Cisco CNS Configuration Engine SSL Security

This chapter discusses the setup and configuration of 128-Bit Secure Sockets Layer (SSL) Encrypted Communications between CNS Agent Enabled Cisco IOS Devices and the Cisco CNS Configuration Engine.

This chapter contains the following sections:

CNS Agent and Configuration Engine Security

Security in communication between the Cisco Configuration Engine server and the enabled CNS agent devices (routers) involves three basic functions:

- Identification—Unique CNS agent ID. At a minimum, the CNS agent IDs are required for a device to communicate with the Cisco Configuration Engine server.
- Authentication—Unique CNS password. The Authentication feature consists of a CNS password that the CNS agents present to the Cisco Configuration Engine server as part of any communication handshake.
- Encryption—128-Bit SSL / Shared PKI SSL Trust point Certificates. Secure Sockets Layer (SSL) protocol. The Encryption feature consists of the industry standard Secure Sockets Layer (SSL) protocol, which protects communications between the CNS agent devices and the Cisco Configuration Engine server.

While device identification is mandatory, authorization and encryption are optional features. Of the two optional features, you can enable either or both of them at any time. Encryption does not require authentication, and authentication does not require encryption.

Each security feature is configured and handled separately by both the Cisco Configuration Engine server and the CNS agent devices.

CNS ID, Password Authorization, SSL Encryption

The CNS Configuration Engine has settings for CPE Device Identification, Authorization, and Encryption. Each of these features is configured and handled separately by both the CNS Configuration Engine and the CNS CPE Devices communicating with the CNS Configuration Engine.

The CPE Device CNS Agent IDs are required for a CNS Agent enabled CPE device to communicate with the CNS Configuration Engine server. The Authorization feature consists of a CNS Password that the CNS Agent enabled devices present to the CNS Server. If you choose Encryption, CNS Agent to CNS Configuration Engine communication negotiations takes place as the first of the communication sequence. Only when the SSL Encryption is successful, the CNS Identification and CNS Authentication protocols are passed. When you enable all the options on the CNS Configuration Engine Server, the options should be successfully passed by the CNS Agent CPE Devices or the device is not allowed to connect to the server for any purpose and will be rejected.

The following sections provide more information:

- Identification, page 7-62
- Authentication, page 7-63
- Encryption, page 7-64

Identification

This is a mandatory setting. Each CNS Enabled CPE device (router) must have a unique ID assigned to it before it can start communication with the CNS Configuration Engine. You can configure several CNS agents on a single router. Each agent must have a unique ID assigned to it.

To configure CNS agent IDs on a CNS agent device, enter the following command, beginning in global configuration mode:

cns id string <*unique string*>

cns id string <*unique string for event agent*> **event**

cns id string <*unique string for image agent*> **image**

Example

```
Router#enable
Router#configure terminal
Router(config)#cns id string my-asset-tag1
Router(config)#cns id string my-asset-tag1 event
Router(config)#cns id string my-asset-tag1 image
Router(config)#end
```

On the Cisco Configuration Engine server, when setting up a new device object through the user interface, the administrator must specify these CNS agent IDs. The Cisco Configuration Engine server will not accept any agent connection unless the CNS agent device and the IDs are already configured on the server.

Authentication

The Authentication feature consists of a CNS password that the CNS agent device presents to the Cisco Configuration Engine server as part of any communication handshake. The CNS password is used in two ways:

- It is assigned at the CNS Configuration Engine Server as a global one-time-use password which is known to the CNS Configuration Engine administrator.
- This one-time-use password is then also placed in the CNS Agent Enabled device's configuration by the device administrator before the device attempts to connect to the CNS Configuration Engine Server.

On the CNS Configuration Engine web User Interface, the radio button under the Devices menu labeled Resync Device allows the administrator to reset any CPE Devices unique password as a global one-time-use password. Then, before the CNS agent device attempts to connect to the Cisco Configuration Engine server, the administrator must enter this one-time-use password in the CNS agent device configuration. If a device is out of sync with the server, it can be reset and the server will re-assign a new random value upon successful connection.

The following prompt in the CNS Configuration Engine Setup program sets the server to expect CNS Passwords from the CPE Devices:

```
Authentication settings:
------------------------
Cisco IOS Devices are normally authenticated before being allowed to connect to the Event
Gateway/Config Server. Disabling authentication will increase security risk.
Enable authentication (y/n)? [n] y
```

During setup of the Cisco Configuration Engine server, the administrator must assign this CNS password as a global one-time-use password. Then, before the CNS agent device attempts to connect to the Cisco Configuration Engine server, the administrator must enter this one-time-use password in the CNS agent device configuration.

In the Cisco Configuration Engine server Setup program, authentication is enabled when you answer y at the "Enable authentication" prompt (see Authentication Settings, page 2-18). This configures the Cisco Configuration Engine server to expect the password from the CNS agent device. After authentication is enabled, the administrator must use the Cisco Configuration Engine user interface to reconfigure the actual password. You can set the password by using the CNS Configuration Engine web UI under the menu **Tools > Security Mgr > BootStrap**.

This password can be used for the initial CPE Device connections to the CNS Configuration Engine Server. After each CPE device has been identified and authenticated, the CNS Configuration Engine server generates the password and automatically assigns a random password for the CPE device.

Note The random cns password command has been intentionally hidden for additional security. You can use the cns password command to set or reset the initial password, but you cannot view the password value after it has been set.

To configure the CNS password on the CPE device, enter the following command, beginning in global configuration mode:

cns password <password>

Example

```
Router(config)# cns password fgfg123
Router(config)# end
```

Encryption

The CNS communications between CNS Agent-enabled devices and the CNS Configuration Engine can be encrypted with 128-Bit SSL Protocol strong encryption. This brings all the benefits of the industry standard Secure Sockets Layer Encryption Protocol to the communications between CNS Agent enabled devices and the CNS Configuration Engine.

The CNS Configuration Engine Setup program prompts related to the SSL Encryption features are Shown below

```
...other prompts....
Encryption settings:
--------------------
Enable cryptographic (crypto) operation between Event Gateway(s)/Config
server and device(s) (y/n)? [n] y
Certificates already exist. Overwrite (y/n)? y
Enter certificate FTP server (hostname.domainname or IP address): cert-host.mydomain.com
Enter username used for FTP server: cnsie
Enter FTP password: ******
Re-enter FTP password: ******
Enter absolute pathname of remote key file: /tftpboot/server.key
Enter absolute pathname of remote certificate file: /tftpboot/server.cer
Enabling plaintext operation will increase security risk.
Enable plaintext operation between Config Server and devices/GUI
administration (y/n)? [y] n
Enable plaintext operation between Event Gateway and devices (y/n)? [y] n
Enter port number for https web access: [443]
...other prompts....
```

SSL Host Communication Basics

To take part in SSL based communications, the hosts system should have the following:

- A common PKI Certificate Server (CS) / Certificate Authority (CA) that is trusted to sign and issue Digital Certificates to use with SSL (Trustpoint).
- Hostname
- DNS Server IP Address (Name Resolution)
- DNS Domain Name(Name Resolution)
- Date and Time zone Settings
- NTP Date and Time Updates
- SSL Trust Point Signed SSL Certificates.

Every host that takes part in SSL Communications needs to have an accurate date and time, along with some fundamental Host Name Resolution capabilities. They should have a strong encryption based Operating Systems to enable 128-Bit SSL Export Grade Encrypted communications.

