# Generating and Installing SSL Certificates on the Cisco ISA500

This application note describes how to generate and install SSL certificates on the Cisco ISA500 security appliance. It includes the following topics:


- Certificate Overview
- Generating a Certification Authority and Root Certificate
- Generating a Certificate Signing Request and Installing a Signed Certificate
- Installing a Self-Signed Certificate on the ISA500
- Activating and Verifying the Certificate
- For More Information


## Certificate Overview

Digital certificates and key pairs are a form of digital identification for user authentication. Certificates can be issued for a variety of functions such as Web user authentication, Web server authentication, secure email (using Secure/Multipurpose Internet Mail Extensions, also called S/MIME), Internet Protocol security (IPsec), Transport Layer Security (TLS), and code signing.

A client or server certificate includes the name of the issuing authority and digital signature, the serial number, the name of the client or server that the certificate was issued for, the public key, and time stamp that indicate the certificate's expiration date.

A public key certificate, usually just called a certificate, is a digitally-signed statement that binds the value of a public key to the identity of the person, device, or service that holds the corresponding private key. Most certificates are based on the X.509v3 certificate standard.

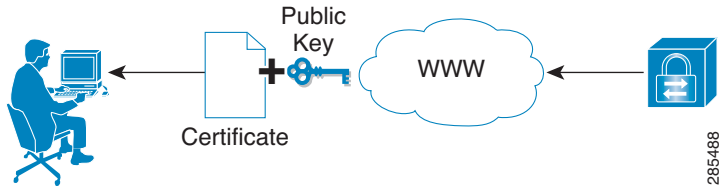
Certificate Authorities (CAs), such as GoDaddy or VeriSign issue certificates. A CA also provides a trusted CA certificate to verify that a client or server certificate originated from the CA. The CA certificate includes the CA distinguished name, public key, and digital signature.

The recipient of the CA digital certificate verifies it is issued by valid CA, and then obtains the public key and identification information held within the certificate. With this information, the recipient can send an encrypted reply.

### How Certificates Work

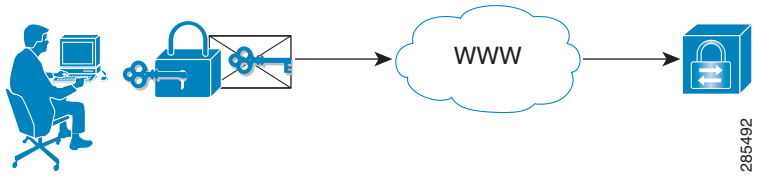
Step 1. A client (browser) send a request from a secure webpage (for example: https://mycompany.com).

Step 2. The web server sends its public key and certificate.

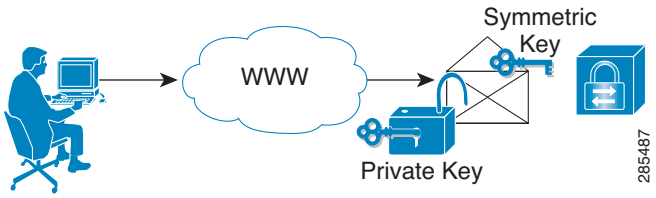



Step 3. The browser verifies whether the certificate was issued by an untrusted or trusted source (such as Verisign), confirms that the certificate is still valid, and verifies that the information is relevant to the site. For an untrusted certificate, the browser prompts an “exception” that asks the user to accept or reject the certificate.

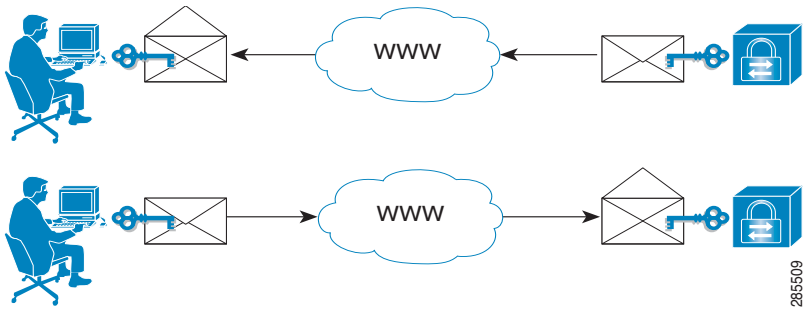
Step 4. Once the certificate is verified and accepted, the browser generates a random symmetric key and encrypted symmetric key information by using the public key. The browser then sends the keys to the server with the encrypted URL in addition to other encrypted HTTP data.




Step 5. Using its private key, the web server decrypts the package to obtain the symmetric key.




Step 6. Both the browser and the server are now using same the symmetric key. This key is used to encrypt and decrypt package data exchanged by the browser and server until the session is ended.



### Generating a Certification Authority and Root Certificate

To create your own SSL certificates, you need a Certification Authority (CA). A CA is required to sign a digital certificate.

