



# CHAPTER 41

## Configuring Certificates

---

This chapter describes how to configure certificates. CAs are responsible for managing certificate requests and issuing digital certificates. A digital certificate contains information that identifies a user or device. Some of this information can include a name, serial number, company, department, or IP address. A digital certificate also contains a copy of the public key for the user or device. A CA can be a trusted third party, such as VeriSign, or a private (in-house) CA that you establish within your organization.

This chapter includes the following sections:

- [Public Key Cryptography, page 41-1](#)
- [Certificate Configuration, page 41-5](#)
- [The Local CA, page 41-17](#)

## Public Key Cryptography

This section includes the following topics:

- [About Public Key Cryptography, page 41-1](#)
- [Certificate Scalability, page 41-2](#)
- [About Key Pairs, page 41-2](#)
- [About Trustpoints, page 41-3](#)
- [About CRLs, page 41-3](#)
- [Supported CA Servers, page 41-5](#)

## About Public Key Cryptography

Digital signatures, enabled by public key cryptography, provide a means to authenticate devices and users. In public key cryptography, such as the RSA encryption system, each user has a key pair containing both a public and a private key. The keys act as complements, and anything encrypted with one of the keys can be decrypted with the other.

In simple terms, a signature is formed when data is encrypted with a private key. The signature is attached to the data and sent to the receiver. The receiver applies the public key of the sender to the data. If the signature sent with the data matches the result of applying the public key to the data, the validity of the message is established.

This process relies on the receiver having a copy of the public key of the sender and having a high degree of certainty that this key belongs to the sender, not to someone pretending to be the sender.

Obtaining the public key of a sender is normally handled out-of-band or through an operation done at installation. For instance, most web browsers are configured with the root certificates of several CAs by default. For VPN, the IKE protocol, a component of IPSec, can use digital signatures to authenticate peer devices before setting up security associations.

## Certificate Scalability

Without digital certificates, you must manually configure each IPSec peer for every peer with which it communicates, and every new peer you add to a network would thus require a configuration change on every peer with which you need it to communicate securely.

When you use digital certificates, each peer is enrolled with a CA. When two peers attempt to communicate, they exchange certificates and digitally sign data to authenticate each other. When a new peer is added to the network, you enroll that peer with a CA and none of the other peers need modification. When the new peer attempts an IPSec connection, certificates are automatically exchanged and the peer can be authenticated.

With a CA, a peer authenticates itself to the remote peer by sending a certificate to the remote peer and performing some public key cryptography. Each peer sends its unique certificate which was issued by the CA. This process works because each certificate encapsulates the public key for the associated peer and each certificate is authenticated by the CA, and all participating peers recognize the CA as an authenticating authority. This is called IKE with an RSA signature.

The peer can continue sending its certificate for multiple IPSec sessions, and to multiple IPSec peers, until the certificate expires. When its certificate expires, the peer administrator must obtain a new one from the CA.

CAs can also revoke certificates for peers that no longer participate in IPSec. Revoked certificates are not recognized as valid by other peers. Revoked certificates are listed in a CRL, which each peer may check before accepting a certificate from another peer.

Some CAs have an RA as part of their implementation. An RA is a server that acts as a proxy for the CA so that CA functions can continue when the CA is unavailable.

## About Key Pairs

Key pairs are RSA keys.

- RSA keys can be used for SSH or SSL.
- SCEP enrollment supports the certification of RSA keys.
- For the purposes of generating keys, the maximum key modulus for RSA keys is 2048 bits. The default size is 1024. Many SSL connections using identity certificates with RSA key pairs that exceed 1024 bits can cause a high CPU usage on the security appliance and rejected clientless logins.
- For signature operations, the supported maximum key size is 4096 bits.
- You can generate a general purpose RSA key pair, used for both signing and encryption, or you can generate separate RSA key pairs for each purpose.

Separate signing and encryption keys helps reduce exposure of the keys. This is because SSL uses a key for encryption but not signing but IKE uses a key for signing but not encryption. By using separate keys for each, exposure of the keys is minimized.

## About Trustpoints

Trustpoints let you manage and track CAs and certificates. A trustpoint is a representation of a CA or identity pair. A trustpoint contains the identity of the CA, CA-specific configuration parameters, and an association with one enrolled identity certificate.

