Configure Site-to-Site Tunnel between FTD and StrongSwan Server

Contents

Introduction

This document describes how to configure a Site-To-Site IKEv2 VPN connection between Cisco FTD and StrongSwan using Certification Authentication.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Site-To-Site VPN
- Certificate Authentication (IKEv2)
- Public Key Infrastructure (PKI)
- Basic knowledge of StrongSwan

Components Used

The information in this document is based on these software and hardware versions:

- Cisco FTD running version 7.2.0 build 18
- Cisco FMC running version 7.2.0 build 18
- Ubuntu Server Running Version 20.04 (Focal Fossa)

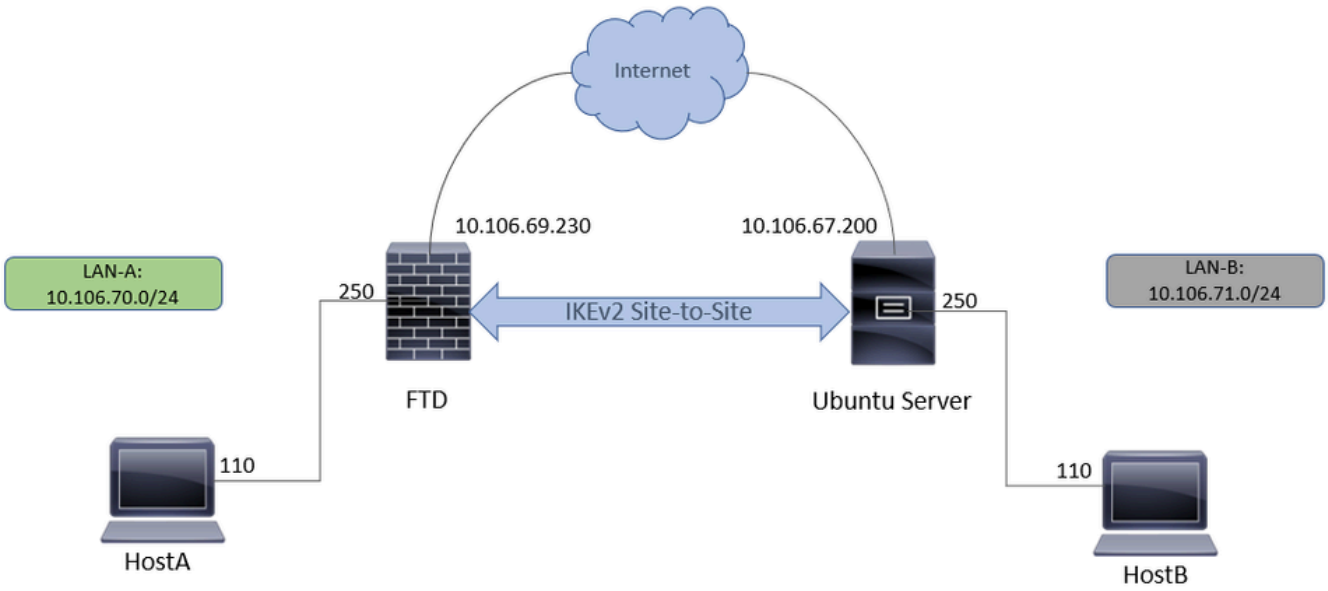
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Configure

Scenario

In this setup, **HOST-A in LAN-A** wants to communicate with **HOST-B in LAN-B**. This traffic must be encrypted and sent over an IKEv2 tunnel between FTD and the Ubuntu Server running StrongSwan. Both peers authenticate each other with **Certificate Authentication**.

Network Diagram



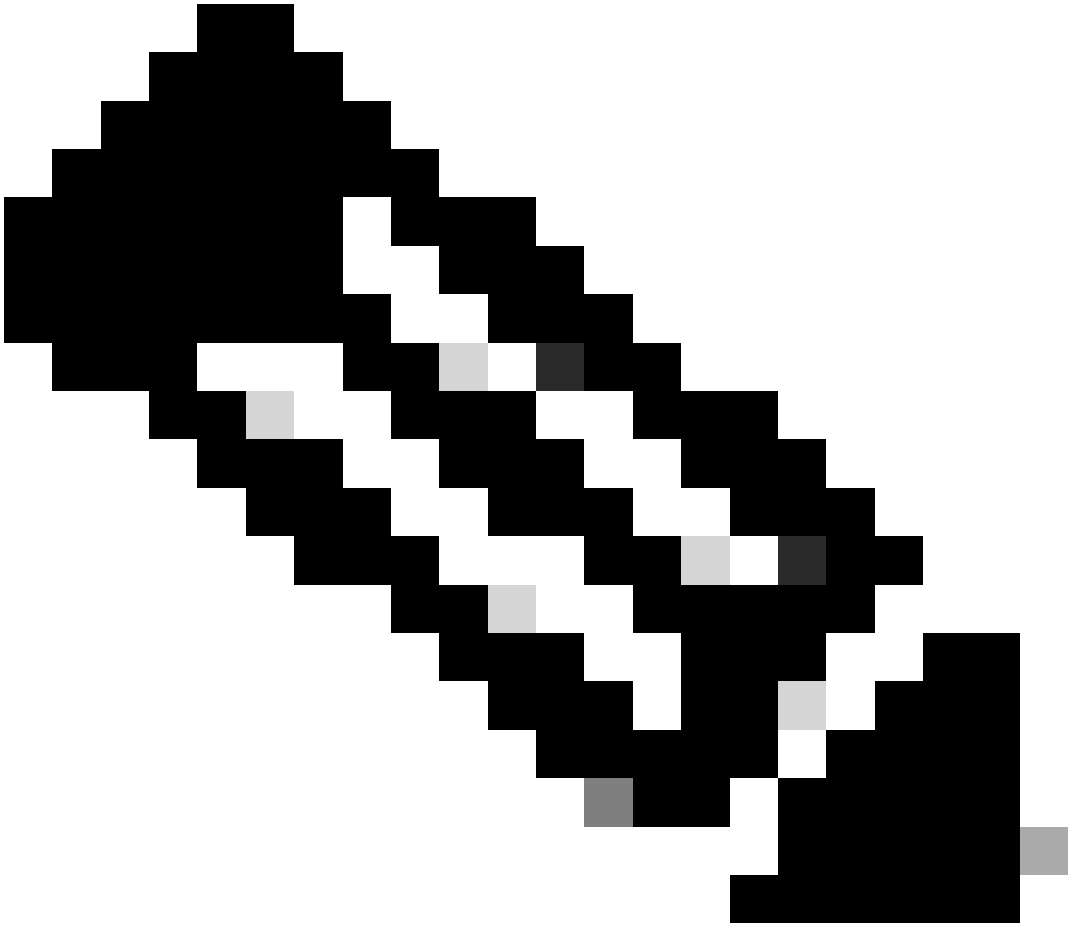
FMC Configuration

Getting a Certificate for FTD

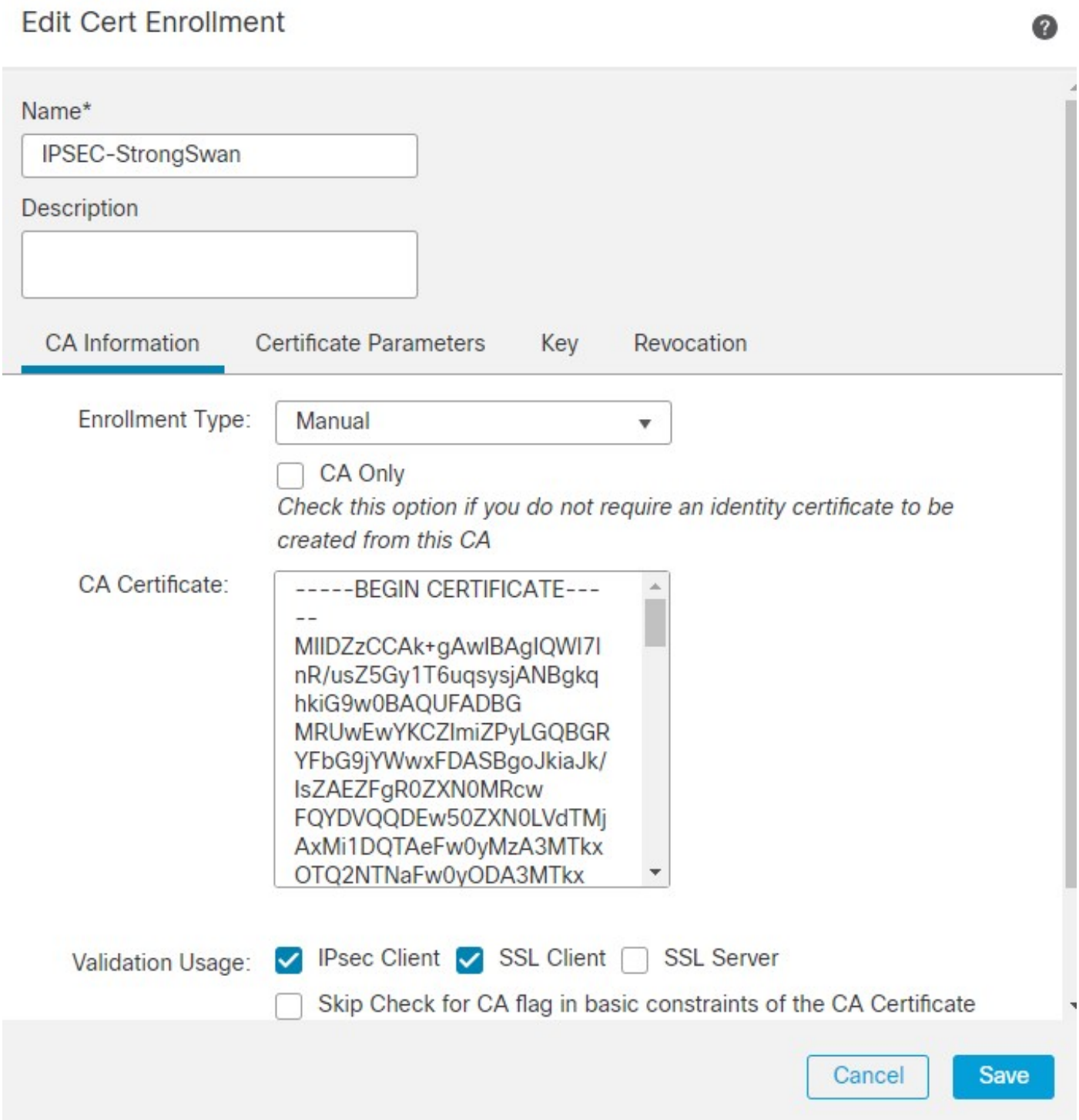
1. From the FMC, navigate to Objects > Object Management > PKI > Cert Enrollment.

