Installing Custom CA Certificates and Importing SUDI Certificate

The Cisco IoT FND OVA is bundled with keys and certificates which is stored in a keystore. The following values are the default values of the keys and certificates:

- On the Cisco IoT FND OVA Linux Host:

  Keystore Location: /opt/fnd/data/

  Keystore Name: cgms_keystore.selfsigned

- On Cisco IoT FND container:

  Keystore Location: /opt/cgms/server/cgms/conf/

  Keystore Name: cgms_keystore

- **Default Password: Public123!**

Important

- This is the default password for both the files.
- Both these files have the same content.
- When Cisco IoT FND container is restarted, the values of /opt/cgms/server/cgms/conf/cgms_keystore file in Cisco IoT FND container is overwritten by /opt/fnd/data/cgms_keystore file. If /opt/fnd/data/cgms_keystore file is not present in host, then /opt/fnd/data/cgms_keystore.selfsigned file is used.

When Cisco IoT FND OVA is a new installation, each certificate/key entry is referenced by an alias name in the keystore. The default alias are:

- cisco_sudi (cisco root CA certificate with 2029 expiry)
- jmarconi (cisco certificate)
- cgms (self signed certificate that is used by Cisco IoT FND when communicating with devices it has to manage)

Note This keystore is specific for certificates used for Cisco IoT FND communication with its managed devices. There is a different keystore for web certificate.

Custom cgms_keystore

The cgms certificate in /opt/cgms/server/cgms/conf/cgms_keystore file in Cisco IoT FND container and /opt/fnd/data/cgms_keystore.selfsigned file of the linux host has by default self signed certificate of Cisco IoT FND. There are two options to build a custom cgms_keystore in /opt/fnd/data location on linux host, where the Cisco IoT FND certificate of the customer organisation can be imported and stored.

We can either copy the existing /opt/fnd/data/cgms_keystore.selfsigned file on the Linux host or build it from scratch. After the cgms_keystore file is present on the linux host, if both /opt/fnd/data/cgms_keystore.selfsigned and /opt/fnd/data/cgms_keystore files are present, then /opt/fnd/data/cgms_keystore takes precedence.

Note NTP is a mandatory requirement for Public Key Infrastructure. Hence NTP should be in sync between the issuing Certificate Authority (CA) server, Cisco IoT FND, TPS, and FAR/HER. If hostname or IP address has to be changed for the Cisco IoT FND host, it has to be done before certificate for Cisco IoT FND is issued and hence it should be done before starting to build cgms_keystore.

The SAN field in Cisco IoT FND certificate is a mandatory requirement and contains the hostname of the Cisco IoT FND server. Any change in hostname or IP address is listed in SAN field (if IP address is also present in the SAN field), then the certificate should be reissued. Depending on the PnP type used, the SAN field contains the hostname of the Cisco IoT FND or the IP address or both.

The cgms_keystore should contain the below mandatory certificates/keys:

- Issuing CA certificate of the organisation – This is the certificate of the issuing CA server of the organisation. The issuing CA server can be a root CA server or intermediate CA server. If it is an intermediate CA, it is recommended to import root CA and also intermediate CA certificates into the keystore.
- Cisco IoT FND device certificate is issued for Cisco IoT FND by issuing CA server.
- Cisco SUDI with 2029 expiry date – This is the cisco manufacturer certificate for Cisco IoT FND issued by Cisco with expiry date 2029.
- Cisco SUDI with 2099 expiry date – This is the cisco manufacturer certificate for Cisco IoT FND issued by Cisco with expiry date 2099.

The below option shows how to build cgms_keystore file from scratch that contains the required certificates and keys.