You can purchase a certificate generated by a trusted CA or you can generate your own by using a third- party tool such as OpenSSL. OpenSSL is a cryptography toolkit that implements the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) network protocols and related cryptography.

When creating a certificate, the CA produces a root certificate and private key. The root certificate along with its private key can be used to sign other certificates or with a Certificate Signing Request (CSR). All root CA certificates are self-signed.

The following example shows how to use Ubuntu Linux OS and the OpenSSL tool to generate an SSL certificate.

**NOTE** Before generating an OpenSSL CA you may want to edit your openssl.cnf file to save time. This file is used each time that you use OpenSSL and stores the default information that you are prompted with during the certificate process.

Step 1. From Ubuntu, install the OpenSSL package.

```
root@ubuntu> apt-get install openssl
```

The OpenSSL package includes a perl script called “CA.pl.” This script supplies the relevant command line arguments to the openssl command for common certificate operations.

Step 2. Locate and add this file under the /usr/lib/ssl or /usr/lib/ssl/misc directory. Modify the script as shown in the example. This modification sets the OpenSSL environment variable from /etc/openssl.cnf and directs all output to the /var/ssl directory.

```
$SSLEAY_CONFIG="-config /etc/openssl.cnf";
…
#$CATOP="./demoCA";
$CATOP="/var/ssl";
```

Step 3. Add the CA.pl file to the /var/ssl directory.

```
root@ubuntu:/usr/lib/ssl> cp CA.pl /var/ssl/CA.pl
```

Step 4. Add the openssl.cnf file to the openssl.cnf directory.

```
root@ubuntu:/usr/lib/ssl> cp openssl.cnf /etc/openssl.cnf
```

Step 5. (Optional) Edit the openssl.cnf file and modify the default values to your own preferences. We recommend that you copy or back up the CA.pl file and the openssl.cnf file before editing this file.

```
Dir = /var/ssl              # Where everything is kept
….
countryName                   = Country Name (2 letter code)
countryName_default      = US
countryName_min= 2
countryName_max= 2
stateOrProvinceName= State or Province Name (full name)
stateOrProvinceName_default     = TX
localityName= Locality Name (eg, city)
localityName_default= RCDN
organizationName= Organization Name (eg, company)
organizationName_default = Cisco SBTG
```

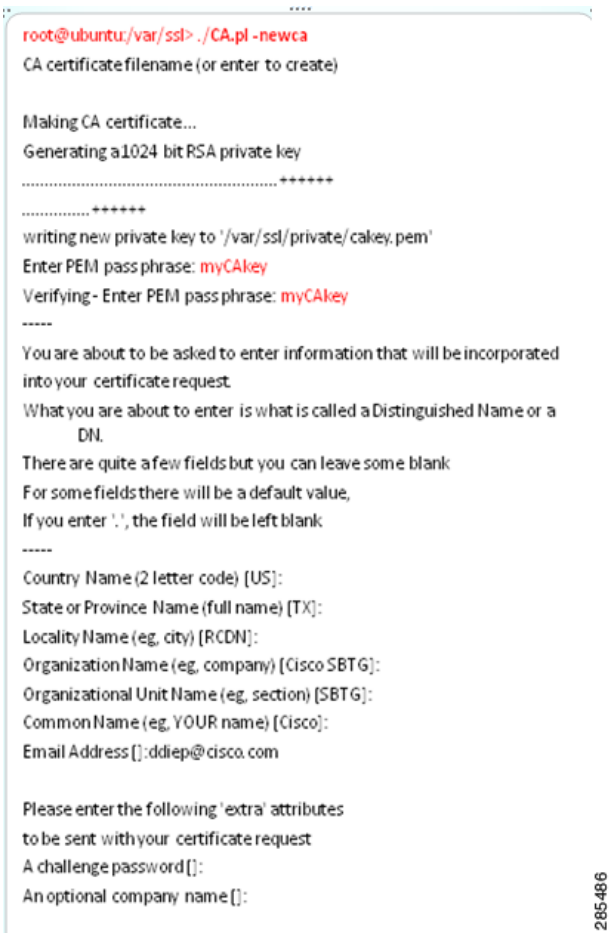
You are now ready to create the CA.

Step 6. From the /var/ssl directory enter the following command:

```
root@ubuntu:/var/ssl> ./CA.pl   -newca
```

Step 7. Enter the PEM passphrase.