2. Click Add Cert Enrollment.

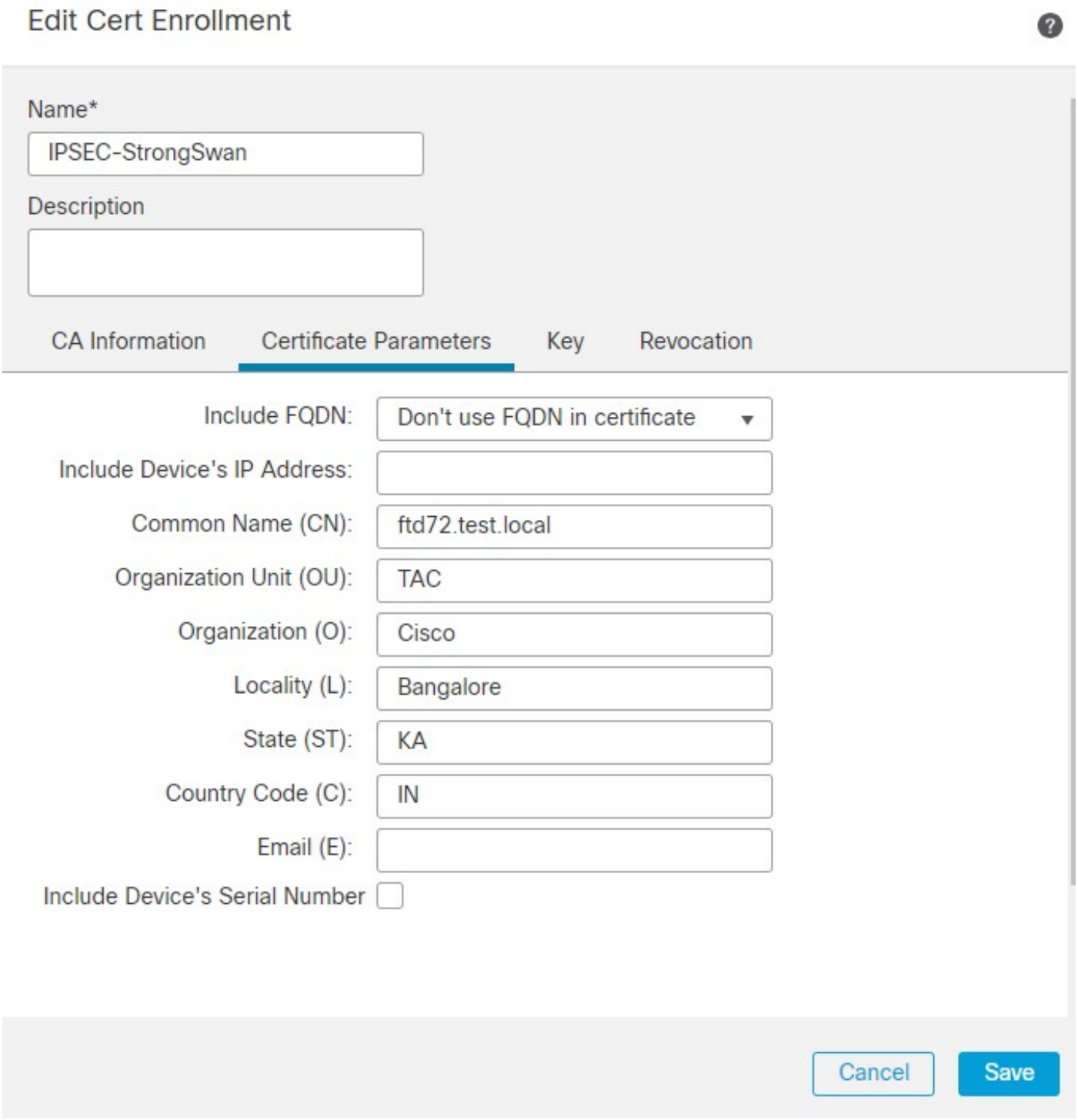
3. The Name section is a mandatory field; give a name for the Trustpoint.
4. The Manual Cert Enrollment is used. On the CA information tab, paste the Issuer Certificate.



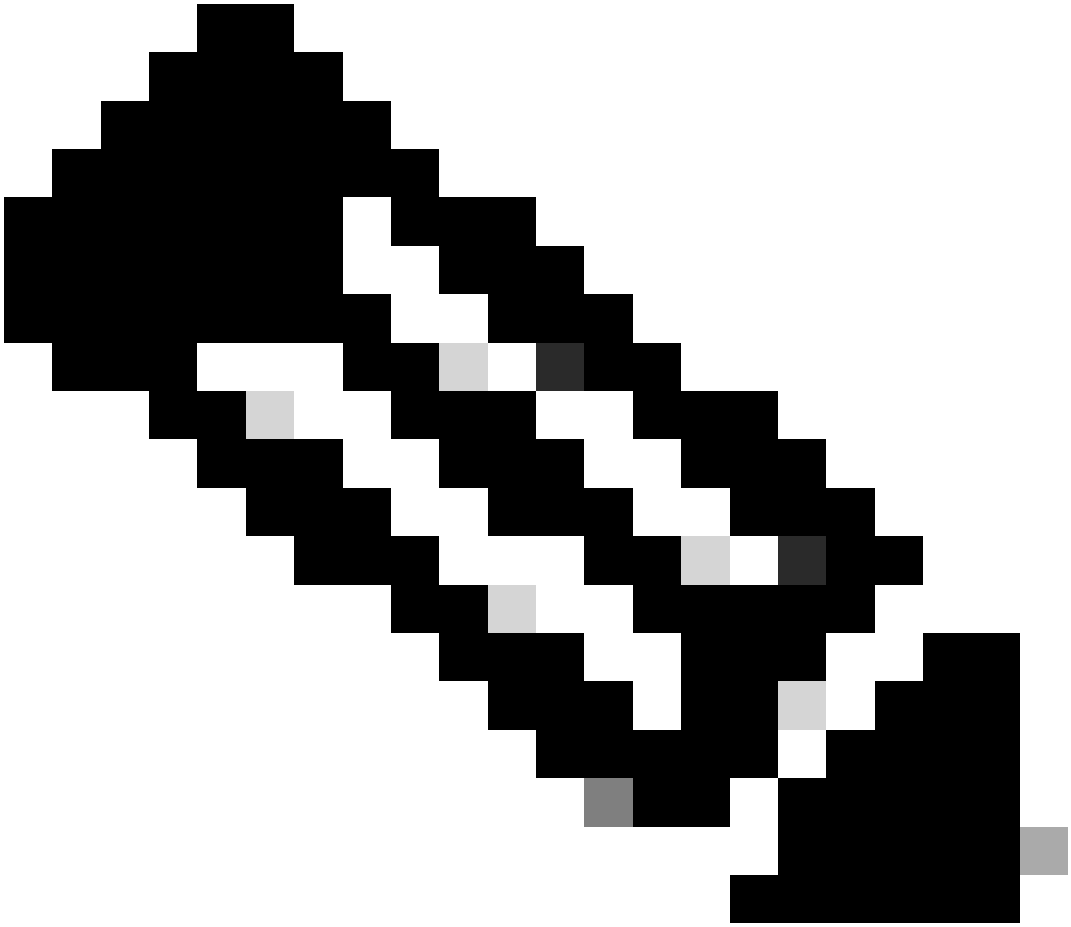
Note: If you do not have an Issuer certificate, you can continue generating CSR without it, and after you get your CSR signed from the CA, edit the trustpoint as mentioned in Step 1. and paste the CA information as described in Step 4.



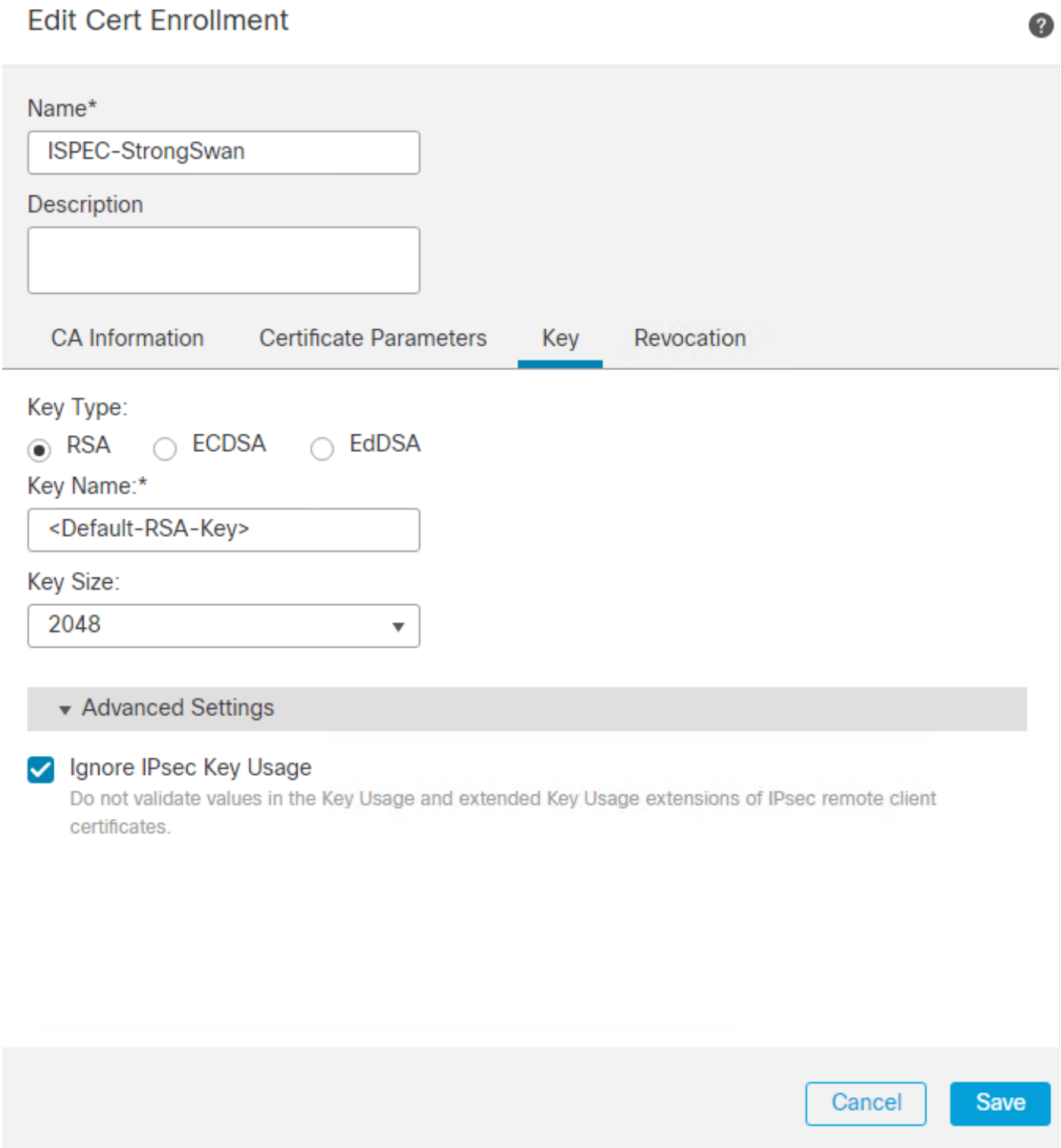
5. In the Certificate Parameters field, enter the parameters as per requirement.



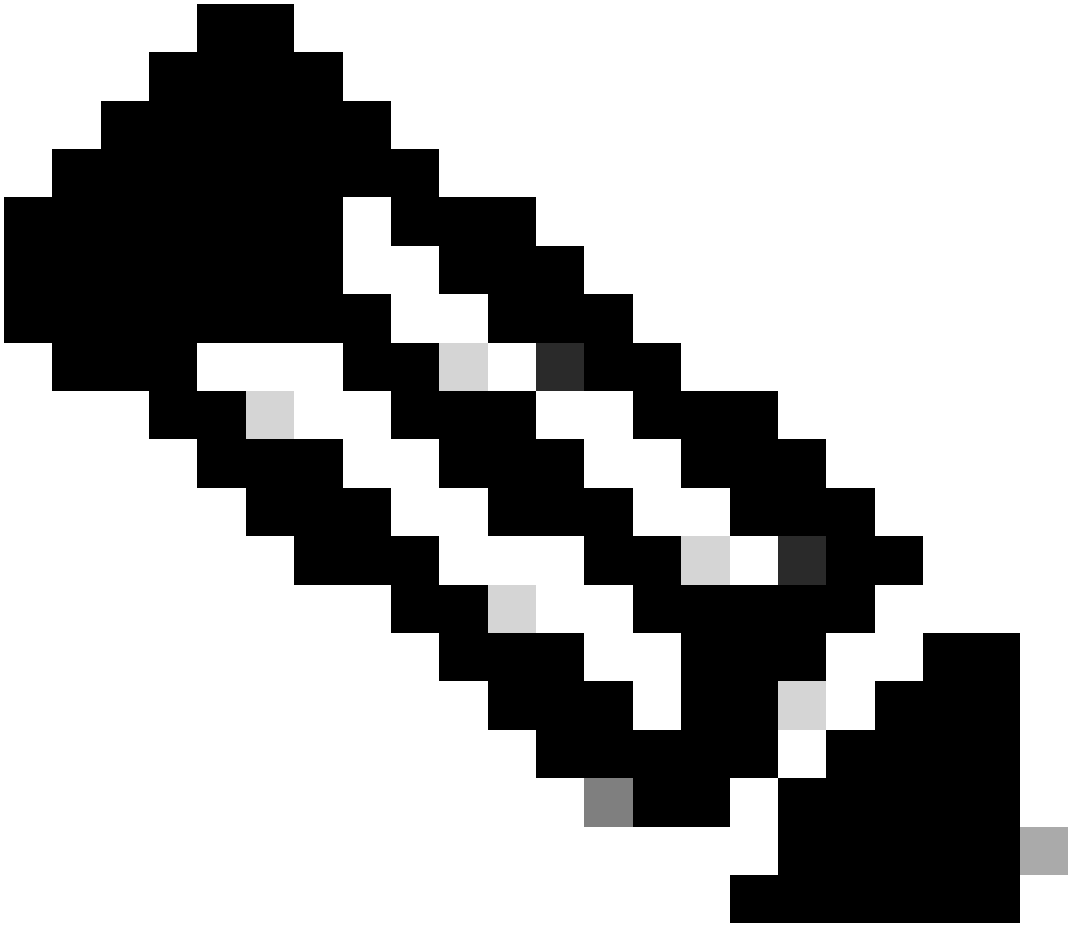
6. In the Key Field, you can use the default RSA keypair or generate a new one by editing the **Key Name** field.