Procedure

Step 1 Change directory to /opt/fnd/data on linux host.

```
# cd /opt/fnd/data
```

Importing Root/Issuing CA Certificate

Step 2 Importing any certificate using keytool command creates the keystore file, if keystore file does not exist. Note that the name of the file has to be cgms_keystore as Cisco IoT FND refers the file with this name. Copy the issuing CA certificate of your organisation to any location (using scp or any other file transfer method). In this illustration, it is copied to /root/rootca.pem. The certificate can be of the format .cer or .crt or .pem. In this illustration, the issuing CA is the root CA and hence the alias name root is used.

```
# keytool -import -trustcacerts -keystore /opt/fnd/data/cgms_keystore -alias
 root -file /root/rootca.pem
```

Convert the keystore from jks to pkcs12.

```
# keytool -importkeystore -srckeystore /opt/fnd/data/cgms_keystore
-destkeystore /opt/fnd/data/cgms_keystore -deststoretype pkcs12
```

Verify that the file has been created by listing the contents of the keystore.

```
# keytool -list -keystore /opt/fnd/data/cgms_keystore
```

Importing Cisco IoT FND Certificate

Step 3 Import the Cisco IoT FND certificate.

Note
Cisco IoT FND certificate has to be imported only with alias name of cgms.

The below steps tells how to generate a key pair .csr file that can be presented to the issuing CA server for a certificate to be granted for Cisco IoT FND server. The .csr file is required by few issuing CA servers of few PKI vendors. If the .csr certificate is given to the issuing CA server, then a certificate is generated based on the contents of the .csr file. If a certificate of Cisco IoT FND has already been issued, use the following steps, if the Cisco IoT FND certificate issued has .pem or .cer or .crt extension. If the Cisco IoT FND certificate has .pfx extension, follow step 4.

a) Generate a key pair and .csr file .

```
# keytool -genkeypair -keyalg RSA -keysize 2048 -alias cgms
-ext "SAN=dns.labfnd.cisco.com, ip:1.0.0.1" -keystore /opt/fnd/data/cgms_keystore
-dname CN=labfnd, OU=iotescblr, O=cisco, L=Bengaluru, ST=Karnataka, C=IN"
```

Note
The key size in this example is 2048, but 4096 can also be used.

```
# keytool -certreq -file labfnd.csr -keystore
/opt/fnd/data/cgms_keystore -alias cgms -ext "SAN=dns:labfnd.cisco.com,ip:1.0.0.1"
```

Note
This .csr file is then presented to the issuing CA server and a certificate is obtained for Cisco IoT FND server.

b) Copy the issued certificate to FND server in any location. In this sample, it is copied to /opt/fnd/data as labfnd.pem file. Import the certificate using below command.

```
# keytool -import -trustcacerts -keystore /opt/fnd/data/cgms_keystore
-alias cgms -file /root/labfnd.pem
```

Step 4 If the Cisco IoT FND issued certificate issued has .pfx format, then we will have a .pfx file instead of the .pem file. If it is a .pfx file, check the alias name of the .pfx file and then import it using the alias cgms in the cgms_keystore.

- Find the alias name of the pfx file. In this case, the nms.pfx is copied to the current location.

```
# keytool -list -v -keystore /opt/fnd/data/nms.pfx -srcstoretype
 pkcs12 | grep Alias
```

- Import the pfx into the cgms_keystore with alias cgms. In this sample, "le-IoT FND-8f0908aa-dc8d-4101-a526-93b4eaad9481" is the alias present in the .pfx file.

```
# keytool -importkeystore -v -srckeystore /opt/fnd/data/nms.pfx
-destkeystore /opt/fnd/data/cgms_keystore -srcalias
 le-IoT FND-8f0908aa-dc8d-4101-a526-93b4eaad9481 -destalias cgms
```

Importing SUDI with 2029 Expiry

Step 5 The SUDI certificate with 2029 expiry is present in /opt/fnd/data directory as cisco-sudi-ca.pem. Import this file to cgms_keystore.

```
# keytool -import -trustcacerts -alias cisco_sudi -file
/opt/fnd/data/cisco-sudi-ca.pem -keystore /opt/fnd/data/cgms_keystore
```

Importing SUDI with 2099 Expiry

Step 6 The updated SUDI certificate with 2099 expiry is present in Cisco IoT FND container as cisco-ca.pem file, copy this to /opt/fnd/data in linux host.

```
docker cp fnd-container:/opt/cgms/server/cgms/conf/ciscosudi/cisco-ca.pem
/opt/fnd/data/
```

