Renewal of FMC Sftunnel CA Certificate for FTD Connectivity

Contents

Introduction

This document describes the renewal of Firepower Management Center (FMC) sftunnel Certificate Authority (CA) certificate in relationship with the Firepower Threat Defense (FTD) connectivity.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Firepower Threat Defense
- Firepower Management Center
- Public Key Infrastructure (PKI)

Components used

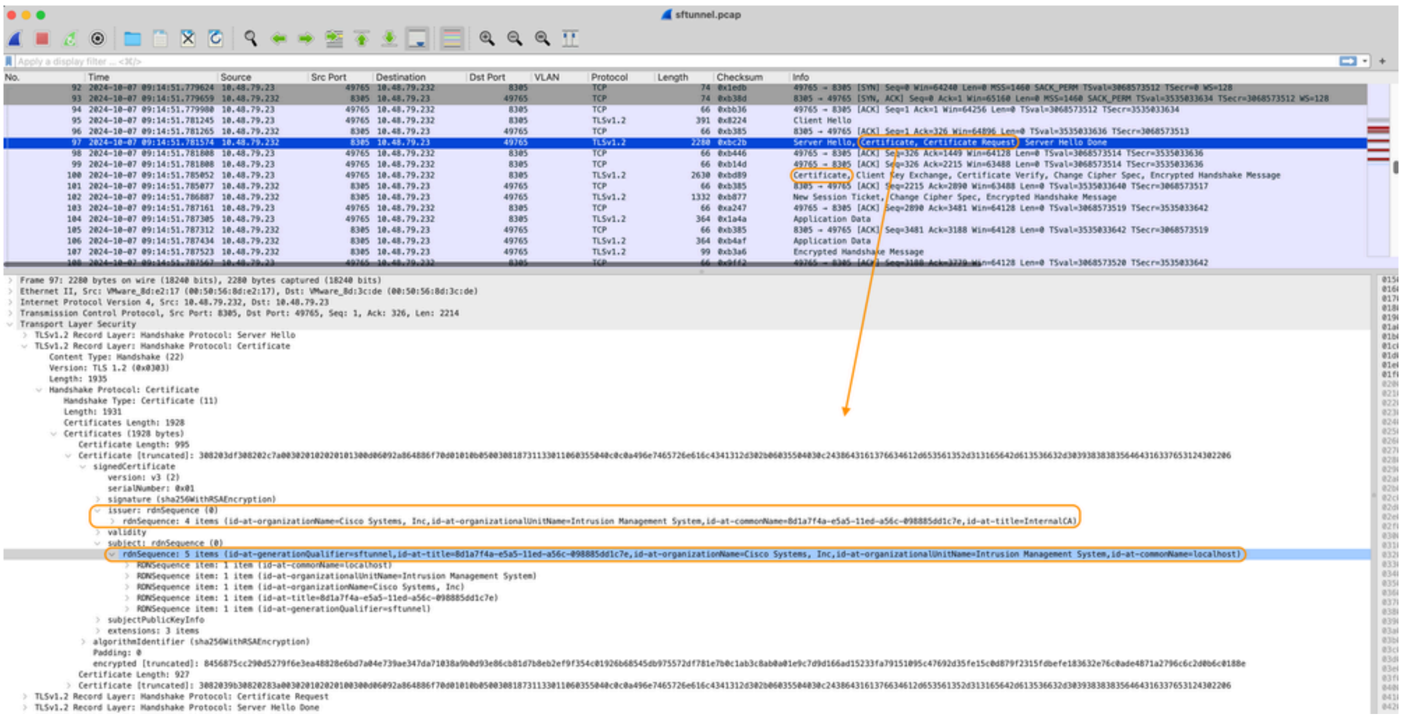
This document is not restricted to specific software and hardware versions.

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 Information

FMC and FTD communicate with eachother over sftunnel (Sourcefire tunnel). This communication uses certificates to make the conversation secure over a TLS session. More information on the sftunnel and how it does get established can be found on this link.

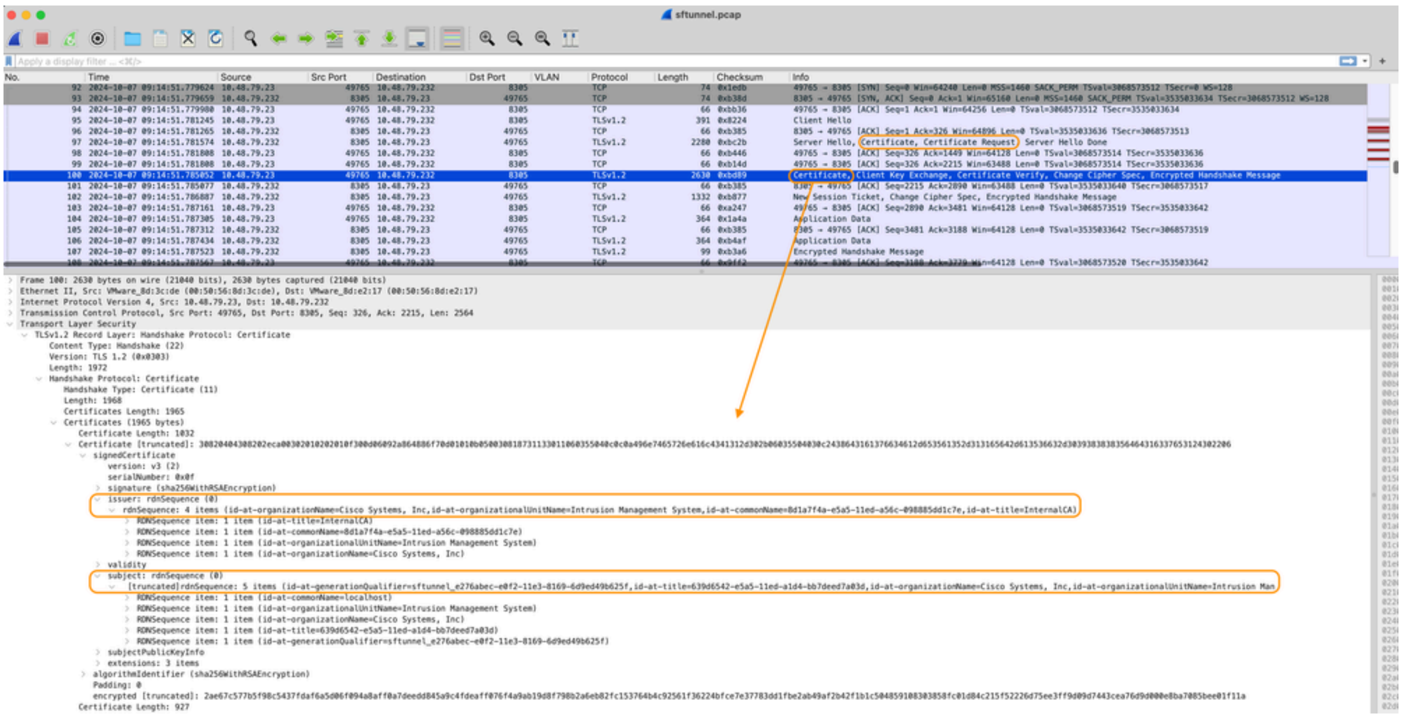
From the packet capture, you can see that the FMC (10.48.79.232 in this example) and FTD (10.48.79.23) are exchanging certificates with eachother. They do this in order to validate that they talk with the correct device and there is no eavesdropping or Man-In-The-Middle (MITM) attack. The communication is encrypted using those certificates and only the party that has the associated private key for that certificate is able to decrypt it again.



```
No.  Time                        Source        Src Port Destination   Dst Port VLAN Protocol Length Checksum Info
 92  2024-10-07 09:14:51.779624  10.48.79.23   49765    10.48.79.232  8305          TCP       74    0x1edb   49765 → 8305 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3068573512 TSecr=0 WS=128
 93  2024-10-07 09:14:51.779659  10.48.79.232  8305     10.48.79.23   49765         TCP       74    0xb38d   8305 → 49765 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3535033634 TSecr=3068573512 WS=128
 94  2024-10-07 09:14:51.779980  10.48.79.23   49765    10.48.79.232  8305          TCP       66    0xbb36   49765 → 8305 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3068573512 TSecr=3535033634
 95  2024-10-07 09:14:51.781245  10.48.79.23   49765    10.48.79.232  8305          TLSv1.2  391    0x8224   Client Hello
 96  2024-10-07 09:14:51.781265  10.48.79.232  8305     10.48.79.23   49765         TCP       66    0xb385   8305 → 49765 [ACK] Seq=1 Ack=326 Win=64896 Len=0 TSval=3535033636 TSecr=3068573513
 97  2024-10-07 09:14:51.781574  10.48.79.232  8305     10.48.79.23   49765         TLSv1.2 2280    0xbc2b   Server Hello, Certificate, Certificate Request, Server Hello Done
 98  2024-10-07 09:14:51.781808  10.48.79.23   49765    10.48.79.232  8305          TCP       66    0xb446   49765 → 8305 [ACK] Seq=326 Ack=1449 Win=64128 Len=0 TSval=3068573514 TSecr=3535033636
 99  2024-10-07 09:14:51.781808  10.48.79.23   49765    10.48.79.232  8305          TCP       66    0xb14d   49765 → 8305 [ACK] Seq=326 Ack=2215 Win=63488 Len=0 TSval=3068573514 TSecr=3535033636
100  2024-10-07 09:14:51.785052  10.48.79.23   49765    10.48.79.232  8305          TLSv1.2 2630    0xbd89   Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
101  2024-10-07 09:14:51.785077  10.48.79.232  8305     10.48.79.23   49765         TCP       66    0xb385   8305 → 49765 [ACK] Seq=2215 Ack=2890 Win=63488 Len=0 TSval=3535033640 TSecr=3068573517
102  2024-10-07 09:14:51.786887  10.48.79.232  8305     10.48.79.23   49765         TLSv1.2 1332    0xb877   New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
103  2024-10-07 09:14:51.787161  10.48.79.23   49765    10.48.79.232  8305          TCP       66    0xa247   49765 → 8305 [ACK] Seq=2890 Ack=3481 Win=64128 Len=0 TSval=3068573519 TSecr=3535033642
104  2024-10-07 09:14:51.787305  10.48.79.23   49765    10.48.79.232  8305          TLSv1.2  364    0x1a4a   Application Data
105  2024-10-07 09:14:51.787312  10.48.79.232  8305     10.48.79.23   49765         TCP       66    0xb385   8305 → 49765 [ACK] Seq=3481 Ack=3188 Win=64128 Len=0 TSval=3535033642 TSecr=3068573519
106  2024-10-07 09:14:51.787434  10.48.79.232  8305     10.48.79.23   49765         TLSv1.2  364    0xb4af   Application Data
107  2024-10-07 09:14:51.787523  10.48.79.232  8305     10.48.79.23   49765         TLSv1.2   99    0xb3a6   Encrypted Handshake Message
108  2024-10-07 09:14:51.787567  10.48.79.23   49765    10.48.79.232  8305          TCP       66    0x9ff2   49765 → 8305 [ACK] Seq=3188 Ack=3779 Win=64128 Len=0 TSval=3068573520 TSecr=3535033642
```

```
> Frame 97: 2280 bytes on wire (18240 bits), 2280 bytes captured (18240 bits)
> Ethernet II, Src: VMware_8d:e2:17 (00:50:56:8d:e2:17), Dst: VMware_8d:3c:de (00:50:56:8d:3c:de)
> Internet Protocol Version 4, Src: 10.48.79.232, Dst: 10.48.79.23
> Transmission Control Protocol, Src Port: 8305, Dst Port: 49765, Seq: 1, Ack: 326, Len: 2214
v Transport Layer Security
  > TLSv1.2 Record Layer: Handshake Protocol: Server Hello
  v TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 1935
    v Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 1931
        Certificates Length: 1928
      v Certificates (1928 bytes)
          Certificate Length: 995
        v Certificate [truncated]: 308203df308202c7a003020102020101300d06092a864886f70d01010b0500308187311330110603550 40c0c0a496e7465726e616c4341312d302b06035504030c2438643161376634612d653561352d313165642d613536632d30393838383564643163376531243022 06
          v signedCertificate
              version: v3 (2)
              serialNumber: 0x01
            > signature (sha256WithRSAEncryption)
            v issuer: rdnSequence (0)
              > rdnSequence: 4 items (id-at-organizationName=Cisco Systems, Inc,id-at-organizationalUnitName=Intrusion Management System,id-at-commonName=8d1a7f4a-e5a5-11ed-a56c-098885dd1c7e,id-at-title=InternalCA)
            > validity
            v subject: rdnSequence (0)
              v rdnSequence: 5 items (id-at-generationQualifier=sftunnel,id-at-title=8d1a7f4a-e5a5-11ed-a56c-098885dd1c7e,id-at-organizationName=Cisco Systems, Inc,id-at-organizationalUnitName=Intrusion Management System,id-at-commonName=localhost)
                > RDNSequence item: 1 item (id-at-commonName=localhost)
                > RDNSequence item: 1 item (id-at-organizationalUnitName=Intrusion Management System)
                > RDNSequence item: 1 item (id-at-organizationName=Cisco Systems, Inc)
                > RDNSequence item: 1 item (id-at-title=8d1a7f4a-e5a5-11ed-a56c-098885dd1c7e)
                > RDNSequence item: 1 item (id-at-generationQualifier=sftunnel)
            > subjectPublicKeyInfo
            > extensions: 3 items
          > algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
            encrypted [truncated]: 8456875cc290d5279f6e3ea48828e6bd7a04e739ae347da71038a9b0d93e86cb81d7b8eb2ef9f354c01926b68545db975572df781e7b0c1ab3c8ab0a01e9c7d9d166ad15233fa79151095c47692d35fe15c0d879f2315fdbefe183632e76c0ade4871a2796c6c2d0b6c0188e
          Certificate Length: 927
        > Certificate [truncated]: 3082039b30820283a003020102020100300d06092a864886f70d01010b0500308187311330110603550 40c0c0a496e7465726e616c4341312d302b06035504030c2438643161376634612d653561352d313165642d613536632d30393838383564643163376531243022 06
  > TLSv1.2 Record Layer: Handshake Protocol: Certificate Request
  > TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
```