**Note:** If you are using the openssl.cnf file, the DN information is already populated. If not, you will need to manually enter this information.



in this example, myCAkey is the private key passphrase. Make sure that you save this password as it is required when signing a Certificate Sign Request (CSR).

```
Using configuration from /etc/openssl.cnf
Enter pass phrase for /var/ssl/private/cakey.pem: myCAkey
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number:
        ef:50:6e:c6:2a:7b:e5:a7
    Validity
        Not Before: Jul 19 21:10:52 2011 GMT
        Not After : Jul 18 21:10:52 2014 GMT
    Subject:
        countryName           = US
        stateOrProvinceName   = TX
        organizationName      = Cisco SBTG
        organizationalUnitName  = SBTG
        commonName            = Cisco
        emailAddress          = ddiep@cisco.com
    X509v3 extensions:
        X509v3 Subject Key Identifier:
            84:C4:5D:5D:15:43:F2:68:42:1F:EB:DD:F1:53:79:31:6E:E2:B3:E3
        X509v3 Authority Key Identifier:
            keyid:84:C4:5D:5D:15:43:F2:68:42:1F:EB:DD:F1:53:79:31:6E:E2:B3:E3
            DirName:/C=US/ST=TX/O=Cisco
SBTG/OU=SBTG/CN=Cisco/emailAddress=ddiep@cisco.com
            serial:EF:50:6E:C6:2A:7B:E5:A7

        X509v3 Basic Constraints:
            CA:TRUE
Certificate is to be certified until Jul 18 21:10:52 2014 GMT (1095 days)

Write out database with 1 new entries
Data Base Updated
```

285501

Step 8. After running the CA.pl script, the root certificate (cacert.pem) and private key (cakey.pem) are created under the /var/ssl directory.These two files are used to verify and sign the certificate signing request (CSR).

- The cacert.pem file is the root CA certificate which also contains the public key.
- The cakey.pem file is the private CA key and is used to sign the user certificate request.

Example of a root certificate (cacert.pem)

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            dd:ff:d8:21:be:54:ba:34
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, ST=TX, O=Cisco SBTG, OU=SBTG, CN=Cisco
        Validity
            Not Before: Jul 21 23:29:50 2011 GMT
            Not After : Jul 20 23:29:50 2014 GMT
        Subject: C=US, ST=TX, O=Cisco SBTG, OU=SBTG, CN=Cisco
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:da:a5:59:d6:44:99:48:85:10:bd:4e:41:b3:ef:
                    b0:a6:ec:65:1a:cd:0b:6e:7f:84:5b:5b:39:32:45:
                    fe:99:c9:2f:76:93:c1:4c:b0:41:06:0c:a6:4c:30:
                    96:72:99:f9:81:f4:ec:d9:c3:0d:86:5f:3e:ed:82:
                    4d:97:21:d8:4c:2b:75:13:1a:af:f3:cd:47:d4:f3:
                    43:67:da:e0:3c:15:e3:70:ba:39:9c:0d:46:58:c6:
                    6a:48:48:19:1d:5c:88:98:d1:5f:6b:25:59:0c:d2:
                    ee:ab:d2:02:f9:f1:90:65:c4:b5:39:9e:19:6f:78:
                    5f:6f:3e:ff:76:b2:28:f1:fb
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:


AF:AA:9B:B8:C5:59:E0:AF:E0:69:69:C3:B6:B9:C6:46:A1:77:7A:17
            X509v3 Authority Key Identifier:
                keyid:AF:AA:9B:B8:C5:59:E0:AF:E0:69:69:C3:B6:B9:C6:46:A1:77:7A:17
                DirName:/C=US/ST=TX/O=Cisco SBTG/OU=SBTG/CN=Cisco
                serial:DD:FF:D8:21:BE:54:BA:34