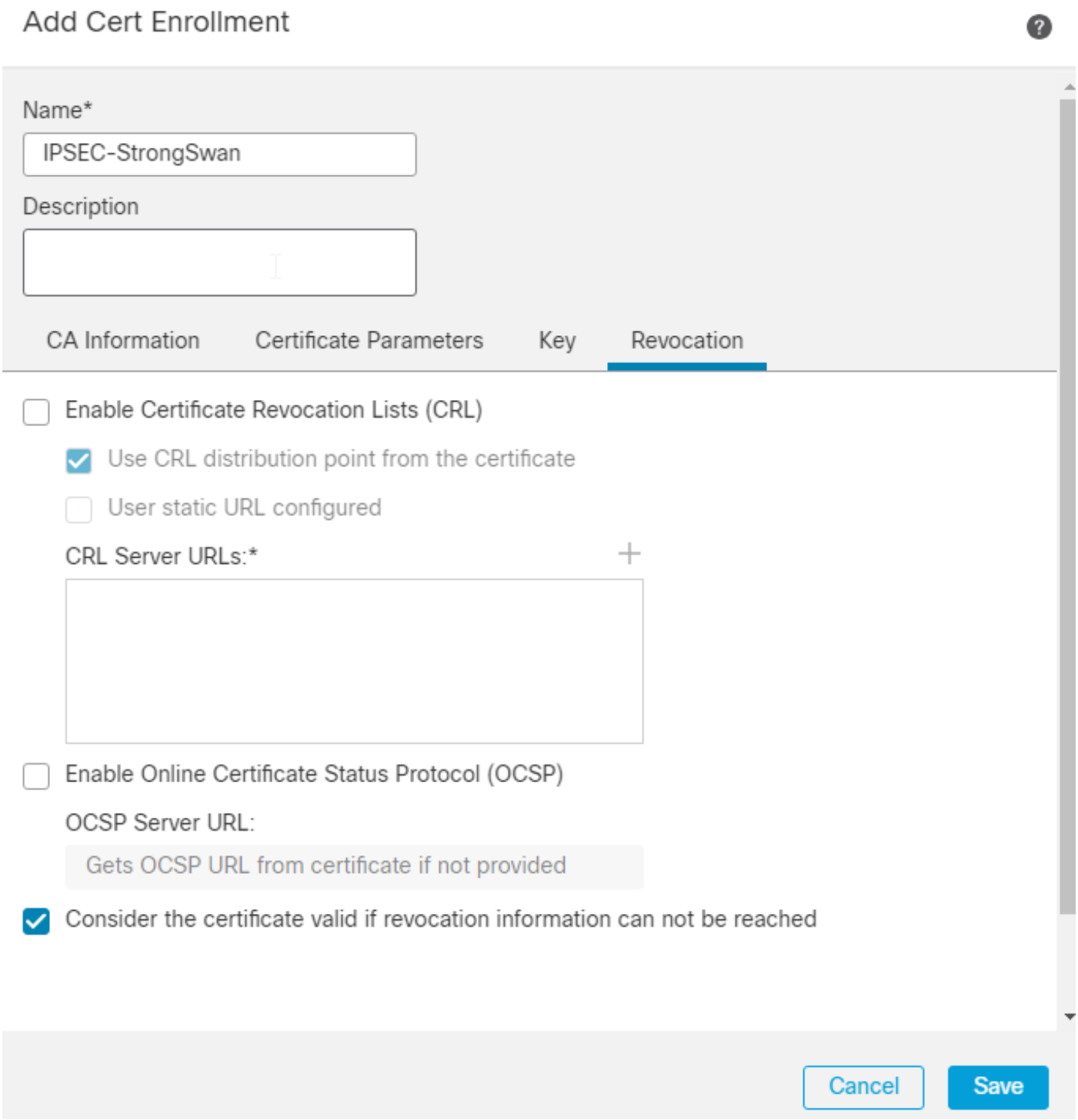
Note: If you use a Windows Certificate Authority (CA), the default Application Policies extension is IP security IKE intermediate. If you are using this default setting, you must choose the Ignore IPsec Key Usage option in the Advanced Settings section on the Key tab in the PKI Certificate Enrollment dialog box for the object you choose. Otherwise, the endpoints cannot complete the site-to-site VPN connection.



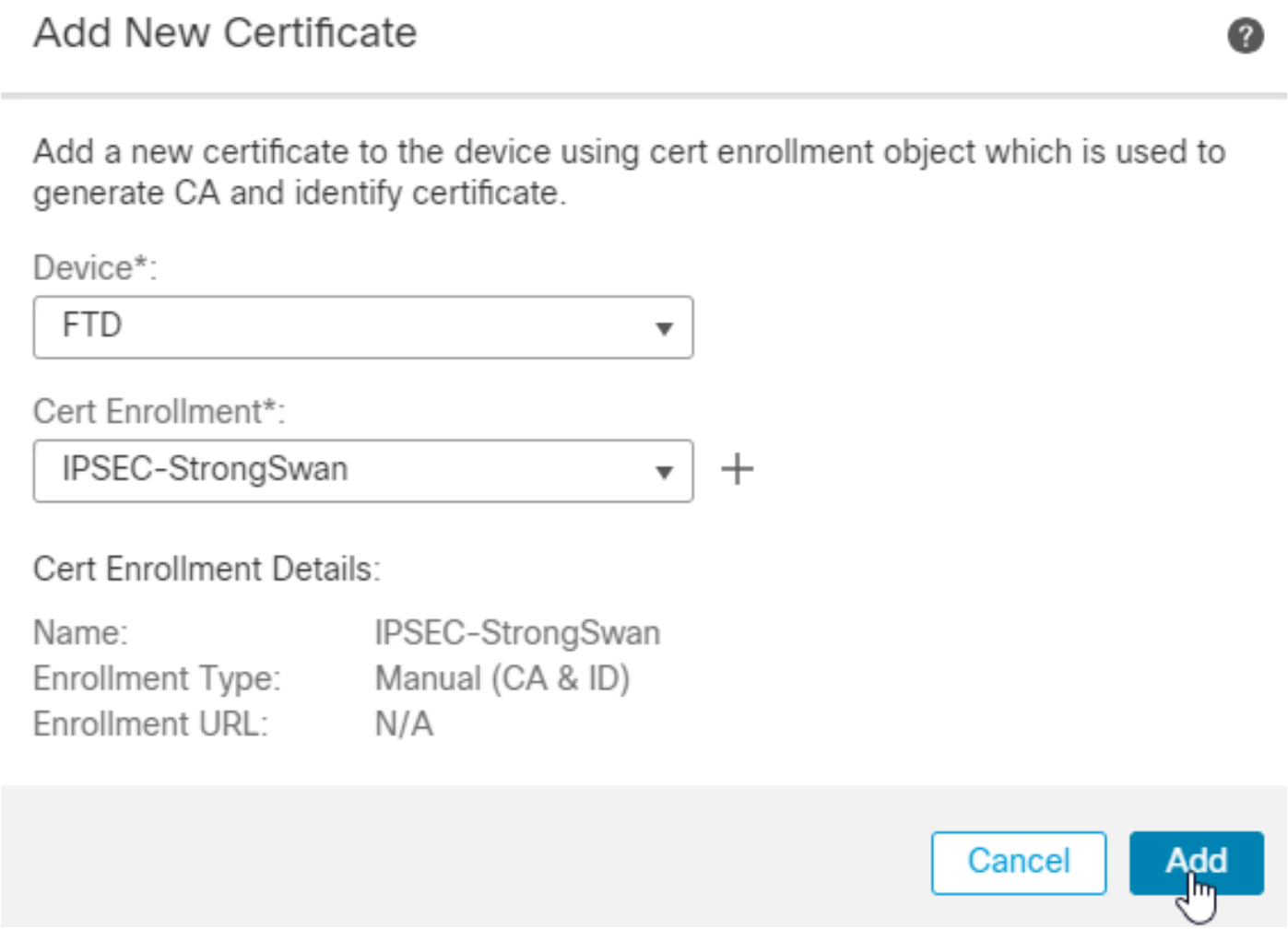
7. In the Revocation Field, choose the checkbox next to Consider the Certificate valid if revocation information can not be reached. No CRL or OCSP checks are used. Click **Save**.



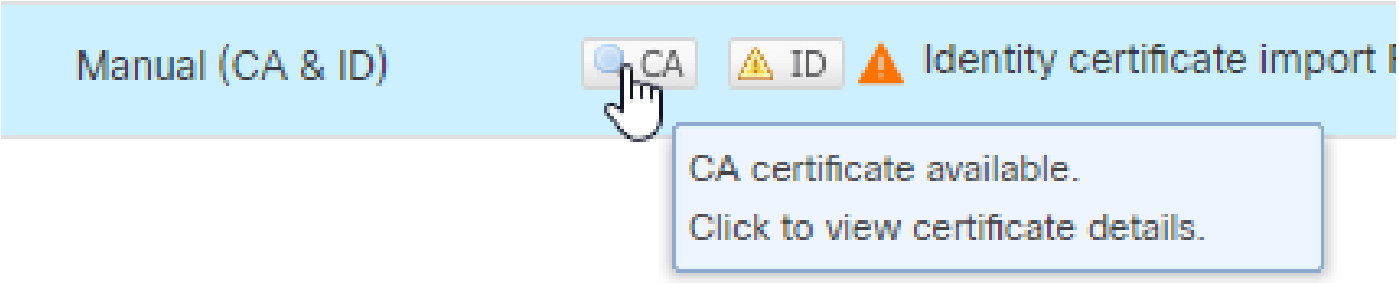
Note: If the device is able to reach the CRL or OCSP servers from the FTD, then you can enable the extensive revocation check in order to get the status of the certificate. The **Consider the Certificate valid if revocation information can not be reached checkbox** is enabled only when there is no connectivity between the CRL server and the FTD device. This is checked by default on the FMC.