Certificate_exchange_server_cert



```
sftunnel.pcap
No.  Time                        Source        Src Port  Destination   Dst Port  VLAN  Protocol  Length  Checksum  Info
 92  2024-10-07 09:14:51.779624  10.48.79.23      49765  10.48.79.232      8305        TCP           74  0x1edb    49765 → 8305 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3068573512 TSecr=0 WS=128
 93  2024-10-07 09:14:51.779659  10.48.79.232      8305  10.48.79.23      49765        TCP           74  0xb38d    8305 → 49765 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3535033634 TSecr=3068573512 WS=128
 94  2024-10-07 09:14:51.779980  10.48.79.23      49765  10.48.79.232      8305        TCP           66  0xbb36    49765 → 8305 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3068573512 TSecr=3535033634
 95  2024-10-07 09:14:51.781245  10.48.79.23      49765  10.48.79.232      8305        TLSv1.2      391  0x8224    Client Hello
 96  2024-10-07 09:14:51.781265  10.48.79.232      8305  10.48.79.23      49765        TCP           66  0xb385    8305 → 49765 [ACK] Seq=1 Ack=326 Win=64896 Len=0 TSval=3535033636 TSecr=3068573513
 97  2024-10-07 09:14:51.781574  10.48.79.232      8305  10.48.79.23      49765        TLSv1.2     2280  0xbc2b    Server Hello, Certificate, Certificate Request, Server Hello Done
 98  2024-10-07 09:14:51.781808  10.48.79.23      49765  10.48.79.232      8305        TCP           66  0xb446    49765 → 8305 [ACK] Seq=326 Ack=1449 Win=64128 Len=0 TSval=3068573514 TSecr=3535033636
 99  2024-10-07 09:14:51.781808  10.48.79.23      49765  10.48.79.232      8305        TCP           66  0xb14d    49765 → 8305 [ACK] Seq=326 Ack=2215 Win=63488 Len=0 TSval=3068573514 TSecr=3535033636
100  2024-10-07 09:14:51.785052  10.48.79.23      49765  10.48.79.232      8305        TLSv1.2     2630  0xbd89    Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
101  2024-10-07 09:14:51.785077  10.48.79.232      8305  10.48.79.23      49765        TCP           66  0xb385    8305 → 49765 [ACK] Seq=2215 Ack=2890 Win=63488 Len=0 TSval=3535033640 TSecr=3068573517
102  2024-10-07 09:14:51.786887  10.48.79.232      8305  10.48.79.23      49765        TLSv1.2     1332  0xb877    New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
103  2024-10-07 09:14:51.787161  10.48.79.23      49765  10.48.79.232      8305        TCP           66  0xa247    49765 → 8305 [ACK] Seq=2890 Ack=3481 Win=64128 Len=0 TSval=3068573519 TSecr=3535033642
104  2024-10-07 09:14:51.787305  10.48.79.23      49765  10.48.79.232      8305        TLSv1.2      364  0x1a4a    Application Data
105  2024-10-07 09:14:51.787312  10.48.79.232      8305  10.48.79.23      49765        TCP           66  0xb385    8305 → 49765 [ACK] Seq=3481 Ack=3188 Win=64128 Len=0 TSval=3535033642 TSecr=3068573519
106  2024-10-07 09:14:51.787434  10.48.79.232      8305  10.48.79.23      49765        TLSv1.2      364  0xb4af    Application Data
107  2024-10-07 09:14:51.787523  10.48.79.232      8305  10.48.79.23      49765        TLSv1.2       99  0xb3a6    Encrypted Handshake Message
108  2024-10-07 09:14:51.787567  10.48.79.23      49765  10.48.79.232      8305        TCP           66  0x9ff2    49765 → 8305 [ACK] Seq=3188 Ack=3729 Win=64128 Len=0 TSval=3068573520 TSecr=3535033642
```

```
> Frame 100: 2630 bytes on wire (21040 bits), 2630 bytes captured (21040 bits)
> Ethernet II, Src: VMware_8d:3c:de (00:50:56:8d:3c:de), Dst: VMware_8d:e2:17 (00:50:56:8d:e2:17)
> Internet Protocol Version 4, Src: 10.48.79.23, Dst: 10.48.79.232
> Transmission Control Protocol, Src Port: 49765, Dst Port: 8305, Seq: 326, Ack: 2215, Len: 2564
v Transport Layer Security
  v TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 1972
    v Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 1968
        Certificates Length: 1965
      v Certificates (1965 bytes)
          Certificate Length: 1032
        v Certificate [truncated]: 30820404308202eca003020102020f300d06092a864886f70d01010b05003081873113301106035504 0c0c0a496e7465726e616c4341312d302b06035504030c2438643161376634612d653561352d313165642d613536632d30393838383564643163376531243022 06
          v signedCertificate
              version: v3 (2)
              serialNumber: 0x0f
            > signature (sha256WithRSAEncryption)
            v issuer: rdnSequence (0)
              v rdnSequence: 4 items (id-at-organizationName=Cisco Systems, Inc,id-at-organizationalUnitName=Intrusion Management System,id-at-commonName=8d1a7f4a-e5a5-11ed-a56c-098885dd1c7e,id-at-title=InternalCA)
                > RDNSequence item: 1 item (id-at-title=InternalCA)
                > RDNSequence item: 1 item (id-at-commonName=8d1a7f4a-e5a5-11ed-a56c-098885dd1c7e)
                > RDNSequence item: 1 item (id-at-organizationalUnitName=Intrusion Management System)
                > RDNSequence item: 1 item (id-at-organizationName=Cisco Systems, Inc)
            > validity
            v subject: rdnSequence (0)
              v  [truncated]rdnSequence: 5 items (id-at-generationQualifier=sftunnel_e276abec-e0f2-11e3-8169-6d9ed49b625f,id-at-title=639d6542-e5a5-11ed-a1d4-bb7deed7a03d,id-at-organizationName=Cisco Systems, Inc,id-at-organizationalUnitName=Intrusion Man
                > RDNSequence item: 1 item (id-at-commonName=localhost)
                > RDNSequence item: 1 item (id-at-organizationalUnitName=Intrusion Management System)
                > RDNSequence item: 1 item (id-at-organizationName=Cisco Systems, Inc)
                > RDNSequence item: 1 item (id-at-title=639d6542-e5a5-11ed-a1d4-bb7deed7a03d)
                > RDNSequence item: 1 item (id-at-generationQualifier=sftunnel_e276abec-e0f2-11e3-8169-6d9ed49b625f)
            > subjectPublicKeyInfo
            > extensions: 3 items
          > algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
            encrypted [truncated]: 2ae67c577b5f98c5437fdaf6a5d06f094a8aff0a7deedd845a9c4fdeaff076f4a9ab19d8f798b2a6eb82fc153764b4c92561f36224bfce7e37783dd1fbe2ab49af2b42f1b1c504859108303858fc01d84c215f52226d75ee3ff9d09d7443cea76d9d000e8ba7085bee01f11a
          Certificate Length: 927
```

Certificate_exchange_client_cert

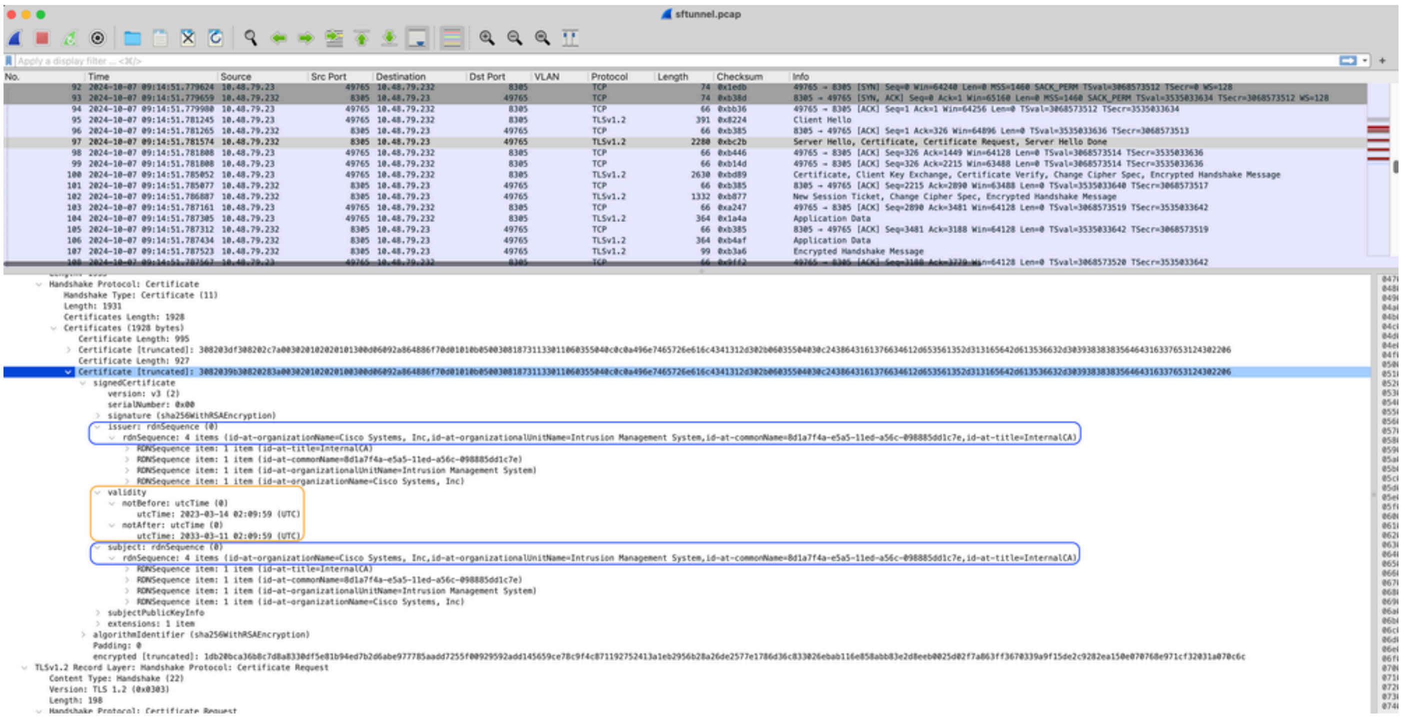
You can see the certificates are signed by the same InternalCA (Issuer) Certificate Authority (CA) which is set up on the FMC system. The configuration is defined on the FMC on **/etc/sf/sftunnel.conf** file which contains something like:

```
proxyssl {
  proxy_cert   /etc/sf/keys/sftunnel-cert.pem;            ---> Certificate provided by FMC to FTD f
  proxy_key    /etc/sf/keys/sftunnel-key.pem;
  proxy_cacert /etc/sf/ca_root/cacert.pem;                ---> CA certificate (InternalCA)
  proxy_crl    /etc/sf/ca_root/crl.pem;
  proxy_cipher 1;
  proxy_tls_version TLSv1.2;
};
```

This indicates the CA that is used to sign all certificates for sftunnel (both the FTD and FMC one) and the certificate used by the FMC to send to all of the FTDs. This certificate is signed by the InternalCA.

When FTD registers to the FMC, the FMC also creates a certificate to push to the FTD device that is used for the further communication on the sftunnel. This certificate is also signed by the same Internal CA certificate. On FMC, you can find that certificate (and private key) under **/var/sf/peers/<UUID-FTD-device>** and potentially under **certs_pushed** folder and is called **sftunnel-cert.pem** (**sftunnel-key.pem** for private key). On FTD, you can find those under **/var/sf/peers/<UUID-FMC-device>** with same naming convention.

However each certificate also has a validity period for security purposes. When inspecting the InternalCA certificate, we can see as well the validity period which is 10 years for the FMC InternalCA as shown from the packet capture.



```
sftunnel.pcap

No.  Time                        Source        Src Port  Destination   Dst Port  VLAN  Protocol  Length  Checksum  Info
 92  2024-10-07 09:14:51.779624  10.48.79.23      49765  10.48.79.232      8305        TCP           74  0x1edb    49765 → 8305 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3068573512 TSecr=0 WS=128
 93  2024-10-07 09:14:51.779659  10.48.79.232      8305  10.48.79.23      49765        TCP           74  0xb38d    8305 → 49765 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3535033634 TSecr=3068573512 WS=128
 94  2024-10-07 09:14:51.779980  10.48.79.23      49765  10.48.79.232      8305        TCP           66  0xbb36    49765 → 8305 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3068573512 TSecr=3535033634
 95  2024-10-07 09:14:51.781245  10.48.79.23      49765  10.48.79.232      8305        TLSv1.2      391  0x8224    Client Hello
 96  2024-10-07 09:14:51.781265  10.48.79.232      8305  10.48.79.23      49765        TCP           66  0xb385    8305 → 49765 [ACK] Seq=1 Ack=326 Win=64896 Len=0 TSval=3535033636 TSecr=3068573513
 97  2024-10-07 09:14:51.781574  10.48.79.232      8305  10.48.79.23      49765        TLSv1.2     2280  0xbc2b    Server Hello, Certificate, Certificate Request, Server Hello Done
 98  2024-10-07 09:14:51.781808  10.48.79.23      49765  10.48.79.232      8305        TCP           66  0xb446    49765 → 8305 [ACK] Seq=326 Ack=1449 Win=64128 Len=0 TSval=3068573514 TSecr=3535033636
 99  2024-10-07 09:14:51.781808  10.48.79.23      49765  10.48.79.232      8305        TCP           66  0xb14d    49765 → 8305 [ACK] Seq=326 Ack=2215 Win=63488 Len=0 TSval=3068573514 TSecr=3535033636
100  2024-10-07 09:14:51.785052  10.48.79.23      49765  10.48.79.232      8305        TLSv1.2     2630  0xbd89    Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
101  2024-10-07 09:14:51.785077  10.48.79.232      8305  10.48.79.23      49765        TCP           66  0xb385    8305 → 49765 [ACK] Seq=2215 Ack=2890 Win=63488 Len=0 TSval=3535033640 TSecr=3068573517
102  2024-10-07 09:14:51.786887  10.48.79.232      8305  10.48.79.23      49765        TLSv1.2     1332  0xb877    New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
103  2024-10-07 09:14:51.787161  10.48.79.23      49765  10.48.79.232      8305        TCP           66  0xa247    49765 → 8305 [ACK] Seq=2890 Ack=3481 Win=64128 Len=0 TSval=3068573519 TSecr=3535033642
104  2024-10-07 09:14:51.787305  10.48.79.23      49765  10.48.79.232      8305        TLSv1.2      364  0x1a4a    Application Data
105  2024-10-07 09:14:51.787312  10.48.79.232      8305  10.48.79.23      49765        TCP           66  0xb385    8305 → 49765 [ACK] Seq=3481 Ack=3188 Win=64128 Len=0 TSval=3535033642 TSecr=3068573519
106  2024-10-07 09:14:51.787434  10.48.79.232      8305  10.48.79.23      49765        TLSv1.2      364  0xb4af    Application Data
107  2024-10-07 09:14:51.787523  10.48.79.232      8305  10.48.79.23      49765        TLSv1.2       99  0xb3a6    Encrypted Handshake Message
108  2024-10-07 09:14:51.787567  10.48.79.23      49765  10.48.79.232      8305        TCP           66  0x9ff2    49765 → 8305 [ACK] Seq=3188 Ack=3779 Win=64128 Len=0 TSval=3068573520 TSecr=3535033642
```