After you have defined a trustpoint, you can reference it by name in commands requiring that you specify a CA. You can configure many trustpoints.



### Note

If a security appliance has multiple trustpoints that share the same CA, only one of these trustpoints sharing the CA can be used to validate user certificates. Use the **support-user-cert-validation** command to control which trustpoint sharing a CA is used for validation of user certificates issued by that CA.

For automatic enrollment, a trustpoint must be configured with an enrollment URL and the CA that the trustpoint represents must be available on the network and must support SCEP.

You can export and import the keypair and issued certificates associated with a trustpoint in PKCS12 format. This is useful if you wish to manually duplicate a trustpoint configuration on a different security appliance.

## About Revocation Checking

When a certificate is issued, it is valid for a fixed period of time. Sometimes a CA revokes a certificate before this time period expires; for example, due to security concerns or a change of name or association. CAs periodically issue a signed list of revoked certificates. Enabling revocation checking forces the security appliance to check that the CA has not revoked a certificate every time it uses that certificate for authentication.

When you enable revocation checking, during the PKI certificate validation process the security appliance checks certificate revocation status. It can use either CRL checking or Online Certificate Status Protocol or both, with the second method you set in effect only when the first method returns an error, for example, that the server is unavailable.

With CRL checking, the security appliance retrieves, parses, and caches Certificate Revocation Lists, which provide a complete list of revoked certificates. OCSP offers a more scalable method of checking revocation status in that it localizes certificate status on a Validation Authority, which it queries for the status of a specific certificate.

## About CRLs

Certificate Revocation Lists provide the security appliance with one means of determining whether a certificate that is within its valid time range has been revoked by its issuing CA. CRL configuration is a part of the configuration of a trustpoint.

You can configure the security appliance to make CRL checks mandatory when authenticating a certificate (**revocation-check crl** command). You can also make the CRL check optional by adding the **none** argument (**revocation-check crl none** command), which allows the certificate authentication to succeed when the CA is unavailable to provide updated CRL data.

The security appliance can retrieve CRLs from CAs using HTTP, SCEP, or LDAP. CRLs retrieved for each trustpoint are cached for a length of time configurable for each trustpoint.

When the security appliance has cached a CRL for more than the length of time it is configured to cache CRLs, the security appliance considers the CRL too old to be reliable, or “stale”. The security appliance attempts to retrieve a newer version of the CRL the next time a certificate authentication requires checking the stale CRL.

The security appliance caches CRLs for a length of time determined by the following two factors:

- The number of minutes specified with the **cache-time** command. The default value is 60 minutes.
- The NextUpdate field in the CRLs retrieved, which may be absent from CRLs. You control whether the security appliance requires and uses the NextUpdate field with the **enforcenextupdate** command.

The security appliance uses these two factors as follows:

- If the NextUpdate field is not required, the security appliance marks CRLs as stale after the length of time defined by the **cache-time** command.
- If the NextUpdate field is required, the security appliance marks CRLs as stale at the sooner of the two times specified by the **cache-time** command and the NextUpdate field. For example, if the cache-time command is set to 100 minutes and the NextUpdate field specifies that the next update is 70 minutes away, the security appliance marks CRLs as stale in 70 minutes.

If the security appliance has insufficient memory to store all CRLs cached for a given trustpoint, it deletes the least recently used CRL to make room for a newly retrieved CRL.

For information about configuring CRL behavior for a trustpoint, see the [“Configuring CRLs for a Trustpoint” section on page 41-13](#).

## About OCSP

Online Certificate Status Protocol provides the security appliance with a means of determining whether a certificate that is within its valid time range has been revoked by its issuing CA. OCSP configuration is a part of the configuration of a trustpoint.

OCSP localizes certificate status on a Validation Authority (an OCSP server, also called the *responder*) which the security appliance queries for the status of a specific certificate. It provides better scalability and more up-to-date revocation status than does CRL checking. It helps organizations with large PKI installations deploy and expand secure networks.

You can configure the security appliance to make OCSP checks mandatory when authenticating a certificate (**revocation-check ocsf** command). You can also make the OCSP check optional by adding the **none** argument (**revocation-check ocsf none** command), which allows the certificate authentication to succeed when the Validation Authority is unavailable to provide updated OCSP data.