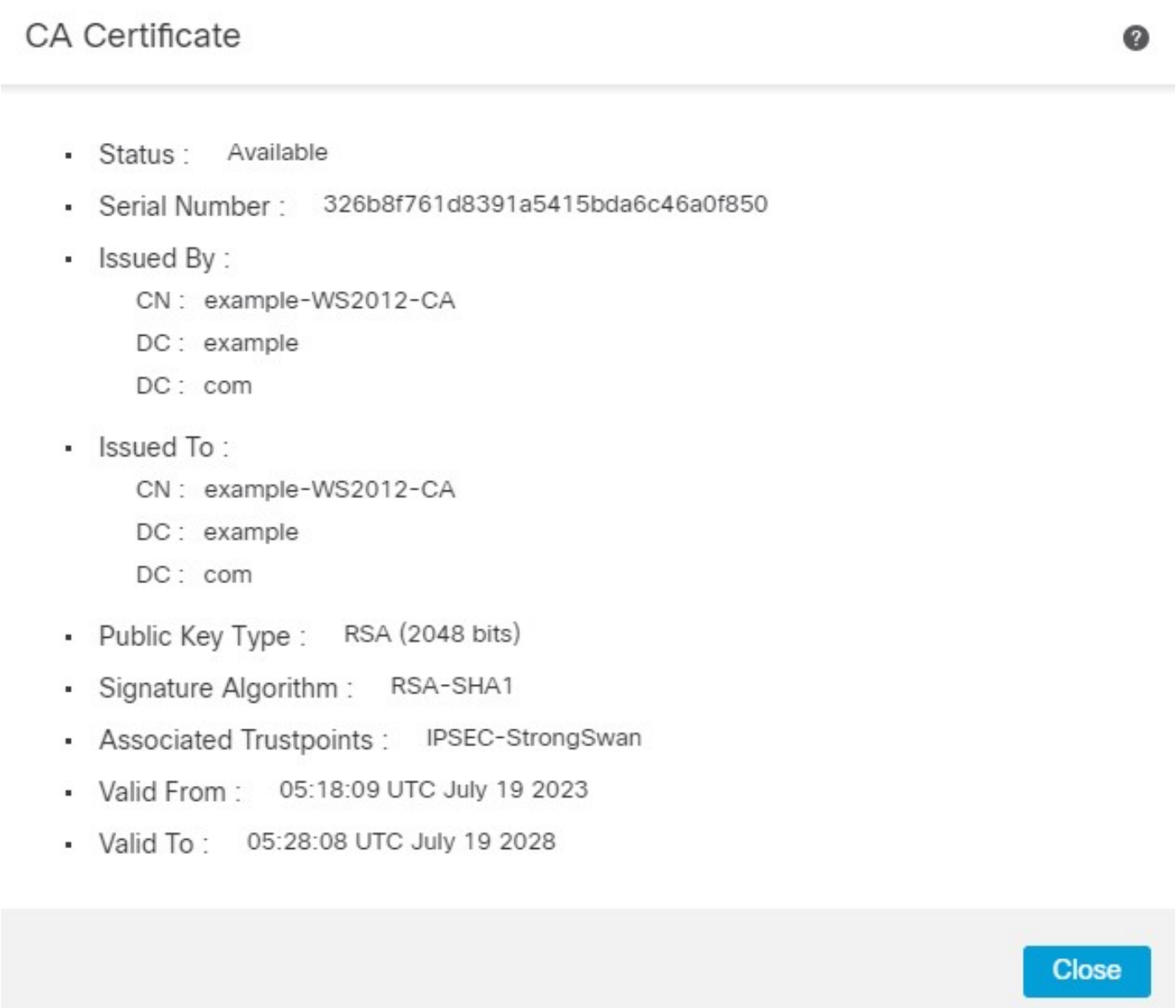
8. Next, navigate to Devices > Certificates, click Add, and choose the FTD device and the Trustpoint you created. Then, click Add.



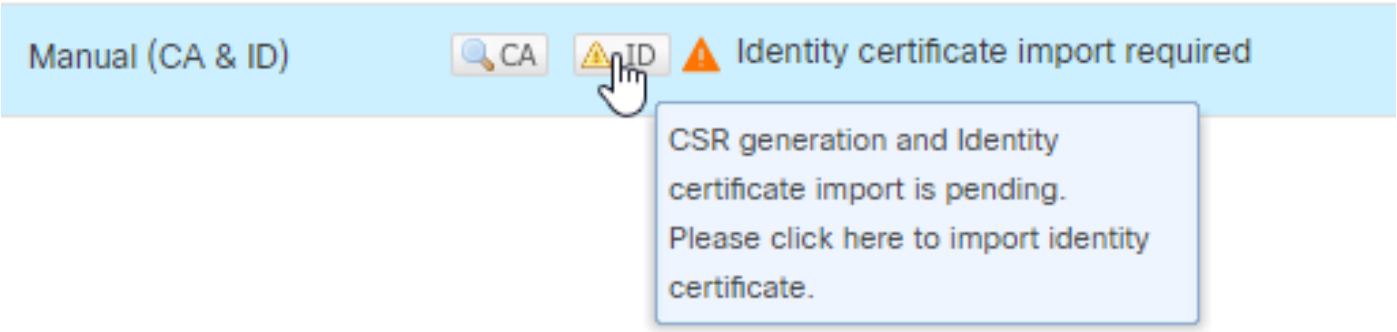
9. You can check the Issuer Certificate by clicking on the magnifying glass icon marked as CA.

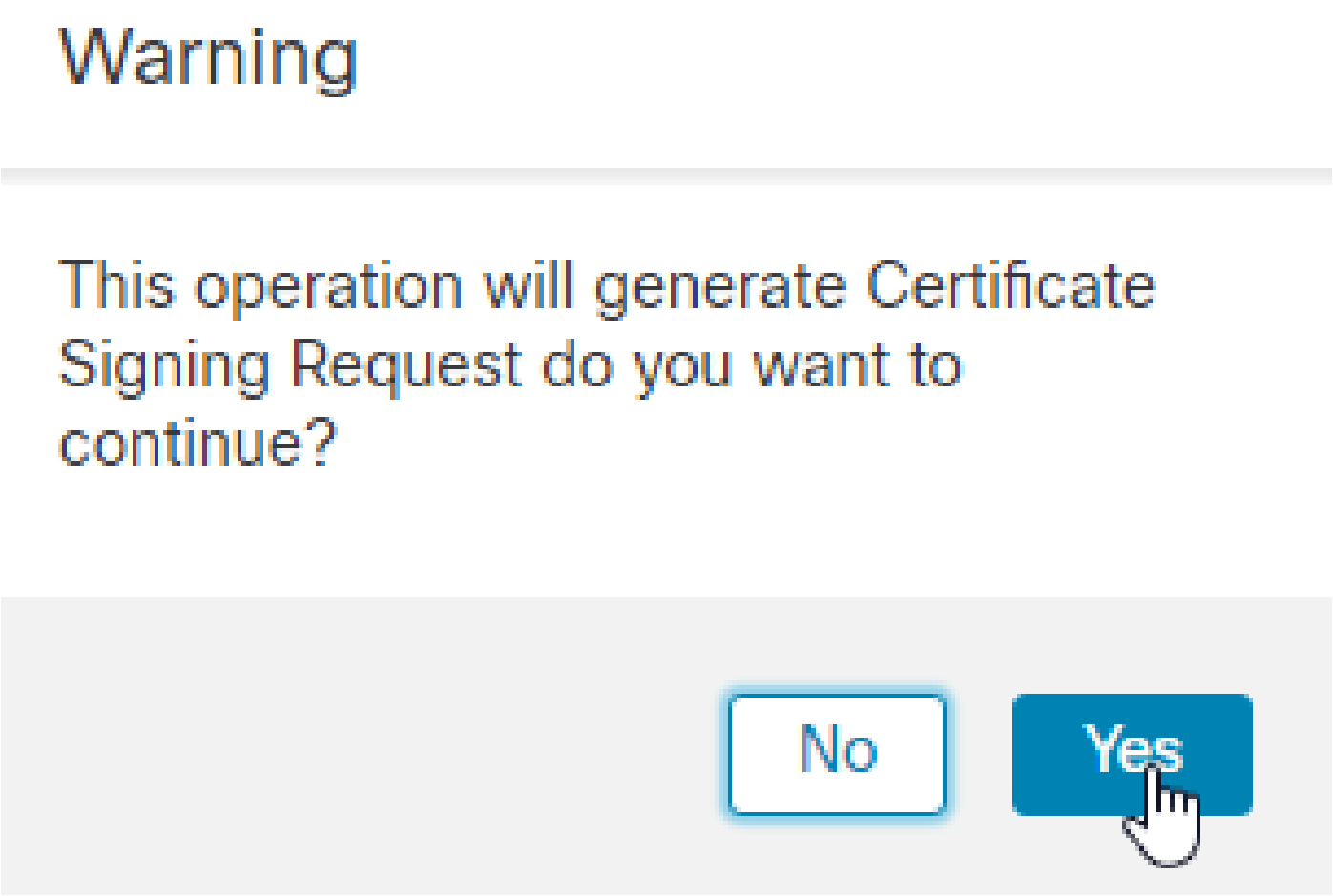


10. You get a similar output.



11. In the next step, you must click the ID field and you get a popup to generate a CSR. Click Yes.





12. Once you have the Identity Certificate File back from the CA, you can import the same using the Browse Identity Certificate and clicking Import.

Import Identity Certificate

Step 1

Send Certificate Signing Request (CSR) to the Certificate Authority.

Certificate Signing Request (Copy the CSR below and send to the Certificate Authority):


```
-----BEGIN CERTIFICATE REQUEST-----
MIIC2zCCAcMCAQAwZzEMMAoGA1UECwwDVEFDMQ4wDAYDVQQKDAVDaXNjbzEZMBcG
A1UEAwwQZnRkNzludGVzdC5sb2NhbDESMBAGA1UEBwwJQmFuZ2Fsb3JlMQswCQYD
VQQIDAJLQTELMAkGA1UEBhMCSU4wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQC3XgIC7ad6h2Uza2BaOBYummYiZvYwjIlzNA/YAckM0Mu8HW6+frDbIJXZ
J+s+WKhLVRcZ9Ad2OAtw0KqTwD3iXRAionMzBpMWNbS6/VpIp4mxL+iOKhTtQBZf
```

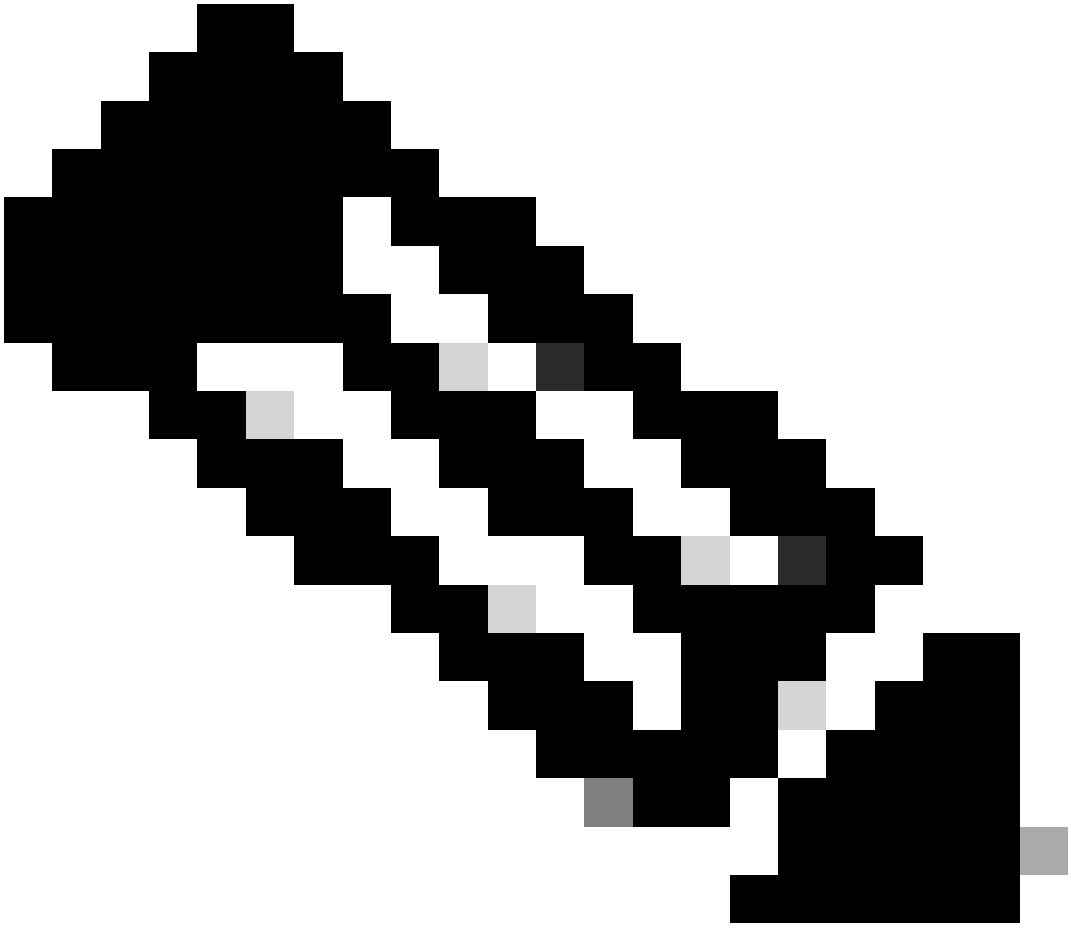

Step 2

Once certificate authority responds back with identity certificate file, import it to device.