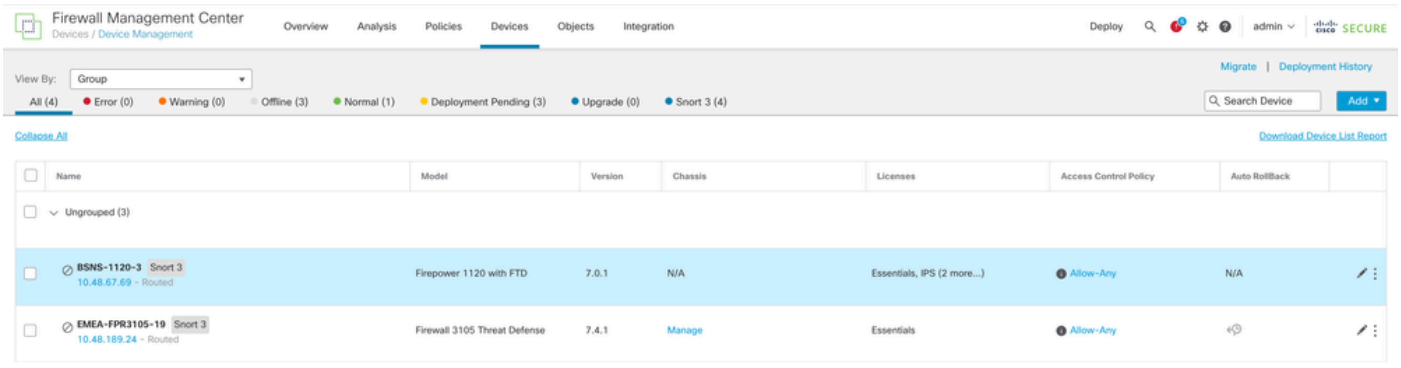
```
  ∨ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 1931
      Certificates Length: 1928
    ∨ Certificates (1928 bytes)
        Certificate Length: 995
      > Certificate [truncated]: 308203df308202c7a003020102020101300d06092a864886f70d01010b05003081873113301106035504 0c0c0a496e7465726e616c4341312d302b06035504030c2438643161376634612d653561352d313165642d613536632d3039383838356464316337653124302206
        Certificate Length: 927
      ∨ Certificate [truncated]: 3082039b30820283a003020102020100300d06092a864886f70d01010b05003081873113301106035504 0c0c0a496e7465726e616c4341312d302b06035504030c2438643161376634612d653561352d313165642d613536632d3039383838356464316337653124302206
        ∨ signedCertificate
            version: v3 (2)
            serialNumber: 0x00
          > signature (sha256WithRSAEncryption)
          ∨ issuer: rdnSequence (0)
            ∨ rdnSequence: 4 items (id-at-organizationName=Cisco Systems, Inc,id-at-organizationalUnitName=Intrusion Management System,id-at-commonName=8d1a7f4a-e5a5-11ed-a56c-098885dd1c7e,id-at-title=InternalCA)
              > RDNSequence item: 1 item (id-at-title=InternalCA)
              > RDNSequence item: 1 item (id-at-commonName=8d1a7f4a-e5a5-11ed-a56c-098885dd1c7e)
              > RDNSequence item: 1 item (id-at-organizationalUnitName=Intrusion Management System)
              > RDNSequence item: 1 item (id-at-organizationName=Cisco Systems, Inc)
          ∨ validity
            ∨ notBefore: utcTime (0)
                utcTime: 2023-03-14 02:09:59 (UTC)
            ∨ notAfter: utcTime (0)
                utcTime: 2033-03-11 02:09:59 (UTC)
          ∨ subject: rdnSequence (0)
            ∨ rdnSequence: 4 items (id-at-organizationName=Cisco Systems, Inc,id-at-organizationalUnitName=Intrusion Management System,id-at-commonName=8d1a7f4a-e5a5-11ed-a56c-098885dd1c7e,id-at-title=InternalCA)
              > RDNSequence item: 1 item (id-at-title=InternalCA)
              > RDNSequence item: 1 item (id-at-commonName=8d1a7f4a-e5a5-11ed-a56c-098885dd1c7e)
              > RDNSequence item: 1 item (id-at-organizationalUnitName=Intrusion Management System)
              > RDNSequence item: 1 item (id-at-organizationName=Cisco Systems, Inc)
          > subjectPublicKeyInfo
          > extensions: 1 item
        > algorithmIdentifier (sha256WithRSAEncryption)
          Padding: 0
          encrypted [truncated]: 1db20bca36b8c7d8a8330df5e81b94ed7b2d6abe977785aadd7255f00929592add145659ce78c9f4c871192752413a1eb2956b28a26de2577e1786d36c833026ebab116e858abb83e2d8eeb0025d02f7a863ff3670339a9f15de2c9282ea150e070768e971cf32031a070c6c
∨ TLSv1.2 Record Layer: Handshake Protocol: Certificate Request
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 198
  ∨ Handshake Protocol: Certificate Request
```

FMC-InternalCA_validity

Problem

The FMC InternalCA certificate is only valid for 10 years. After the expiry time, the remote system does not trust this certificate anymore (as well as certificates signed by it) and this leads to sftunnel communication issues between FTD and FMC devices (and also FMC HA communication). This means as well that several key functionalities like connection events, malware lookups, identity based rules, policy deployments and many other things are not working.

The devices do show up as disabled on the FMC UI under the **Devices > Device Management** tab when the sftunnel is not connected. The issue that relates to this expiry is tracked on Cisco bug ID CSCwd08098. Note though that all systems are affected, even when you run a fixed release of the defect. More information on this fix is found in the Solution section.



	Name	Model	Version	Chassis	Licenses	Access Control Policy	Auto RollBack
	Ungrouped (3)						
	BSNS-1120-3 Snort 3 10.48.67.69 - Routed	Firepower 1120 with FTD	7.0.1	N/A	Essentials, IPS (2 more...)	Allow-Any	N/A
	EMEA-FPR3105-19 Snort 3 10.48.189.24 - Routed	Firewall 3105 Threat Defense	7.4.1	Manage	Essentials	Allow-Any	

Disabled-devices

The FMC does not automatically refresh the CA and republish the certificates to the FTD devices. And there is also no FMC health alert which indicates that the certificate expires. Cisco bug ID CSCwd08448 is tracked in this regards to provide a health alert on the FMC UI in the future.

What happens after the expiry date?

Initially nothing happens and the sftunnel communication channels continue to operate as before. However when the sftunnel communication between FMC and FTD devices gets broken and it tries to re-establish the connection, it does fail and you can observe log lines on the messages log file that point to the certificate expiry. Note that the sftunnel communication (used for High-Availability (HA) sync) between primary and secondary FMC is also potentially impacted as described in this section.

Log lines from FTD device from **/ngfw/var/log/messages**:

```
Sep 20 04:10:47 FTD-hostname SF-IMS[50792]: [51982] sftunneld:sf_ssl [INFO] Initiating IPv4 connection 
Sep 20 04:10:47 FTD-hostname SF-IMS[50792]: [51982] sftunneld:sf_ssl [INFO] Wait to connect to 8305 (IP
Sep 20 04:10:47 FTD-hostname SF-IMS[50792]: [51982] sftunneld:sf_ssl [INFO] Connected to 10.10.200.31 f
Sep 20 04:10:47 FTD-hostname SF-IMS[50792]: [51982] sftunneld:sf_ssl [ERROR] -Error with certificate at
Sep 20 04:10:47 FTD-hostname SF-IMS[50792]: [51982] sftunneld:sf_ssl [ERROR]   issuer   = /title=Intern
Sep 20 04:10:47 FTD-hostname SF-IMS[50792]: [51982] sftunneld:sf_ssl [ERROR]   subject  = /title=Intern
Sep 20 04:10:47 FTD-hostname SF-IMS[50792]: [51982] sftunneld:sf_ssl [ERROR]   err 10:certificate has e
Sep 20 04:10:47 FTD-hostname SF-IMS[50792]: [51982] sftunneld:sf_ssl [ERROR] SSL_renegotiate error: 1: 
Sep 20 04:10:47 FTD-hostname SF-IMS[50792]: [51982] sftunneld:sf_ssl [ERROR] Connect:SSL handshake fail
Sep 20 04:10:47 FTD-hostname SF-IMS[50792]: [51982] sftunneld:sf_ssl [WARN] SSL Verification status: ce
```

Log lines from FMC device from **/var/log/messages**:

```
Sep 20 03:14:23 FMC-hostname SF-IMS[1504]: [4171] sftunneld:sf_ssl [INFO] VERIFY ssl_verify_callback_in
Sep 20 03:14:23 FMC-hostname SF-IMS[1504]: [4171] sftunneld:sf_ssl [ERROR] SSL_renegotiate error: 1: er
Sep 20 03:14:23 FMC-hostname SF-IMS[1504]: [4171] sftunneld:sf_ssl [WARN] establishConnectionUtil: SSL 
Sep 20 03:14:23 FMC-hostname SF-IMS[1504]: [4171] sftunneld:sf_ssl [WARN] establishConnectionUtil: SSL 
Sep 20 03:14:23 FMC-hostname SF-IMS[1504]: [4171] sftunneld:sf_ssl [WARN] establishConnectionUtil: SSL 
Sep 20 03:14:23 FMC-hostname SF-IMS[1504]: [4171] sftunneld:sf_ssl [INFO] establishConnectionUtil: Fail
Sep 20 03:14:23 FMC-hostname SF-IMS[1504]: [4171] sftunneld:sf_ssl [ERROR] establishSSLConnection: Unab
Sep 20 03:14:23 FMC-hostname SF-IMS[1504]: [4171] sftunneld:sf_ssl [ERROR] establishSSLConnection: ret_
Sep 20 03:14:23 FMC-hostname SF-IMS[1504]: [4171] sftunneld:sf_ssl [ERROR] establishSSLConnection: iret
Sep 20 03:14:23 FMC-hostname SF-IMS[1504]: [4171] sftunneld:sf_ssl [ERROR] establishSSLConnection: Fail
```

The sftunnel communication can be broken due to various reasons:

- Communication loss because of loss of network connectivity (potentially only temporary)
- Reboot of FTD or FMC
    - Expected ones: manual reboot, upgrades, manual restart of sftunnel process on FMC or FTD (for example by pmtool restartbyid sftunnel)
    - Unexpected ones: tracebacks, power outage

Because there are so many possibilities that can break the sftunnel communication, it is highly advised to correct on the situation as quickly as possible, even when currently all FTD devices are properly connected despite the expired certificate.

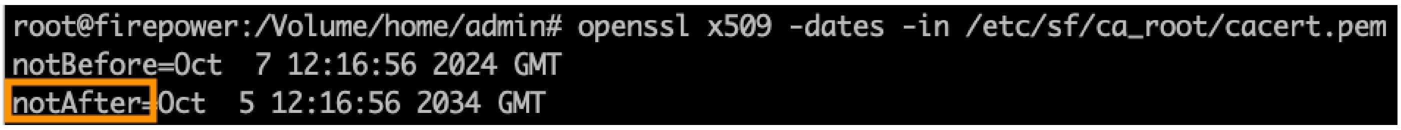
How to quickly verify if the certificate is expired or when it does expire?

The easiest way is to run these commands on the FMC SSH session:

```
expert
```

```
sudo su
cd /etc/sf/ca_root
openssl x509 -dates -noout -in cacert.pem
```

This shows you the Validity elements of the certificate. The main relevant part here is the "notAfter" which shows that the certificate here is valid till 5th of October 2034.



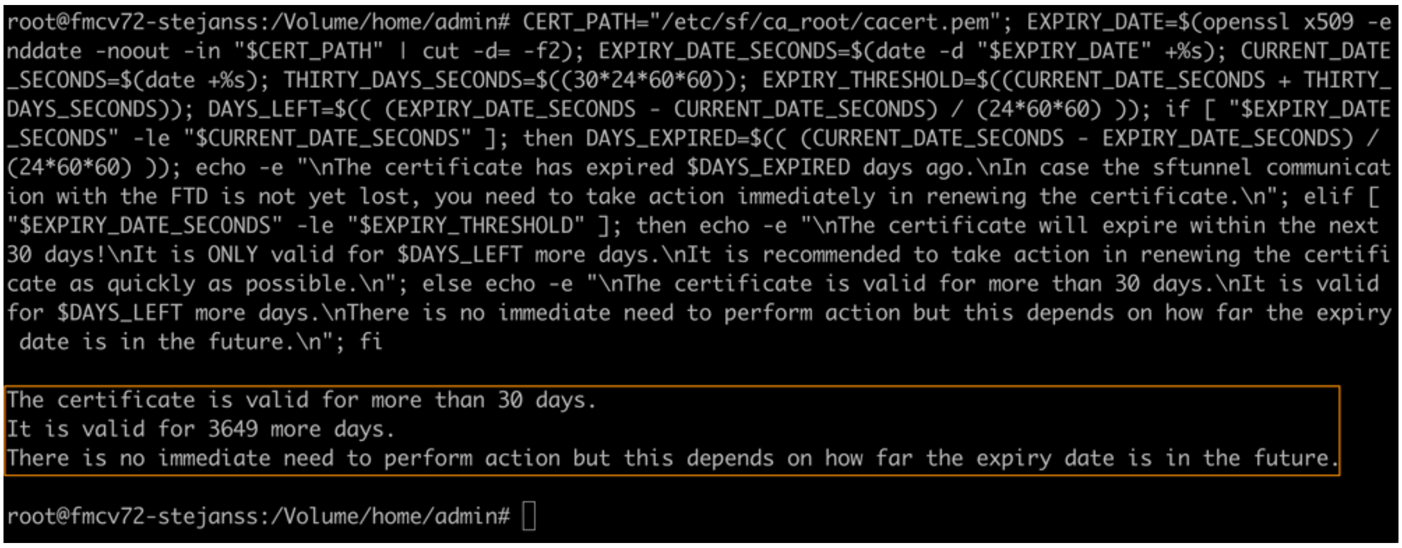
```
root@firepower:/Volume/home/admin# openssl x509 -dates -in /etc/sf/ca_root/cacert.pem
notBefore=Oct  7 12:16:56 2024 GMT
notAfter=Oct  5 12:16:56 2034 GMT
```

NotAfter

If you prefer a single command to be ran that immediately gives you the amount of days that the certificate is still valid for, you can use this:

```
CERT_PATH="/etc/sf/ca_root/cacert.pem"; EXPIRY_DATE=$(openssl x509 -enddate -noout -in "$CERT_PATH" | c
```

An example of a setup where the certificate is still valid for multiple years is shown.



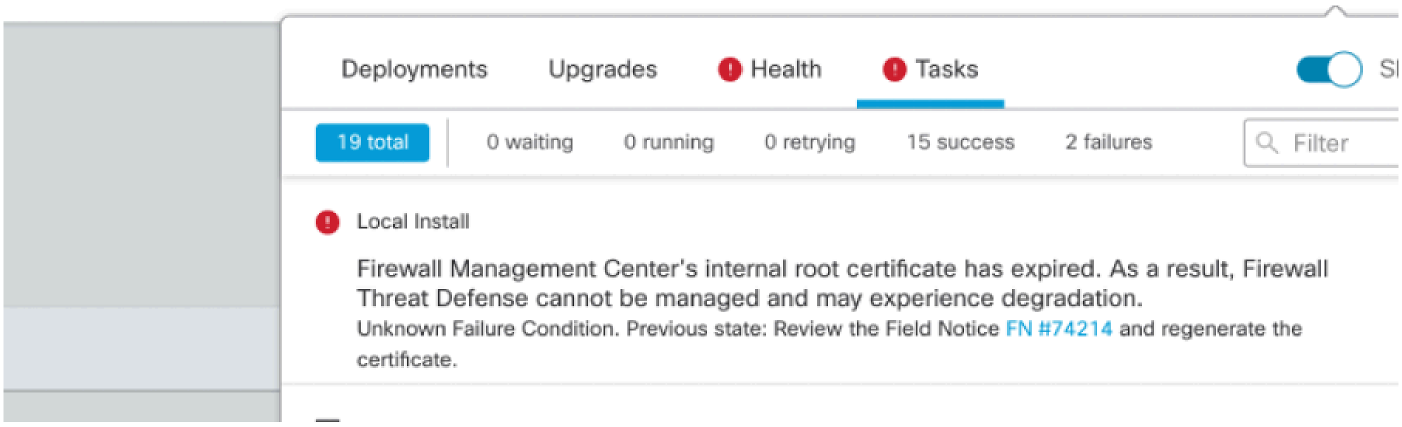
```
root@fmcv72-stejanss:/Volume/home/admin# CERT_PATH="/etc/sf/ca_root/cacert.pem"; EXPIRY_DATE=$(openssl x509 -e
nddate -noout -in "$CERT_PATH" | cut -d= -f2); EXPIRY_DATE_SECONDS=$(date -d "$EXPIRY_DATE" +%s); CURRENT_DATE
_SECONDS=$(date +%s); THIRTY_DAYS_SECONDS=$((30*24*60*60)); EXPIRY_THRESHOLD=$((CURRENT_DATE_SECONDS + THIRTY_
DAYS_SECONDS)); DAYS_LEFT=$(( (EXPIRY_DATE_SECONDS - CURRENT_DATE_SECONDS) / (24*60*60) )); if [ "$EXPIRY_DATE
_SECONDS" -le "$CURRENT_DATE_SECONDS" ]; then DAYS_EXPIRED=$(( (CURRENT_DATE_SECONDS - EXPIRY_DATE_SECONDS) /
(24*60*60) )); echo -e "\nThe certificate has expired $DAYS_EXPIRED days ago.\nIn case the sftunnel communicat
ion with the FTD is not yet lost, you need to take action immediately in renewing the certificate.\n"; elif [
"$EXPIRY_DATE_SECONDS" -le "$EXPIRY_THRESHOLD" ]; then echo -e "\nThe certificate will expire within the next
30 days!\nIt is ONLY valid for $DAYS_LEFT more days.\nIt is recommended to take action in renewing the certifi
cate as quickly as possible.\n"; else echo -e "\nThe certificate is valid for more than 30 days.\nIt is valid
for $DAYS_LEFT more days.\nThere is no immediate need to perform action but this depends on how far the expiry
 date is in the future.\n"; fi

The certificate is valid for more than 30 days.
It is valid for 3649 more days.
There is no immediate need to perform action but this depends on how far the expiry date is in the future.

root@fmcv72-stejanss:/Volume/home/admin# 
```

Certificate_expiry_validation_command

How do I get notified in the future about an upcoming certificate expiry?