Four Steps to CNS SSL Communication

Using the CNS Agent and CNS Configuration Engine, you can establish a trusted and encrypted communication channel between Cisco IOS devices and the CNS Configuration Engine by using the CNS Agent SSL Encryption feature. You can encrypt the transmission of all IOS Syslog Messages including Firewall and IDS Sensor traffic, Configuration Updates, Statistics Gathering, and Device Inventory over a single 128-Bit SSL Connection.

There are four basic steps to set up the SSL Communications between the CNS Agents and CNS Configuration Engine:

Step 1 Obtain the Server Self-Signed (root) Certificate. In the first part of the deployment the Certificate Authority (CA) must be setup and you should have a Self-Signed Certificate.

Step 2 CNS Configuration Engine SSL Enrollment.

The CNS Configuration Engine needs to enroll and acquire its own unique SSL Certificate from the Cisco IOS CA Server. The CPE device checks the CNS Configuration Engine's SSL Certificate and the Root CA Signature against its own SSL Certificate in order to setup the SSL Connectivity. This enrollment is done over a File Copy or Terminal/Console cut and paste method.

Note In versions 1.3.2 and 1.4 of the CNS Configuration Engine there is no SCEP protocol enrollment client.

Step 3 Set the Cisco IOS Trustpoint.

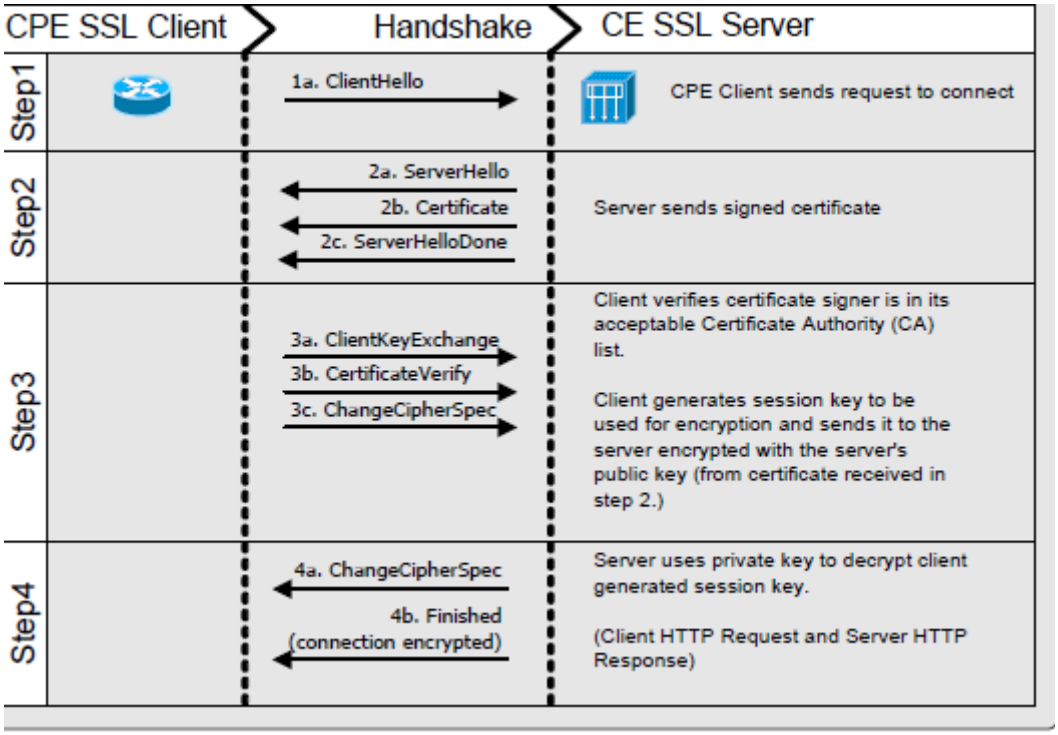
Cisco CNS Agent CPE Devices only need a copy of the Root CA's Certificate self-signed certificate. This is simply known as *Setting the Cisco IOS Trustpoint* as they are not required to enroll for their own individual certificate. You can use the same Root CA's Self-Signed certificate for CNS Configuration Engine Server Certificate validation. This is similar to how PC web browsers work today.

Cisco CPE Devices are designed to primarily utilize the Cisco SCEP Enrollment Protocol to set the SSL Trustpoints or enroll a CPE Device for its own Certificate. The use of the SSL Protocol in Cisco Devices was introduced into Cisco IOS in 1999-2000 and the SCEP Protocol for Certificate Enrollment and related functions was developed in a joint venture by Cisco Systems Inc. and Verisign Inc.

Step 4 Turn on the SSL.

On the CNS Configuration Engine re-run the Setup program to enable Encryption on the Server end. The CNS Agents in the CPE device(s) need to reconfigure the command to use the encrypt keyword.

Figure 7-1 CNS 128-Bit SSL Client & Server HandShake



Running SSL Encrypted Communication

After setting up the encryption on the server end, the Cisco IOS Device and the CNS Configuration Engine are prepared to begin the SSL Encrypted Communications. Any CNS Agent inbound service connection which attempts to initiate the connection with the keyword *encrypt* in its IOS command will be encrypted over SSL.

Cisco IOS v12.3(4)T Certificate Server

Before you configure the CNS Configuration Engine or an SSL Cisco IOS Device getting any certificates or setting the device SSL trustpoints, you should setup the SSL Certificate Server (Trustpoint).

The Cisco IOS Certificate Server supports Simple Certificate Enrollment Protocol (SCEP) over HTTP as its primary Certificate enrollment protocol.

Today, Cisco IOS SSL Client Devices use the SCEP as their primary Certificate Server enrollment protocol to set their SSL Cisco IOS Trustpoints. The CNS Configuration Engine uses a manual enrollment (terminal write) to enroll and obtain its SSL Certificate from the Certificate Server as it does not support the SCEP protocol or other automated enrollment protocols.

Engine Certificate Enrollment IOS Command

For the CNS Configuration Engine Certificate Request, the following syntax on the Cisco IOS CS is required to have it accept the certificate request from a Terminal or Screen dump:

crypto pki server {cs-label} request pkcs10 terminal pem

Cisco IOS Certificate Enrollment IOS Command

For the Cisco IOS CPE Device Certificate request by means of SCEP, use the following command:

crypto pki server {cs-label} request pkcs10 url {url}

The URL is the path that the Cisco IOS CPE Device uses to set its Trustpoint by means of SCEP.

Setting up the Cisco IOS Certificate Server

These are short version of IOS CS setup and configuration that you need to substitute with your own server name and issuer-name values. Further documentation covering the setup of the IOS Certificate Server in detail is available at Cisco.com.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip http server
Router(config)#crypto pki server IOS-CA-10090
Router(cs-server)#issuer-name CN=My Company,L=San Francisco CA,C=us
Router(cs-server)#grant auto
% This will cause all certificate requests to be automatically granted.

Are you sure you want to do this? [yes/no]: yes
Router(cs-server)#no shutdown
% Once you start the server, you can no longer change some of % the configuration.
Are you sure you want to do this? [yes/no]: yes
% Generating 1024 bit RSA keys ...[OK]

Mar 13 02:15:37.029: %SSH-5-ENABLED: SSH 1.99 has been enabled
% Certificate Server enabled.
Router(cs-server)#end
```

Viewing the IOS Certificate Server Self-Signed (root) Certificate

To view the Certificate Authorities own (self-signed) SSL Certificate on your IOS Certificate Server, use the command below. This is the Certificate that will be digitally signed and authenticate all of the SSL Certificates issued by this Certificate Authority.

Show Crypto CA Certificate

The following example shows how the **crypto ca certificate** command is used at the router prompt.