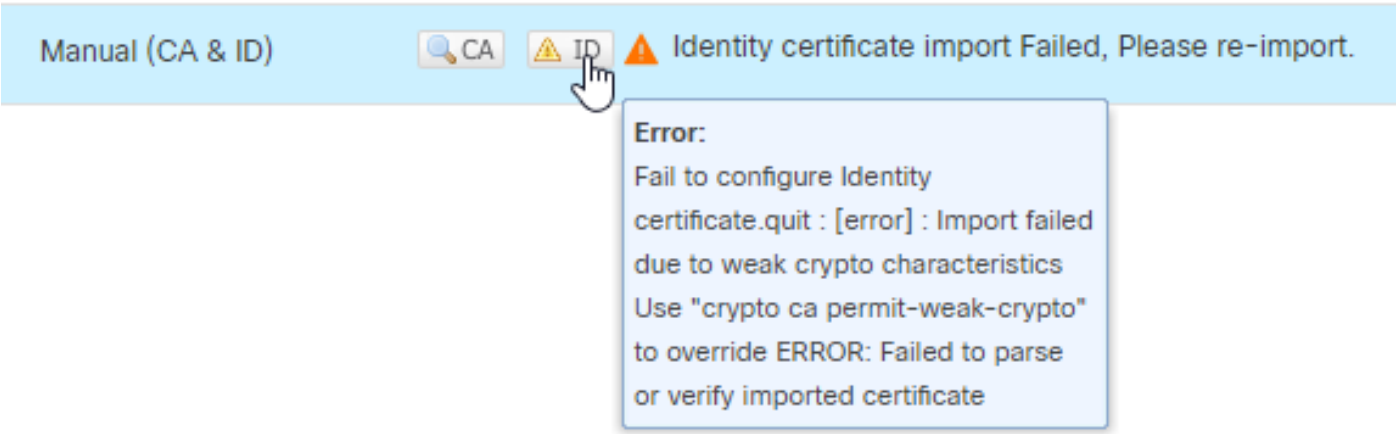
Identity Certificate File: ftd72.test.local.cer

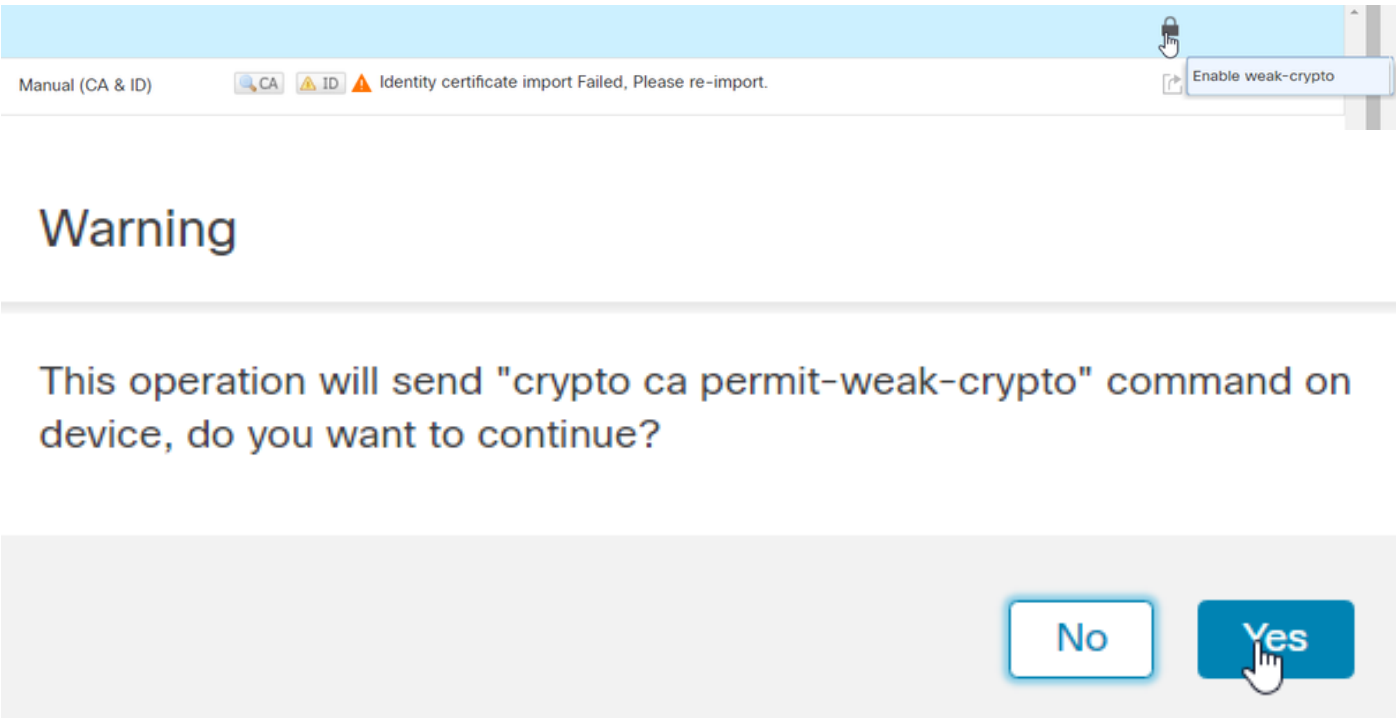
Browse Identity Certificate

Cancel Import



Note: If you receive an error regarding Import failed due to weak crypto characteristics, use the Enable weak-crypto option as shown and once you receive the popup, click Yes in order to continue.

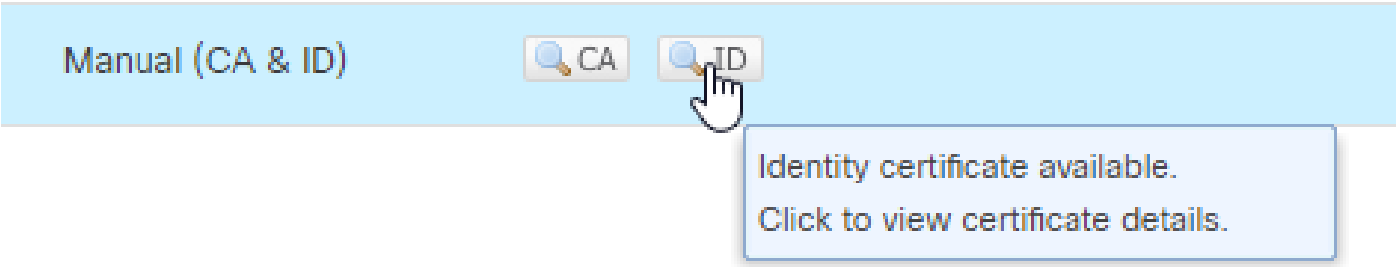


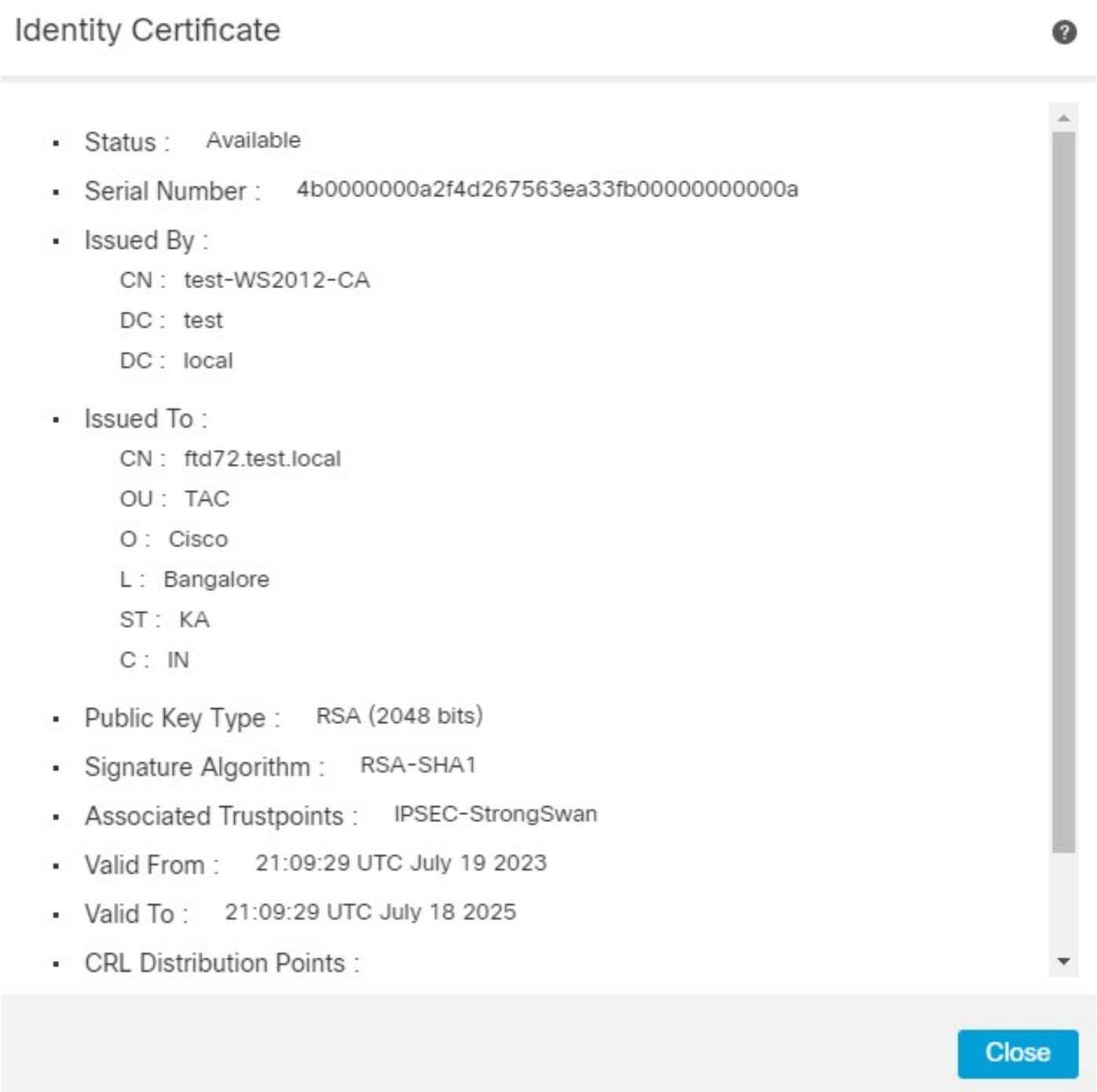


13. Repeat the steps in order to generate the CSR, and import the Identity cert.

There is no need to submit the CSR again since nothing about the device was changed. You can directly import the issued certificate by navigating to the CA.