With recent VDB updates (399 or higher), you are alerted automatically when your certificate expires within 90 days. Therefore you do not need to manually track on this yourself as you are alerted when you are close to the expiry time. This then shows up on the FMC web page in two forms. Both ways refer to the field notice page.

The first method is through **Task Tab**. This message is sticky and available to the user unless explicitly closed. The notification pop up also shows up and is available until explicitly closed by the user. It does always show up as an error.

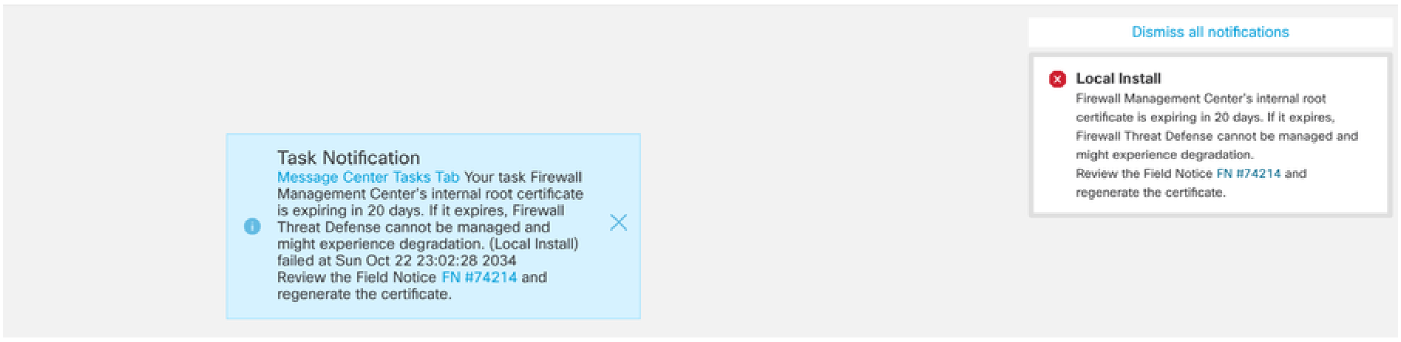

Deployments Upgrades Health Tasks

19 total | 0 waiting | 0 running | 0 retrying | 15 success | 2 failures

Local Install

Firewall Management Center's internal root certificate has expired. As a result, Firewall Threat Defense cannot be managed and may experience degradation.
Unknown Failure Condition. Previous state: Review the Field Notice FN #74214 and regenerate the certificate.

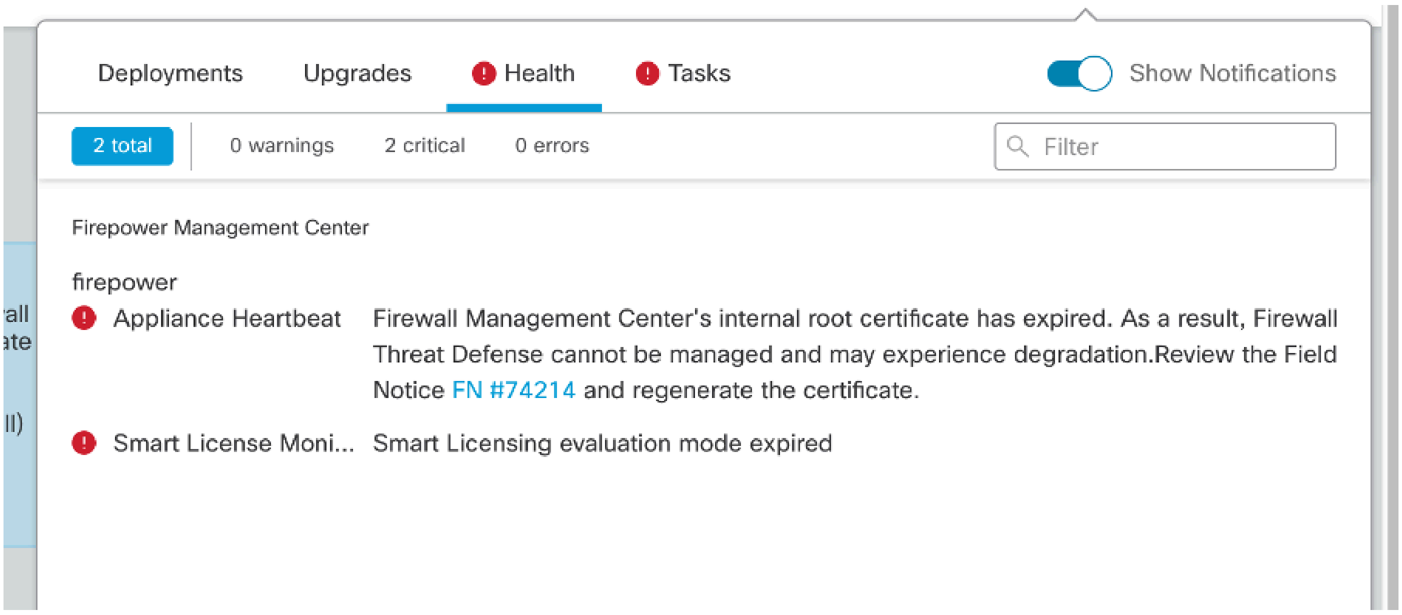
Expiry Notification on Task Tab


Task Notification
Message Center Tasks Tab Your task Firewall Management Center's internal root certificate is expiring in 20 days. If it expires, Firewall Threat Defense cannot be managed and might experience degradation. (Local Install) failed at Sun Oct 22 23:02:28 2034
Review the Field Notice FN #74214 and regenerate the certificate.

Dismiss all notifications

Local Install
Firewall Management Center's internal root certificate is expiring in 20 days. If it expires, Firewall Threat Defense cannot be managed and might experience degradation.
Review the Field Notice FN #74214 and regenerate the certificate.

The second method is through **Health Alert**. This shows up in the Health tab however this is not sticky and replaces or removes when health monitor is run which by default is every 5 minutes. It also shows up a notification pop up which needs to be explicitly closed by the user. This can show up both as error (when expired) as a warning (when going to expire).


Deployments Upgrades Health Tasks Show Notifications

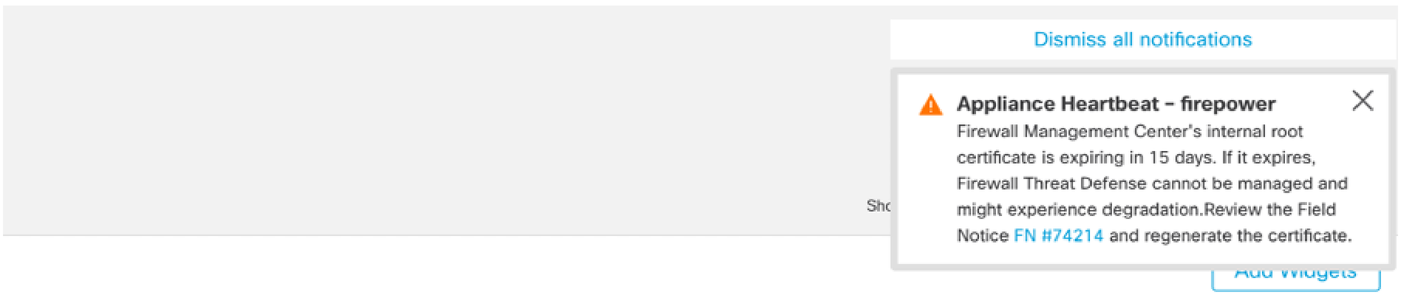
2 total | 0 warnings | 2 critical | 0 errors

Firepower Management Center

firepower

Appliance Heartbeat	Firewall Management Center's internal root certificate has expired. As a result, Firewall Threat Defense cannot be managed and may experience degradation.Review the Field Notice FN #74214 and regenerate the certificate.
Smart License Moni...	Smart Licensing evaluation mode expired

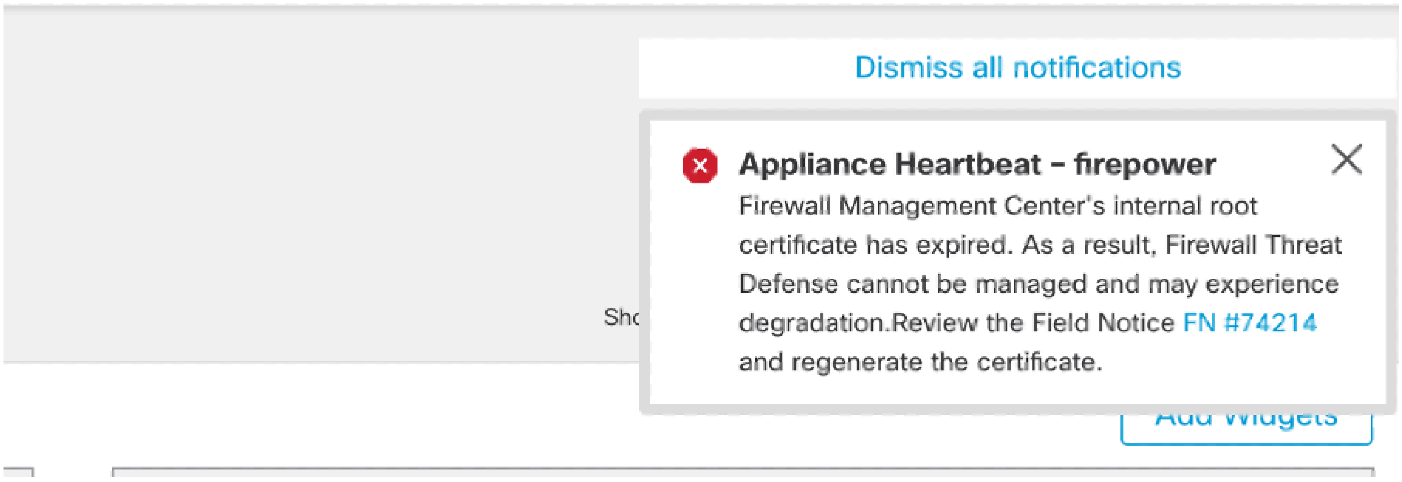
Expiry notification on Health tab


Dismiss all notifications

Appliance Heartbeat - firepower

Firewall Management Center's internal root certificate is expiring in 15 days. If it expires, Firewall Threat Defense cannot be managed and might experience degradation.Review the Field Notice FN #74214 and regenerate the certificate.

Warning notification on Health Alert Pop Up


Dismiss all notifications

Appliance Heartbeat - firepower

Firewall Management Center's internal root certificate has expired. As a result, Firewall Threat Defense cannot be managed and may experience degradation.Review the Field Notice FN #74214 and regenerate the certificate.

Error notification on Health Alert Pop Up

Solution 1 - Certificate has not yet expired (ideal scenario)

This is the best situation as then depending on the certificate expiry, we still have time. Either we take the fully automated approach (recommended) that has a dependency on the FMC version or we take on a more manual approach which requires TAC interaction.

Recommended approach

This is the situation where no down time and least amount of manual operations is expected in normal circumstances.

Before proceeding, you must install the hotfix for your particular version as listed here. The benefit here is that those hotfixes do not require a reboot of the FMC and thus potential broken sftunnel communication when the certificate is expired already. The available hotfixes (download links are for virtual FMC, for hardware FMC look on the appropriate download pages for the versions) are:

- 7.0.0 - 7.0.6 : Hotfix FK - 7.0.6.99-9
- 7.1.x : no fixed release as end of software maintenance
- 7.2.0 - 7.2.9 : Hotfix FZ - 7.2.9.99-4
- 7.3.x : Hotfix AE - 7.3.1.99-4
- 7.4.0 - 7.4.2 : Hotfix AO - 7.4.2.99-5
- 7.6.0 : Hotfix B - 7.6.0.99-5

Once the hotfix is installed, the FMC now contains the **generate_certs.pl** script that:

1. Regenerates the InternalCA
2. Recreates the sftunnel certificates signed by this new InternalCA

3. Pushes the new sftunnel certificates and private keys over to the respective FTD devices (and secondary FMC when applicable) (when the sftunnel is operational)

Note: The **generate_certs.pl** script currently checks whether critical operations are running. If not, then it fails to run.

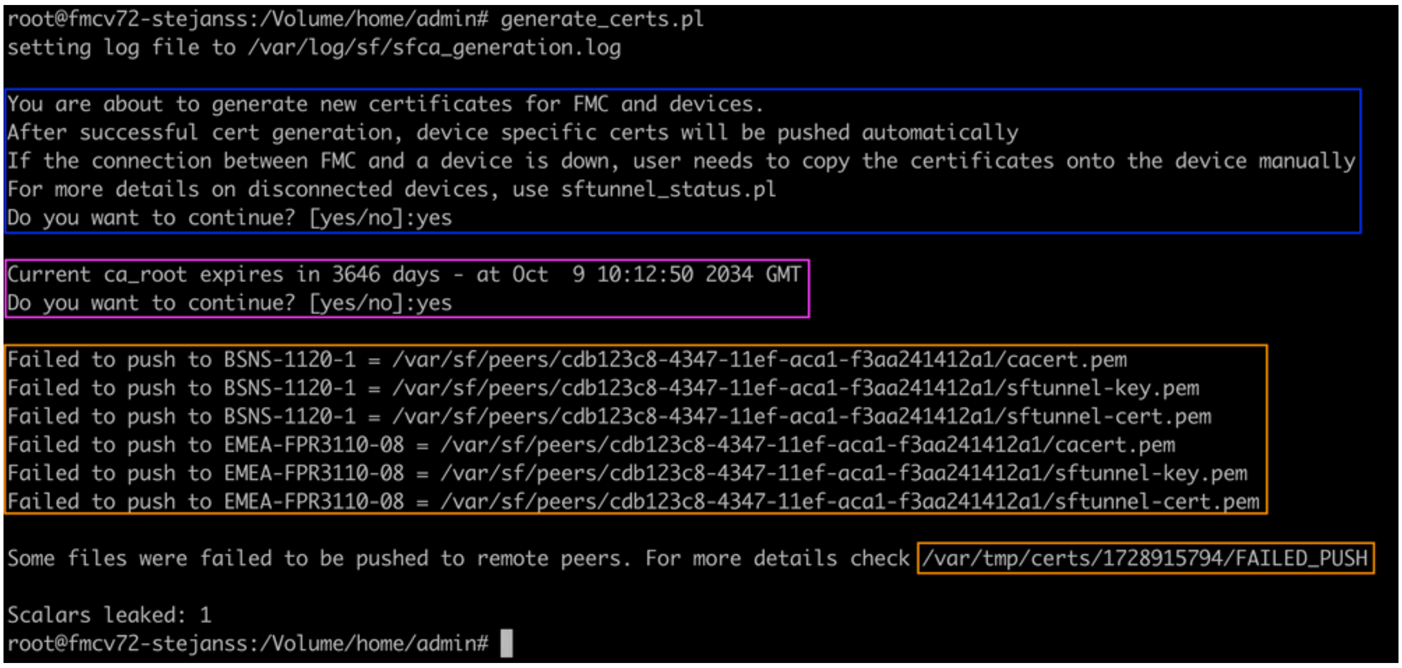
Critical operations can be: Smart agent not registered or registration in progress, Backup/Restore task in progress, SRU update task in progress, VDB update task in progress, Domain task in progress, HA Operation in progress or Upgrade is running.