            X509v3 Basic Constraints:
                CA:TRUE
    Signature Algorithm: sha1WithRSAEncryption
         51:0e:de:68:45:15:c1:6d:b1:b1:5f:7d:a7:d1:64:0b:49:0:
         8b:8d:14:56:4a:a5:c2:03:47:18:ed:99:bf:b1:0a:a4:47:66:
         79:80:dd:7b:55:e3:7c:62:63:a8:d7:39:c4:44:26:e5:0a:35:
         5d:cb:08:ab:2e:da:76:8f:d2:2a:e1:98:7:bac:9d:a3:4b:d5:
         a2:29:86:d1:f4:5e:70:8b:2b:96:6c:f8:96:0a:1c:33:4a:4d:
         e0:98:f9:09:d3:77:88:f8:56:d5:d5:da:9f:71:32:a5:d7:75:
         dc:d2:df:bb:63:6d:b1:0e:d2:09:e8:bb:62:7c:3f:8d:b9:48:
         a8:a7
-----BEGIN CERTIFICATE-----
MIICyzCCAjSgAwIBAgIJAN3/2CG+VLoOMAOGCSqGSIb3DQEBBQUAMIE4xCzAJBgNV
BAYTAlVTMQswCQYDVQQIEwJUWDETMBEGA1UEChMKQ2lzY28gU0JUUFpzENMAsGA1UE
CxMEU0JUUFpzE0MAwGA1UEAxMFQ2lzY28wHhcNMTEwNzIxMjMyOTUwWhcNMTQwNzIw
MjMyOTUwWjB0MQswCQYDVQQGEwJVUzELMAkGA1UECBMCVFgxETAPBgNVBAoTCENp
c2NvMFNCVExDTALBgNVBAsTBFNCVExDjAMBgNVBAMTBUNpc2NvMIGfMA0GCSqG
SIb3DQEBAQUAA4GNADCBiQKBgQDaaVnWPJIlkPC9TlGz77Cm7GUazOtef4PbWzky
Pf6Zy592l8FMzEBGDKZMMJZymfmB9OzZww2GXz7tgk2XkhMH3UTGq/zzUfU8ONa
2uA8FeNwujmcDUZVmp8BIdXlvOV9dVMMOv6rQgLS8ZBl+LUSnhheF9vP/9 2
zij+wIDAQABo4GwMIGtMB0GA1UdDgQWBBSvqpu4xVngr+BpacPmuc2GoXd6FzB+
BgNVHSMEdzB1gBSvqpu4xVngr+BpacPmuc2GoXd6F6FSpFAwTjELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAlRYMRMwEQYDVQQKEwpDaXNjby BTQlFHMQOwCwYDVQQLEwRT
QlFHMQ4wDAYDVQQDEwVDaXNjb4IJAN3/2CG+VLoOMAwGA1UdEwQFMAMBAf8wDQYJ
KoZIhvcNAQEFBQADgYEAUQ7eaEUVwW2rs\99p9FIC4SQi4OUVlqIwgNHGO2Zv7EK
pEdmeVDde1XDfGJjqNc5xBQm5Qo1Xcskjy7ado/SKwGvie6ydoOvVoimGOfPeclr
Inc4IgocM0pN4Jj5CdN3iPhW1fXan3Eypdd13NLfu2NtrQ7SCeI7Vw/jbIkqNc=
-----END CERTIFICATE-----
```

285506

Example of a private key (cakey.pem)

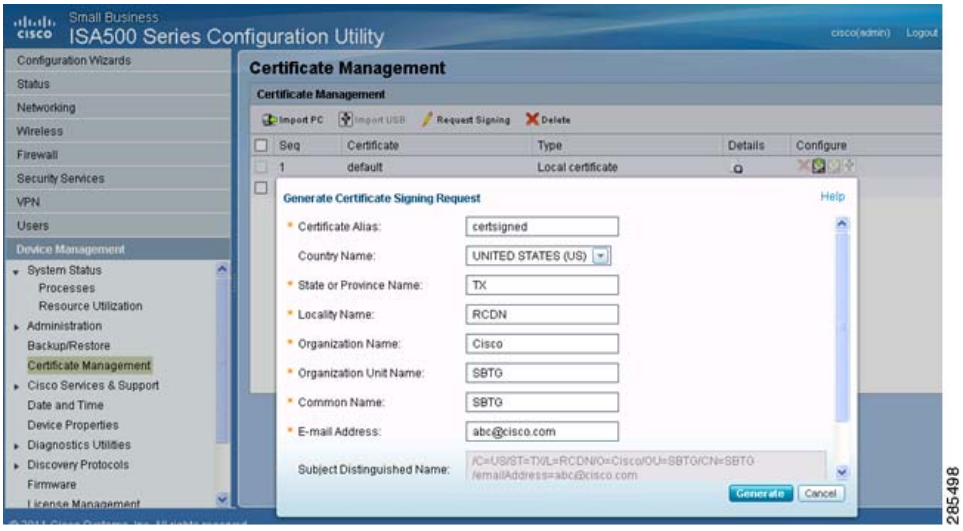

## Generating a Certificate Signing Request and Installing a Signed Certificate

You can configure the ISA500 to generate a certificate signing request (CSR). A CSR contains all the information required to create your digital certificate including the contact information, the common name for which the signed certificate is issued, and the public key of the server that will use the certificate. The CSR is then signed by the root CA or by a trusted CA such as GoDaddy or VeriSign.

Step 1. From the ISA500 Configuration Utility, choose **Device Management > Certificate Management**.

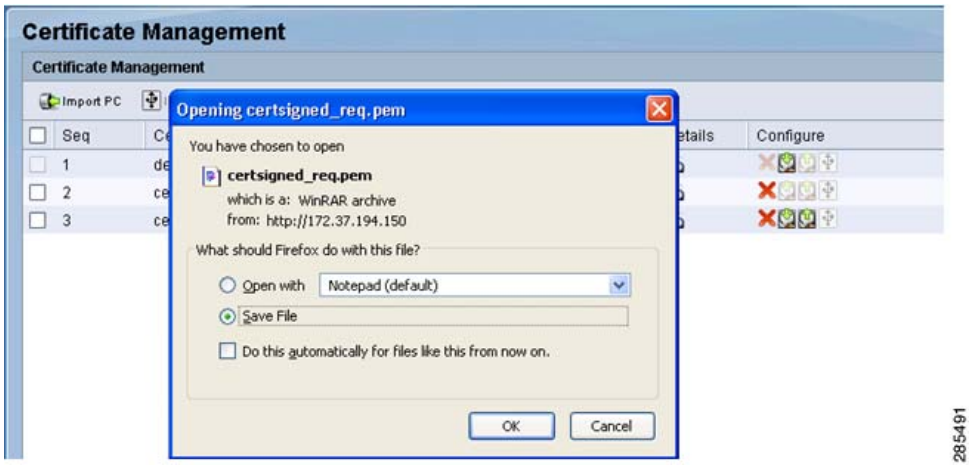
a. Click **Request Signing** and enter the information for the required fields.

b. Click **Generate** to create a a CSR (.pem) file.




The CSR is added to the Certificate Management table.


Step 2. Click the **Download** button to download the CSR file to your local machine. In this example, the CSR file is identified as certsigned_req.pem.

Example of a certificate signing request

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBLzCB2gIBADB1MQswCQYDVQQGEwJVUzELMAkGA1UECBMCVFgxDTALBgNVBAcT
BFJDRE4xDjAMBgNVBAoTBUNpc2NvMQ0wCwYDVQQLEwRTQlRHMQ0wCwYDVQQDEwRT
QlRHMRwwGgYJKoZIhvcNAQkBFg1hYmNAY2lzY28uY29tMFwwDQYJKoZIhvcNAQEB
BQADSwAwSAJBAN6OzrqrgkPzqJkSkbRnc8oBOyACCUQFC55u/rPbZ4goeW8erOfR
8fvQnc1O2oPJB62XBdQZHFh2SUvmuLFeWqsCAwEAAaAAMA0GCSqGSIb3DQEBBQUA
A0EAyLM6zg9dCS/tvCIUh1K+x2EWYU1IWr57KToV8MeS+J52CgRhnGjPuYz19A5u
mZ6/3W/C9N0vHq5BWGqg1vSP5w==
-----END CERTIFICATE REQUEST-----
```