Our implementation of OCSP provides three ways to define the OCSP server URL. The security appliance uses these servers in the following order:

1. The OCSP URL defined in a match certificate override rule (**match certificate** command).
2. The OCSP URL configured in the **ocsp url** command.
3. The AIA field of the client certificate.



### Note

To configure a trustpoint to validate a self-signed OCSP responder certificate, you import the self-signed responder certificate into its own trustpoint as a trusted CA certificate. Then you configure the **match certificate** command in the client certificate validating trustpoint to use the trustpoint that contains the

self-signed OCSP responder certificate to validate the responder certificate. The same applies for configuring validating responder certificates external to the validation path of the client certificate.

The OCSP server (responder) certificate typically signs the OCSP response. After receiving the response, the security appliance tries to verify the responder certificate. The CA normally sets the lifetime of its OCSP responder certificate to a relatively short period to minimize the chance of it being compromised. The CA typically also includes an `ocsp-no-check` extension in the responder certificate indicating that this certificate does not need revocation status checking. But if this extension is not present, the security appliance tries to check its revocation status using the same method specified in the trustpoint. If the responder certificate is not verifiable, revocation checks fails. To avoid this possibility, configure **revocation-check none** in the responder certificate validating trustpoint, while configuring **revocation-check ocsp** for the client certificate.

---

## Supported CA Servers

The security appliance supports the following CA servers:

- Cisco IOS CS
- Baltimore Technologies
- Entrust
- Microsoft Certificate Services
- Netscape CMS
- RSA Keon
- VeriSign

## Certificate Configuration

This section describes how to configure the security appliance with certificates and other procedures related to certificate use and management.

This section includes the following topics:

- [Preparing for Certificates, page 41-5](#)
- [Configuring Key Pairs, page 41-6](#)
- [Configuring Trustpoints, page 41-7](#)
- [Obtaining Certificates, page 41-9](#)
- [Configuring CRLs for a Trustpoint, page 41-13](#)
- [Exporting and Importing Trustpoints, page 41-15](#)
- [Configuring CA Certificate Map Rules, page 41-16](#)

## Preparing for Certificates

Before you configure a security appliance with certificates, ensure that the security appliance is configured properly to support certificates. An improperly configured security appliance can cause enrollment to fail or for enrollment to request a certificate containing inaccurate information.

To prepare a security appliance for certificates, perform the following steps:

- 
- Step 1** Ensure that the hostname and domain name of the security appliance are configured correctly. You can use the **show running-config** command to view the hostname and domain name as currently configured. For information about configuring the hostname, see the [“Setting the Hostname” section on page 9-2](#). For information about configuring the domain name, see the [“Setting the Domain Name” section on page 9-2](#).
- Step 2** Be sure that the security appliance clock is set accurately before configuring the CA. Certificates have a date and time that they become valid and that they expire. When the security appliance enrolls with a CA and gets a certificate, the security appliance checks that the current time is within the valid range for the certificate. If it is outside that range, enrollment fails. For information about setting the clock, see the [“Setting the Date and Time” section on page 9-2](#).
- 

## Configuring Key Pairs

This section includes the following topics:

- [Generating Key Pairs, page 41-6](#)
- [Removing Key Pairs, page 41-7](#)

## Generating Key Pairs

Key pairs are RSA keys, as discussed in the [“About Key Pairs” section on page 41-2](#). You must generate key pairs for the types of certification you want to use.

To generate key pairs, perform the following steps:

- 
- Step 1** Generate the types of key pairs needed for your PKI implementation. To do so, perform the following steps, as applicable:

- a. If you want to generate RSA key pairs, use the **crypto key generate rsa** command.

```
hostname/contexta(config)# crypto key generate rsa
```

If you do not use additional keywords this command generates one general purpose RSA key pair. Because the key modulus is not specified, the default key modulus of 1024 is used. You can specify other modulus sizes with the **modulus** keyword.



**Note** Many SSL connections using identity certificates with RSA key pairs that exceed 1024 bits can cause a high CPU usage on the security appliance and rejected clientless logins.