```
Router#show crypto ca certificates
CA Certificate
    Status: Available
    Certificate Serial Number: 01
    Certificate Usage: Signature
    Issuer:
        cn=My Company
        l=San Francisco CA
        c=us
    Subject:
        cn=My Company
        l=San Francisco CA
        c=us
    Validity Date:
        start date: 02:15:39 UTC Mar 13 2004
        end date: 02:15:39 UTC Mar 13 2007
    Associated Trustpoints: IOS-CA-10090
```

Show Crypto PKI Server

The following example shows how the **crypto pki server** command is used at the router prompt.

```
Router#show crypto pki server
Certificate Server IOS-CA-10090:
    Status: enabled, configured
    CA cert fingerprint: 7F1AEE23 9067BD38 97137AE7 24C80C37
    Granting mode is: auto
    Last certificate issued serial number: 0x1
    CA certificate expiration timer: 02:15:39 UTC Mar 13 2007
    CRL NextUpdate timer: 02:15:49 UTC Mar 20 2004
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

IOS Certificate Server Enrollment

This section describes the IOS certificate server enrollment procedure.

Certificate Enrollment using SCEP Protocol

SCEP is the recommended method of Certificate Enrollment for all Cisco IOS Devices. An example of IOS client configuration for SCEP enrollment is shown below:

```
crypto ca identity SSLrootCA
enrollment url http://myIOSCAserver.com
exit
```

Certificate Enrollment using Terminal Copy and Paste

The command line enrollment requires access to the IOS command. The certificate request and reply both will be issued on the console session screen and not saved to any file. The issued results can simply be copied to file as text and saved for use in the CNS Configuration Engine. To obtain a certificate, use the following command:

```
Router#crypto pki server iosca request pkcs10 terminal
<<paste request>>
<<copy certificate>>
```

Sample Terminal Enrollment

The following example shows how to use a certificate request for a CNS Configuration Engine:

Note Your certificate request can have PEM headers and footers.

```
Cisco-1711#crypto pki server IOS-CA-10090 request pkcs10 terminal
% Enter Base64 encoded or PEM formatted PKCS10 enrollment request.
% End with a blank line or "quit" on a line by itself.
-----BEGIN CERTIFICATE
REQUEST----MIIDQDCCAigCAQAwgawxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlh
MREwDwYDVQQHEwhTYW4gSm9zZTEbMBkGA1UEChMSQ2lzY28gU3lzdGVtcyBJbmMu
MSYwJAYDVQQLEx1OTVRHIENOUyBDb25maWd1cmF0aW9uIEVuZ2luZTENMAsGA1UE
AxMEb3B1czEhMB8GCSqGSIb3DQEJARYSZW1pa3VsaWNAY2lzY28uY29tMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA84y2NE0C1GY5nnubkG6u2ZYAxK7h
Qy+9UPSfeVYHE3EWuQNuZljoXOd40+/FTfeOEdhh9fsertl+wNavHXwpngDfhx59
dbRduYekAmL7kPVute1BtVB6yJJ7yWRoGW3pG5PinMPXwNQhSRZHgBVBusWtdsD/
/clo7Mly4F2vqYmSVaE0SHLR1QiJKO+0iOx9iBzGxH8S2BpfdjwBKwO2MzRFgPhV
PgjtMvNDA47DgwfcuSiO6liW1Cv/g3GgzBT/G6sWzf6ah2+kTLQq9XXim5b9J2Hf
IIniAwohSq+kHo8f/cbq+WT/BwLaU163oZXiNKUxeYBYCuzZ7GGMkRHVwQIDAQAB
oE4wGAYJKoZIhvcNAQkHMQsTCWNyaW1zb241NTAyBgkqhkiG9w0BCQIxJRMjTmV0
d29yayBNYW5hZ2VtZW50IFRlY2hub2xvZ3kgR3JvdXAwDQYJKoZIhvcNAQEEBQAD
ggEBABtrhmhJYP3iD5iencZzQr4P8bwDjtYAS4B0EpFAfeymGV5ZSKMM3pLEbac8
iVL2EbLwB1G25O24PiFOmysw60+ehmR4TR1yI3pMZ7oTn71md4a3z04GAw6o9q4u
xPEWx07LckkGdmaZAd5t9+hLrBWTq+gZmZszVkenhBPIEdDh8gN0XKM5xd9IVtio
jZNJVeXPlVxWFT/nIz+BaVolJenuqrB6wgWjMoAfvknr2v9axIf8P6j80vPSX3wU
H+JUNCYTa4hXoX8sK9xeoHpqNyvVMz6ntAgHe+JjJyfRlLxq/hnbcem83lyr2Mix
% Granted certificate:
MIIDBzCCAnCgAwIBAgIBAjANBgkqhkiG9w0BAQQFADA9MQswCQYDVQQGEwJ1czEZ
MBcGA1UEBxMQU2FuIEZyYW5jaXNjbyBDQTETMBEGA1UEAxMKTXkgQ29tcGFueTAe
Fw0wNDAzMTMwMjIzNDRaFw0wNTAzMTMwMjIzNDRaMIGsMQswCQYDVQQGEwJVUzET
MBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxGzAZBgNVBAoT
EkNpc2NvIFN5c3RlbXMgSW5jLjEmMCQGA1UECxMdTk1URyBDTlMgQ29uZmlndXJh
dGlvbiBFbmdpbmUxDTALBgNVBAMTBG9wdXMxITAfBgkqhkiG9w0BCQEWEmVtaWt1
bGljQGNpc2NvLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAPOM
tjRNAtRmOZ57m5BurtmWAMSu4UMvvVD0n3lWBxNxFrkDbmZY6FzneNPvxU33jhHY
YfX7Hq7ZfsDWrx18KZ4A34cefXW0XbmHpAJi+5D1brXtQbVQesiSe8lkaBlt6RuT
4pzD18DUIUkWR4AVQbrFrXbA//3JaOzJcuBdr6mJklWhNEhy0dUIiSjvtIjsfYgc
xsR/EtgaX3Y8ASsDtjM0RYD4VT4I7TLzQwOOw4MH3LkojupYltQr/4NxoMwU/xur
Fs3+modvpEy0KvV14puW/Sdh3yCJ4gMKIUqvpB6PH/3G6vlk/wcC2lNet6GV4jSl
MXmAWArs2exhjJER1cECAwEAAaMjMCEwHwYDVR0jBBgwFoAUUmc3z2kTZuJ3JJAW
BQGz2gYFGuQwDQYJKoZIhvcNAQEEBQADgYEAHOsmp3H2wii8NaoYV8uXsbIYyk5V
KDIpB3EX6G74b6MG0egzH+39HYJT7S7uevyPEbMg1xusJoeRmUGl0GricJcmlPUL
cqqSt+nueOpiZs0W1pwqunqYTkTy3DP1oyxSWA1Xe9sIJQXcPvppj+7KvpIvckCk
RqgVxWG1aPcBTYI=
```

