 3 and Step 4 below.**

To determine if you use a Self-Signed Certificate:

1. Enter the show running-config | begin crypto command on your device.
2. Look for the crypto PKI trust-point configuration.
3. In the crypto PKI trust-point configuration, look for the trust-point enrollment configuration. The trust-point enrollment must be configured for " **selfsigned** " to be impacted. Additionally, the Self-Signed Certificate must also appear in the configuration. Notice that the trust-point name does not contain the words "self-signed" as shown in this next example.

```
<#root>

crypto pki trust-point TP-self-signed-XXXXXXXX
  enrollment

selfsigned

  subject-name cn=IOS-Self-Signed-Certificate-662415686   revocation-check none
rsakeypair TP-self-signed-662415686 ! ! crypto pki certificate chain TP-self-signed-XXXXXXXX certi

self-signed

 01
3082032E 31840216 A0030201 02024101 300D0609 2A864886 F70D0101 05050030   30312E30 2C060355 040313
2D536967 6E65642D 43657274   ...   ECA15D69 11970A66 252D34DC 760294A6 D1EA2329 F76EB905 6A5153C9
D19BFB22 9F89EE23 02D22D9D 2186B1A1 5AD4
```

If the trust-point enrollment is *not* **configured for** "selfsigned"; **the device is NOT impacted by this field notice**. No action is required.

If the trust-point enrollment *is* **configured for "selfsigned" and if the Self-Signed Certificate appears in the configuration**; the **device can be impacted** by this field notice. Continue to Step 4.

4. If you determined in Step 3 that the trust-point enrollment is configured for "selfsigned" and that the Self-Signed Certificate appears in the configuration, then check to see if the Self-Signed Certificate is applied to a feature on the device.

   Various features that can be tied to the SSC are shown in these sample configurations:

- For **HTTPS Server**, this text must be present:

```
ip http secure-server
```

Additionally, a trust-point can also be defined as shown in the next code example. If this command is not present, the default behavior is to use the Self-Signed Certificate.

```
ip http secure-trust-point TP-self-signed-XXXXXXXX
```

If a trust-point is defined and it points to a certificate other than the Self-Signed Certificate, you are not impacted.

For **HTTPS Server**, the impact of the expired certificate is minor because Self-Signed Certificates are already untrusted by web browsers and generate a warning even when they are not expired. The presence of an expired certificate can change the warning you receive in the browser.

- For **SIP over TLS**, this text is present in the configuration file:

```
voice service voip
 sip
  session transport tcp tls
!
sip-ua
crypto signaling default trust-point <self-signed-trust-point-name>
! or
crypto signaling remote-addr a.b.c.d /nn trust-point <self-signed-trust-point-name>
!
```

- For **Cisco Unified CME** with encrypted signaling enabled, this text is present in the configuration file:

```
telephony-service
 secure-signaling trust-point <self-signed-trust-point-name>
 tftp-server-credentials trust-point <self-signed-trust-point-name>
```

- For **Cisco Unified SRST** with encrypted signaling enabled, this text is present in the configuration file:

```
credentials
 trust-point <self-signed-trust-point-name>
```

- For **Cisco IOS** dspfarm **resource**s (Conference, Media Termination Point, or Transcoding) with encrypted signaling enabled, this text is present in the configuration file:

```
dspfarm profile 1 conference security
trust-point <self-signed-trust-point-name>
!
dspfarm profile 2 mtp security
trust-point <self-signed-trust-point-name>
!
dspfarm profile 3 transcode security
 trust-point <self-signed-trust-point-name>
!
sccp ccm 127.0.0.1 identifier 1 priority 1 version 7.0 trust-point <self-signed-trust-point-name>
!
```

- For **STCAPP ports** configured with encrypted signaling, this text is present in the configuration file:

```
stcapp security trust-point <self-signed-trust-point-name>
stcapp security mode encrypted
```

- For **Cisco Unified Communications Gateway Services API in Secure Mode**, this text is present in the configuration file:

```
uc secure-wsapi
ip http secure-server
ip http secure-trust-point TP-self-signed-XXXXXXXX
```