You can also assign a label to each key pair using the **label** keyword. The label is referenced by the trustpoint that uses the key pair. If you do not assign a label, the key pair is automatically labeled <Default-RSA-Key>.

```
hostname/contexta(config)# crypto key generate rsa label key-pair-label
```

- Step 2** (Optional) Use the **show crypto key mypubkey** command to view key pair(s). The following example shows an RSA general-purpose key:

```
hostname/contexta(config)# show crypto key mypubkey
Key pair was generated at: 16:39:47 central Feb 10 2005
Key name: <Default-RSA-Key>
Usage: General Purpose Key
Modulus Size (bits): 1024
Key Data:

30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 00ea51b7
0781848f 78bccac2 4a1b5b8d 2f3e30b4 4cae9f86 f4485207 159108c9 f5e49103
9eeb0f5d 45fd1811 3b4aafce 292b3b64 b4124a6f 7a777b08 75b88df1 8092a9f8
5508e9e5 2c271245 7fd1c0c3 3aaf1e04 c7c4efa4 600f4c4a 6afe56ad c1d2c01c
e08407dd 45d9e36e 8cc0bfef 14f9e6ac eca141e4 276d7358 f7f50d13 79020301 0001
Key pair was generated at: 16:34:54 central Feb 10 2005
```

- Step 3** Save the key pair you have generated. To do so, save the running configuration by entering the **write memory** command.

## Removing Key Pairs

To remove key pairs, use the **crypto key zeroize command in global configuration mode**.

The following example removes RSA key pairs:

```
hostname(config)# crypto key zeroize rsa
WARNING: All RSA keys will be removed.
WARNING: All device certs issued using these keys will also be removed.

Do you really want to remove these keys? [yes/no] y
hostname(config)#
```

## Configuring Trustpoints

For information about trustpoints, see the [“About Trustpoints” section on page 41-3](#).

To configure a trustpoint, perform the following steps:

- Step 1** Create a trustpoint corresponding to the CA from which the security appliance needs to receive its certificate.

```
hostname/contexta(config)# crypto ca trustpoint trustpoint
```

For example, to declare a trustpoint called Main:

```
hostname/contexta(config)# crypto ca trustpoint Main
hostname/contexta(config-ca-trustpoint)#
```

Upon entering this command, you enter the Crypto ca trustpoint configuration mode.

- Step 2** Specify the enrollment method to be used with this trustpoint.

To specify the enrollment method, do one of the following items:

- To specify SCEP enrollment, use the **enrollment url** command to configure the URL to be used for SCEP enrollment with the trustpoint you declared. For example, if the security appliance requests certificates from trustpoint Main using the URL `http://10.29.67.142:80/certsrv/mscep/mscep.dll`, then the command would be as follows:

```
hostname/contexta(config-ca-trustpoint)# enrollment url
http://10.29.67.142:80/certsrv/mscep/mscep.dll
```

- To specify manual enrollment, use the **enrollment terminal** command to indicate that you will paste the certificate received from the CA into the terminal.

**Step 3** As needed, specify other characteristics for the trustpoint. The characteristics you need to define depend upon your CA and its configuration. You can specify characteristics for the trustpoint using the following commands. Refer to the *Cisco Security Appliance Command Reference* for complete descriptions and usage guidelines of these commands.

- **accept-subordinates**—Indicates whether CA certificates subordinate to the CA associated with the trustpoint are accepted if delivered during phase one IKE exchange when not previously installed on the device.
- **crl required | optional | nocheck**—Specifies CRL configuration options. When you enter the **crl** command with the **optional** keyword included within the command statement, certificates from peers can still be accepted by your security appliance even if the CRL is not accessible to your security appliance.



**Note** If you chose to enable required or optional CRL checking, be sure you configure the trustpoint for CRL management, which should be completed after you have obtained certificates. For details about configuring CRL management for a trustpoint, see the [“Configuring CRLs for a Trustpoint” section on page 41-13](#).