Note if a PEM formated certificate is expected manually add the following PEM headers:

```
-----BEGIN CERTIFICATE-----
<<hex data>>
-----END CERTIFICATE-----

Hence the issued request would end up with headers above and below it's first and last
data lines:
-----BEGIN CERTIFICATE----MIIDBzCCAnCgAwIBAgIBAjANBgkqhkiG9w0BAQQFADA9MQswCQYDVQQGEwJ1czEZ
MBcGA1UEBxMQU2FuIEZyYW5jaXNjbyBDQTETMBEGA1UEAxMKTXkgQ29tcGFueTAe
Fw0wNDAzMTMwMjIzNDRaFw0wNTAzMTMwMjIzNDRaMIGsMQswCQYDVQQGEwJVUzET
MBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxGzAZBgNVBAoT
EkNpc2NvIFN5c3RlbXMgSW5jLjEmMCQGA1UECxMdTk1URyBDTlMgQ29uZmlndXJh
dGlvbiBFbmdpbmUxDTALBgNVBAMTBG9wdXMxITAfBgkqhkiG9w0BCQEWEmVtaWt1
bGljQGNpc2NvLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAPOM
tjRNAtRmOZ57m5BurtmWAMSu4UMvvVD0n3lWBxNxFrkDbmZY6FzneNPvxU33jhHY
YfX7Hq7ZfsDWrx18KZ4A34cefXW0XbmHpAJi+5D1brXtQbVQesiSe8lkaBlt6RuT
4pzD18DUIUkWR4AVQbrFrXbA//3JaOzJcuBdr6mJklWhNEhy0dUIiSjvtIjsfYgc
xsR/EtgaX3Y8ASsDtjM0RYD4VT4I7TLzQwOOw4MH3LkojupYltQr/4NxoMwU/xur
Fs3+modvpEy0KvV14puW/Sdh3yCJ4gMKIUqvpB6PH/3G6vlk/wcC2lNet6GV4jS1
MXmAWArs2exhjJER1cECAwEAAaMjMCEwHwYDVR0jBBgwFoAUUmc3z2kTZuJ3JJAW
BQGz2gYFGuQwDQYJKoZIhvcNAQEEBQADgYEAHOsmp3H2wii8NaoYV8uXsbIYyk5V
KDIpB3EX6G74b6MG0egzH+39HYJT7S7uevyPEbMg1xusJoeRmUGl0GricJcmlPUL
cqqSt+nueOpiZs0W1pwqunqYTkTy3DP1oyxSWA1Xe9sIJQXcPvppj+7KvpIvckCk
RqgVxWG1aPcBTYI=
-----END CERTIFICATE----
```

Sample Commands and Output

The section provides the sample commands and the output.

Crypto Key and SSL Certificate Request Creation

The following steps will describe you the basics of RSA Key generation and SSL Certificate Request Generation on the CNS CE. The generated certificate is referred to as the *Certificate Signing Request* and is used to apply for a valid signed SSL Certificate from your Certificate Authority in return.

Logon to the Console or Terminal of your CNS Configuration Engine and enter the commands in the order they are presented here.

Note You have to substitute the sample filenames with your own actual filenames as required. The output listed here is actual OpenSSL output created in a test environment and your actual output data can vary slightly. In the following text, both the command input and any output generated is listed as captured on the screen. Make sure that the CNS Configuration Engine is installed and setup with the network TCP/IP connectivity.

Generating RSA Keys

Log into CNS Configuration Engine Console or Terminal, enter the following commands in order to generate an RSA Keypair and a Certificate Signing Request:

- **% openssl genrsa -out /root/server.key 1024**

```
Generating RSA private key, 1024 bit long modulus
.................++++++
.....................................................++++++
e is 65537 (0x10001)
```

Changing the File Ownership rights of the RSA keys

To change the files ownership rights of the RSA keys, use the following command.

- **% chown -v root:root /root/server.key**

```
changed ownership of `/root/server.key' to root:root
```

- % chmod -v 400 /root/server.key

```
mode of `/root/server.key' changed to 0400 (r--------)
```

Generating an SSL Certificate Signing Request

To generate an SSL Certificate Signing request, use the following command.

- **% openssl req -new -key /root/server.key -out /root/server.csr**

```
Using configuration from /usr/share/ssl/openssl.cnfer.key -out /root/server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Berkshire]:California
Locality Name (eg, city) [Newbury]:San Jose
Organization Name (eg, company) [My Company Ltd]:Cisco Systems Inc.
Organizational Unit Name (eg, section) []:Network Management Technology Group
Common Name (eg, your name or your server's hostname) []:opus
Email Address []:administrator@cisco.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:cisco123
An optional company name []:.
```

After you complete the above steps, create a v2.0 Privacy Enhanced Mail (PEM) formatted SSL Certificate Signing Request on the local CNS Configuration Engine file system. The SSL Certificate should be signed by your Certificate Server/Authority. The *server.csr* file can now be transferred to the certificate authority for signing.

The following example shows the unformatted file.

"-----BEGIN CERTIFICATE REQUEST-----" and "-----END

CERTIFICATE REQUEST-----" header and footer text on a line each by themselves.

```
[root@opus root]# more /root/server.csr
-----BEGIN CERTIFICATE
REQUEST----MIICETCCAXoCAQAwgbcxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlh
MREwDwYDVQQHEwhTYW4gSm9zZTEbMBkGA1UEChMSQ2lzY28gU3lzdGVtcyBJbmMu
MSwwKgYDVQQLEyNOZXR3b3JrIE1hbmFnZW1lbnQgVGVjaG5vbG9neSBHcm91cDEN
MAsGA1UEAxMEb3B1czEmMCQGCSqGSIb3DQEJARYXYWRtaW5pc3RyYXRvckBjaXNj
by5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALSbTB5cijXDFzGmGxDK
z5FpR1I8PfzJI/2lEfOUzKwffzbUmKESLOTTQ3g1Y7Qbh7lZBU1rVsc4I1pwHyKu
JOGNhG8wJCUau3ErmhExEM/OunD6zXU1VT/CSvMzG2e6l5JHEBIuZyL/LWEiaA+0
+7NiqI/xgsYPAUVdheEOqgkdAgMBAAGgGTAXBgkqhkiG9w0BCQcxChMIY2lzY28x
MjMwDQYJKoZIhvcNAQEEBQADgYEAQfwCg/fceFdy/xgpps6GSkRt8EB6gsMwNv2E
Cp+FQR0CK9NcpNwNezevbhqpNoaVhsmXgfbAw8mVxJWLJeLe1Bhf9GBXPwIttqLJ
IyfNZfagXMkW+S9z53MnPXg49RaT07itYkqe/1h6RV4TeHYjhPkHGuFeb9GsKM4X
B35lEeo=
-----END CERTIFICATE REQUEST----
```

Certificate Request Digitally Signed and Issued

Copy the SSL Certificate Request *server.csr* file to your Certificate Server and submit it for signing by the CS/CA. The CS/CA Administrator will verify the CNS Configuration Engine Certificate Request with the CS/CA's Root Certificate according to their policy and return the signed/valid SSL Certificate to you as a '*server.cer*' file in a Privacy Enhanced Mail (PEM) format. After you receive the certificate, copy or place the signed Certificate in you preferred directory location on the CNS Configuration Engine or on an external FTP Server that has IP FTP Client connectivity access.

