Provisioning Certificates

The Secure Socket Layer (SSL) protocol secures the network communication and allows data to be encrypted before transmission and provides security. Many application servers and web servers support the use of keystores for SSL configuration. The use of SSL between the switches and KMC requires provisioning of Public Key Infrastructure.

This chapter includes the following topics:

Information About Public Key Infrastructure Certificates

A certificate is an electronic document that you use to identify a server, a company, or some other entity and to associate that identity with a public key.

Certificate authority (CA) are entities that validate identities and issue certificates. The certificate that the CA issues binds a particular public key to the name of the entity that the certificate identifies (such as the name of a server or device). Only the public key that the certificate certifies works with the corresponding private key that is possessed by the entity that the certificate identifies. Certificates help prevent the use of fake public keys for impersonation.

Prerequisites for SSL

Before configuring SSL, consider the following:

- You must install a third-party tool such as the freely available OpenSSL application to generate keys, certificates, and certificate signing requests. Download OpenSSL for Windows from the following link:

  http://gnuwin32.sourceforge.net/packages/openssl.ht
  After installing in Windows, by default, openssl.exe is located at c:\openssl\bin.
- Ensure that the time in all the switches, DCNM-SAN and the system running the OpenSSL commands, are all synchronized.

- Provide different identities for the CA certificate and KMC certificate.
- Only JRE1.6 JAVA keytool is supported for importing PKCS12 certificates to Java Keystores (JKS) files.

Configuring SSL Using CLI

This section describes the following SSL configuration topics:

Creating the CA Certificate

Your organization might already have a CA certificate. If you are requesting the CA from a security administrator, indicate that you need the CA certificate in PEM format, and you will need them to sign certificates as part of configuring SME. If you do not have or want to use an existing CA, you can create a new one by using an OpenSSL command.

This command is used to create the Certificate Authority (CA). This command creates a certificate (identify plus public key) and a private key. The private key must always be protected. In a typical enterprise organization, the private key should already exist.

Create a CA certificate using the OpenSSL application. Enter the following command for the 365-day certificate:

OpenSSL> **req -x509 -days 365 -newkey rsa:2048 -out cacert.pem -outform PEM**

This command creates two files: a cacert.pem file and a privkey.pem file in the directory with OpenSSL.exe. The cacert.pem file is the certificate. The privkey.pem file must be stored in a safe location.

Configuring Trust points

This sequence of steps must be done for all of the switches managed by a DCNM-SAN server. Ensure that the same trustpoint name is used for all the switches.

To configure truspoints, follow these steps:

Step 1 Enter the configuration mode.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

Step 2 Create a trust point named my_ca.

```
switch(config)# crypto ca trustpoint my_ca
```

Step 3 Create an RSA key pair for the switch in the trustpoint submode.

```
switch(config-trustpoint)# rsakeypair my_ca_key 2048
```

Step 4 Exit the trustpoint submode.

```
switch(config-trustpoint)# exit
```

Step 5 Authenticate the cacert.pem file for the trustpoint by cutting and pasting the contents of the cacert.pem created in Step 1.

```
switch(config)# crypto ca authenticate my_ca
input (cut & paste) CA certificate (chain) in PEM format;
end the input with a line containing only END OF INPUT :
```

```
----BEGIN CERTIFICATE----
MIIDnjCCAwegAwIBAgIBADANBgkqhkiG9w0BAQQFADCBlzELMAkGA1UEBhMCVVMx
EzARBgNVBAgTCkNhbGlmb3JuaWExETAPBgNVBAcTCFNhbiBKb3NlMRowGAYDVQQK
ExFDaXNjbyBTeXN0ZW1zIEluYzEOMAwGA1UECxMFRGV2ZWwxETAPBgNVBAMTCG1h
bWFzc2V5MSEwHwYJKoZIhvcNAQkBFhJtYW1hc3NleUBjaXNjby5jb20wHhcNMDcx
MTIyMDgzNDM1WhcNMDgxMTIxMDgzNDM1WjCBlzELMAkGA1UEBhMCVVMxEzARBgNV
BAgTCkNhbGlmb3JuaWExETAPBgNVBAcTCFNhbiBKb3NlMRowGAYDVQQKExFDaXNj
byBTeXN0ZW1zIEluYzEOMAwGA1UECxMFRGV2ZWwxETAPBgNVBAMTCG1hbWFzc2V5
MSEwHwYJKoZIhvcNAQkBFhJtYW1hc3NleUBjaXNjby5jb20wgZ8wDQYJKoZIhvcN
AQEBBQADgY0AMIGJAoGBAMbZAv0+Ka/FS3/jwdaqItc8Ow3alpw9gyqEzA3uFLjN
tXSfHRu9OsrP5tliHHlJP+fezeAUuVfmMTPrOIxURcF2c7Yq1Ux5s4Ua3cMGf9BG
YBRbhO8Filt2mGDqY5u0mJY+eViR69MZk8Ouj+gRxQq83fB8MqJG39f1BedRcZLB
AgMBAAGjgfcwgfQwHQYDVR0OBBYEFGXsBg7f7FJcL/741j+M2dgI7rIyMIHEBgNV
HSMEgbwwgbmAFGXsBg7f7FJcL/741j+M2dgI7rIyoYGdpIGaMIGXMQswCQYDVQQG
EwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxGjAY
BgNVBAoTEUNpc2NvIFN5c3RlbXMgSW5jMQ4wDAYDVQQLEwVEZXZlbDERMA8GA1UE
AxMIbWFtYXNzZXkxITAfBgkqhkiG9w0BCQEWEm1hbWFzc2V5QGNpc2NvLmNvbYIB
ADAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBAUAA4GBAFmDucZlBZFJk09IihEm
5wd4oouxHsKPQroyG/CYShv1XXAyEGytxuCAITDzMq2IJiFbZt0kIiyuP9YRQLNR
z47G4IRJGp5J2HnOc2cdF8Mc0DDApdgndUiIX/lv7vuQfyxqX45oSncwQct3y38/
FPEbcRgZgnOgwcrqBzKV0Y3+
----END CERTIFICATE----
END OF INPUT
Fingerprint(s): MD5 Fingerprint=1E:18:10:69:7B:C1:CC:EA:82:08:67:FB:90:7D:58:EB
Do you accept this certificate? [yes/no]:yes
```