- **crl configure**—Enters CRL configuration mode.
- **default enrollment**—Returns all enrollment parameters to their system default values. Invocations of this command do not become part of the active configuration.
- **email address**—During enrollment, asks the CA to include the specified email address in the Subject Alternative Name extension of the certificate.
- **enrollment retry period**—(Optional) Specifies a retry period in minutes. This characteristic only applies if you are using SCEP enrollment.
- **enrollment retry count**—(Optional) Specifies a maximum number of permitted retries. This characteristic only applies if you are using SCEP enrollment.
- **enrollment terminal**—Specifies cut and paste enrollment with this trustpoint.
- **enrollment url URL**—Specifies automatic enrollment (SCEP) to enroll with this trustpoint and configures the enrollment URL.
- **fqdn fqdn**—During enrollment, asks the CA to include the specified fully qualified domain name in the Subject Alternative Name extension of the certificate.
- **id-cert-issuer**—Indicates whether the system accepts peer certificates issued by the CA associated with this trustpoint.
- **ip-address ip-address**—During enrollment, asks the CA to include the IP address of the security appliance in the certificate.
- **keypair name**—Specifies the key pair whose public key is to be certified.
- **match certificate map**—Configures OCSP URL overrides and trustpoints to use to validate OCSP responder certificates
- **ocsp disable-nonce**—Disable the nonce extension on an OCSP request; the nonce extension cryptographically binds requests with responses to avoid replay attacks.

- **ocsp url**—Configures an OCSP server for the security appliance to use to check all certificates associated with a trustpoint rather than the server specified in the AIA extension of the client certificate.
- **password string**—Specifies a challenge phrase that is registered with the CA during enrollment. The CA typically uses this phrase to authenticate a subsequent revocation request.
- **revocation-check**—Sets one or more methods for revocation checking: CRL, OCSP, and none.
- **subject-name X.500 name**—During enrollment, asks the CA to include the specified subject DN in the certificate. If a DN string contains a comma, enclose the value string with double quotes (for example, O="Company, Inc.>").
- **serial-number**—During enrollment, asks the CA to include the security appliance serial number in the certificate.
- **support-user-cert-validation**—If enabled, the configuration settings to validate a remote user certificate can be taken from this trustpoint, provided that this trustpoint is authenticated to the CA that issued the remote certificate.
- **exit**—Leaves the mode.

**Step 4** Save the trustpoint configuration. To do so, save the running configuration by entering the **write memory** command.

---

## Obtaining Certificates

The security appliance needs a CA certificate for each trustpoint and one or two certificates for itself, depending upon the configuration of the keys used by the trustpoint. If the trustpoint uses separate RSA keys for signing and encryption, the security appliance needs two certificates, one for each purpose. In other key configurations, only one certificate is needed.

The security appliance supports enrollment with SCEP and with manual enrollment, which lets you paste a base-64-encoded certificate directly into the terminal. For site-to-site VPNs, you must enroll each security appliance. For remote access VPNs, you must enroll each security appliance and each remote access VPN client.

This section includes the following topics:

- [Obtaining Certificates with SCEP, page 41-9](#)
- [Obtaining Certificates Manually, page 41-11](#)
- [Using Extended Keys for Certificates, page 41-13](#)

### Obtaining Certificates with SCEP

This procedure provides steps for configuring certificates using SCEP. Repeat these steps for each trustpoint you configure for automatic enrollment. When you have completed this procedure, the security appliance will have received a CA certificate for the trustpoint and one or two certificates for signing and encryption purposes. If you use general-purpose RSA keys, the certificate received is for signing and encryption. If you use separate RSA keys for signing and encryption, the security appliance receives separate certificates for each purpose.

**Note**

Whether a trustpoint uses SCEP for obtaining certificates is determined by the use of the **enrollment url** command when you configure the trustpoint (see the [“Configuring Trustpoints”](#) section on page 41-7).

To obtain certificates with SCEP, perform the following steps:

- Step 1** Obtain the CA certificate for the trustpoint you configured.

```
hostname/contexta(config)# crypto ca authenticate trustpoint
```

For example, using trustpoint named Main, which represents a subordinate CA:

```
hostname/contexta(config)# crypto ca authenticate Main
```

```
INFO: Certificate has the following attributes:
```

```
Fingerprint:      3736ffc2 243ecf05 0c40f2fa 26820675
```

```
Do you accept this certificate? [yes/no]: y
```

```
Trustpoint 'Main' is a subordinate CA and holds a non self signed cert.
```

```
Trustpoint CA certificate accepted.
```

- Step 2** Enroll the security appliance with the trustpoint. This process retrieves a certificate for signing data and, depending upon the type of keys you configured, for encrypting data.