Viewing the CNS Configuration Engine Certificate Contents

To view the CNS Configuration Engine Certificate contents, run the following command on the CNS Configuration Engine console (as root). In the output, you should look at the serial: number:

```
[root@nugi root]# openssl x509 -noout -text -in /etc/tibgate/server.crt
......other data.......
X509v3 Authority Key Identifier:
keyid:28:B6:86:CF:E5:52:C9:8C:23:BA:C2:A2:A0:22:F1:DA:5E:77:53:30
DirName:/Email=administrator@mycompany.com/C=US/ST=CA/L=San Jose/O=Company Co
Inc./OU=Dept/CN=Personnel
serial:4D:00:F1:83:1F:8D:56:AC:4F:63:BF:0A:CA:AB:4F:00
.....other data.......
```

Preview the Issued SSL Certificate

After you copy the CNS Configuration Engine Certificate onto the CNS Configuration Engine, to preview the signed contents of the CNS Configuration Engine Certificate, use the following OpenSSL command on the Certificate server.cer file.

root@opus root]# openssl x509 -noout -text -in /root/server.cer

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=us, L=San Francisco CA, CN=My Company
        Validity
            Not Before: Mar 13 02:23:44 2004 GMT
            Not After : Mar 13 02:23:44 2005 GMT
        Subject: C=US, ST=California, L=San Jose, O=My Company., OU=Personnel Dept,
    CN=myhostname/Email=administrator@company.com
    co.com
        Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
        RSA Public Key: (2048 bit)
            Modulus (2048 bit):
                00:f3:8c:b6:34:4d:02:d4:66:39:9e:7b:9b:90:6e:
                ae:d9:96:00:c4:ae:e1:43:2f:bd:50:f4:9f:79:56:
                07:13:71:16:b9:03:6e:66:58:e8:5c:e7:78:d3:ef:
                c5:4d:f7:8e:11:d8:61:f5:fb:1e:ae:d9:7e:c0:d6:
                af:1d:7c:29:9e:00:df:87:1e:7d:75:b4:5d:b9:87:
                a4:02:62:fb:90:f5:6e:b5:ed:41:b5:50:7a:c8:92:
                7b:c9:64:68:19:6d:e9:1b:93:e2:9c:c3:d7:c0:d4:
                21:49:16:47:80:15:41:ba:c5:ad:76:c0:ff:fd:c9:
                68:ec:c9:72:e0:5d:af:a9:89:92:55:a1:34:48:72:
                1:d5:08:89:28:ef:b4:88:ec:7d:88:1c:c6:c4:7f:
                12:d8:1a:5f:76:3c:01:2b:03:b6:33:34:45:80:f8:
                55:3e:08:ed:32:f3:43:03:8e:c3:83:07:dc:b9:28:
                8e:ea:58:96:d4:2b:ff:83:71:a0:cc:14:ff:1b:ab:
            16:cd:fe:9a:87:6f:a4:4c:b4:2a:f5:75:e2:9b:96:
```

```
            fd:27:61:df:20:89:e2:03:0a:21:4a:af:a4:1e:8f:
            1f:fd:c6:ea:f9:64:ff:07:02:da:53:5e:b7:a1:95:
            e2:34:a5:31:79:80:58:0a:ec:d9:ec:61:8c:91:11:
            d5:c1
        Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Authority Key Identifier:
        keyid:52:67:37:CF:69:13:66:E2:77:24:90:16:05:01:B3:DA:06:05:1A:E4
Signature Algorithm: md5WithRSAEncryption
    1c:eb:26:a7:71:f6:c2:28:bc:35:aa:18:57:cb:97:b1:b2:18:
    ca:4e:55:28:32:29:07:71:17:e8:6e:f8:6f:a3:06:d1:e8:33:
    1f:ed:fd:1d:82:53:ed:2e:ee:7a:fc:8f:11:b3:20:d7:1b:ac:
    26:87:91:99:41:a5:d0:6a:e2:70:97:26:94:f5:0b:72:aa:92:
    b7:e9:ee:78:ea:62:66:cd:16:d6:9c:2a:ba:7a:98:4e:44:f2:
    dc:33:f5:a3:2c:52:58:0d:57:7b:db:08:25:05:dc:3e:fa:69:
    8f:ee:ca:be:92:2f:72:40:a4:46:a8:15:c5:61:b5:68:f7:01:
    4d:82
[root@opus root]#
```

CNS Configuration Engine Certificate Request

This is an example of keys and a certificate request on a test CNS Configuration Engine. You can create a certificate request on your CNS Configuration Engine (*.csr) and the CA signs the certificate and sends you back a valid certificate (*.cer).

```
[root@opus root]# openssl req -new -key /root/server.key -out /root/my-cnsce.csr
Using configuration from /usr/share/ssl/openssl.cnf
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [GB]:us
State or Province Name (full name) [Berkshire]:California
Locality Name (eg, city) [Newbury]:San Francisco
Organization Name (eg, company) [My Company Ltd]:Company Co Inc.
Organizational Unit Name (eg, section) []:Department
Common Name (eg, your name or your server's hostname) []:my-cnsce
Email Address []:administrator@company.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:2FCE2F9EF109E
An optional company name []:Company Co Inc.
[root@nugi root]#
```

Enabling SSL in the Configuration Engine

Log into the CNS Configuration Engine Console or Terminal as root, run the CNS Configuration Engine Setup program a nd entering the command Setup. This allows you to make changes to the CNS Configuration Engine Setup encryption settings, and only apply those changes you have made. In this case you will be enabling SSL on that host and identifying the key and certificate locations.

Sample CNS Configuration Engine Setup for SSL

The following example shows the screen from my SSL enabled CNS Configuration Engine Setup:

```
======CNS Configuration Engine SETUP=====================

Please review the following parameters:
username for user-level shell account: admin
password for user-level shell account:
eth0 IP address: 10.1.2.8
eth0 network mask: 255.255.255.0
eth0 default gateway IP address: 10.1.2.7 eth1 IP address:
primary DNS server IP address: 10.1.2.3
secondary DNS server IP address (optional):
Configuration Engine login name: gui-admin
Configuration Engine login password: ******
internal LDAP server password: *********
Enable cryptographic (crypto) operation between Event Gateway(s)/Config server and
device(s) (y/n)? yes
Certificates already exist. Overwrite (y/n)? no
Enable plaintext operation between Config Server and devices/GUI administration (y/n)? no
Enable plaintext operation between Event Gateway and devices (y/n)? no
Enable authentication (y/n)? no
NSM directive (none, default, http): default
Enable Event Gateway debug log (y/n)?no
log file rotation timer (minutes, 0 = no rotation): 15
max log file size (Kbytes): 3072
the max versions of log file (0-99): 1
number of Event Gateways that will be started with crypto operation: 1
number of Event Gateways that will be started with plaintext operation: 1
CNS Event Bus Network Parameter: 10.1.25.18
CNS Event Bus Service Parameter: 7500
Re-configure IMGW (y/n)? no

 Warning: setup cannot be aborted while committing changes.

Commit changes (y/n):yes
```

Cisco IOS SSL

This section describes how to set the Cisco IOS SSL.

Setting the Cisco IOS SSL Trustpoint

In Cisco IOS you can set the trustpoint over a network by using the SCEP, or you can screen dump it by using the 'terminal' option. To paste it over a terminal, it needs to be in a specific format (encoded). It should be in Base-64 encoded format for the terminal entry method.

CPE SSL Trustpoint Using SCEP

The following example shows how to obtain the Trustpoint by using the SCEP Protocol method.