- For **SSLVPN**, this text is present in the configuration file:

```
webvpn gateway <gw name>
 ssl trust-point TP-self-signed-XXXXXXXX

OR

crypto ssl policy <policy-name>
  pki trust-point <trust-point-name> sign
```

- For **ISAKMP and IKEv2**, the Self-Signed Certificate can be used if any of the configurations is present (further analysis of the configuration is required in order to determine if the feature uses the Self-Signed Certificate versus a different certificate):

```
crypto isakmp policy <number>
 authentication pre-share | rsa-encr < NOT either of these
!
crypto ikev2 profile <prof name>
 authentication local rsa-sig
 pki trust-point TP-self-signed-xxxxxx
```

```
!
crypto isakmp profile <prof name>
 ca trust-point TP-self-signed-xxxxxx
```

- For **SSH Server**, It is extremely unlikely that you can leverage certificates to authenticate the SSH sessions. However, you can check your configuration to verify this. You must have all three lines shown in the next code example in order to be impacted.

> **Note**: If you leveraged username and password combination to SSH into your device then you are NOT impacted.

```
ip ssh server certificate profile
 ! Certificate used by server
 server
  trust-point sign TP-self-signed-xxxxxx
```

- For **RESTCONF**, this text is present in the configuration file:

```
restconf
! And one of the following
ip http secure-trust-point TP-self-signed-XXXXXXXXX
! OR
ip http client secure-trust-point TP-self-signed-XXXXXXXX
```

Solution(s)

The solution is to upgrade the Cisco IOS or Cisco IOS XE software to a release that includes the fix:

- Cisco IOS XE Software Release 16.9.1 and later
- Cisco IOS Software Release 15.6(3)M7 and later; 15.7(3)M5 and later; or 15.8(3)M3 and later

After you upgrade the software, you must regenerate the Self-Signed Certificate and export it to any devices that can require the certificate in their trust-store.

Three workarounds are available if an immediate software upgrade is not feasible:

1. Obtain a valid certificate from a 3rd part Certificate Authority (CA).
2. Use the Cisco IOS CA Server to generate a new certificate.
3. Use OpenSSL to generate a new Self-Signed Certificate.

1. Obtain a Valid Certificate from a 3rd Party Certificate Authority (CA)

Install a certificate from a certificate authority. Common CAs include: Comodo, Let's Encrypt, RapidSSL, Thawte, Sectigo, GeoTrust, Symantec, and so on. With this workaround, a certificate request is generated and displayed by Cisco IOS. The administrator then copies the request, submits it to a third-party CA, and retrieves the result.

Note: Use of a CA to sign certificates is considered to be a security best-practice. This procedure is provided as a workaround in this field notice; however, it is preferable to continue to use the third-party CA-signed certificate after you apply this workaround, rather than to use a Self-Signed Certificate.

To install a certificate from a third-party CA:

1. Create a Certificate Signing Request (CSR):

```
<#root>

Router#

configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#

crypto pki trustpoint TEST


Router(ca-trustpoint)#

enrollment term pem


Router(ca-trustpoint)#

subject-name CN=TEST


Router(ca-trustpoint)#

revocation-check none


Router(ca-trustpoint)#

rsakeypair TEST


Router(ca-trustpoint)#

exit


Router(config)#

crypto pki enroll TEST


   % Start certificate enrollment ..
   % The subject name in the certificate will include: CN=TEST
   % The subject name in the certificate will include: Router.cisco.com
   % The serial number in the certificate will be: FTX1234ABCD
   % Include an IP address in the subject name? [no]: n
   Display Certificate Request to terminal? [yes/no]:

yes


   Certificate Request follows:
```

```
   -----BEGIN CERTIFICATE REQUEST-----
A Base64 Certificate is displayed here. Copy it, along with the ---BEGIN and ---END lines.
   -----END CERTIFICATE REQUEST-----

   ---End - This line not part of the certificate request---
```

2. Submit the CSR to the third-party CA.

Note: The procedure to submit the CSR to a third-party CA and retrieve the certificate that results varies based on the CA that is used. Consult the documentation for your CA for instructions on how to perform this step.