- Step 3** To perform enrollment, use the **crypto ca enroll** command. Before entering this command, contact your CA administrator because the administrator may need to authenticate your enrollment request manually before the CA grants its certificates.

```
hostname(config)# crypto ca enroll trustpoint
```

If the security appliance does not receive a certificate from the CA within 1 minute (the default) of sending a certificate request, it resends the certificate request. The security appliance continues sending a certificate request every 1 minute until a certificate is received.

**Note**

If the fully qualified domain name configured for the trustpoint is not identical to the fully qualified domain name of the security appliance, including the case of the characters, a warning appears. If needed, you can exit the enrollment process, make any necessary corrections, and enter the **crypto ca enroll** command again.

The following enrollment example performs enrollment with the trustpoint named Main:

```
hostname(config)# crypto ca enroll Main
%
% Start certificate enrollment ..
% 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: 2b0rn0t2b
Re-enter password: 2b0rn0t2b
% The subject name in the certificate will be: securityappliance.example.com
% The fully-qualified domain name in the certificate will be:
securityappliance.example.com
% Include the device serial number in the subject name? [yes/no]: no
Request certificate from CA [yes/no]: yes
% Certificate request sent to Certificate authority.
```

**Note**

The password is required if the certificate for the security appliance needs to be revoked, so it is crucial that you remember this password. Note it and store it in a safe place.

You must enter the **crypto ca enroll** command for each trustpoint with which the security appliance needs to enroll.

**Note**

If your security appliance reboots after you issued the **crypto ca enroll** command but before you received the certificate, reissue the **crypto ca enroll** command and notify the CA administrator.

- Step 4** Verify that the enrollment process was successful using the **show crypto ca certificate** command. For example, to show the certificate received from trustpoint Main:

```
hostname/contexta(config)# show crypto ca certificate Main
```

The output of this command shows the details of the certificate issued for the security appliance and the CA certificate for the trustpoint.

- Step 5** Save the configuration using the **write memory** command:

```
hostname/contexta(config)# write memory
```

## Obtaining Certificates Manually

This procedure provides steps for configuring certificates using manual certificate requests. Repeat these steps for each trustpoint you configure for manual enrollment. When you have completed this procedure, the security appliance will have received a CA certificate for the trustpoint and one or two certificates for signing and encryption purposes. If you use general-purpose RSA keys, the certificate received is for signing and encryption. If you use separate RSA keys for signing and encryption, the certificates received are used for each purpose exclusively.

**Note**

Whether a trustpoint requires that you manually obtain certificates is determined by the use of the **enrollment terminal** command when you configure the trustpoint (see the [“Configuring Trustpoints” section on page 41-7](#)).

To obtain certificates manually, perform the following steps:

- Step 1** Obtain a base-64 encoded CA certificate from the CA represented by the trustpoint.
- Step 2** Import the CA certificate. To do so, use the **crypto ca authenticate** command. The following example shows a CA certificate request for the trustpoint Main.

```
hostname (config)# crypto ca authenticate Main
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVCqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
[ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit
```

```
INFO: Certificate has the following attributes:
Fingerprint:      24b81433 409b3fd5 e5431699 8d490d34
```

```
Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.
```

```
% Certificate successfully imported
hostname (config)#
```

- Step 3** Generate a certificate request. To do so, use the **crypto ca enroll** command. The following example shows a certificate and encryption key request for the trustpoint Main, which is configured to use manual enrollment and general-purpose RSA keys for signing and encryption.

```
hostname (config)# crypto ca enroll Main
% Start certificate enrollment ..

% The fully-qualified domain name in the certificate will be:
securityappliance.example.com

% Include the device serial number in the subject name? [yes/no]: n

Display Certificate Request to terminal? [yes/no]: y
Certificate Request follows:

MIIBoDCCAQkCAQAwIzEhMB8GCSqGSIb3DQEJAhYSRmVyYWxQaXguY2lzY28uY29t
[ certificate request data omitted ]
jF4waw68eOxQxVmdgMWeQ+RbIOYmvt8g6hnBTrd0GdqqjVlt

---End - This line not part of the certificate request---

Redisplay enrollment request? [yes/no]: n
hostname (config)#
```



**Note** If you use separate RSA keys for signing and encryption, the **crypto ca enroll** command displays two certificate requests, one for each key. To complete enrollment, acquire a certificate for all certificate requests generated by the **crypto ca enroll** command.