Therefore you cannot run this script when you only use Classic Licenses on your FMC (or any of the listed operations need to complete first) in which case you need to contact Cisco TAC to bypass this check, regenerate the certificates and then undo the bypass again.

Therefore it is recommended (if possible) to:

1. Install the applicable hotfix for your version train
2. Take a backup on the FMC
3. Validate all current sftunnel connections using **sftunnel_status.pl** script on the FMC (from **expert** mode)

4. Run the script from expert mode using **generate_certs.pl**
5. Inspect the outcome to validate if any manual operations are required (when devices are not connected to FMC) [explained further below]
6. Run **sftunnel_status.pl** from the FMC to validate that all of the sftunnel connections are running fine



```
root@fmcv72-stejanss:/Volume/home/admin# generate_certs.pl
setting log file to /var/log/sf/sfca_generation.log

You are about to generate new certificates for FMC and devices.
After successful cert generation, device specific certs will be pushed automatically
If the connection between FMC and a device is down, user needs to copy the certificates onto the device manually
For more details on disconnected devices, use sftunnel_status.pl
Do you want to continue? [yes/no]:yes

Current ca_root expires in 3646 days - at Oct  9 10:12:50 2034 GMT
Do you want to continue? [yes/no]:yes

Failed to push to BSNS-1120-1 = /var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1/cacert.pem
Failed to push to BSNS-1120-1 = /var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1/sftunnel-key.pem
Failed to push to BSNS-1120-1 = /var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1/sftunnel-cert.pem
Failed to push to EMEA-FPR3110-08 = /var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1/cacert.pem
Failed to push to EMEA-FPR3110-08 = /var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1/sftunnel-key.pem
Failed to push to EMEA-FPR3110-08 = /var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1/sftunnel-cert.pem

Some files were failed to be pushed to remote peers. For more details check /var/tmp/certs/1728915794/FAILED_PUSH

Scalars leaked: 1
root@fmcv72-stejanss:/Volume/home/admin#
```

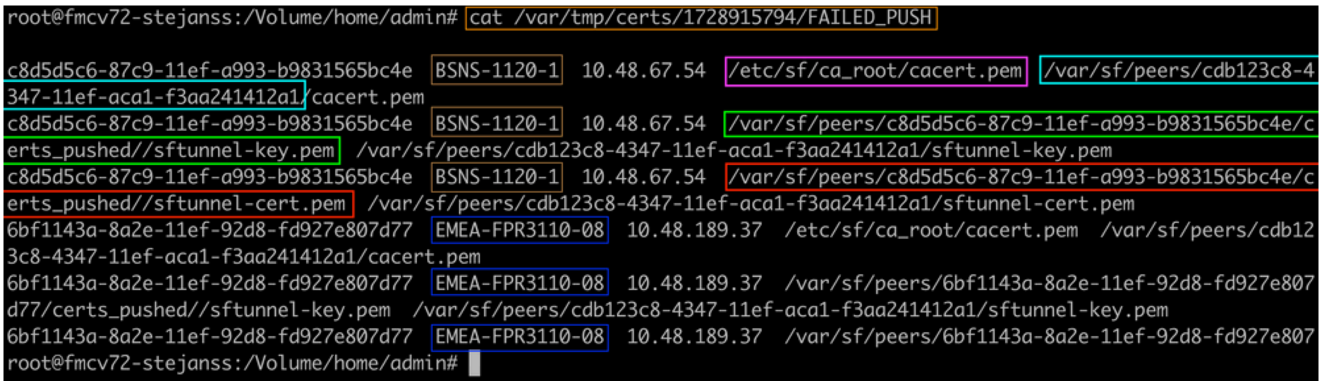
Generate_certs.pl script

Note: When you have FMC running in High-Availability (HA), you need to perform the operation first on the primary node and then on the secondary node as it uses those certificates as well to communicate between the FMC nodes. The InternalCA on both FMC nodes is different so they do have different expiry times. You can find more information about the certificates in FMC HA on [this section](#).

On the example here you see that it creates up a log file on **/var/log/sf/sfca_generation.log**, indicates to use **sftunnel_status.pl**, indicates the expiry time on the InternalCA and indicates for any failures on it. Here for example it failed to push the certificates over to device BSNS-1120-1 and EMEA-FPR3110-08 device, which is expected because the sftunnel was down for those devices.

In order to correct the sftunnel for the failed connections, you run the next steps:

1. On FMC CLI, open the FAILED_PUSH file using **cat /var/tmp/certs/1728303362/FAILED_PUSH** (number value represents unix time, so check the output of previous command in your system) which has the next format: **FTD_UUID FTD_NAME FTD_IP SOURCE_PATH_ON_FMC DESTINATION_PATH_ON_FTD**



```
root@fmcv72-stejanss:/Volume/home/admin# cat /var/tmp/certs/1728915794/FAILED_PUSH

c8d5d5c6-87c9-11ef-a993-b9831565bc4e  BSNS-1120-1  10.48.67.54  /etc/sf/ca_root/cacert.pem  /var/sf/peers/cdb123c8-4
347-11ef-aca1-f3aa241412a1/cacert.pem
c8d5d5c6-87c9-11ef-a993-b9831565bc4e  BSNS-1120-1  10.48.67.54  /var/sf/peers/c8d5d5c6-87c9-11ef-a993-b9831565bc4e/c
erts_pushed//sftunnel-key.pem  /var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1/sftunnel-key.pem
c8d5d5c6-87c9-11ef-a993-b9831565bc4e  BSNS-1120-1  10.48.67.54  /var/sf/peers/c8d5d5c6-87c9-11ef-a993-b9831565bc4e/c
erts_pushed//sftunnel-cert.pem  /var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1/sftunnel-cert.pem
6bf1143a-8a2e-11ef-92d8-fd927e807d77  EMEA-FPR3110-08  10.48.189.37  /etc/sf/ca_root/cacert.pem  /var/sf/peers/cdb12
3c8-4347-11ef-aca1-f3aa241412a1/cacert.pem
6bf1143a-8a2e-11ef-92d8-fd927e807d77  EMEA-FPR3110-08  10.48.189.37  /var/sf/peers/6bf1143a-8a2e-11ef-92d8-fd927e807
d77/certs_pushed//sftunnel-key.pem  /var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1/sftunnel-key.pem
6bf1143a-8a2e-11ef-92d8-fd927e807d77  EMEA-FPR3110-08  10.48.189.37  /var/sf/peers/6bf1143a-8a2e-11ef-92d8-fd927e807
root@fmcv72-stejanss:/Volume/home/admin#
```

FAILED_PUSH

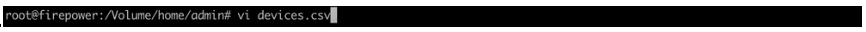
2. Transfer over those new certificates (cacert.pem / sftunnel-key.pem / sftunnel-cert.pem) from the FMC to the FTD devices
   ===Automatic approach===

   The hotfix installation also provides the **copy_sftunnel_certs.py** and **copy_sftunnel_certs_jumpserver.py** scripts that automate the transfer of the various certificates to systems for which the sftunnel was not up while the certificates were regenerated. This can also be used for systems that had a broken sftunnel connection because the certificate was expired already.

   You can use the **copy_sftunnel_certs.py** script when the FMC itself has SSH access to the various FTD systems (and potentially the secondary FMC if applicable). If that is not the case, you can download the script (**/usr/local/sf/bin/copy_sftunnel_certs_jumpserver.py**) from the FMC to a jump server that has SSH access to both the FMC(s) and FTD devices and run the Python script from there. If that is also not possible, then suggest to run the manual approach shown next. The next examples show the **copy_sftunnel_certs.py** script being used, but the steps are the same for the **copy_sftunnel_certs_jumpserver.py** script.

   A. Create a CSV file on the FMC (or jump server) that contains the device info (device_name, IP address, admin_username, admin_password) that is used to make the SSH connection.

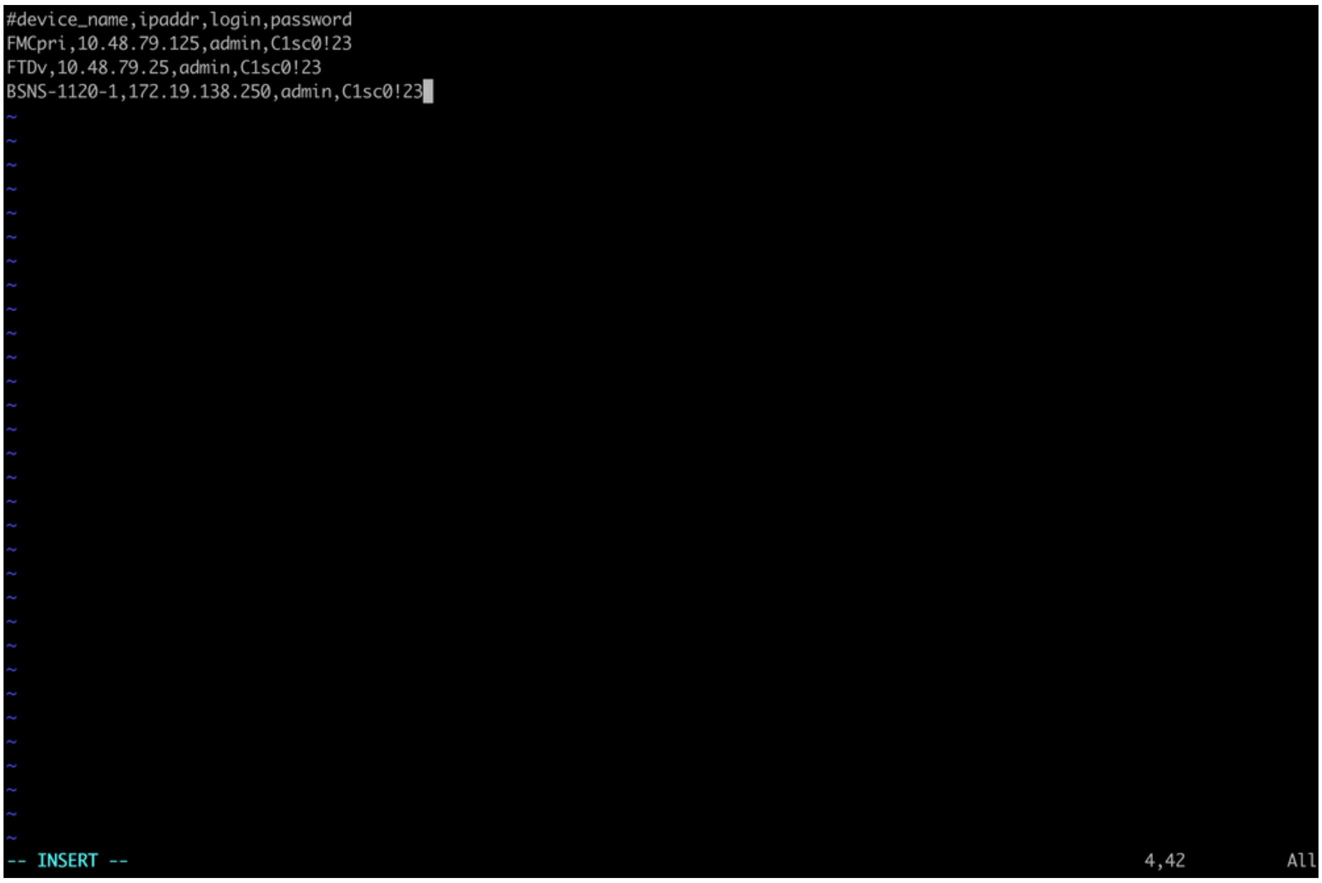
   When you run this from a remote server like a jump server for Primary FMC, make sure to add in the primary FMC details as the first entry followed by all managed FTD and secondary FMC. When you run this from a remote server like a jump server for Secondary FMC, make sure to add in the secondary FMC details as the first entry followed by all managed FTD. More info on the certificates in FMC HA can be found in [this section]().

   i. Create a file using **vi devices.csv**.

```
root@firepower:/Volume/home/admin# vi devices.csv
```

   vi devices.csv

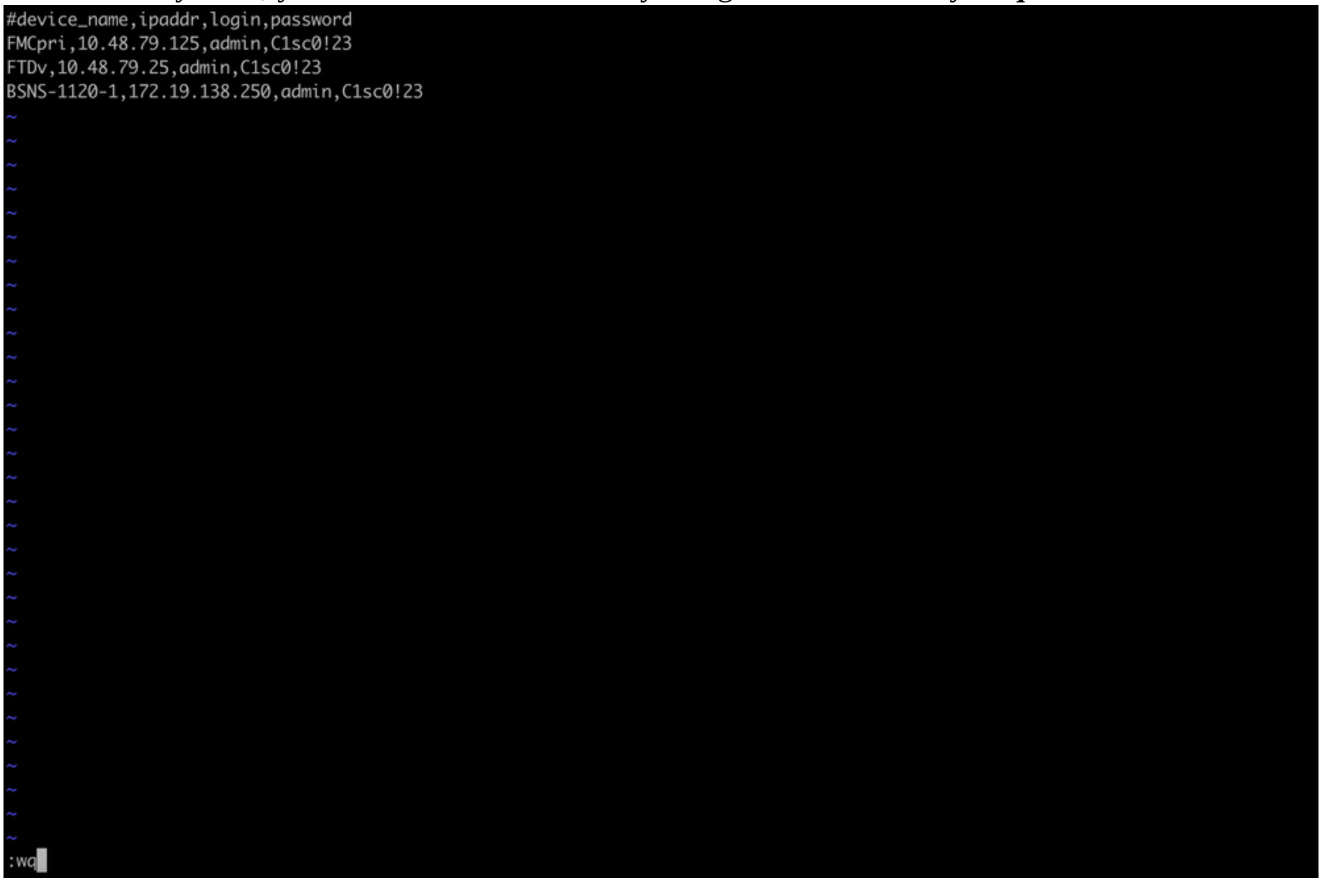
   ii. This opens up the empty file (not shown) and you fill in the details as shown after you use **i** letter on keyboard to go into INTERACTIVE mode (seen at bottom of the screen).



```
#device_name,ipaddr,login,password
FMCpri,10.48.79.125,admin,C1sc0!23
FTDv,10.48.79.25,admin,C1sc0!23
BSNS-1120-1,172.19.138.250,admin,C1sc0!23
~
-- INSERT --                                              4,42          All
```