---

3. Download the new identity certificate for the router along with the CA certificate.

4. Install the CA certificate on the device:

<#root>

```
Router#
```

configure terminal

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

crypto pki auth TEST

```
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
REMOVED
-----END CERTIFICATE-----

Certificate has the following attributes:
   Fingerprint MD5: 79D15A9F C7EB4882 83AC50AC 7B0FC625
   Fingerprint SHA1: 0A80CC2C 9C779D20 9071E790 B82421DE B47E9006

% Do you accept this certificate? [yes/no]:
```

yes

```
trust-point CA certificate accepted.
% Certificate successfully imported
```

5. Install the identity certificate on the device:

<#root>

```
Router(config)#
```

crypto pki import TEST certificate

```
Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
REMOVED
-----END CERTIFICATE-----

% Router Certificate successfully imported
```

2. Use the Cisco IOS CA Server to Generate a New Certificate

Use the local Cisco IOS Certificate Authority server to generate and sign a new certificate.

Note:The local CA server feature is not available on all products.

<#root>

Router#

configure terminal

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

ip http server

```
Router(config)#
```

crypto pki server IOS-CA

```
Router(cs-server)#
```

grant auto

```
Router(cs-server)#
```

database level complete

```
Router(cs-server)#
```

no shut

```
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key
% or type Return to exit
Password: <password>

Re-enter password: <password>
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

% Certificate Server enabled.
```

<#root>

Router#

show crypto pki server IOS-CA Certificates

Serial Issued date Expire date Subject Name
1 21:31:40 EST Jan 1 2020 21:31:40 EST Dec 31 2022 cn=IOS-CA

<#root>

Router#

configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#

crypto pki trustpoint TEST

Router(ca-trustpoint)#

enrollment url http://<local interface ip>:80

     <<<< Replace <local interface ip> with the IP address of an interface on the router
Router(ca-trustpoint)#

subject-name CN=TEST

Router(ca-trustpoint)#

revocation-check none

Router(ca-trustpoint)#

rsakeypair TEST

Router(ca-trustpoint)#

exit

Router#

configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#

crypto pki auth TEST

Certificate has the following attributes:
Fingerprint MD5: C281D9A0 337659CB D1B03AA6 11BD6E40
Fingerprint SHA1: 1779C425 3DCEE86D 2B11C880 D92361D6 8E2B71FF

```
% Do you accept this certificate? [yes/no]:
```

yes

```
Trustpoint CA certificate accepted.

Router(config)#
```

crypto pki enroll TEST

```
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
password to the CA Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please take note of it.

Password: <passsword>
Re-enter password: <password>

% The subject name in the certificate will include: CN=TEST
% The subject name in the certificate will include: Router.cisco.com
% Include the router serial number in the subject name? [yes/no]: yes
% The serial number in the certificate will be: FTX1234ABCD
% Include an IP address in the subject name? [no]: no

Request certificate from CA? [yes/no]:
```

yes

```
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate verbose TEST' command will show the fingerprint
```

3. Use OpenSSL to Generate a New Self-signed Certificate

Use OpenSSL to generate a PKCS12 certificate bundle and import the bundle to Cisco IOS.

LINUX, UNIX or MAC (OSX) Example

<#root>

User@linux-box$ openssl req -newkey rsa:2048 -nodes -keyout tmp.key -x509 -days 4000 -out tmp.cer -subj "/CN=SelfSignedCert" &> /dev/null && openssl pkcs12 -export -in tmp.cer -inkey tmp.key -out tmp.bin -passout pass:

Cisco123

```
 && openssl pkcs12 -export -out certificate.pfx -password pass:
```