14. You can now view the Identity Certificate by clicking on the magnifying glass icon marked as ID.

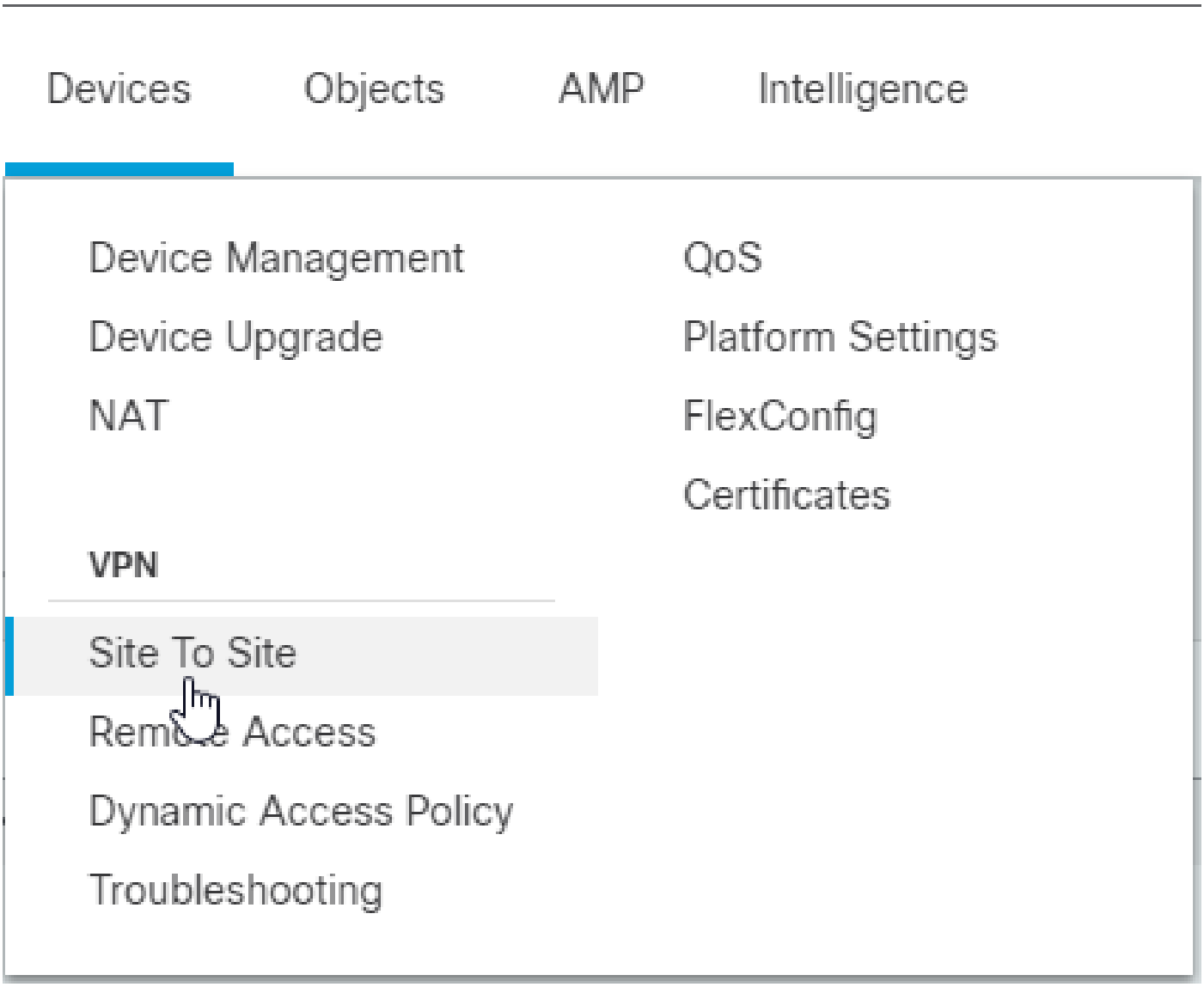




15. The Certificate is added successfully.

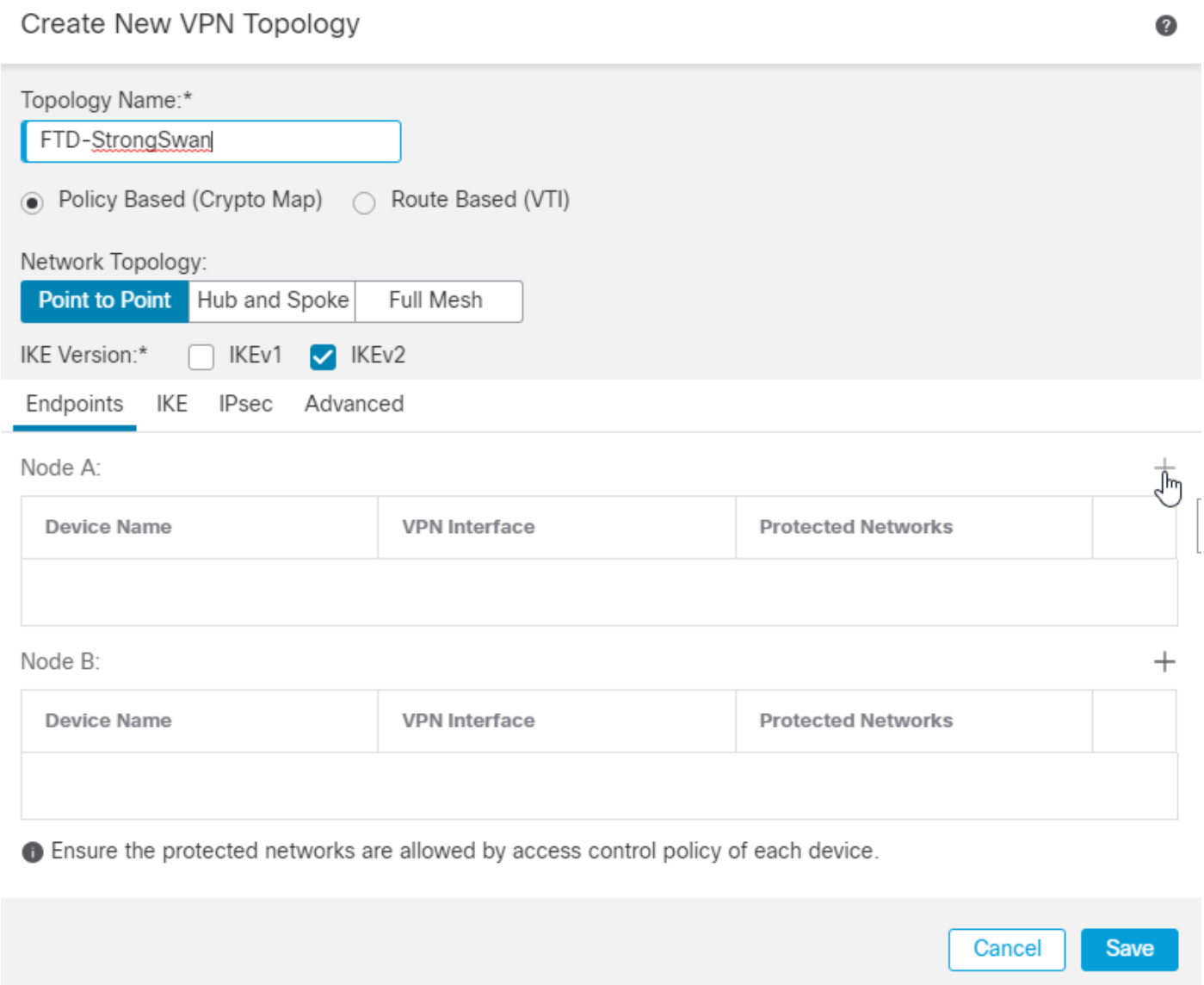
VPN Configuration

1. Navigate to Devices > Site to Site VPN .



2. Click Add > VPN Tunnel.

3. Enter the Topology Name which is a Mandatory Field. Policy based (Crypto Map), Point-to-Point Topology, and IKEv2 are selected by default and you must use these.
4. In the Endpoints Section, click the + icon next to Node A.



5. Choose the FTD device as Node A, and the VPN terminating interface is the outside interface.
6. In the Protected Networks Field, choose the Subnet/IP Address (Network), and click the + icon.

Add Endpoint

Device:*

FTD

Interface:*

outside

IP Address:*

10.106.69.230

This IP is Private

Connection Type:

Bidirectional

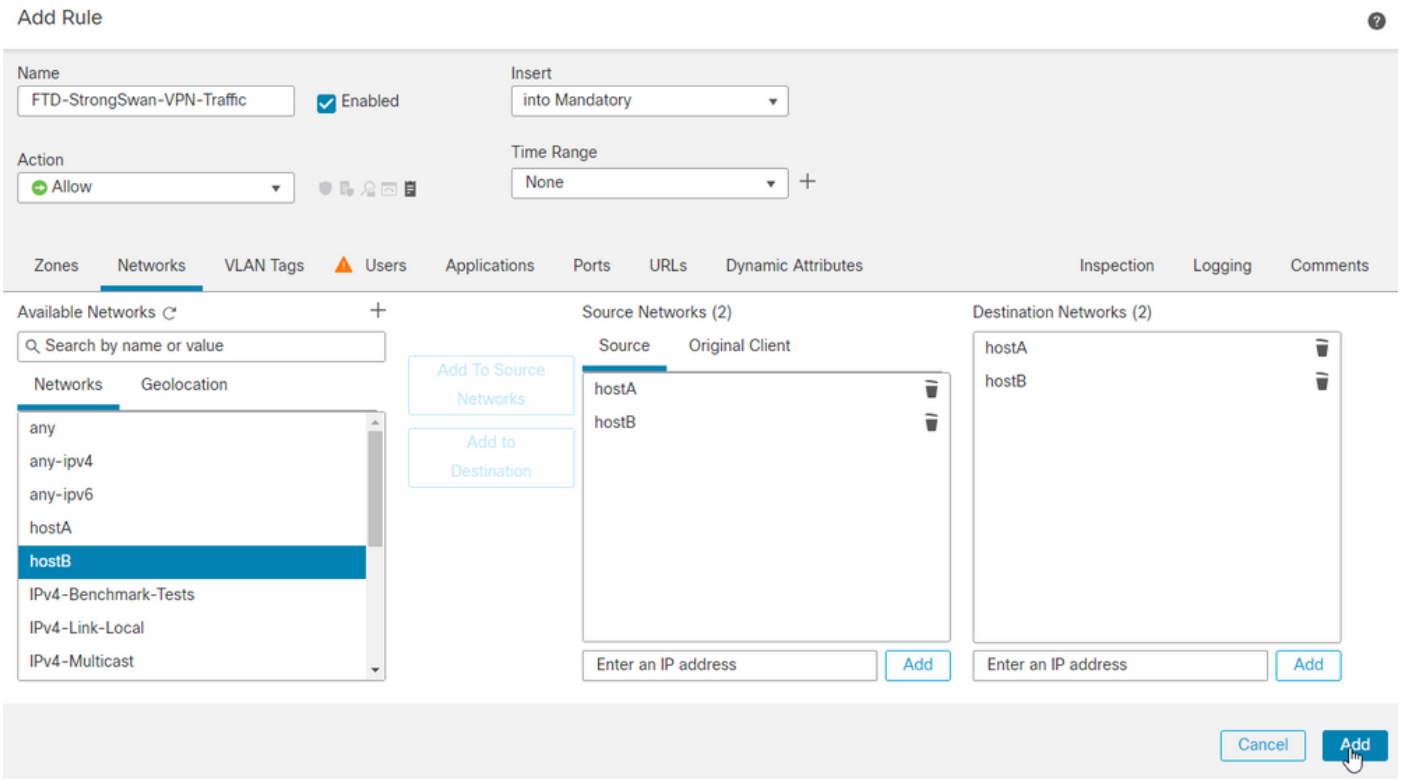
Certificate Map:

Protected Networks:*

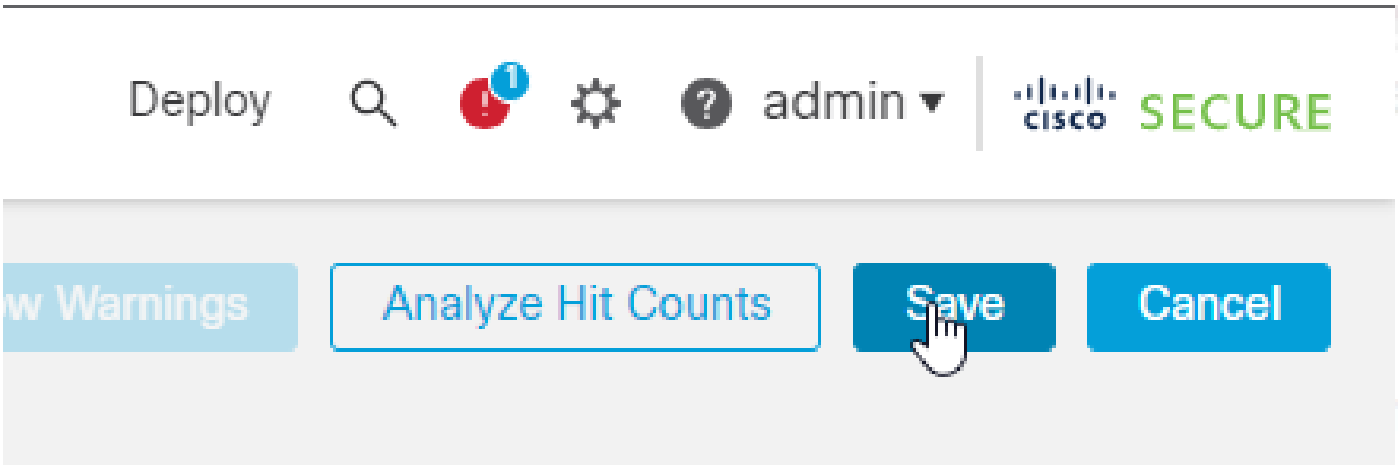
Subnet / IP Address (Network) Access List (Extended)

Advance Settings

. If you have configured zones, choose the relevant ones and add them for source and destination. Then click Add .