```
!
Router(config)#crypto ca trustpoint company.com
Router(config)#enrollment mode ra
Router(config)#enrollment url http://my-iosca:80/
Router(config)#usage ssl-client
Router(config)#revocation-check none
Router(config)#crypto ca authenticate att.com
```

```
Certificate has the following attributes:
Fingerprint: 1D74D54A 464207FD 817901A4D 67B5112B
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
!
crypto ca certificate chain company.com
certificate ca 4D69F1831F8D56AF4F63BF0F1B6B4F9B
308202F2 3082029C A0030201 0202104D 69F1821F 8D56AC4F 63BF0ACA AB4F9F30
0D06092A 864886F7 0D010105 05003081 97312139 1F06092A 864886F7 0D010901
1612656D 696B756C 69634063 6973636E 2E636F6D 310B3009 06035504 06130255
53310B30 09060355 04081302 43413111 300F0603 55040713 084D696C 70697461
73311B30 19060355 040A1312 43697365 6F205379 7374656D 7320496E 632E3113
30110603 55040B13 0A434E53 2D494E77 4D425531 13301106 03550403 130A494E
534D4255 2D434131 301E170D 30333035 32383232 34333533 5F970D30 35303532
38323235 3231325A 30819731 21301F06 092A8648 86F70D01 09011612 656D696B
756C6963 40636973 636F2E63 6F6D310B 30090603 55040613 02123331 0B300906
03550408 13024411 3111300F 06035504 0713084D 696C7069 74617331 1B301906
0355040A 13124369 73636F20 53797374 656D7320 496E632E 31133011 06035504
0B130A43 4E532D49 4E534D42 55311330 11060355 0403130A 494E534D 42552D43
4131305C 300D0609 2A864886 F70D0101 01050003 4B003048 02410097 2E2CED5E
B0F1194F 572E3275 F54F8C92 1A489BD4 CAD7701B 2275EA6B 3520DC29 7D17A1C4
AC6EB48C B03B1AF1 8CDAB68A A31FB15A 55325A45 032106E6 8214B902 03010001
A381C130 81BE300B 0603551D 0F040403 0201C630 0F060355 1D130101 FF040530
030101FF 301D0603 551D0E04 16041428 B686CFE5 52C98C23 BAC2A2A0 22F1DA5E
77533030 6D060355 1D1F0466 3064302F A02DA02B 86296874 74703A2F 2F67696C
6C696761 6E2F4365 7274456E 726F6C6C 2F494E53 4D42552D 4341312E 63726C30
31A02FA0 2D862B66 696C653A 2F2F5C5C 67696C6C 6967616E 5C436572 74456E72
6F6C6C5C 494E534D 42552D43 41312E63 726C3010 06092B06 01040182 37150104
03020100 300D0609 2A864886 F70D0101 05050003 41012C82 FA96ED42 329B01B7
83A48A06 4CEDFE36 2CF9706D 1ED1B7C8 882B0B4C 039A4E15 14B410FC 4258FFED
C8069002 76446090 CA22B642 B4207869 F495F8F2 BA51
quit
```

CPE SSL Trustpoint Using Terminal Copy/Paste

The following example shows how to set the trustpoint over the console by using a terminal session.

```
!
Router(config)#crypto ca trustpoint company.com
Router(config)#enrollment mode ra
Router(config)#enrollment terminal
Router(config)#usage ssl-client
Router(config)#revocation-check none
Router(config)#
!
Router(config)#crypto ca authenticate company.com
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE----MIIC8jCCApygAwIBAgIQTWnxgx+
NVqxPY78KyqtPnzANBgkqhkiG9w0BAQUFADCB
lzEhMB8GCSqGSIb3DQEJARYSZW1pa3VsaWNAY2lzY28uY29tMQswCQYDVQQGEwJV
UzELMAkGA1UECBMCQ0ExETAPBgNVBAcTCE1pbHBpdGFzMRswGQYDVQQKExJDaXNj
byBTeXN0ZW1zIEluYy4xEzARBgNVBAsTCkNOUy1JTlNNQlUxEzARBgNVBAMTCklO
U01CVS1DQTEwHhcNMDMwNTI4MjI0MzUzWhcNMDUwNTI4MjI1MjEyWjCBlzEhMB8G
CSqGSIb3DQEJARYSZW1pa3VsaWNAY2lzY28uY29tMQswCQYDVQQGEwJVUzELMAkG
A1UECBMCQ0ExETAPBgNVBAcTCE1pbHBpdGFzMRswGQYDVQQKExJDaXNjbyBTeXN0
ZW1zIEluYy4xEzARBgXMAAsTCkNOUy1JTlNNQlUxEzARBgNVBAMTCklOU01CVS1D
QTEwXDANBgkqhkiG9w0BAQEFAANLADBIAkEAly4s7V6w8RlPVy4ydfVPjJIaSJvU
ytdwGyJ16ms1INwpfRehxKxutIywOxrxjNq2iqMfsVpVMlpFAyEG5oIUuQIDAQAB
o4HBMIG+MAsGA1UdDwQEAwIBxjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBQo
tobP5VLJjCO6wqKgIvHaXndTMDBtBgNVHR8EZjBkMC+gLaArhilodHRwOi8vZ2ls
bGlnYW4vQ2VydEVucm9sbC9JTlNNQlUtQ0ExLmNybDAxoC+gLYYrZmlsZTovL1xc
```

```
Z2lsbGlnYW5cQ2VydEVucm9sbFxJTlNNQlUtQ0ExLmNybDAQBgkrBgEEAYI3FQEE
AwIBADANBgkqhkiG9w0BAQUFAANBACyC+pbtQjKbAbeDpIoGTO3+Niz5cG0e0bfI
iCsLTAOaThUUtBD8Qlj/7cgGkAJ2RGCQyiK2QrQgeGn0lfjyukE=
-----END CERTIFICATE----quit
Certificate has the following attributes:
Fingerprint: 1D74D54A B64207FD 81831A4D 1EDF56194
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
% Certificate successfully imported
```

Show Crypto CA Trustpoint IOS Command

The following example shows how to get the following output, and the key field in this example is the Serial Number.

```
Router#show crypto ca trustpoints
Trustpoint company-IOS-CA:
Subject Name: mytrustpoint
CN = Department
OU = Personnel
O = Company Co Inc.
L = San Francisco
ST = CA
C = US
EA = administrator@company.com
Serial Number: 4D69F1831F8Ef1344F63BF0ACAAB4F9F
Certificate configured.
CEP URL: <http://my-iosca>
```

OpenSSL Certificate Formats

This section describes the OpenSSL Certificate formats.

Certificate and Key Formats

The certificate and key formats are:

- PEM
- DER
- PKCS#12

PEM

This is the default format used by OpenSSL and the only format usable by CNS Configuration Engine v1.4 and earlier. This format can contain all the private keys (RSA and DSA), public keys (RSA and DSA) and (x509) certificates. PEM stores the data in Base64 encoded DER format, surrounded by ascii headers, so this format is suitable for text mode transfers between systems.

DER

DER format can contain all the private keys, public keys and certificates. DER stores according to the ASN1 DER format. It is the default format for most browsers.

PKCS#12

PKCS#12 is also known as PFX files. They can contain all the private keys, public keys and certificates.It stores the data in a binary format.

For more information see http://www.drh-consultancy.demon.co.uk/pkcs12faq.html/.

The format of a X509 PEM certificate is:

(Header Info)

```
------BEGIN (TRUSTED|X509) CERTIFICATE----(
Certificate Data)
------END (TRUSTED|X509) CERTIFICATE---
```

Converting Certificate Formats with OpenSSL

This section describes the procedure to convert the certificate formats with the OpenSSL tools. For more information refer the OpenSSL documentation.

OpenSSL To PKCS#12

The following example shows how to convert OpenSSL to PKCS#12.

```
openssl pkcs12 -export -in pem-certificate-and-key-file -out
pkcs-12-certificate-and-key-file
openssl pkcs12 -export -in pem-certificate-file -inkey pem-key-file -out
pkcs-12-certificate-and-key-file
```

OpenSSL From PKCS#12 to PEM

The following example shows how to convert OpenSSL from PKCS#12 to PEM.