devices.csv example

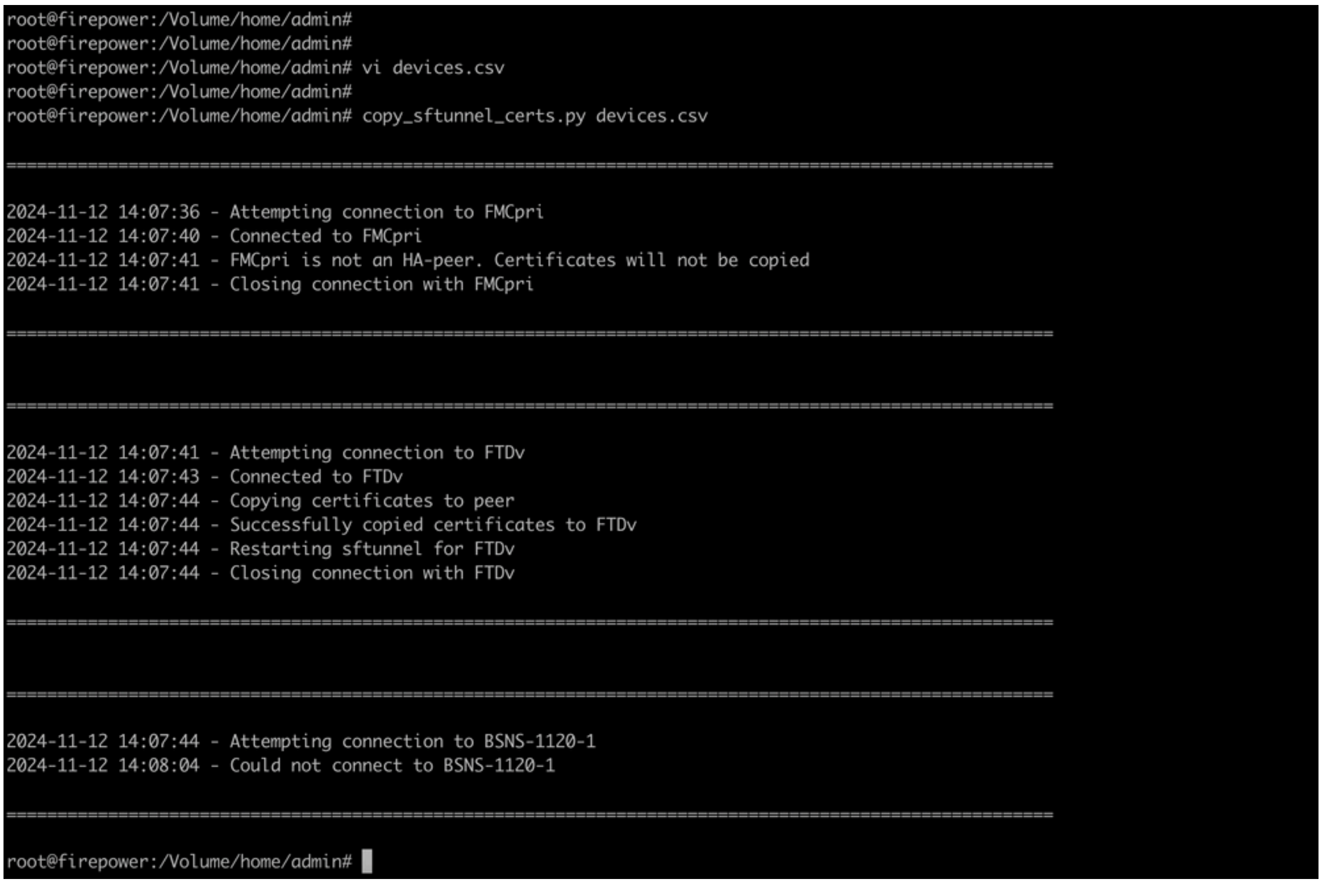
iii. When fully done, you close and save the file by using **ESC** followed by **:wq** and then Enter.



```
#device_name,ipaddr,login,password
FMCpri,10.48.79.125,admin,C1sc0!23
FTDv,10.48.79.25,admin,C1sc0!23
BSNS-1120-1,172.19.138.250,admin,C1sc0!23
~
:wq
```

Save the devices.csv

B. Run the script (from root using **sudo**) with **copy_sftunnel_certs.py devices.csv** and it shows you the outcome. Here it shows that the certificate to FTDv was correctly pushed and that for BSNS-1120-1 it could not make the SSH connection to the device.

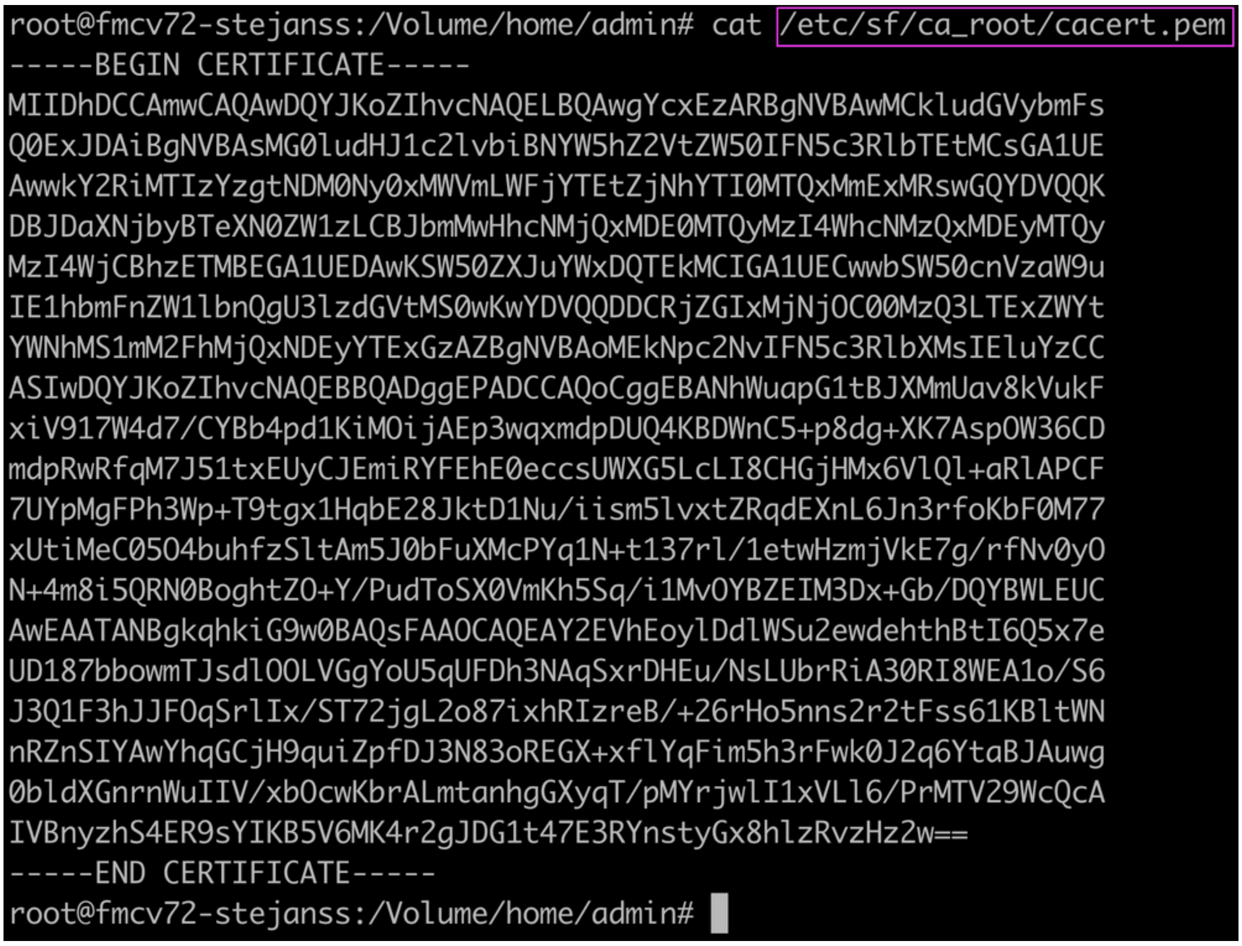


```
root@firepower:/Volume/home/admin#
root@firepower:/Volume/home/admin#
root@firepower:/Volume/home/admin# vi devices.csv
root@firepower:/Volume/home/admin#
root@firepower:/Volume/home/admin# copy_sftunnel_certs.py devices.csv

==============================================================================================

2024-11-12 14:07:36 - Attempting connection to FMCpri
2024-11-12 14:07:40 - Connected to FMCpri
2024-11-12 14:07:41 - FMCpri is not an HA-peer. Certificates will not be copied
2024-11-12 14:07:41 - Closing connection with FMCpri

==============================================================================================



==============================================================================================

2024-11-12 14:07:41 - Attempting connection to FTDv
2024-11-12 14:07:43 - Connected to FTDv
2024-11-12 14:07:44 - Copying certificates to peer
2024-11-12 14:07:44 - Successfully copied certificates to FTDv
2024-11-12 14:07:44 - Restarting sftunnel for FTDv
2024-11-12 14:07:44 - Closing connection with FTDv

==============================================================================================



==============================================================================================

2024-11-12 14:07:44 - Attempting connection to BSNS-1120-1
2024-11-12 14:08:04 - Could not connect to BSNS-1120-1

==============================================================================================

root@firepower:/Volume/home/admin# █
```

copy_sftunnel_certs.py devices.csv

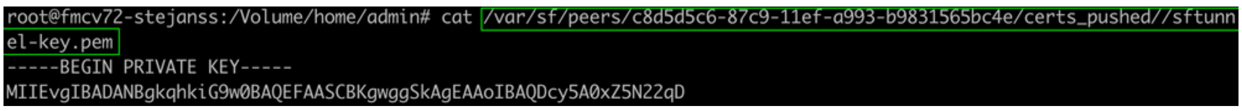
===Manual approach===

1. Print (**cat**) the output each of the files for each FTD (or secondary FMC in HA) impacted (cacert.pem / sftunnel-key.pem (not shown completely for security purposes) / sftunnel-cert.pem) on the FMC CLI by copying the file location from previous output (FAILED_PUSH file).



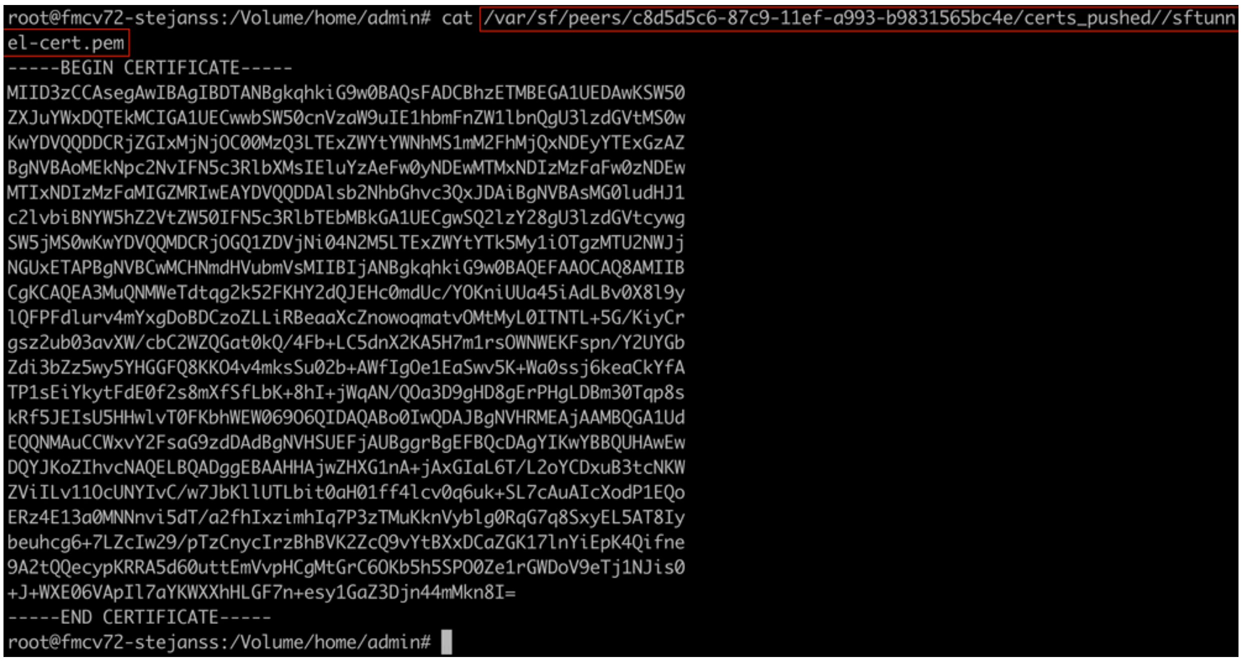
```
root@fmcv72-stejanss:/Volume/home/admin# cat /etc/sf/ca_root/cacert.pem
-----BEGIN CERTIFICATE-----
MIIDhDCCAmwCAQAwDQYJKoZIhvcNAQELBQAwgYcxEzARBgNVBAwMCkludGVybmFs
Q0ExJDAiBgNVBAsMG0ludHJ1c2lvbiBNYW5hZ2VtZW50IFN5c3RlbTEtMCsGA1UE
AwwkY2RiMTIzYzgtNDM0Ny0xMWVmLWFjYTEtZjNhYTI0MTQxMmExMRswGQYDVQQK
DBJDaXNjbyBTeXN0ZW1zLCBJbmMwHhcNMjQxMDE0MTQyMzI4WhcNMzQxMDEyMTQy
MzI4WjCBhzETMBEGA1UEDAwKSW50ZXJuYWxDQTEkMCIGA1UECwwbSW50cnVzaW9u
IE1hbmFnZW1lbnQgU3lzdGVtMS0wKwYDVQQDDCRjZGIxMjNjOC00MzQ3LTExZWYt
YWNhMS1mM2FhMjQxNDEyYTExGzAZBgNVBAoMEkNpc2NvIFN5c3RlbXMsIEluYzCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANhWuapG1tBJXMmUav8kVukF
xiV917W4d7/CYBb4pd1KiMOijAEp3wqxmdpDUQ4KBDWnC5+p8dg+XK7AspOW36CD
mdpRwRfqM7J51txEUyCJEmiRYFEhE0eccsUWXG5LcLI8CHGjHMx6VlQl+aRlAPCF
7UYpMgFPh3Wp+T9tgx1HqbE28JktD1Nu/iism5lvxtZRqdEXnL6Jn3rfoKbF0M77
xUtiMeC05O4buhfzSltAm5J0bFuXMcPYq1N+t137rl/1etwHzmjVkE7g/rfNv0yO
N+4m8i5QRN0BoghtZO+Y/PudToSX0VmKh5Sq/i1MvOYBZEIM3Dx+Gb/DQYBWLEUC
AwEAATANBgkqhkiG9w0BAQsFAAOCAQEAY2EVhEoylDdlWSu2ewdehthBtI6Q5x7e
UD187bbowmTJsdlOOLVGgYoU5qUFDh3NAqSxrDHEu/NsLUbrRiA30RI8WEA1o/S6
J3Q1F3hJJFOqSrlIx/ST72jgL2o87ixhRIzreB/+26rHo5nns2r2tFss61KBltWN
nRZnSIYAwYhqGCjH9quiZpfDJ3N83oREGX+xflYqFim5h3rFwk0J2q6YtaBJAuwg
0bldXGnrnWuIIV/xbOcwKbrALmtanhgGXyqT/pMYrjwlI1xVLl6/PrMTV29WcQcA
IVBnyzhS4ER9sYIKB5V6MK4r2gJDG1t47E3RYnstyGx8hlzRvzHz2w==
-----END CERTIFICATE-----
root@fmcv72-stejanss:/Volume/home/admin# 
```