285497

Step 3. Copy this CSR file into the /var/ssl (Ubuntu) folder. This file is to be signed by the CA that we created above. See Generating a Certification Authority and Root Certificate, page 3.

Step 4. Open the Ubuntu session and use OpenSSL to sign the CSR request (certsigned_req.pem).

In the example below, the fields in red are required. The certsigned_req.pem must match whatever CSR filename that the user generated in Step 3. The -out certsigned.crt file can be any filename. After entering the passphrase, the -out file is automatically generated.

```
root@ubuntu:/var/ssl> openssl x509 -req -days 365 -in certsigned_req.pem -CA
cacert.pem -CAkey ./private/cakey.pem -CAcreateserial -out certsigned.crt
Signature ok
subject=/C=US/ST=TX/L=RCDN/O=Cisco/OU=SBTG/CN=SBTG/emailAddress=abc@cisco.com
Getting CA Private Key
Enter pass phrase for ./private/cakey.pem:myCAkey
```

After the request is signed, the certsigned.crt file is generated. This is a signed certificate ready to be uploaded to the ISA500.

Example of a signed certificate:

```
-----BEGIN CERTIFICATE-----
MIIB5DCCAU0CCQCOZ9dFlEGl5DANBgkqhkiG9w0BAQUFADA8MQswCQYDVQQGEwJV
UzELMAkGA1UECBMCVFgxDTALBgNVBAoTBFNCVEcxETAPBgNVBAMTCERPTiBESUVQ
MB4XDTEyMDUxNjE5NDMzOVoXDTEzMDUxNjE5NDMzOVowdTELMAkGA1UEBhMCVVMx
CzAJBgNVBAgTAlRYMQ0wCwYDVQQHEwRSQ0ROMQ4wDAYDVQQKEwVDaXNjbzENMAsG
A1UECxMEU0JURzENMAsGA1UEAxMEU0JURzEcMBoGCSqGSIb3DQEJARYNYWJjQGNp
c2NvLmNvbTBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQDejs66q4JD86iZEpG0Z3PK
ATsgAglEBQuebv6z22eIKHlvHqzn0fH70J3NTtqDyQetlwXUGRxYdklL5rixXlqr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAFi2n60WzT0AgbFR9NOg4O8Jjh6TL2Pi8
X/nbIPj02+iw5zEkyjmcdmw2yWO4A9Qk54s4fZQwYUC4IX38zV1km236AXz7zp7N
1zxiLdYY7qQheNHlGp6ei23KloxegEf8sT0UC+0eeS/JOX6OUeeI2poEvs6T6IYr
gYZ1PGaArt0=
-----END CERTIFICATE-----
```