```
openssl pkcs12 -in pkcs-12-certificate-file -out pem-certificate-file
openssl pkcs12 -in pkcs-12-certificate-and-key-file -out pem-certificate-and-key-file
```

OpenSSL From PEM/DER to DER/PEM

The following example shows how to convert OpenSSL from PEM/DER to DER/PEMPKCS#12.

```
openssl dsa -inform PEM|DER -outform DER|PEM -in pem-file|der-file -out der-file|pem-file
```

OpenSSL Certificate Formats

The following example shows the OpenSSL certificate formats.

```
OpenSSL From PEM/DER to DER/PEM - RSA Keys
openssl rsa -inform PEM|DER -outform DER|PEM -in pem-file|der-file -out der-file|pem-file
```

Troubleshooting CNS SSL Communications

To view the issued CNS Config Engine certificate contents n the CNS Configuration Engine, run the following OpenSSL command. This sample is taken from a real certificate issued by a IOS CS/CA in our lab. The key value to note here is the “Serial Number:” hex string, which denotes the serial number of the issued certificate.

Viewing the SSL Certificate

The following example shows how to view the SSL certificate.

```
openssl x509 -noout -text -in /etc/tibgate/server.crt
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            15:79:ce:3e:00:00:ef:00:00:04
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: Email=administrator@mycompany.com, C=US, ST=CA, L=San Francisco, O=Company
Co Inc., OU=Department, CN=Personnel
        Validity
            Not Before: May 30 01:52:27 2003 GMT
            Not After : May 30 02:02:27 2004 GMT
        Subject: Email=administrator@mycompany.com, C=us, ST=California, L=San Jose,
O=Company Inc., OU=Department, CN=config-engine
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
        RSA Public Key: (1024 bit)
            Modulus (1024 bit):
                00:c8:5a:71:55:f8:30:21:da:ef:f1:6f:5c:e5:df:
                92:66:be:d2:7f:86:65:e7:e1:de:f4:c2:ac:1e:e1:
                e9:7a:a2:64:20:81:ed:a6:ff:f8:85:ab:fc:63:0f:
                d3:71:93:b1:6b:31:f5:0b:11:64:c1:dc:29:88:7f:
                ab:81:69:bf:f0:81:5c:af:1b:86:9d:14:30:47:fd:
                44:04:ea:3e:e6:e0:2b:7d:33:d4:37:ba:a9:ba:ee:
                29:2f:52:a9:f3:e2:26:60:5d:c7:6d:25:92:80:fe:
                16:07:f8:c9:2d:75:6f:29:4c:17:3c:85:70:ad:c1:
                65:aa:ea:c5:e0:09:47:24:e1
            Exponent: 65537 (0x10001)
    X509v3 extensions:
    X509v3 Subject Key Identifier:
        35:7E:67:7C:B9:AC:79:ED:34:CB:08:DF:AB:1E:C6:0D:FC:41:3B:71
    X509v3 Authority Key Identifier:
        keyid:28:B6:86:CF:E5:52:C9:8C:23:BA:C2:A2:A0:22:F1:DA:5E:77:53:30
        DirName:/Email=administrator@mycompany.com/C=US/ST=CA/L=San Jose/O=Company
Co Inc./OU=Department/CN=Marketing
        serial:4D:69:F1:1F:EF:8E:56:AC:4F:63:BF:0A:CA:AB:4F:9F
    X509v3 CRL Distribution Points:
        URI:http://my-iosca/
        URI:file://\\my-iosca
    Authority Information Access:
        CA Issuers - URI:http://my-iosca
        CA Issuers - URI:file://\\my-iosca
Signature Algorithm: sha1WithRSAEncryption
    24:d7:86:57:95:78:08:60:8d:88:ab:6b:46:76:bc:45:ce:59:
    6c:af:29:43:17:22:a1:78:d0:65:8a:11:79:ef:6b:15:84:8b:
    bf:40:de:9a:08:81:8c:da:ea:e1:0c:fb:bb:0c:8d:96:74:31:
    30:a0:12:de:19:ca:1b:24:60:0d
```

Debug Dump of SSL Transactions

To view the **debug** output of the SSL Transactions, use the following commands. The SSL Dump program is part of OpenSSL toolkit and can help in the process of SSL trustpoint setting and sharing.

```
/opt/CSCOcnsie/tools/ssldump -A -e -N -d -i eth0 port 443
..or more simply..
/opt/CSCOcnsie/tools/ssldump -i eth0 port 443
/opt/CSCOcnsie/tools/ssldump -i eth0 port 11012
..and the operator can easily pipe these to a file as such:
```

```
/opt/CSCOcnsie/tools/ssldump -i eth0 port 443 > ssl_http.log
/opt/CSCOcnsie/tools/ssldump -i eth0 port 11012 > ssl_event.log
```

IOS SSL Device Troubleshooting

The following debug and show commands are available on most of the IOS CPE's that support the SSL Security Layer. These commands can provided the operator all the information they may need to help debug a SSL Client trustpoint setup in the CPE device.

PKI Debug Commands

The following example shows the PKI debug commands.

```
debug crypto pki transactions
debug crypto pki messages
```

SSL Debug Commands

The following example shows the SSL debug commands.

```
debug ssl traffic
debug ssl error
debug ssh hdshake
```

Show Crypto Commands

The following example shows the crypto commands.

```
show crypto key pubkey-chain rsa
show crypto ca trustpoints
show crypto ca certificate
```

Show Crypto Key Pubkey-Chain RSA Command

The following example shows the output of what a Trustpoint certificate public key looks like in the IOS Device:

```
Router#show crypto key pubkey-chain rsa
Codes: M - Manually configured, C - Extracted from certificate

Code     Usage     IP-Address/VRF         Keyring           Name
C       Signing                           default           X.500 DN name:
                              CN = IOS-CA1
                              OU = Marketing
                              O = Company Co Inc.
                              L = San Jose
                              ST = CA
                              C = US
                              EA = administrator@company.com
```

Show Crytpo CA Trustpoint

The following example shows the output of a IOS SSL Trustpoint certificate contents and signature looks like in the IOS Device:

```
Router#show crypto ca trustpoints
Trustpoint cisco-ssl-home:
    Subject Name:
    CN = IOS-CA1
    OU = Marketing
    O = Company Co Inc.
    L = San Jose
    ST = CA
    C = US
    EA = administrator@company.com
        Serial Number: 4D69F1E1D5F8D6AC4F63BF0ACAAB4F9F
Certificate configured.
CEP URL: http://my-iosca
```

Show Crypto CA Certificate

The following example shows the output of the IOS "show crypto ca cert" contents look like in the IOS Device:

```
Router#show crypto ca cert
CA Certificate
    Status: Available
    Certificate Serial Number: 2D19F1841F8D56AC4F63BF0AC1DB4F9F
Certificate Usage: Signature
Issuer:
    CN = IOS-CA1
    OU = Marketing
    O = Company Co Inc.
    L = San Jose
    ST = CA
    C = US
    EA = administrator@company.com
Subject:
    CN = IOS-CA1
    OU = Marketing
    O = Company Co Inc.
    L = San Jose
    ST = CA
    C = US
    EA = administrator@company.com
CRL Distribution Point:
    http://my-iosca
Validity Date:
    start date: 22:43:53 UTC May 28 2003
    end date: 22:52:12 UTC May 28 2005
Associated Trustpoints: ssl-home
```

CNS ID Syntax

The following example shows the CNS ID Syntax.