Step 6 Generate a certificate request for enrolling with the trustpoint created in Step 2. This request will be used by the CA sign the switch's certificate.

```
switch(config)# crypto ca enroll my_ca
Create a challenge password. You will need to verbally provide this
password to the CA Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please make a note of it.
Password:nbv123
The subject name in the certificate will be: ips-vegas8.cisco.com
Include the switch serial number in the subject name? [yes/no]:no
Include an IP address in the subject name [yes/no]:no
The certificate request will be displayed...
----BEGIN CERTIFICATE REQUEST----
MIIBJTCB0AIBADAfMR0wGwYDVQQDExRpcHMtdmVnYXM4LmNpc2NvLmNvbTBcMA0G
CSqGSIb3DQEBAQUAA0sAMEgCQQCeAzv5w9d32YpPfYdNYoFjOW0yRVbYEe+mNHi8
b2VPOVZ6UOFdhIS1Im0/Xv1Bpcuy4TRktu7whNyyvvu3niVdAgMBAAGgTDAVBgkq
hkiG9w0BCQcxCBMGbmJ2MTIzMDMGCSqGSIb3DQEJDjEmMCQwIgYDVR0RAQH/BBgw
FoIUaXBzLXZlZ2FzOC5jaXNjby5jb20wDQYJKoZIhvcNAQEEBQADQQBzPcKE3Eje
TjODnPXNkz1WsU3oUdsuxOT/m1OSBZvhBfHICQZZpfS2ILqaQP16LiZCZydHWViN
Q+9LmHUZ4BDG
----END CERTIFICATE REQUEST----
switch(config)#
```

Step 7 Create a file named switch.csr in the OpenSSL.exe directory. Cut and paste the certificate request created in Step 6. Ensure that you include the BEGIN CERTIFICATE REQUEST and END CERTIFICATE REQUEST lines in the file content.

Step 8 Generate a certificate using the switch certificate request in the OpenSSL application by entering the following command: OpenSSL> x509 -req -days 365 -in switch.csr -CA cacert.pem -CAkey privkey.pem -set_serial 01 -out switch.pem

This is the switch's public certificate, now signed by the CA.

Note If your security administrator controls the CA, you will need to send them the switch.csr file and request that they complete this step and respond with the switch.pem file.

Step 9 Import the signed certificate on the switch by cutting and pasting the contents of the switch.pem file that was created in Step 8.

```
switch(config)# crypto ca import my_ca certificate
input (cut & paste) certificate in PEM format:
----BEGIN CERTIFICATE----
MIIB4jCCAUsCAQEwDQYJKoZIhvcNAQEEBQAwgZcxCzAJBgNVBAYTAlVTMRMwEQYD
VQQIEwpDYWxpZm9ybmlhMREwDwYDVQQHEwhTYW4gSm9zZTEaMBgGA1UEChMRQ2lz
Y28gU3lzdGVtcyBJbmMxDjAMBgNVBAsTBURldmVsMREwDwYDVQQDEwhtYW1hc3Nl
eTEhMB8GCSqGSIb3DQEJARYSbWFtYXNzZXlAY2lzY28uY29tMB4XDTA3MTIxNDAy
MzIzOVoXDTA4MTIxMzAyMzIzOVowHzEdMBsGA1UEAxMUaXBzLXZlZ2FzOC5jaXNj
by5jb20wXDANBgkqhkiG9w0BAQEFAANLADBIAkEAngM7+cPXd9mKT32HTWKBYzlt
MkVW2BHvpjR4vG9lTzlWelDhXYSEtSJtP179QaXLsuE0ZLbu8ITcsr77t54lXQID
AQABMA0GCSqGSIb3DQEBBAUAA4GBAKR3WAAF/9zMb2u9A42I2cB2G5lucSzndc4P
+O4sYZF5pBt7UpyAs1GKAqivGXVq2FJ2JetX78Fqy7jYCzanWm0tck0/G1dSfr/X
lCFXUuVed9de02yqxARSEx8mX4ifqzYHErHdbi+vDAaMzkUEvHWthOuUZ7fvpoNH
+xhRAuBo
----END CERTIFICATE----
```