cacert.pem



```
root@fmcv72-stejanss:/Volume/home/admin# cat /var/sf/peers/c8d5d5c6-87c9-11ef-a993-b9831565bc4e/certs_pushed//sftunn
el-key.pem
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDcy5A0xZ5N22qD
```

sftunnel-key.pem

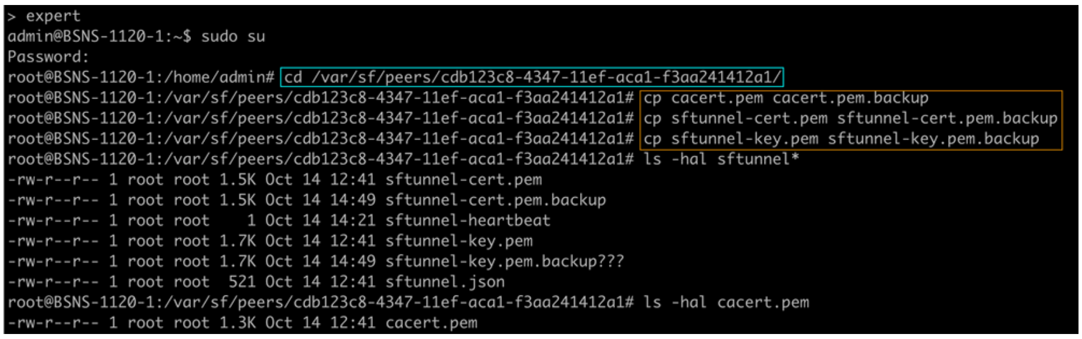


```
root@fmcv72-stejanss:/Volume/home/admin# cat /var/sf/peers/c8d5d5c6-87c9-11ef-a993-b9831565bc4e/certs_pushed//sftunn
el-cert.pem
-----BEGIN CERTIFICATE-----
MIID3zCCAsegAwIBAgIBDTANBgkqhkiG9w0BAQsFADCBhzETMBEGA1UEDAwKSW50
ZXJuYWxDQTEkMCIGA1UECwwbSW50cnVzaW9uIE1hbmFnZW1lbnQgU3lzdGVtMS0w
KwYDVQQDDCRjZGIxMjNjOC00MzQ3LTExZWYtYWNhMS1mM2FhMjQxNDEyYTExGzAZ
BgNVBAoMEkNpc2NvIFN5c3RlbXMsIEluYzAeFw0yNDEwMTMxNDIzMzFaFw0zNDEw
MTIxNDIzMzFaMIGZMRIwEAYDVQQDDAlsb2NhbGhvc3QxJDAiBgNVBAsMG0ludHJ1
c2lvbiBNYW5hZ2VtZW50IFN5c3RlbTEbMBkGA1UECgwSQ2lzY28gU3lzdGVtcywg
SW5jMS0wKwYDVQQMDCRjOGQ1ZDVjNi04N2M5LTExZWYtYTk5My1iOTgzMTU2NWJj
NGUxETAPBgNVBCwMCHNmdHVubmVsMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA3MuQNMWeTdtqg2k52FKHY2dQJEHc0mdUc/YOKniUUa45iAdLBv0X8l9y
lQFPFdlurv4mYxgDoBDCzoZLLiRBeaaXcZnowoqmatvOMtMyL0ITNTL+5G/KiyCr
gsz2ub03avXW/cbC2WZQGat0kQ/4Fb+LC5dnX2KA5H7m1rsOWNWEKFspn/Y2UYGb
Zdi3bZz5wy5YHGGFQ8KKO4v4mksSu02b+AWfIgOe1EaSwv5K+Wa0ssj6keaCkYfA
TP1sEiYkytFdE0f2s8mXfSfLbK+8hI+jWqAN/QOa3D9gHD8gErPHgLDBm30Tqp8s
kRf5JEIsU5HHwlvT0FKbhWEW069O6QIDAQABo0IwQDAJBgNVHRMEAjAAMBQGA1Ud
EQQNMAuCCWxvY2FsaG9zdDAdBgNVHSUEFjAUBggrBgEFBQcDAgYIKwYBBQUHAwEw
DQYJKoZIhvcNAQELBQADggEBAAHHAjwZHXG1nA+jAxGIaL6T/L2oYCDxuB3tcNKW
ZViILv11OcUNYIvC/w7JbKllUTLbit0aH01ff4lcv0q6uk+SL7cAuAIcXodP1EQo
ERz4E13a0MNNnvi5dT/a2fhIxzimhIq7P3zTMuKknVyblg0RqG7q8SxyEL5AT8Iy
beuhcg6+7LZcIw29/pTzCnycIrzBhBVK2ZcQ9vYtBXxDCaZGK17lnYiEpK4Qifne
9A2tQQecypKRRA5d60uttEmVvpHCgMtGrC6OKb5h5SPO0Ze1rGWDoV9eTj1NJis0
+J+WXE06VApIl7aYKWXXhHLGF7n+esy1GaZ3Djn44mMkn8I=
-----END CERTIFICATE-----
root@fmcv72-stejanss:/Volume/home/admin# 
```

sftunnel-cert.pem

2. Open FTD (or secondary FMC when in FMC HA) CLI of each respective FTD on **expert** mode with root privileges through **sudo su** and renew the certificates with the next procedure.
    1. Browse to the location seen on the light blue highlight from FAILED_PUSH output (**cd /var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1** here for example but this is different for each FTD).
    2. Take backups of the existing files.

```
cp cacert.pem cacert.pem.backup
cp sftunnel-cert.pem sftunnel-cert.pem.backup
cp sftunnel-key.pem sftunnel-key.pem.backup
```

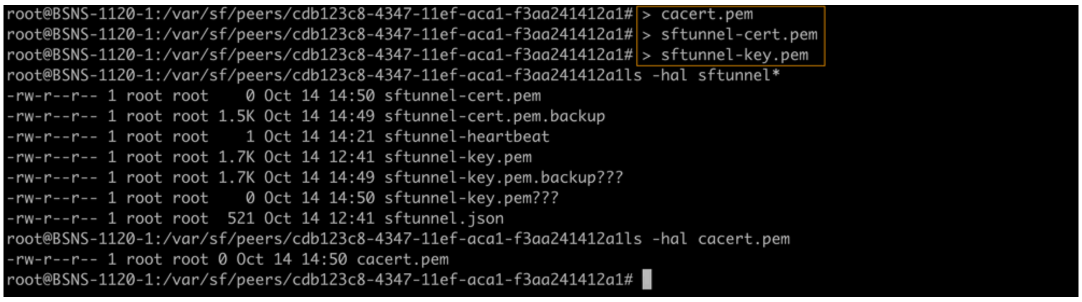


```
> expert
admin@BSNS-1120-1:~$ sudo su
Password:
root@BSNS-1120-1:/home/admin# cd /var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1/
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1# cp cacert.pem cacert.pem.backup
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1# cp sftunnel-cert.pem sftunnel-cert.pem.backup
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1# cp sftunnel-key.pem sftunnel-key.pem.backup
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1# ls -hal sftunnel*
-rw-r--r-- 1 root root 1.5K Oct 14 12:41 sftunnel-cert.pem
-rw-r--r-- 1 root root 1.5K Oct 14 14:49 sftunnel-cert.pem.backup
-rw-r--r-- 1 root root    1 Oct 14 14:21 sftunnel-heartbeat
-rw-r--r-- 1 root root 1.7K Oct 14 12:41 sftunnel-key.pem
-rw-r--r-- 1 root root 1.7K Oct 14 14:49 sftunnel-key.pem.backup???
-rw-r--r-- 1 root root  521 Oct 14 12:41 sftunnel.json
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1# ls -hal cacert.pem
-rw-r--r-- 1 root root 1.3K Oct 14 12:41 cacert.pem
```

Take backups of the current certificates

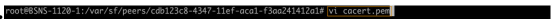
    3. Empty the files so we can write up new content in them.

```
> cacert.pem
> sftunnel-cert.pem
> sftunnel-key.pem
```



```
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1# > cacert.pem
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1# > sftunnel-cert.pem
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1# > sftunnel-key.pem
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1ls -hal sftunnel*
-rw-r--r-- 1 root root    0 Oct 14 14:50 sftunnel-cert.pem
-rw-r--r-- 1 root root 1.5K Oct 14 14:49 sftunnel-cert.pem.backup
-rw-r--r-- 1 root root    1 Oct 14 14:21 sftunnel-heartbeat
-rw-r--r-- 1 root root 1.7K Oct 14 12:41 sftunnel-key.pem
-rw-r--r-- 1 root root 1.7K Oct 14 14:49 sftunnel-key.pem.backup???
-rw-r--r-- 1 root root    0 Oct 14 14:50 sftunnel-key.pem???
-rw-r--r-- 1 root root  521 Oct 14 12:41 sftunnel.json
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1ls -hal cacert.pem
-rw-r--r-- 1 root root 0 Oct 14 14:50 cacert.pem
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1#
```

Empty contents of existing certificate files

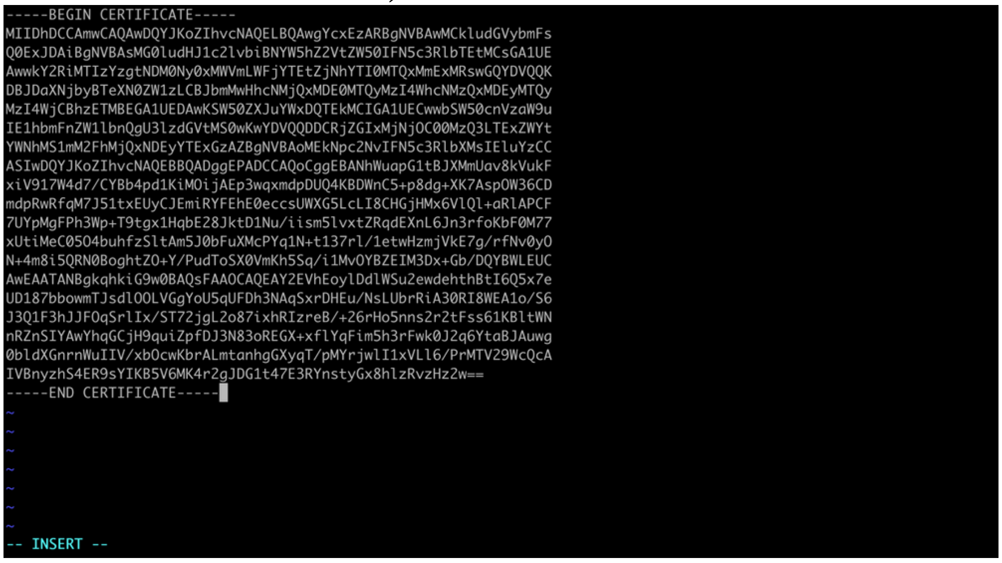
    4. Write the new content (from FMC output) in each of the files individually using **vi cacert.pem / vi sftunnel-cert.pem / vi sftunnel-key.pem** (separate command per file - screenshots only show this for cacert.pem but needs to be repeated for sftunnel-cert.pem and sftunnel-key.pem).

```
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1# vi cacert.pem
```