Step 7 Import the SUDI with 2099 expiry, that is, cisco-ca.pem to cgms_keystore.

```
# keytool -import -trustcacerts -keystore /opt/fnd/data/cgms_keystore
-alias sudi1 -file /opt/fnd/data/cisco-ca.pem
```

Step 8 Copy the **cisco-sudi-ca.pem** and **cisco-ca.pem** to the TPS server using the following command:

```
# scp ./cisco-ca.pem root@10.0.0.1
```

Here's an example output:

```
The authenticity of host '10.0.0.1 (10.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.1' (ECDSA) to the list of known hosts.
root@10.0.0.1's password:
cisco-ca.pem                                          100%   123KB 123.4KB/s   00:01
```

Step 9 Import certificates into the Java KeyStore using the `keytool` utility on the TPS server:

```
# keytool -import -trustcacerts -keystore /opt/cgms-tpsproxy/conf/cgms_keystore -alias cisco_sudi
-file ./cisco-sudi-ca.pem
keytool -import -trustcacerts -keystore /opt/cgms-tpsproxy/conf/cgms_keystore -alias cisco_sudi1
-file ./cisco-ca.pem
```

Step 10 Restart the container.

```
docker stop fnd-container
docker start fnd-container
```

Important
It is not advised to use the restart command. It is best practice to stop the container and then start the container so the services can stop gracefully. Sometimes restart will not be graceful and can lead to operational issues.

Step 11 After restart, verify that the contents of the cgms_keystore in the Cisco IoT FND container has same contents as that of the cgms_keystore in /opt/fnd/data of linux host using the below command.

```
# keytool -list -v -keystore -/opt/fnd/data/cgms_keystore
```

```
# docker exec -it fnd-container keytool -list -v -keystore /opt/cgms/server/cgms/conf/cgms_keystore
```

Step 12 To configure or change the cgms_keystore password, see Changing Password for more information.

- Changing Password, on page 5
- Managing Custom Web Certificates, on page 5
- Installing Custom Browser Certificates, on page 6
- Properties Used in Cisco IoT FND and TPS Configuration, on page 11

Changing Password

The cgms.properties file should contain the password for cgms_keystore, so that IoT FND application can access the cgms_keystore. Hence the first time cgms_keystore is created, encrypt the password of cgms_keystore and provide this encrypted password in cgms.properties file.

If at any time the password for cgms_keystore is changed, then the changed password has to be encrypted again and updated in the cgms.properties file.

Procedure

Step 1 Run the following command to encrypt the password for the new **cgms_keystore**. The sample is provided below.

```
# docker exec -it fnd-container /opt/cgms/bin/encryption_util.sh
encrypt <keystore password>
```

```
# docker exec -it fnd-container /opt/cgms/bin/encryption_util.sh
encrypt cisco123
#2bVvZsq+vsq94YxuAKdaag--
```

Step 2 Modify the **cgms.properties** file in the `/opt/fnd/data` folder, and edit the following line to set the new encrypted **cgms_keystore** password:

```
cgms-keystore-password-hidden=<encrypted new cgms_keystore password>
```

Note: With OVA 4.3.x and above, you can leave the `cgms_keystore.selfsigned` default bundled keystore untouched.

If both the files (**cgms_keystore** and **cgms_keystore.selfsigned**) are present, the **cgms_keystore** will be used by the container.

Managing Custom Web Certificates

Procedure

Step 1 The web certificate details are not retained after a reboot of Cisco IoT FND. You should perform a back up of the following files before you attempt to reboot Cisco IoT FND.

In the /opt/cgms/server/cgms/conf/ directory:

- jbossas.keystore.password
- jbossas.keystore
- VAULT.dat
- vault.keystore
- standalone.xml
- cgms.conf

Step 2 Copy the above files to their respective folders, and restart the Cisco IoT FND.

Installing Custom Browser Certificates

By default Cisco IoT FND installations use a self-signed certificate for HTTP or HTTPS communication using either a client web browser or the NB API client. If required, you can use certificates signed by your CA servers. This section presents installation procedures for these custom certificates.

BEFORE YOU BEGIN

- Clear the client browser cache.
- Remove existing certificates for the NMS server (by IP and DNS) on the client browser

  In Firefox, for example, select **Preferences** > **Advanced** > **Encryption** > **View Certifications**. Remove the certificates in the list for the respective server.
- Choose a common name to use in the signed certificate.