```
Router#enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#cns id ?
    Async          Async interface
    BVI            Bridge-Group Virtual Interface
    CDMA-Ix        CDMA Ix interface
    CTunnel        CTunnel interface
    Dialer         Dialer interface
    Ethernet       IEEE 802.3
    FastEthernet   FastEthernet IEEE 802.3
    Group-Async    Async Group interface
    Lex            Lex interface
    Loopback       Loopback interface
    MFR            Multilink Frame Relay bundle interface
    Multilink      Multilink-group interface
    Tunnel         Tunnel interface
    Vif            PGM Multicast Host interface
    Virtual-PPP Virtual PPP interface
    Virtual-Template Virtual Template interface
    Virtual-TokenRing Virtual TokenRing
    Vlan Catalyst Vlans
    hardware-serial Use hardware serial number as unique ID
    hostname Use hostname as unique ID
    string Use an arbitrary string as the unique ID
```

CNS ID Network Interface Value Lookups

The following example shows the output of the CNS ID network interface value:

```
Router#enable
Router#configure terminal
Router#(config)#cns id FastEthernet 0 ?
ipaddress Use IP address as unique ID
mac-address Use MAC address as unique ID

Router(config)#cns id Async 1 ?
ipaddress Use IP address as unique ID
mac-address Use MAC address as unique ID
```

CNS ID Hardware Serial Number

The following example shows the output of the CNS ID hardware serial number:

```
Router(config)#cns id hardware-serial ?
event Set this ID as the event ID
image Set this ID as the image ID
<cr>
```

Viewing the Motherboard Hardware Serial Number

The following example shows the output of the motherboard hardware serial number:

```
Router#enable
Router#config terminal
Router(config)#do show version
Cisco IOS Software, C1700 Software (C1700-ADVENTERPRISEK9-M), Experimental Version 12.3
Copyright (c) 1986-2004 by Cisco Systems, Inc.
Compiled Fri 13-Feb-04 20:04 by ntimms
ROM: System Bootstrap, Version 12.2(7r)XM4, RELEASE SOFTWARE (fc1)
cisco-1711 uptime is 2 days, 19 hours, 27 minutes
System returned to ROM by reload
System restarted at 07:51:21 UTC Thu Mar 4 2004
System image file is "flash:c1700-adventerprisek9-mz.24.Feb.2004"
This product contains cryptographic features and is subject to United States and local
country laws governing import, export, transfer and use. Delivery of Cisco cryptographic
products does not imply third-party authority to import, export, distribute or use
encryption. Importers, exporters, distributors and users are responsible for compliance
with U.S. and local country laws. By using this product you agree to comply with
applicable laws and regulations. If you are unable to comply with U.S. and local laws,
return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to export@cisco.com.

Cisco 1711 (MPC862P) processor (revision 0x100) with 116939K/14133K bytes of memory.
Processor board ID FOC07271A6Q (2119579075), with hardware revision 0000
MPC862P processor: part number 7, mask 0
1 Ethernet interface
5 FastEthernet interfaces
1 Serial interface
1 terminal line
1 Virtual Private Network (VPN) Module
32K bytes of NVRAM.
32768K bytes of processor board System flash (Read/Write)
Configuration register is 0x2102
```

View the CNS Image ID to Hardware-Serial

The following example shows the output of the CNS Image to hardware-serial:

```
Router#enable
Router#configure terminal
Router(config)#cns id hardware-serial image
Router(config)#do show cns image status
CNS Image Agent ID: FOC07271A6Q
Number of failed upgrades: 0
Number of successful upgrades: 0
Messages received: 8
Receive errors: 1
Bad XML format:1       Not Supported:0    Invalid Parameter:0
Memory exhausted:0     File too large:0   Operation failed:0
File Errors:0          Auth Errors: 0
Transmit Status
TX Attempts:7
Successes:6 Failures 3
Detailed Failures
Memory exhausted:0     queue error:0      external error:0
other error:3
```

CNS Configuration ID to Hardware-Serial

The following example shows the output of the CNS configuration id to hardware-serial:

```
Router#enable
Router#configure terminal
Router(config)#cns id hardware-serial
Router(config)#do show cns config connections
The partial configuration agent is enabled.
Configuration server: 10.1.25.94
Port number: 80
Encryption: disabled
Config id: FOC07271A6Q
Connection Status:
The initial configuration agent is not running.
```

Note The hardware-serial ID is what is listed in the 'show version' command and is retrieved internally by the CNS Agents from the device's motherboard. This may or may not be the same alpha-numeric string as the serial-number printed on the chassis of your particular Cisco IOS Device.

Additional Information Sources

You can refer the following documents for additional information.

Cisco IOS Certificate Server

For more information on Cisco IOS certificate server, see the document at:

http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801d1cb0.html.

Certificate Server Data Sheet

For more information on certificate server data sheet, see the technical document at:

http://www.cisco.com/en/US/tech/tk583/tk372/tech_brief09186a00801e05dc.html.

Cisco IOS PKI

Cisco IOS Software Releases 12.3 T / Security Commands

For more information on Cisco IOS software releases and security commands, see the command reference document at:
http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_command_reference_chapter09186a00801a7f81.html.

SSL Public Key Infrastructure

For more information on SSL public key infrastructure, see the white paper at:

http://www.cisco.com/en/US/tech/tk583/tk618/technologies_white_paper09186a0080179739.shtml.

Certificate Security Attribute-Based Access Control

For more information on certificate security attribute-based access control, see the document at:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00801541ce.html.

Cisco IOS Certificate Server Data Sheet

For more information on Cisco IOS certificate server data sheet, see the document at

http://www.cisco.com/en/US/tech/tk583/tk372/tech_brief09186a00801e05dc.html.

http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/certs_ds.pdf.

Cisco IOS Certificate Server and Software Releases 12.3 T

For more information on Cisco IOS server and software releases 12.4, see the document at

http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801d1cb0.html.

Trusted Root Certification Authority

For more information on trusted rooted certificate authority, see the document at:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1834/products_feature_guide09186a008007fecf.html.

Online Certificate Status Protocol

For more information on online certificate status protocol, see the document at:

http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801a755b.html.

Cisco's SCEP Home Page

For more information on SCEP home page, refer the document at:

http://www.cisco.com/warp/public/cc/pd/sqsw/tech/scep_wp.html.

CISCO IOS Software Releases 12.3 T

For more information on Cisco IOS software releases 12.3 T and RSA Key pair, see the document at:

http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801d1cb4.html.

Trustpoint Command

For more information on trustpoint commands, see the document at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/fttrust.htm.

Certificate Enrollment Enhancements

For more information on certificate enrollment enhancements, see the document at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/ftenrol2.htm.

Configuring Crypto Maps

For more information on configuring crypto maps for DN-based access control, see the document at:

http://www.cisco.com/warp/public/471/vpn_dn.html - tools

Multiple RSA Key Pair Support

For more information on multiple RSA key pair support, see the document at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/ftmltkey.htm -42193.

Simple Certificate Enrollment Protocol:

For more information on simple certificate enrollment protocol, see the document at:

http://www.cisco.com/warp/public/cc/pd/sqsw/tech/scep_wp.htm.

Public Domain OpenSSL

For more information on OpenSSL, see *http://www.openssl.org.*

OpenSSL certificates

For more information on OpenSSL certificate, see *http://www.openssl.org/docs/HOWTO/certificates.txt.*