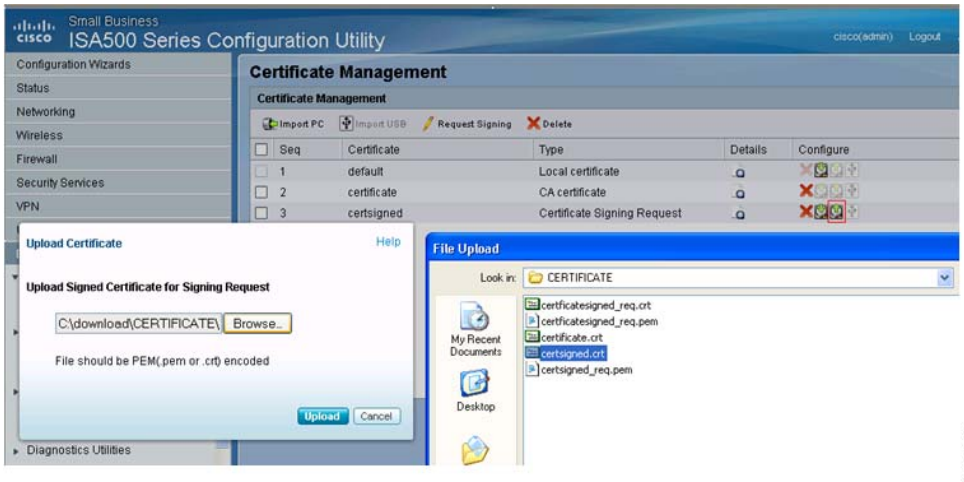
285494

Step 5. To install the signed certificate on the ISA500:

a. Open the Configuration Utility and choose **Device Management > Certificate Management**.

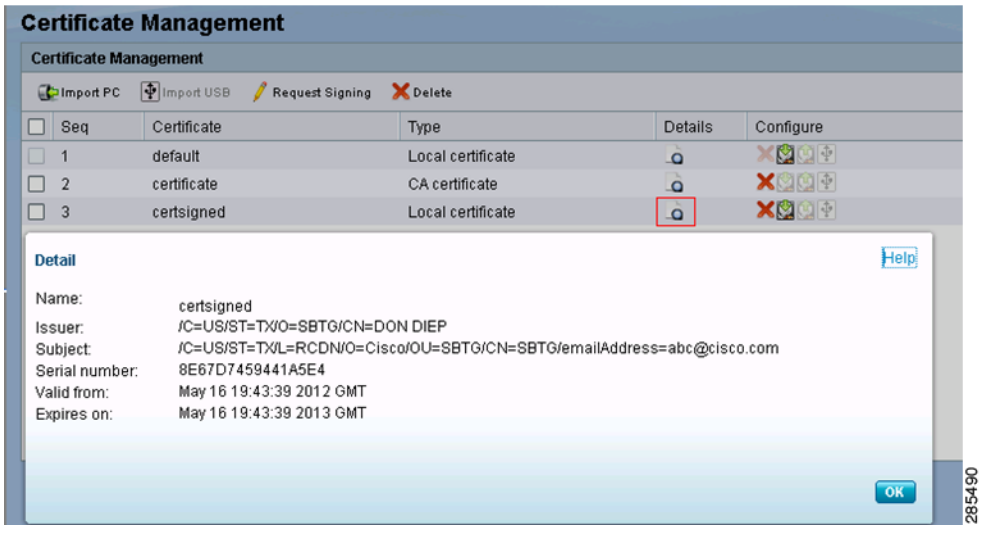
b. Select the CSR entry and click the **Upload** icon.

c. Click **Browse** to locate the signed certificate (certsigned.crt) and click **Upload**.




Step 6. When the signed certificate is successfully uploaded, the CSR status changes from Certificate Signing Request to Local certificate.


Step 7. Click the **Details** icon to view the certificate information.



### Installing a Self-Signed Certificate on the ISA500

This following steps describe how to install a self-signed certificate on the ISA500 by using the same certificate that we generated in Generating a Certification Authority and Root Certificate, page 3. This certificate is signed with its own private key.

Step 1. Open an Ubuntu session.

Step 2. From the /var/ssl directory, enter this command:

```
root@ubuntu:/var/ssl> openssl pkcs12 -keypbe PBE-SHA1-3DES -certpbe PBE-SHA1-
3DES -export -in cacert.pem -inkey /var/ssl/private/cakey.pem -out certp12.p12
```

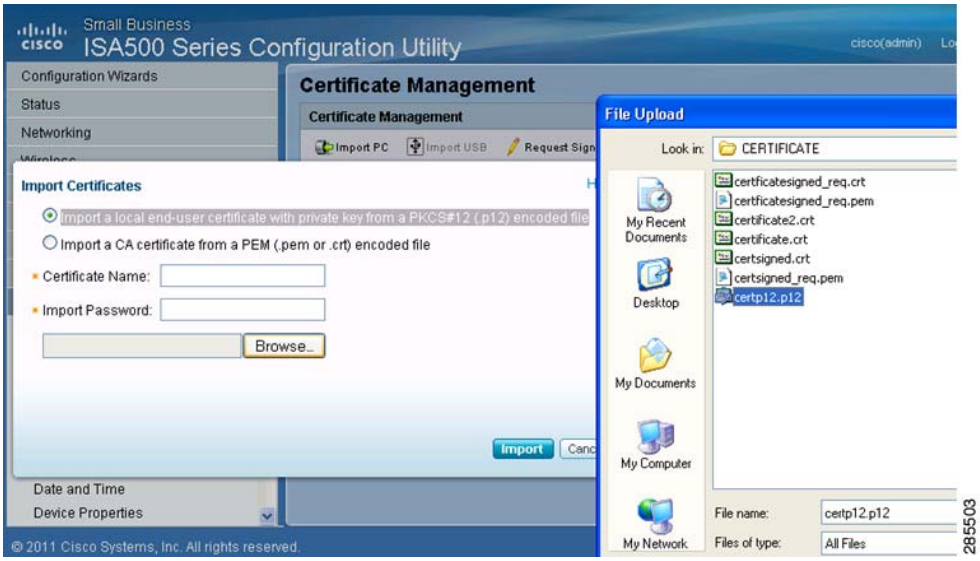
This command generates the certificate in a PKCS#12 format (For example: certp12.p12). PKCSs#12 is a binary format and cannot be viewed or edited.