  This name requires a DNS entry that resolves to the NMS server IP address.
- Generate the new certificates and export them to a.PFX file.

  This file must contain the private keys, public certificate, and CA server certificates.

  See Use Keytool to Create the cgms_keystore File, on page 10 for the procedure to generate the private and public keys for the cgms_keystore file and export them to a .PFX file.

Installing Custom Certificates in the Browser Client

These steps assume that the Java environment has been set to use the Java bundled with Cisco IoT FND in `/opt/cgms/jre`.

Note Update the jbossas.keystore in Cisco IoT FND container. The keystore is present in: `/opt/cgms/server/cgms/conf/`. The name of the keystore is jbossas.keystore.

Procedure

Step 1 Copy the certificate and private key *.pfx file from the host to container.

```
docker cp newcert.pfx fnd-container:/opt/cgms/server/cgms/conf/
```

Step 2 Run the following command to enter the container.

```
docker exec -i -t fnd-container /bin/bash
```

Step 3 Change directory to `/opt/cgms/server/cgms/conf/` in the container.

```
cd /opt/cgms/server/cgms/conf/
```

Step 4 On the NMS server, copy the existing jbossas.keystore, jbossas.keystore.password, vault.keystore, and VAULT.dat files from the `/opt/cgms/server/cgms/conf/` directory to a safe location.

Step 5 Determine the alias in the .PFX file that you plan to import into the new jbossas.keystore file:

```
#keytool -list -v -keystore newcert.pfx -storetype pkcs12
```

Enter the keystore password: keystore_password_when_pfx_file_was_created

```
Keystore type: PKCS12
Keystore provider: SunJSSE

Your keystore contains 1 entry

Alias name: le-nms-a88ef13a-a519-457f-a2e1-0540f5453ee0
Creation date: Feb 23, 2018
Entry type: PrivateKeyEntry
Certificate chain length: 2
Certificate[1]:
…
```

Step 6 Delete the existing jbossas.keystore from the `/opt/cgms/server/cgms/conf/` directory.

Step 7 Import the new custom certificate, in .PFX file format into a new jbossas.keystore file; and, at the same time change the alias name to **jboss**. Follow the prompts:

```
# keytool -importkeystore -v -srckeystore newcert.pfx -srcstoretype pkcs12
-destkeystore /opt/cgms/server/cgms/conf/jbossas.keystore -deststoretype jks
-srcalias le-nms-a88ef13a-a519-457f-a2e1-0540f5453ee0 -destalias
jboss -destkeypass your_keystore_password
Enter destination keystore password: your_keystore_password
Enter source keystore password: keystore_password_when_pfx_file_was_created
[Storing /opt/cgms/server/cgms/conf/jbossas.keystore]
```

Step 8 The keystore password is stored in the `/opt/cgms/server/cgms/conf/VAULT.dat` file.

Perform the following steps to update the keystore password to match the one entered in Step 4 which is *your_keystore_password*.

- IoT FND releases 4.9.x and earlier
- IoT FND releases 4.10.x and later

Cisco IoT FND releases 4.9.x and earlier:

a) Delete vault.keystore file.
b) Delete VAULT.dat file.
c) Create a new vault.keystore file.

```
keytool -genseckey -alias vault -storetype jceks -keyalg AES -keysize 128 -storepass
your_keystore_password -keypass your_keystore_password -keystore
/opt/cgms/server/cgms/conf/vault.keystore
```

d) Update the VAULT.dat file with the new password:

```
/opt/cgms/bin/vault.sh -k /opt/cgms/server/cgms/conf/vault.keystore -p your_keystore_password
-e /opt/cgms/server/cgms/conf -i 50 -s 12345678 -v vault -b keystore_pass -a password -x
your_keystore_password
```

If required you can optionally modify the iteration and salt values.