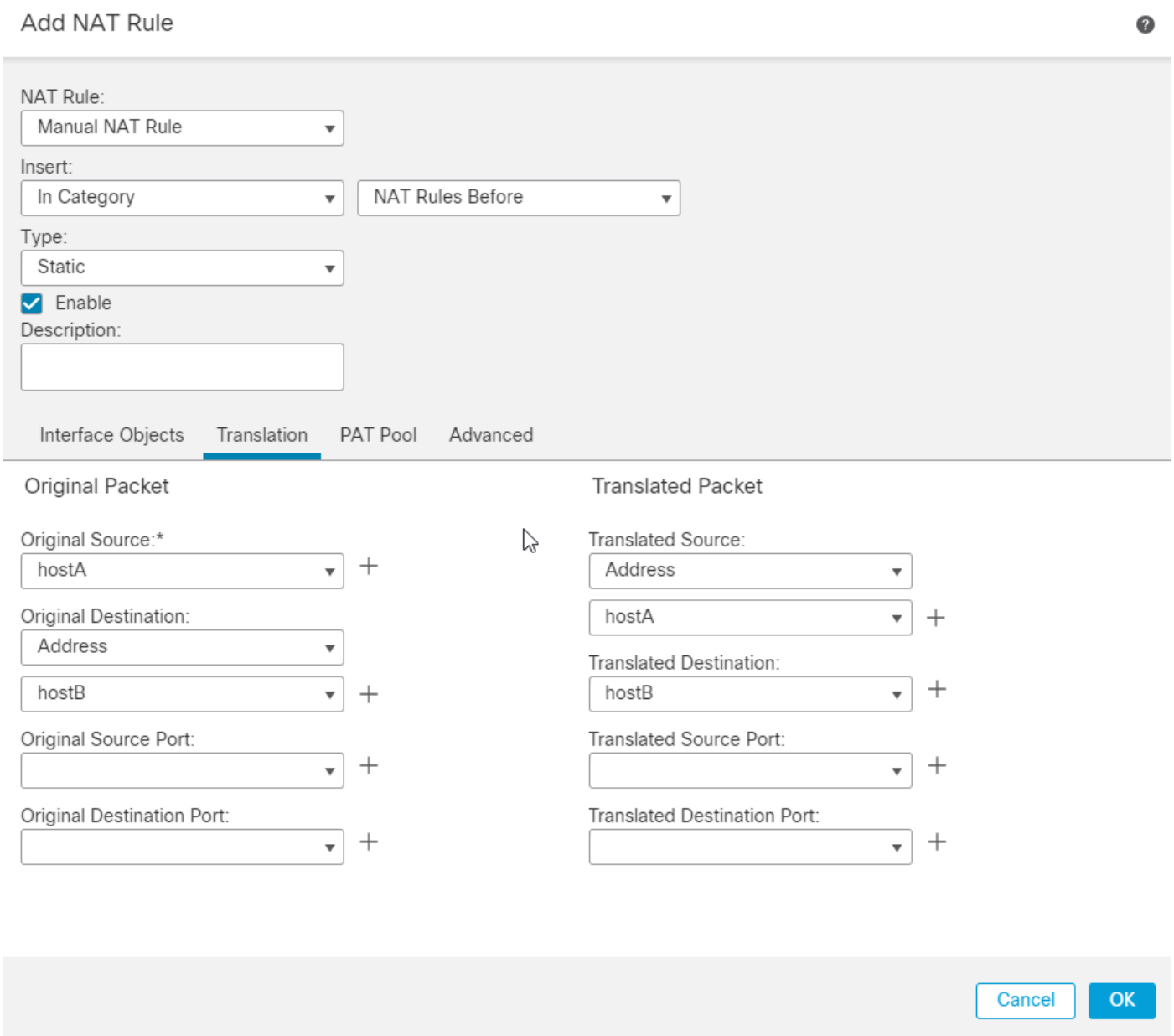
25. After adding the rule, click Save .



26. Finally, you must configure a **No NAT** statement for the VPN traffic to be exempted in case there is any NAT present on the FTD. Navigate to Devices > NAT. Click Add Rule.

27. Add the relevant interface objects and under the translation section choose the Original and Translated source as the VPN-protected network behind the FTD, which is **hostA** in this case. Similarly, for the Original and Translated destinations, choose the VPN-protected network behind the Remote end, which is **hostB** in this case.

28. Under the Advanced section ensure to check the Do not proxy ARP on Destination Interface and Perform Route Lookup for Destination Interface checkboxes, which are required for **No NAT** statements.



StrongSwan Configuration

Getting Certificate

All the commands shown need **sudo** permissions. Contact your system administrator, if you do not have **sudo** access or permissions to install the software. Official StrongSwan example configurations can be found here.

1. Start by updating the system package cache.

```
apt update
```

2. Install StrongSwan and its dependencies. You can find more information about the packages here.

```
apt install strongswan strongswan-pki strongswan-swanctl libcharon-extra-plugins libcharon-extauth-plug
```

3. Check the status of the StrongSwan daemon. The status must show active(running).

```
systemctl status strongswan-starter.service
```

4. If for some reason it is not, enable it and start it with this command.

```
systemctl enable --now strongswan-starter.service
```

5. Next, use the strong-swan pki command line tool in order to generate the Private key and CSR. In order to generate a private key for the server execute this command.

```
pki --gen > sswan.priv.key
```

6. Generate CSR requests for the StrongSwan server. You can modify the --dn argument as per your requirements.

```
pki --req --in sswan.priv.key --dn "CN=sswan.test.local, O=Cisco, OU=TAC, ST=KA, C=IN" --outform pem >
```

7. After getting the CSR signed by the CA, copy the issuer certificate, the identity certificate, and the Private Key to the respective /etc/swanctl directory.

```
cp $HOME/certs/root-ca.cer /etc/swanctl/x509ca/
cp $HOME/certs/sswan.test.local.cer /etc/swanctl/x509/
cp $HOME/certs/sswan.priv.key /etc/swanctl/private/
```

Swanctl Configuration File

<#root>

```
connections {
  strongswan-ftd {
    # Peer IP's
    local_addrs = 10.106.67.200
    remote_addrs = 10.106.69.230
```

```
    local {
      auth = pubkey
      certs = sswan.test.local.cer
      id = "sswan.test.local"
    }

    remote {
      auth = pubkey
      id = "C=IN, ST=KA, L=Bangalore, O=Cisco, OU=TAC, CN=ftd72.test.local"
    }

    children {
      hostB-hostA {


local_ts = 10.106.71.110/32



        remote_ts = 10.106.70.110/32




rekey_time = 28800                  # Phase-2 Lifetime



esp_proposals = aes-sha-modp2048    # Phase-2 Paramters


      }
    }

    mobike = no
    version = 2                             # IKE version 2


reauth_time = 86400                     # Phase-1 Lifetime



proposals = aes-sha-prfsha1-modp2048    # Phase-1 Paramters


  }
}
```

Verify

FTD

1. Check the IKEv2 Phase-1 Paramters.

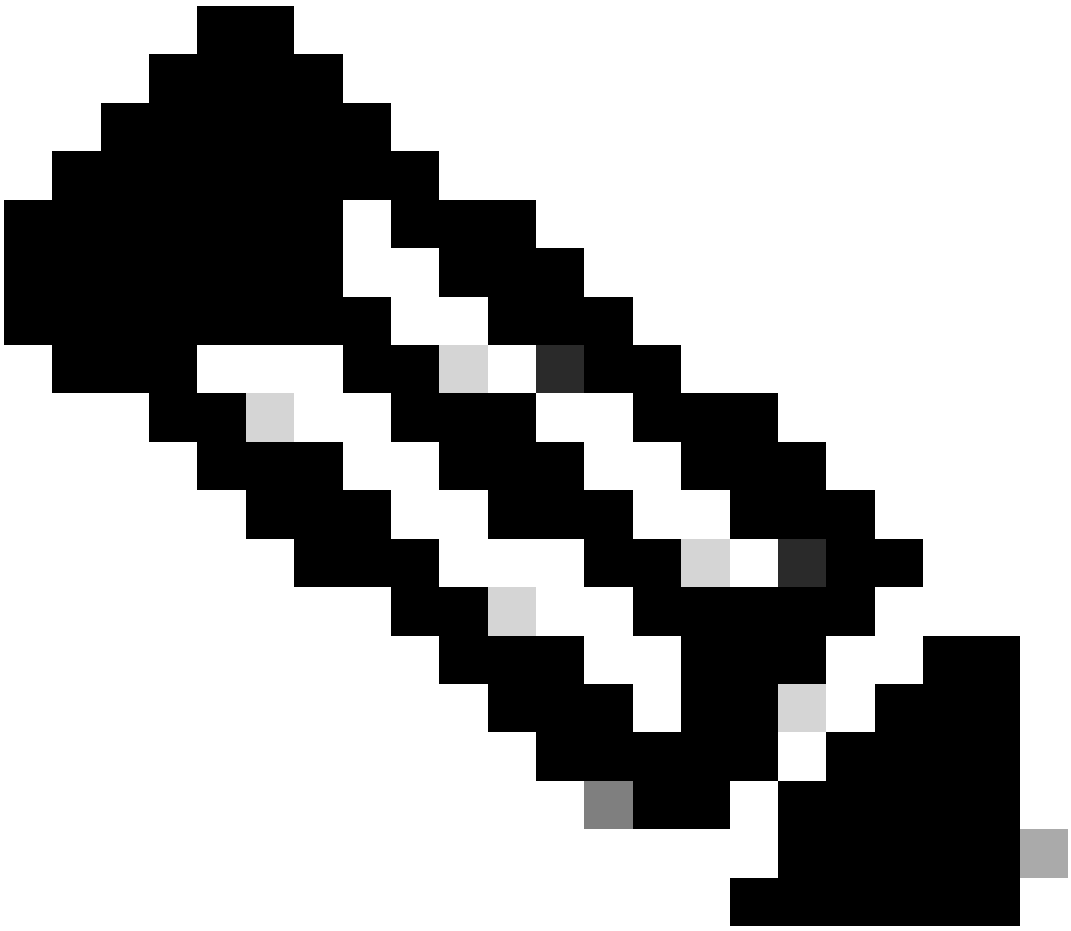
```
firepower# sh run crypto ikev2
crypto ikev2 policy 1
 encryption aes
```

```
 integrity sha
 group 14
 prf sha
 lifetime seconds 86400
crypto ikev2 enable outside
```

2. Check Phase-2 Parameters.

```
firepower# sh run crypto ipsec
crypto ipsec ikev2 ipsec-proposal CSM_IP_2
 protocol esp encryption aes
 protocol esp integrity sha-1
```

3. Check the crypto map configuration.



Note: On FTD 7.2.0 version the default PFS is DH Group 14 (MOD 2048). You can verify the same by running sh run all crypto map.

```
firepower# sh run crypto map
crypto map CSM_outside_map 1 match address CSM_IPSEC_ACL_1
crypto map CSM_outside_map 1 set pfs
crypto map CSM_outside_map 1 set peer 10.106.67.200
crypto map CSM_outside_map 1 set ikev2 ipsec-proposal CSM_IP_2
crypto map CSM_outside_map 1 set trustpoint IPSEC-StrongSwan
crypto map CSM_outside_map 1 set reverse-route
crypto map CSM_outside_map interface outside
```

4. Check the Crypto ACL.

```
firepower# sh access-list CSM_IPSEC_ACL_1
access-list CSM_IPSEC_ACL_1; 1 elements; name hash: 0x1fb1fb7
access-list CSM_IPSEC_ACL_1 line 1 extended permit ip host 10.106.70.110 host 10.106.71.110 (hitcnt=37)
```