Step 3. Import the certificate from your local PC.

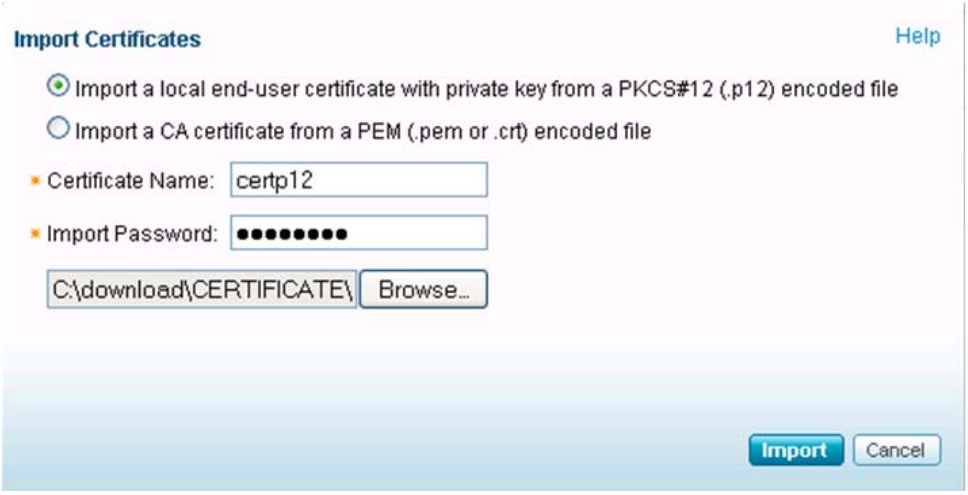
a. Choose **Device Management > Certificate Management**.

b. Select **Import PC**.

c. Click **Browse** and select the certp12.p12 certificate.




d. Enter the **Certificate Name** and **Import Password**. This is same private key password (myCAkey) that you used when generating the CA certificate. See Step 6 on page 4.




e. Click **Import**.

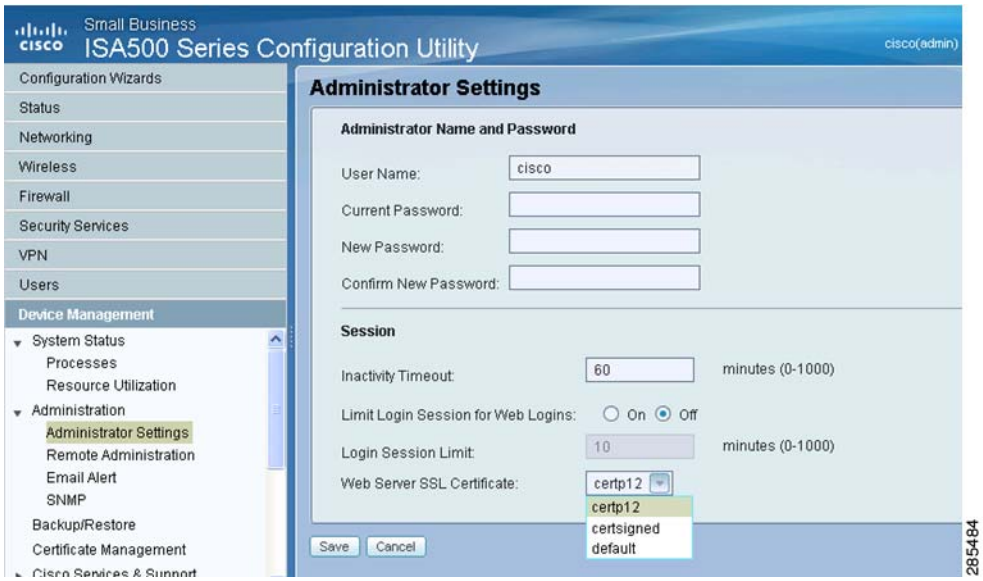
The certificate appears on the Certificate Management page.