vi cacert.pem

        1. Press **i** to go into interactive mode (after vi command is entered and you see an empty file).

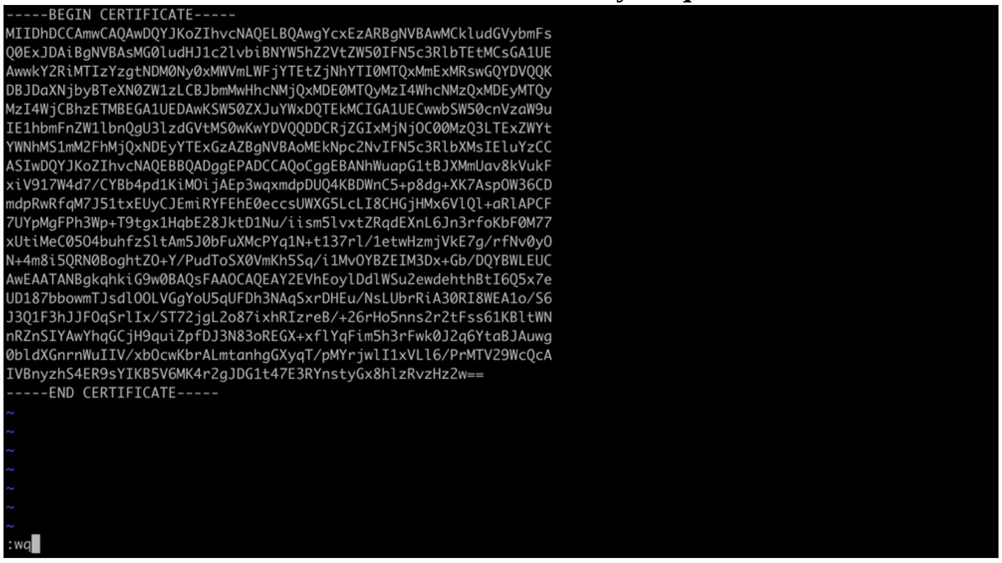
2. Copy paste the entire content (including -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----) in the file.



```
-----BEGIN CERTIFICATE-----
MIIDhDCCAmwCAQAwDQYJKoZIhvcNAQELBQAwgYcxEzARBgNVBAwMCkludGVybmFs
Q0ExJDAiBgNVBAsMG0ludHJ1c2lvbiBNYW5hZ2VtZW50IFN5c3RlbTEtMCsGA1UE
AwwkY2RiMTIzYzgtNDM0Ny0xMWVmLWFjYTEtZjNhYTI0MTQxMmExMRswGQYDVQQK
DBJDaXNjbyBTeXN0ZW1zLCBJbmMwHhcNMjQxMDE0MTQyMzI4WhcNMzQxMDEyMTQy
MzI4WjCBhzETMBEGA1UEDAwKSW50ZXJuYWxDQTEkMCIGA1UECwwbSW50cnVzaW9u
IE1hbmFnZW1lbnQgU3lzdGVtMS0wKwYDVQQDDCRjZGIxMjNjOC00MzQ3LTExZWYt
YWNhMS1mM2FhMjQxNDEyYTExGzAZBgNVBAoMEkNpc2NvIFN5c3RlbXMsIEluYzCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANhWuapG1tBJXMmUav8kVukF
xiV917W4d7/CYBb4pd1KiMOijAEp3wqxmdpDUQ4KBDWnC5+p8dg+XK7AspOW36CD
mdpRwRfqM7J51txEUyCJEmiRYFEhE0eccsUWXG5LcLI8CHGjHMx6VlQl+aRlAPCF
7UYpMgFPh3Wp+T9tgx1HqbE28JktD1Nu/iism5lvxtZRqdEXnL6Jn3rfoKbF0M77
xUtiMeC05O4buhfzSltAm5J0bFuXMcPYq1N+t137rl/1etwHzmjVkE7g/rfNv0yO
N+4m8i5QRN0BoghtZO+Y/PudToSX0VmKh5Sq/i1MvOYBZEIM3Dx+Gb/DQYBWLEUC
AwEAATANBgkqhkiG9w0BAQsFAAOCAQEAY2EVhEoylDdlWSu2ewdehthBtI6Q5x7e
UD187bbowmTJsdlOOLVGgYoU5qUFDh3NAqSxrDHEu/NsLUbrRiA30RI8WEA1o/S6
J3Q1F3hJJFOqSrlIx/ST72jgL2o87ixhRIzreB/+26rHo5nns2r2tFss61KBltWN
nRZnSIYAwYhqGCjH9quiZpfDJ3N83oREGX+xflYqFim5h3rFwk0J2q6YtaBJAuwg
0bldXGnrnWuIIV/xbOcwKbrALmtanhgGXyqT/pMYrjwlI1xVLl6/PrMTV29WcQcA
IVBnyzhS4ER9sYIKB5V6MK4r2gJDG1t47E3RYnstyGx8hlzRvzHz2w==
-----END CERTIFICATE-----
~
~
~
~
~
~
~
-- INSERT --
```

Copy content in vi (INSERT mode)

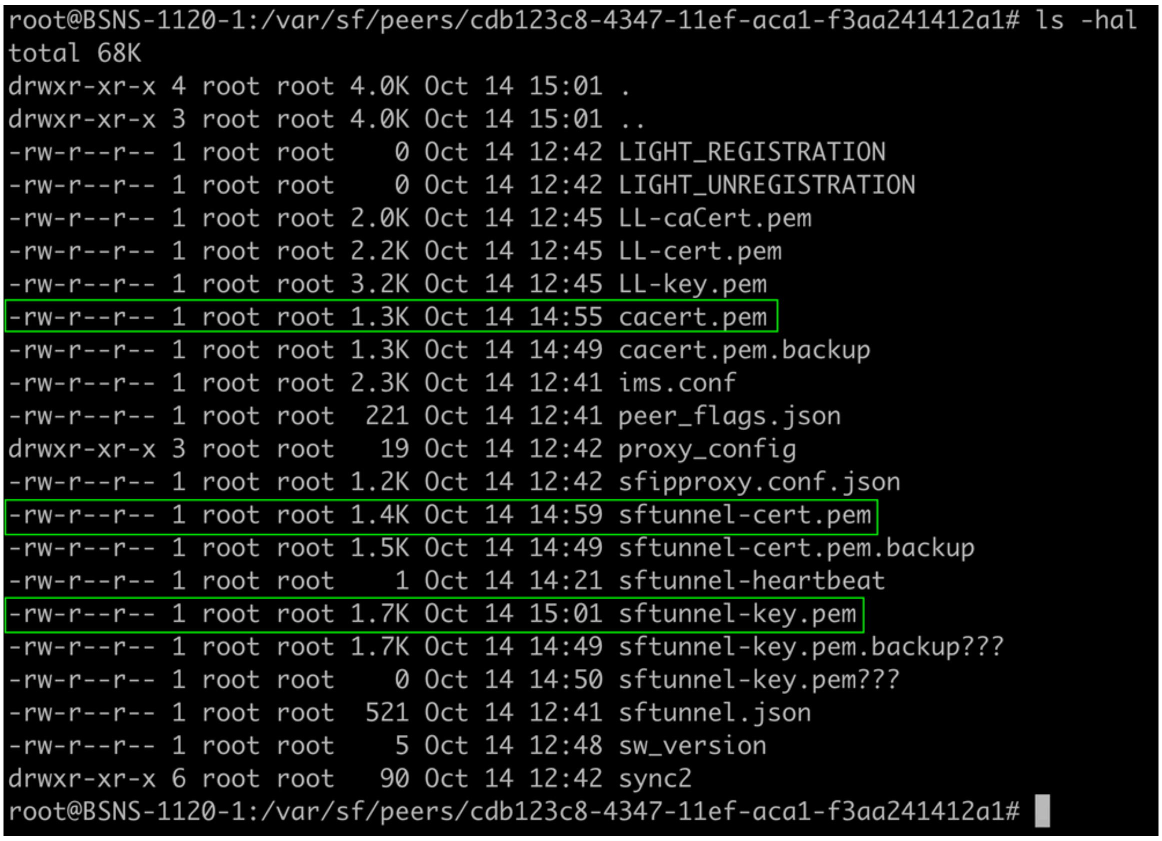
3. Close and write to the file with **ESC** followed by **:wq** and then enter.



```
-----BEGIN CERTIFICATE-----
MIIDhDCCAmwCAQAwDQYJKoZIhvcNAQELBQAwgYcxEzARBgNVBAwMCkludGVybmFs
Q0ExJDAiBgNVBAsMG0ludHJ1c2lvbiBNYW5hZ2VtZW50IFN5c3RlbTEtMCsGA1UE
AwwkY2RiMTIzYzgtNDM0Ny0xMWVmLWFjYTEtZjNhYTI0MTQxMmExMRswGQYDVQQK
DBJDaXNjbyBTeXN0ZW1zLCBJbmMwHhcNMjQxMDE0MTQyMzI4WhcNMzQxMDEyMTQy
MzI4WjCBhzETMBEGA1UEDAwKSW50ZXJuYWxDQTEkMCIGA1UECwwbSW50cnVzaW9u
IE1hbmFnZW1lbnQgU3lzdGVtMS0wKwYDVQQDDCRjZGIxMjNjOC00MzQ3LTExZWYt
YWNhMS1mM2FhMjQxNDEyYTExGzAZBgNVBAoMEkNpc2NvIFN5c3RlbXMsIEluYzCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANhWuapG1tBJXMmUav8kVukF
xiV917W4d7/CYBb4pd1KiMOijAEp3wqxmdpDUQ4KBDWnC5+p8dg+XK7AspOW36CD
mdpRwRfqM7J51txEUyCJEmiRYFEhE0eccsUWXG5LcLI8CHGjHMx6VlQl+aRlAPCF
7UYpMgFPh3Wp+T9tgx1HqbE28JktD1Nu/iism5lvxtZRqdEXnL6Jn3rfoKbF0M77
xUtiMeC05O4buhfzSltAm5J0bFuXMcPYq1N+t137rl/1etwHzmjVkE7g/rfNv0yO
N+4m8i5QRN0BoghtZO+Y/PudToSX0VmKh5Sq/i1MvOYBZEIM3Dx+Gb/DQYBWLEUC
AwEAATANBgkqhkiG9w0BAQsFAAOCAQEAY2EVhEoylDdlWSu2ewdehthBtI6Q5x7e
UD187bbowmTJsdlOOLVGgYoU5qUFDh3NAqSxrDHEu/NsLUbrRiA30RI8WEA1o/S6
J3Q1F3hJJFOqSrlIx/ST72jgL2o87ixhRIzreB/+26rHo5nns2r2tFss61KBltWN
nRZnSIYAwYhqGCjH9quiZpfDJ3N83oREGX+xflYqFim5h3rFwk0J2q6YtaBJAuwg
0bldXGnrnWuIIV/xbOcwKbrALmtanhgGXyqT/pMYrjwlI1xVLl6/PrMTV29WcQcA
IVBnyzhS4ER9sYIKB5V6MK4r2gJDG1t47E3RYnstyGx8hlzRvzHz2w==
-----END CERTIFICATE-----
~
~
~
~
~
~
~
:wq
```

ESC followed by :wq to write to file

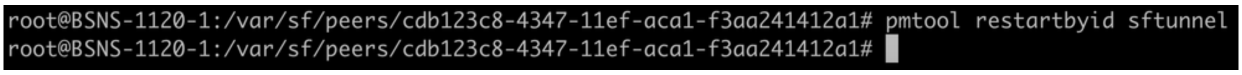
1. Validate that the correct permissions **(chmod 644)** and owners **(chown root:root)** are set for each of the files using **ls -hal**. This is correctly set actually when we update the existing file.



```
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1# ls -hal
total 68K
drwxr-xr-x 4 root root 4.0K Oct 14 15:01 .
drwxr-xr-x 3 root root 4.0K Oct 14 15:01 ..
-rw-r--r-- 1 root root    0 Oct 14 12:42 LIGHT_REGISTRATION
-rw-r--r-- 1 root root    0 Oct 14 12:42 LIGHT_UNREGISTRATION
-rw-r--r-- 1 root root 2.0K Oct 14 12:45 LL-caCert.pem
-rw-r--r-- 1 root root 2.2K Oct 14 12:45 LL-cert.pem
-rw-r--r-- 1 root root 3.2K Oct 14 12:45 LL-key.pem
-rw-r--r-- 1 root root 1.3K Oct 14 14:55 cacert.pem
-rw-r--r-- 1 root root 1.3K Oct 14 14:49 cacert.pem.backup
-rw-r--r-- 1 root root 2.3K Oct 14 12:41 ims.conf
-rw-r--r-- 1 root root  221 Oct 14 12:41 peer_flags.json
drwxr-xr-x 3 root root   19 Oct 14 12:42 proxy_config
-rw-r--r-- 1 root root 1.2K Oct 14 12:42 sfipproxy.conf.json
-rw-r--r-- 1 root root 1.4K Oct 14 14:59 sftunnel-cert.pem
-rw-r--r-- 1 root root 1.5K Oct 14 14:49 sftunnel-cert.pem.backup
-rw-r--r-- 1 root root    1 Oct 14 14:21 sftunnel-heartbeat
-rw-r--r-- 1 root root 1.7K Oct 14 15:01 sftunnel-key.pem
-rw-r--r-- 1 root root 1.7K Oct 14 14:49 sftunnel-key.pem.backup???
-rw-r--r-- 1 root root    0 Oct 14 14:50 sftunnel-key.pem???
-rw-r--r-- 1 root root  521 Oct 14 12:41 sftunnel.json
-rw-r--r-- 1 root root    5 Oct 14 12:48 sw_version
drwxr-xr-x 6 root root   90 Oct 14 12:42 sync2
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1#
```

All certificate files updated with right owners and permissions

3. Restart the sftunnel on each respective FTD (or secondary FMC when in FMC HA) where the sftunnel was not operational for the changes in the certificate to take effect with the command **pmtool restartbyid sftunnel**



```
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1# pmtool restartbyid sftunnel
root@BSNS-1120-1:/var/sf/peers/cdb123c8-4347-11ef-aca1-f3aa241412a1#
```

pmtool restartbyid sftunnel

3. Validate that all FTDs (and secondary FMC) are correctly connected now using **sftunnel_status.pl** output

Solution 2 - Certificate has already expired

In this situation, we have two different scenarios. Either all of the sftunnel connections are still operational or they are not anymore (or partial).

FTDs still connected through sftunnel

We can apply the same procedure as indicated in the Certificate has not yet expired (ideal scenario) - Recommended approach section.

However do NOT upgrade or reboot the FMC (or any FTD) in this situation as it disconnects all of the sftunnel connections and we need to manually run all of the certificate updates on each FTD (and secondary

FMC as covered on this section). The only exception to this one, are the listed Hotfix releases as they do not require a reboot of the FMC.

The tunnels remain connected and the certificates are replaced on each of the FTDs (and secondary FMC when in FMC HA). In case some certificates would fail to populate, it does prompt you with the ones that failed and you need to take the manual approach as indicated earlier on the previous section.

FTDs not connected anymore through sftunnel

Recommended approach

We can apply the same procedure as indicated in the Certificate has not yet expired (ideal scenario) - Recommended approach section. In this scenario, the new certificate does get generated on the FMC but cannot be copied to the devices as the tunnel is already down. This process can be automated with the copy_sftunnel_certs.py / copy_sftunnel_certs_jumpserver.py scripts

If all of the FTD devices are disconnected from the FMC, we can upgrade the FMC in this situation as it does not have an impact on the sftunnel connections. If you still have some devices connected through sftunnel, then be aware that the upgrade of the FMC closes all of the sftunnel connections and they do not come up again due to the expired certificate. The benefit of the upgrade here would be that it does provide you a good guidance on the certificate files that need to be transferred to each of the FTDs (and secondary FMC when in FMC HA).

Manual approach

In this situation, you can then run the **generate_certs.pl** script from the FMC which generates the new certificates but you still need to push them over to each of the FTD (and secondary FMC as covered on this section) devices manually. Depending on the amount of devices, this is doable or can be a tedious task. However when using the copy_sftunnel_certs.py / copy_sftunnel_certs_jumpserver.py scripts this is highly automated.

Appendix

Certificates used in FMC HA

The communication between primary and secondary FMC in high-availability pair happens as well over the sftunnel, similarly to a FMC that communicates over to the FTD devices. The certificate elements (cacert.pem / sftunnel-cert.pem / sftunnel-key.pem) are stored on the FMC in the same place under **/var/sf/peers/<UUID-FMC>**. However it is always the primary FMC that acts as the manager for the secondary FMC. So in this way, the secondary FMC is like a FTD device from the perspective of the primary FMC. Therefore you only see the certificate on the secondary FMC and not on the files on the primary FMC on the **/var/sf/peers/** folder.

Often your secondary FMC has been created at a later time than your primary FMC and thus the internal CA can still be valid and not yet expired unlike your primary FMC. Therefore a manual role switch of the active FMC could be an option to minimize the impact in those situations where the secondary FMC would still have a valid certificate and thus associated sftunnel communications with all of the FTDs. This would then still allow you to deploy configurations while in the meantime then also prepare the fix or update on the certificate of the primary FMC. However the communication between both FMC devices could be broken as well when the sftunnel was down after the certificate expiry of the primary FMC.

In the process of renewing the certificates, there are two different scenarios:

1. Primary FMC CA certificate is expired: as the secondary FMC is seen like a FTD to the primary FMC, the certificate renewal needs to push out the updated certificates created on the primary FMC not only to the FTD devices but also towards the secondary FMC.

2. Secondary FMC CA certificate is expired: in this situation the only certificate updates that are required, are the ones to the FTD devices. Because the sftunnel communication between both FMCs in HA is based on the primary FMC CA certificate.