Cisco123

```
 -inkey
tmp.key -in tmp.cer && rm tmp.bin tmp.key tmp.cer && openssl base64 -in certificate.pfx

MIII8QIBAzCCCLcGCSqGSIb3DQEHAaCCCKgEggikMIIIoDCCA1cGCSqGSIb3DQEH
BqCCA0gwggNEAgEAMIIDPQYJKoZIhvcNAQcBMBwGCiqGSIb3DQEMAQYwDgQIGnxm
t5r28FECAggAgIIDEKyw1OsmucdQGt1c0DdfYXwUo8BwaBnzQvN0ClawXNQln2bT
```

```
vrhus6LfRvVxBNPeQz2ADgLikGxatwV5EDgooM+IEucKDURGLEotaRrVU5Wk3EGM
mjC6Ko9OaM3OvhAGEEXrk26cq+OWsEuF3qudggRYv2gIBcrJ2iUQNFsBIrvlGHRo
FphOTqhVaAPxZS7hOB3OcK1tMKHOIa8EwygyBvQPfjjBT79QFgeexIJFmUtqYX/P
<OUTPUT OMITTED FOR BREVITY>
tT6r4SuibYKu6HV45ffjSzOimcJI+D9LKhLWR6pK/k5ge8v7aK9/rsVbjavbdy7b
CSqGSIb3DQEJFTEWBBS96DY/gRfN1dSx46P1EqjPvSYiETAxMCEwCQYFKw4DAhoF
AAQU+EX0kNvuNz6XmFxXER8wlqKTGvgECA+D+Z81uwafAgIIAA==
```

Cisco IOS or Cisco IOS XE Router Example

<#root>

Router#

configure terminal

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

crypto pki trustpoint TEST

```
Router(ca-trustpoint)#
```

enrollment terminal

```
Router(ca-trustpoint)#
```

revocation-check none

```
Router(ca-trustpoint)#
```

exit

```
R1(config)#
```

crypto pki import TEST pkcs12 terminal password Cisco123

```
Enter the base 64 encoded pkcs12.
End with a blank line or the word "quit" on a line by itself:
MIII8QIBAzCCCLcGCSqGSIb3DQEHAaCCCKgEggikMIIIoDCCA1cGCSqGSIb3DQEH
BqCCA0gwggNEAgEAMIIDPQYJKoZIhvcNAQcBMBwGCiqGSIb3DQEMAQYwDgQItyCo
Vh05+0QCAggAgIIDENUWY+UeuY5sIRZuoBi2nEhdIPd1th/auBYtX79aXGiz/iEW
<OUTPUT OMITTED FOR BREVITY>
IY1l273y9bC3qPVJOUGoQW8SGfarqEjaqxdAet66E5V6u9Yvd4oMsIYGsa7Om+FN
CsUVj+ll5hzGjK78LOycXWpH4gDOGYBVf+D7mgWqaqZvxYUoEkOrTMmW5zElMCMG
CSqGSIb3DQEJFTEWBBSgiBJIYpJLzo/GYNOsesZh3wGmPTAxMCEwCQYFKw4DAhoF
AAQUdeUrLIC2uo/mbyE86he5+qEjmPYECKu76GWaeKb7AgIIAA==
quit

CRYPTO_PKI: Imported PKCS12 file successfully.

R1(config)#
```

4. Verify that the New Certificate is Installed

<#root>

```
R1#
```

show crypto pki certificates TEST

```
Load for five secs: 5%/1%; one minute: 2%; five minutes: 3%
Time source is SNTP, 15:04:37.593 UTC Mon Dec 16 2019
CA Certificate
  Status: Available
  Certificate Serial Number (hex): 00A16966E46A435A99
  Certificate Usage: General Purpose
  Issuer:
    cn=SelfSignedCert
  Subject:
    cn=SelfSignedCert
  Validity Date:
    start date: 14:54:46 UTC Dec 16 2019
    end   date: 14:54:46 UTC Nov 28 2030
```

Note: Self-signed certificates expire on 00:00 1 Jan 2020 UTC and you cannot create them after that time.

Q&A

Q: What is the issue?

Self-signed X.509 PKI certificates generated on products that run affected Cisco IOS or Cisco IOS XE versions expire on 01/01/2020 00:00:00 UTC. New Self-Signed Certificates cannot be created on affected devices after 01/01/2020 00:00:00 UTC. Any service that relies on these Self-Signed Certificates can no longer work after the certificate expires.

Q: What is the impact to a client network if a Self-Signed Certificate expires for their product?