Example Output:

```
=========================================================================
JBoss Vault
JBOSS_HOME:/opt/cgms
JAVA: java
=========================================================================

Dec 20, 2018 3:25:29 PM org.picketbox.plugins.vault.PicketBoxSecurityVault init
INFO: PBOX000361: Default Security Vault Implementation Initialized and Ready
Secured attribute value has been stored in vault.
Please make note of the following:
********************************************
Vault Block:keystore_pass
Attribute Name:password
Configuration should be done as follows:
VAULT::keystore_pass::password::1
********************************************
Vault Configuration in AS7 config file:
********************************************
...
</extensions>
<vault>
<vault-option name=="KEYSTORE_URL"
value="/opt/cgms/server/cgms/conf/vault.keystore"/>
<vault-option name="KEYSTORE_PASSWORD" value="MASK-
VKsAwH928fwt.3H2qUwOG"/>
<vault-option name="KEYSTORE_ALIAS" value="vault"/>
<vault-option name="SALT" value="12345678"/>
<vault-option name="ITERATION_COUNT" value="50"/>
<vault-option name="ENC_FILE_DIR" value="/opt/cgms/server/cgms/conf/"/>
</vault><management>...
********************************************
```

The vault.keystore contains the reference to VAULT.dat and VAULT.dat stores and hides the jboss keystore password. This command creates a new VAULT.dat file that contains the new jboss.keytsore password.

Cisco IoT FND releases 4.10.x and later:

a) Delete `vault.keystore` file.
b) Delete VAULT.dat file.
c) Create a new vault.keystore file:

```
/opt/cgms/jre/bin/keytool -genseckey -alias vault -storetype jceks
-keyalg AES -keysize 256 -storepass keystore -dname "CN=IoTFND, OU=IoT, O=Cisco Systems,
L=San Jose, ST=CA, C=US" -keypass keystore -validity 730 -keystore vault.keystore
```

d) Update the VAULT.dat file with the new password:

```
/opt/cgms/bin/vault.sh --keystore /opt/cgms/server/cgms/conf/vault.keystore
 --keystore-password keystore --alias vault --vault-block keystore_pass
--attribute password --sec-attr keystore --enc-dir /opt/cgms/server/cgms/conf/
 --iteration 50 --salt 12345678 -n
```

If required you can optionally modify the iteration and salt values.

Expected Output:

```
=========================================================

  JBoss Vault

  JBOSS_HOME: /opt/cgms

  JAVA: /opt/cgms/jre/bin/java

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

WFLYSEC0047: Secured attribute value has been stored in Vault.
Please make note of the following:
********************************************
Vault Block:keystore_pass
Attribute Name:password
Configuration should be done as follows:
VAULT::keystore_pass::password::1
********************************************
WFLYSEC0048: Vault Configuration commands in WildFly for CLI:
********************************************
For standalone mode:
/core-service=vault:add(vault-options=[("KEYSTORE_URL" =>
"/opt/cgms/server/cgms/conf/vault.keystore"),
("KEYSTORE_PASSWORD" => "MASK-0VKsAwH928fwt.3H2qUwOG"),("KEYSTORE_ALIAS" => "vault"),
("SALT" => "12345678"),("ITERATION_COUNT" => "50"),("ENC_FILE_DIR" =>
"/opt/cgms/server/cgms/conf/")])
********************************************
For domain mode:
/host=the_host/core-service=vault:add(vault-options=[("KEYSTORE_URL" =>
"/opt/cgms/server/cgms/conf/vault.keystore"),
("KEYSTORE_PASSWORD" => "MASK-0VKsAwH928fwt.3H2qUwOG"),("KEYSTORE_ALIAS" => "vault"),
("SALT" => "12345678"),("ITERATION_COUNT" => "50"),("ENC_FILE_DIR" =>
 "/opt/cgms/server/cgms/conf/")])
********************************************
```

The vault.keystore contains the reference to VAULT.dat and VAULT.dat stores and hides the jboss keystore password. This command creates a new VAULT.dat file that contains the new jboss.keytsore password.

Step 9 Copy `/opt/cgms/standalone/configuration/standalone.xml` and standalone-cluster.xml to a safe location.

Step 10 Update the `/opt/cgms/standalone/configuration/standalone.xml` or standalone-cluster.xml file.