You now have a fully configured trustpoint on the switch: A defined trustpoint, a recognized CA, a public/private key pair, and a CA signed certificate identifying the switch. The signed certificate can be used for PKI communications with all entities that recognize the CA. Repeat steps 1 through 9 for every switch in the fabric.

Removing Trustpoints

This sequence of steps must be done for all of the switches to remove the crypto CA signed trustpoints.

To remove the trustpoints, follow these steps:

Step 1 Enter the configuration mode.

Example:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

Step 2 Enter into the trustpoint mode.

Example:

```
switch(config)# crypto ca trustpoint my_ca
```

Step 3 Remove the certificate corresponding to the trustpoint.

Example:
```
switch(config-trustpoint)# delete certificate force
```

Step 4 Remove an RSA keypair for the switch in the trustpoint submode.

Example:
```
switch(config-trustpoint)# no rsakeypair my_ca_key
```

Step 5 Remove the CA certificate corresponding to the trustpoint.

Example:
```
switch(config-trustpoint)# delete ca-certificate
```

Step 6 Exit the trustpoint submode.

Example:
```
switch(config-trustpoint)# exit
```

Step 7 Removing the trustpoint that is configured.

Example:
```
switch(config)# no crypto ca trustpoint my_ca
```

Generating KMC Certificate

To generate the KMC certificate, follow these steps. Generate KMC certificate by entering the following commands in the OpenSSL application:

Step 1 Create the KCM Server's private key.
OpenSSL> genrsa -out sme_kmc_server.key 2048

Step 2 Create a certificate signing request using the private key from Step 1.
OpenSSL> req -new -key sme_kmc_server.key -out sme_kmc_server.csr -config openssl.conf

Step 3 Using the certificate and private key, create a signed certificate for the KMC Server.
OpenSSL> x509 -req -days 365 -in sme_kmc_server.csr -CA cacert.pem -CAkey privkey.pem -CAcreateserial -out sme_kmc_server.cert

Note If your security administrator controls the CA, you will need to send them the sme_kmc_server.csr and request that they complete this step and respond with the sme_kmc_server.cert.

Step 4 Export the signed KMC certificate to pkcs12 format.
OpenSSL> pkcs12 -export -in sme_kmc_server.cert -inkey sme_kmc_server.key -out sme_kmc_server.p12

Step 5 Import this PKCS12 keystore to Java Keystores using JAVA keytool (JRE 1.6).
"<JAVA_HOME>\bin\keytool" -importkeystore -srckeystore sme_kmc_server.p12 -srcstoretype PKCS12 -destkeystore sme_kmc_server.jks -deststoretype JKS

Note Remember the password because it needs to be updated in the properties file.

Step 6 Import the CA certificate to Java Keystores using JAVA keytool (JRE 1.6).
"<JAVA_HOME>\bin\keytool" -importcert -file cacert.pem -keystore sme_kmc_trust.jks -storetype JKS

Step 7 Place these keystore files in the <install path>dcm\fm\conf\cert directory.

Step 8 Modify the KMC SSL settings in the Key Manager Settings in DCNM-SAN Web Client.

Step 9 Restart the DCNM-SAN server.

Note You can also use sme_kmc_server.p12 as KMC certificate and cacert.pem as KMC trust certificate instead of using Java keystores created in Step 5 and 6.

Note User need to place the keystore files for every DCNM upgrade if a cluster is up and running with SSL ON Option. DCNM Upgrade donot retain keystore files.

Feature History for SSL

The below table lists the release history for this feature.

Table 1: Feature History for SSL

Feature Name	Releases	Feature Information
Software change	5.2(1)	In Release 5.2(1), Fabric Manager is changed to DCNM for SAN (DCNM-SAN).
	4.1(1c)	In Release 4.1(1b) and later, the MDS SAN-OS software is changed to MDS NX-OS software. The earlier releases are unchanged and all references are retained.
Generating and installing self-signed certificates	4.1(1c)	In Release 4.1(1c) and later, the SSL configuration when KMC is separated from Fabric Manager Server.
Introduction to Secure Socket Layer (SSL)	3.3(1c)	Describes how to configure SSL for SME and edit SSL settings in the SME wizard.