Any affected product's functionality that relies on the Self-Signed Certificates can no longer work after the certificate expires. Please see the Field Notice for additional detail.

Q: How do I know if I am affected by this issue?

The Field Notice provides instructions to determine if you use a Self-Signed Certificate and whether your configuration is affected by this issue. Please see the "How To Identify Affected Products" section in the Field Notice.

Q: Is there a script that I can run to see if I am affected?

Yes. Use Cisco CLI Analyzer, run an System Diagnostic run. If the certificate is present and that is used an alert can be shown. https://cway.cisco.com/cli/

Q. Has Cisco provided software fixes for this issue?

Yes. Cisco has released software fixes for this issue as well as workarounds in the event a software upgrade is not immediately feasible. Please see the Field Notice for complete detail.

Q: Does this issue affect any Cisco product that use a certificate?

No. This issue affects **only products that use Self-Signed Certificates generated by specific versions of Cisco IOS or Cisco IOS XE** with the certificate applied to a service on the product. Products that use Certificates generated by a Certificate Authority (CA) are not impacted by this issue.

Q: Do Cisco products use only Self-Signed Certificates?

No. Certificates can be generated by either an external 3rd-party Certificate Authority or on the Cisco IOS or Cisco IOS XE device itself as a Self-Signed Certificate. Specific user requirements can require the use of Self-Signed Certificates. Certificates generated by a Certificate Authority (CA) are not impacted by this issue.

Q. Why did this issue occur?

Unfortunately, despite the best efforts of technology vendors, software defects do still occur. When a bug is discovered in any Cisco technology, we are committed to transparency and to provide our users the information they need to protect their network.

In this case, the issue is caused by a known software bug in which affected versions of Cisco IOS and Cisco IOS XE can always set the Self-Signed Certificate's expiration date to 01/01/2020 00:00:00 UTC. After this date, the certificate expires and is invalid, which could impact product functionality.

Q: Why was an expiration date of January 1, 2020 00:00:00 UTC chosen?

Certificates commonly have an expiration date. In the case of this software bug, the January 1, 2020 date was used during Cisco IOS and Cisco IOS XE software development over 10 years ago and is a human error.

Q: What products are affected by this issue?

Any Cisco product that run Cisco IOS releases prior to 15.6(03)M07, 15.7(03)M05, 15.8(03)M03, and 15.9(03)M and any Cisco product that run Cisco IOS XE releases prior to 16.9.1

Q: What do users need to do?

You need to review the field notice to assess whether you are impacted by this issue and, if so, to use the Workaround/Solution instructions to mitigate this issue.

Q: Is this issue a security vulnerability?

No. This is not a security vulnerability, and there is no risk to the integrity of the product.

Q: Is SSH affected?

No. SSH does use RSA keypairs but does not utilize certificates except in a rare configuration. For Cisco IOS to utilize certificates the next configuration must be present.

```
ip ssh server certificate profile
   server
      trust-point sign TP-self-signed-xxxxxx
```

Q: What fixed versions are available for the Classic Catalyst 2K, 3K, 4K, 6K platforms?

For Polaris based platforms(3650/3850/Catalyst 9K series), fix is available 16.9.1 onwards
For CDB platform, fix is available 15.2(7)E1a onwards

For the other Classic Switching Platforms:

Commits are in progress but we do not have posted CCO Release. Next CCO Release can have the fix.

In the interm please utilize one of the other available workarounds.

Q: Is WAAS affected?

WAAS continues to operate properly and optimize traffic, however, AppNav-XE & the Central Manager went offline to the device that has an expired Self-Signed Certificate. This means you cannot monitor AppNav-Cluster or change any policies for WAAS. In summary, WAAS continues to work properly, but management and monitoring is suspended until the certificate issue is resolved. To resolve the issue, a new certificate can need to be generated on Cisco IOS and then imported into the Central Manager.

Related Information

- See FN70489 Field Notice: FN - 70489 - PKI Self-Signed Certificate Expiration in Cisco IOS and Cisco IOS XE Software
- See Cisco bug ID CSCvi48253