Depending on whether the Cisco IoT FND server is standalone or clustered, you can update the respective file.

a) Replace the keystore password with the output in Step 5.

```
<vault>
<vault-option name="KEYSTORE_URL"
value="/opt/cgms/server/cgms/conf/vault.keystore"/>
<vault-option name="KEYSTORE_PASSWORD" value="MASK-VKsAwH928fwt.3H2qUwOG"/>
<vault-option name="KEYSTORE_ALIAS" value="vault"/>
<vault-option name="SALT" value="12345678"/>
<vault-option name="ITERATION_COUNT" value="50"/>
<vault-option name="ENC_FILE_DIR" value="/opt/cgms/server/cgms/conf/"/>
</vault>
```

b) Edit standalone.xml or standalone-cluster.xml and replace the existing <vault> section with the above. Save and exit.

Step 11 Restart the CGMS service by running the command:

```
/etc/init.d/cgms restart
```

Step 12 Use your browser to connect to the NMS server.

Step 13 Accept and add the new certificates.

Step 14 Use your browser to log in to Cisco IoT FND.

Note
Back up the following files to the local host before you restart or upgrade the container to prevent it from replacing the custom web certificates with self-signed certificates.

- jbossas.keystore,
- jbossas.keystore.password,
- VAULT.dat,
- vault.keystore,
- standalone.xml, and
- standalone-cluster.xml.

Use Keytool to Create the cgms_keystore File

Use the instructions provided in this section to create the cgms_keystore file for both Cisco IoT FND and the TPS proxy.

Procedure

Step 1 Enter the following command on the Cisco IoT FND or TPS proxy server as root to view the contents of the .PFX file.

```
[root@ tps_server
~]# keytool -list -v -keystore nms_cert.pfx -srcstoretype pkcs12
```

Note
You can get the alias name from the .PFX file during import.

Step 2 Enter the keystore password when prompted.

The password is the same as the one you used for creating the .PFX file.

The alias name is displayed as part of the information as given in this Example 1. You can use this alias name in the command to import the certificates into the cgms_keystore file.

Step 3 Enter the following command to import the certificates into the cgms_keystore file:

```
keytool -importkeystore -v -srckeystore
filename.pfx -srcstoretype pkcs12
-destkeystore cgms_keystore -deststoretype jks -srcalias alias_name
-destalias cgms
```

```
-destkeypass
keystore_password
```

Step 4 Enter the destination keystore password when prompted.

Step 5 Re-enter the keystore password when prompted.

Step 6 Enter the password you used earlier while creating the .PFX file (either *nms_cert.pfx* or *tps_cert.pfx*) when prompted for the source keystore password.

Example 1:

To view the *nms_cert.pfx* file and access the alias name, enter the following commands as root:

```
# keytool -list -v -keystore nms_cert.pfx -srcstoretype pkcs12
Enter keystore password: keystore
Keystore type: PKCS12
Keystone provider: SunJSSE
Your keystore contains 1 entry
Alias name: le-cgnms-75edd1e3-7e65-41b4-97f1-a913ebf21c8b
Creation date: March 29.2018
Entry type: PrivateKey Entry
Certificate chain length: 3
Certificate[1]:
Owner: CN=nms.sgbu.cisco.com
Issuer: CN=cisco-RSA-SUBCA-CA, DC=cisco, DC=com
...
```

Note
This example diplays the steps for the *nms_cert.pfx*. To view the details on the *tps_cert.pfx* and import the certificates to the TPS proxy, use the same commands but replace the references to *nms_cert.pfx* with *tps_cert.pfx*, and use the alias name from the *tps_cert.pfx* file.

Example 2:

To import the certificates to the **cgms_keystore** file on Cisco IoT FND, enter the following commands as root:

```
# keytool -importkeystore -v
-srckeystore nms_cert.pfx -srcstoretype pkcs12
-destkeystore cgms_keystore -deststoretype jks -srcalias
le-cgnms-75edd1e3-7e65-41b4-97f1-a913ebf21c8b -destalias cgms
-destkeypass
keystore_password

Enter destination keystore password: keystore_password
Re-enter new password: keystore_password
Enter source keystore password: keystore
...Storing cgms_keystore
```

Note
The **storing** *cgms* **_keystore** text indicates successful completion.