5. Check the tunnel status.

<#root>

```
firepower# sh vpn-sessiondb det l2l

Session Type: LAN-to-LAN Detailed

Connection  : 10.106.67.200
Index       : 61 IP Addr : 10.106.67.200
Protocol    : IKEv2 IPsec
Encryption  : IKEv2:

(1)AES128 IPsec: (1)AES128


Hashing     : IKEv2:

(1)SHA1 IPsec: (1)SHA1


Bytes Tx    : 0
Bytes Rx    : 0
Login Time  : 12:16:25 UTC Mon Jul 17 2023
Duration    : 0h:11m:30s
Tunnel Zone : 0

IKEv2 Tunnels: 1
IPsec Tunnels: 1

IKEv2:
  Tunnel ID : 61.1
  UDP Src Port : 500 UDP Dst Port : 500

  Rem Auth Mode: rsaCertificate



  Loc Auth Mode: rsaCertificate
```

```
Encryption : AES128 Hashing : SHA1


  Rekey Int (T): 86400 Seconds Rekey Left(T): 85710 Seconds


  PRF : SHA1 D/H Group : 14


  Filter Name :

IPsec:
  Tunnel ID : 61.2
  Local Addr : 10.106.70.110/255.255.255.255/0/0
  Remote Addr : 10.106.71.110/255.255.255.255/0/0
  Encryption : AES128 Hashing : SHA1

  Encapsulation: Tunnel PFS Group : 14



Rekey Int (T): 28800 Seconds Rekey Left(T): 28110 Seconds


  Rekey Int (D): 4608000 K-Bytes Rekey Left(D): 4608000 K-Bytes
  Idle Time Out: 30 Minutes Idle TO Left : 29 Minutes
  Conn Time Out: 1032728 Minutes Conn TO Left : 1032714 Minutes
  Bytes Tx : 600 Bytes Rx : 880


Pkts Tx : 10 Pkts Rx : 10
```

6. Check IPSEC SA counters.

<#root>

```
firepower# sh cry ipsec sa
interface: outside
    Crypto map tag: CSM_outside_map, seq num: 1, local addr: 10.106.69.230

    access-list CSM_IPSEC_ACL_1 extended permit ip host 10.106.70.110 host 10.106.71.110
    Protected vrf:
    local ident (addr/mask/prot/port): (10.106.70.110/255.255.255.255/0/0)
    remote ident (addr/mask/prot/port): (10.106.71.110/255.255.255.255/0/0)
    current_peer: 10.106.67.200



    #pkts encaps: 10

,

 #pkts encrypt: 10

,

 #pkts digest: 10
```

```
    #pkts decaps: 10

,

 #pkts decrypt: 10

,

 #pkts verify: 10
```

StrongSwan

1. Check the connections loaded. If no connections are seen run the swanctl --load-all.

<#root>

```
root@strongswan:~# swanctl --list-conn
strongswan-ftd: IKEv2, reauthentication every 86400s, no rekeying
  local: 10.106.67.200
  remote: 10.106.69.230
local public key authentication:


id: sswan.test.local


  certs: C=IN, ST=KA, O=Cisco, OU=TAC, CN=sswan.test.local
remote public key authentication:


id: C=IN

,

 ST=KA

,

 L=Bangalore

,

 O=Cisco

,

 OU=TAC

,

 CN=ftd72.test.local


hostB-hostA: TUNNEL, rekeying every 28800s
  local: 10.106.71.110/32
  remote: 10.106.70.110/32
```

2. Check the SA status of the child.

<#root>

```
root@strongswan:~# swanctl --list-sas
strongswan-ftd: #11, ESTABLISHED, IKEv2, 791c5a5633f9ea83_i a4e0487769c49dad_r*
  local 'sswan.test.local' @ 10.106.67.200[500]
  remote 'C=IN, ST=KA, L=Bangalore, O=Cisco, OU=TAC, CN=ftd72.test.local' @ 10.106.69.230[500]
```

AES_CBC-128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048

```
  established 279s ago, reauth in 83226s
  hostB-hostA: #8, reqid 6,
```

INSTALLED

,

TUNNEL

,

ESP:AES_CBC-128/HMAC_SHA1_96

```
    installed 279s ago, rekeying in 25753s, expires in 31401s
    in cc01a2a7, 600 bytes, 10 packets, 10s ago
    out 3594c049, 600 bytes, 10 packets, 10s ago
    local 10.106.71.110/32
    remote 10.106.70.110/32
```

Troubleshoot

FTD

```
debug crypto condition peer 10.106.67.200
debug crypto ikev2 platfform 127
debug crypto ikev2 protocol 127
debug crypto ipsec 127
```

Peer ID validation is turned on.

<#root>

```
==== OUTPUT OMITTED ====
IKEv2-PLAT-4: (203): Peer ID check started, received ID type: IPv4 address
IKEv2-PLAT-4: (203): Peer ID check: failed to retreive IP from SAN
IKEv2-PLAT-4: (203): Peer ID check: failed to retreive DNS name from SAN
IKEv2-PLAT-4: (203): Peer ID check: failed to retreive RFC822 name from SAN
IKEv2-PLAT-4: (203): retrieving SAN for peer ID check
IKEv2-PLAT-2: (203): Peer ID check failed
IKEv2-PROTO-2: (203): Failed to locate an item in the database
IKEv2-PROTO-7: (203): SM Trace-> SA: I_SPI=40DC7DC3A0BDF20D R_SPI=E02399BAC06E0944 (I) MsgID = 00000001
```

IKEv2-PROTO-4: (203): Verification of peer's authentication data FAILED

```
IKEv2-PROTO-7: (203): SM Trace-> SA: I_SPI=40DC7DC3A0BDF20D R_SPI=E02399BAC06E0944 (I) MsgID = 00000001

IKEv2-PROTO-4: (203): Auth exchange failed

IKEv2-PROTO-2: (203): Auth exchange failed

IKEv2-PROTO-2: (203): Auth exchange failed

IKEv2-PROTO-7: (203): SM Trace-> SA: I_SPI=40DC7DC3A0BDF20D R_SPI=E02399BAC06E0944 (I) MsgID = 00000001
IKEv2-PROTO-7: (203): SM Trace-> SA: I_SPI=40DC7DC3A0BDF20D R_SPI=E02399BAC06E0944 (I) MsgID = 00000001
IKEv2-PLAT-7: Negotiating SA request deleted
IKEv2-PLAT-7: Decrement count for outgoing negotiating
IKEv2-PROTO-7: (203): SM Trace-> SA: I_SPI=40DC7DC3A0BDF20D R_SPI=E02399BAC06E0944 (I) MsgID = 00000001
IKEv2-PROTO-4: (203): Abort exchange
IKEv2-PROTO-4: (203): Deleting SA
==== OUTPUT OMITTED ====
```

StrongSwan

<#root>

```
root@strongswan:~# swanctl --log
01[NET] received packet: from 10.106.69.230[500] to 10.106.67.200[500] (574 bytes)
01[ENC] parsed IKE_SA_INIT request 0 [ SA KE No V V N(NATD_S_IP) N(NATD_D_IP) N(FRAG_SUP) V ]
01[IKE] received Cisco Delete Reason vendor ID
01[IKE] received Cisco Copyright (c) 2009 vendor ID
01[IKE] received FRAGMENTATION vendor ID
01[IKE] 10.106.69.230 is initiating an IKE_SA

01[CFG] selected proposal: IKE:AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048

01[IKE] sending cert request for "DC=local, DC=test, CN=test-WS2012-CA"
01[ENC] generating IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ N(FRAG_SUP) N(CH
01[NET] sending packet: from 10.106.67.200[500] to 10.106.69.230[500] (481 bytes)
06[NET] received packet: from 10.106.69.230[500] to 10.106.67.200[500] (528 bytes)
06[ENC] parsed IKE_AUTH request 1 [ EF(1/5) ]
06[ENC] received fragment #1 of 5, waiting for complete IKE message
07[NET] received packet: from 10.106.69.230[500] to 10.106.67.200[500] (528 bytes)
07[ENC] parsed IKE_AUTH request 1 [ EF(2/5) ]
07[ENC] received fragment #2 of 5, waiting for complete IKE message
12[NET] received packet: from 10.106.69.230[500] to 10.106.67.200[500] (528 bytes)
12[ENC] parsed IKE_AUTH request 1 [ EF(3/5) ]
12[ENC] received fragment #3 of 5, waiting for complete IKE message
11[NET] received packet: from 10.106.69.230[500] to 10.106.67.200[500] (528 bytes)
11[ENC] parsed IKE_AUTH request 1 [ EF(4/5) ]
11[ENC] received fragment #4 of 5, waiting for complete IKE message
09[NET] received packet: from 10.106.69.230[500] to 10.106.67.200[500] (208 bytes)
09[ENC] parsed IKE_AUTH request 1 [ EF(5/5) ]
09[ENC] received fragment #5 of 5, reassembled fragmented IKE message (2012 bytes)
09[ENC] parsed IKE_AUTH request 1 [ V IDi CERT CERTREQ AUTH SA TSi TSr N(INIT_CONTACT) N(ESP_TFC_PAD_N)
09[IKE] received cert request for "DC=local, DC=test, CN=test-WS2012-CA"
09[IKE] received end entity cert "C=IN, ST=KA, L=Bangalore, O=Cisco, OU=TAC, CN=ftd72.test.local"
09[CFG] looking for peer configs matching 10.106.67.200[%any]...10.106.69.230[C=IN, ST=KA, L=Bangalore,
```

09[CFG] selected peer config 'strongswan-ftd'

```
09[CFG] using certificate "C=IN, ST=KA, L=Bangalore, O=Cisco, OU=TAC, CN=ftd72.test.local"
09[CFG] using trusted ca certificate "DC=local, DC=test, CN=test-WS2012-CA"
09[CFG] reached self-signed root ca with a path length of 0
09[CFG] checking certificate status of "C=IN, ST=KA, L=Bangalore, O=Cisco, OU=TAC, CN=ftd72.test.local"
09[CFG] fetching crl from 'ldap:///CN=test-WS2012-CA,CN=ws2012,CN=CDP,CN=Public%20Key%20Services,CN=Ser
09[LIB] LDAP bind to 'ldap:///CN=test-WS2012-CA,CN=ws2012,CN=CDP,CN=Public%20Key%20Services,CN=Services
09[CFG] crl fetching failed
09[CFG] certificate status is not available
09[IKE] authentication of 'C=IN, ST=KA, L=Bangalore, O=Cisco, OU=TAC, CN=ftd72.test.local' with RSA sig
09[IKE] received ESP_TFC_PADDING_NOT_SUPPORTED, not using ESPv3 TFC padding
09[IKE] authentication of 'sswan.test.local' (myself) with RSA signature successful
09[IKE] IKE_SA strongswan-ftd[11] established between 10.106.67.200[sswan.test.local]...10.106.69.230[C
09[IKE] scheduling reauthentication in 83505s
09[IKE] maximum IKE_SA lifetime 92145s
09[IKE] sending end entity cert "C=IN, ST=KA, O=Cisco, OU=TAC, CN=sswan.test.local"
09[CFG] selected proposal: ESP:AES_CBC_128/HMAC_SHA1_96/NO_EXT_SEQ
09[IKE] CHILD_SA hostB-hostA{8} established with SPIs cc01a2a7_i 3594c049_o and TS 10.106.71.110/32 ===
09[ENC] generating IKE_AUTH response 1 [ IDr CERT AUTH SA TSi TSr N(AUTH_LFT) ]
09[ENC] splitting IKE message (1852 bytes) into 2 fragments
09[ENC] generating IKE_AUTH response 1 [ EF(1/2) ]
09[ENC] generating IKE_AUTH response 1 [ EF(2/2) ]
09[NET] sending packet: from 10.106.67.200[500] to 10.106.69.230[500] (1248 bytes)
09[NET] sending packet: from 10.106.67.200[500] to 10.106.69.230[500] (672 bytes)
12[NET] received packet: from 10.106.69.230[500] to 10.106.67.200[500] (76 bytes)
12[ENC] parsed INFORMATIONAL request 2 [ ]
12[ENC] generating INFORMATIONAL response 2 [ ]
```