## Activating and Verifying the Certificate

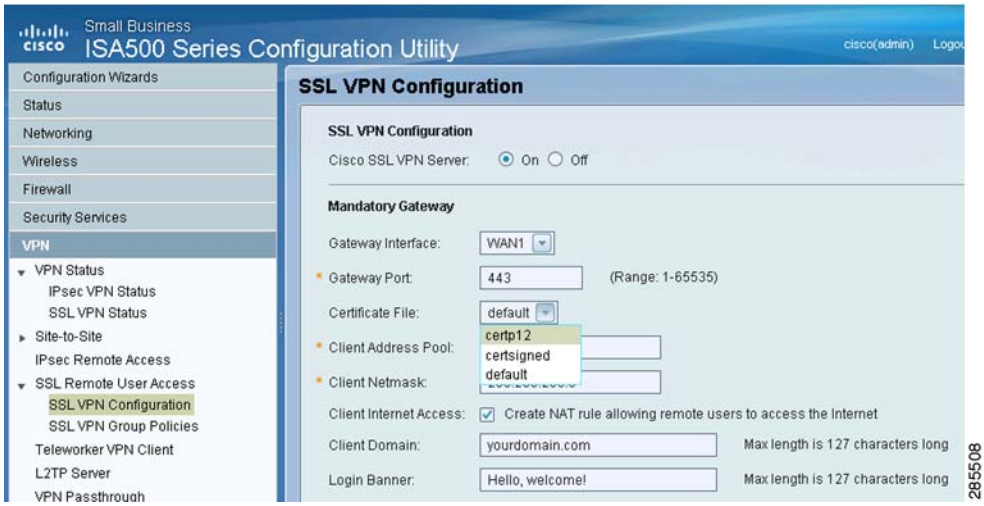
The next steps show how to activate the certificate for Web login users or to authenticate users who try to access your network resource through the SSL VPN tunnels. By default, the default certificate is used, or you can choose an imported certificate for authentication. The following example shows the **certp12** certificate that you imported in the previous section.

### Activating the Certificate

Step 1. For a SSL Web login, choose **Device Management > Administration > Administration Settings**. Choose the **certp12** certificate from the Web Server SSL Certificate drop-down menu and click **Save**.

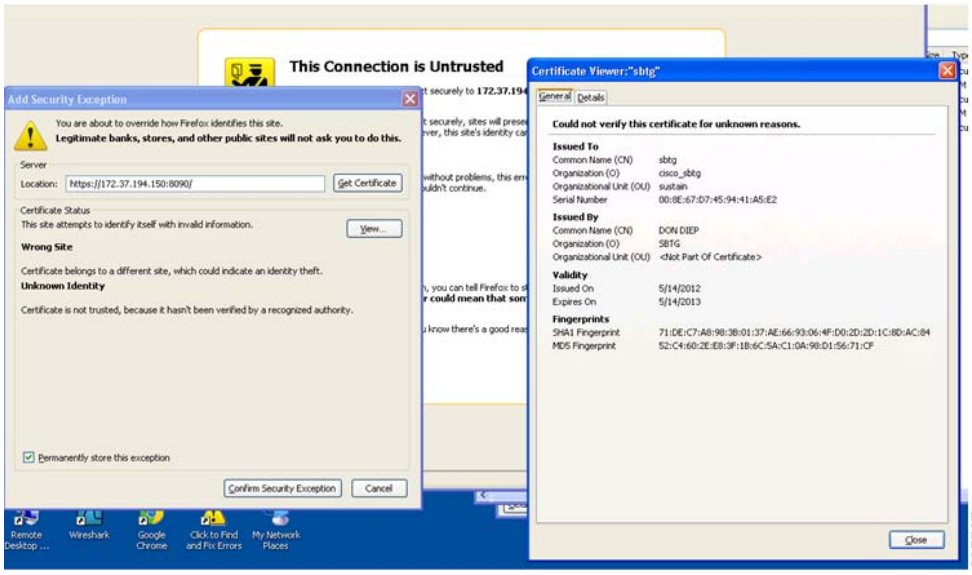



Step 2. For a SSL VPN connection, choose **VPN > SSL Remote User Access > SSL VPN Configuration**. Choose **certp12** from the Certificate File drop-down menu and click **Save**.



### Verifying the Certificate

To verify that the new certificate has taken effect, log in to the ISA500 by using HTTPS. The browser will prompt you for an exception because this certificate is arriving from an untrusted source. Click **View** to review the certificate content before accepting or canceling the security exception.



## For More Information

| Product Resources | Location |
|---|---|
| Product Documentation | www.cisco.com/go/isa500resources |
| Cisco Small Business Support Community | www.cisco.com/go/smallbizsupport |
| Cisco Small Business Support and Resources | www.cisco.com/go/smallbizhelp |
| Phone Support Contacts | www.cisco.com/go/sbsc |
| Firmware Downloads | www.cisco.com/go/isa500software |
| Cisco Partner Central for Small Business (Partner Login Required) | www.cisco.com/web/partners/sell/smb |
| Cisco Small Business Home | www.cisco.com/smb |