Properties Used in Cisco IoT FND and TPS Configuration

The Cisco Gateway Management System (CGMS) and TPSPROXY properties that are available for configuration in Cisco IoT FND and TPS are as given:

CGMS Properties

Property Name	Example Value	Description
cgms-keystore-password-hidden=	< encrypted >	Encrypted password for the cgms keystore. Encrypt or decrypt with encryption_util.sh.
hsm-keystore-name=	testGroup1	HSM partition name.
hsm-keystore-password=	< encrypted >	Encrypted HSM partition password.
security-module=	ssm/hsm	Type of security module being used.
ssm-host=	<ipv4 address >	IP Address of SSM server.
ssm-port=	8445	Port of SSM server.
ssm-keystore-alias=	ssm_csmp	Alias name for SSM certificate in keystore.
ssm-keystore-password=	< encrypted >	Encrypted password for the SSM keystore.
ssm-key-password=	< encrypted >	Encrypted key for the SSM key.
multicast-interface-address=	< ipv6 address >	Cisco IoT FND IPv6 source address for multicast traffic.
dhcpV4ClientListenAddresses=	<ipv4 address >	IPv4 address on your Cisco IoT FND server used to exchange DHCPv4 messages.
dhcpV6ClientListenAddresses=	< ipv6 address >	IPv6 address on your Cisco IoT FND server used to exchange DHCPv6 messages.
OptimizeTunnelProv=	true/false	Indicates whether or not to lock the HER during tunnel provisioning.
allowed-outage-skew=	5000	Allow outage skew in seconds, for outage or restoration events.
rf.validate-firmware-tlvs=	true/false	Skips CG-Mesh device firmware validation.
googleMapsClientId=	< Client ID >	Google maps client ID.
googleMapsApiKey=	< API key >	Google maps API key.
enable-bootstrap-service=	true/false	Used to enable PNP bootstrapping service.

Property Name	Example Value	Description
scep-url=	http(s) :// < url of SCEP server >	URL of SCEP server.
ca-fingerprint=	< fingerprint of CA certificate >	Fingerprint of CA certificate.
proxy-bootstrap-ip=	<ipv4/v6 address or FQDN >	PNP server identity sent by Cisco IoT FND to the PNP agent.
bootstrap-fnd-alias=	subca	Alias name assigned to the CA certificate from the issuer in the Cisco IoT FND keystore.
pnp-server-port=	9125	PNP server port, default is 9125.
pnp-install-trustpool=	true/false	Send the CA bundle file which includes well known public CA certificates.
reload-during-bootstrap=	true/false	Indicates whether or not to reload a device after PNP boostrapping.
router-file-upload-retries	0	Number of retries for router file upload job
router-firmware-upload-retries	0	Number of retries for the firmware upload job.
router-firmware-install-retries	0	Number of retries for the firmware install job.
collect-cellular-link-metrics	true/false	Indicates whether or not to collect cellular metrics.
collect-cellular-link-metrics-interval	30	Interval for cellular metrics.
router-firmware-upload-timeout-minutes=	30	Firmware upload job timeout duration in minutes.
router-firmware-install-timeout-minutes=	60	Firmware install job timeout duration in minutes.
cgr-ha-fetch-mesh-key-attempts	3	Number of attempts to fetch the mesh keys.
cgr-ha-fetch-mesh-key-delay-mins	1	Number of minutes or interval between mesh-key-attempts.

TPSPROXY Properties

Property Name	Example Value	Description
cgdm-tpsproxy-addr=	<ipv4/v6 address or FQDN >	Source IP address of messages coming from the TPSProxy.
cgdm-tpsproxy-subject=	CN="common_name", OU="organizational_unit", O="organization", L="location", ST="state", C="country"	The exact certificate subject contained in the TPSPROXY's certificate.
bootstrap-proxy-listen-port=	9125	Port on which TPS is listening for HTTP traffic.
inbound-bsproxy-destination=	<ipv4/v6 address or FQDN >	IP address and port to forward info received from the router over HTTP.
outbound-proxy-allowed-addresses=	<ipv4/v6 address or FQDN >	Comma separated list of FQDN/IP addresses, the proxy allows outbound messages to originate from it.