- Step 4** For each request generated by the **crypto ca enroll** command, obtain a certificate from the CA represented by the applicable trustpoint. Be sure the certificate is in base-64 format.

- Step 5** For each certificate you receive from the CA, use the **crypto ca import certificate** command. The security appliance prompts you to paste the certificate to the terminal in base-64 format.



**Note** If you use separate RSA key pairs for signing and encryption, perform this step for each certificate separately. The security appliance determines automatically whether the certificate is for the signing or encryption key pair. The order in which you import the two certificates is irrelevant.

The following example manually imports a certificate for the trustpoint Main:

```
hostname (config)# crypto ca import Main certificate
% The fully-qualified domain name in the certificate will be:
securityappliance.example.com

Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself
[ certificate data omitted ]
quit
INFO: Certificate successfully imported
hostname (config)#
```

- Step 6** Verify that the enrollment process was successful using the **show crypto ca certificate** command. For example, to show the certificate received from trustpoint Main:

```
hostname/contexta(config)# show crypto ca certificate Main
```

The output of this command shows the details of the certificate issued for the security appliance and the CA certificate for the trustpoint.

- Step 7** Save the configuration using the **write memory** command:

```
hostname/contexta(config)# write memory
```

---

## Using Extended Keys for Certificates

After you have upgraded the security appliance (for example, from Version 7.2 to Version 8.0), you can connect to it with the Cisco AnyConnect VPN Client using an older certificate with an extended key OID. To allow you to migrate older certificates, two new submode options to the **crypto ca trustpoint** command have been added. To enable this feature, perform the following steps:

- Step 1** Enter the **crypto ca trustpoint** command to enter the crypto ca trustpoint configuration mode:

```
hostname (config)# crypto ca trustpoint trustpoint-name
hostname(config-ca-trustpoint)
```

- Step 2** To perform key usage checks for an SSL certificate, enter the following submode option:

```
hostname(config-ca-trustpoint) no ignore-ssl-keyusage
```

- Step 3** To perform key usage checks for an IPsec certificate, enter the following submode option:

```
hostname(config-ca-trustpoint) no ignore-ipsec-keyusage
```

---

## Configuring CRLs for a Trustpoint

If you want to use mandatory or optional CRL checking during certificate authentication, you must perform CRL configuration for each trustpoint. For more information about CRLs, see the [“About CRLs” section on page 41-3](#).

To configure CRLs for a trustpoint, perform the following steps:

- Step 1** Enter crypto ca trustpoint configuration mode for the trustpoint whose CRL configuration you want to modify. To do so, enter the **crypto ca trustpoint** command.

- Step 2** If you have not already enabled CRLs, you can do so now by using the **crl** command with either the **required** or **optional** keyword. If you specify the **required** keyword, certificate authentication with this trustpoint cannot succeed if the CRL is unavailable.

- Step 3** Enter the **crl configure** command.

```
hostname/contexta(config-ca-trustpoint)# crl configure
hostname/contexta(config-ca-crl)#
```

Upon entering this command, you enter the crl configuration mode for the current trustpoint.

**Tip**

To set all CRL configuration options to their default values, use the **default** command. At any time while performing CRL configuration, if you want to start over, enter this command and restart this procedure.

**Step 4** Configure the retrieval policy with the **policy** command. The following keywords for this command determine the policy.

- **cdp**—CRLs are retrieved only from the CRL distribution points specified in authenticated certificates.

**Note**

SCEP retrieval is not supported by distribution points specified in certificates.

- **static**—CRLs are retrieved only from URLs you configure.
- **both**—CRLs are retrieved from CRL distribution points specified in authenticated certificates and from URLs you configure.

**Step 5** If you used the keywords **static** or **both** when you configured the CRL policy, you need to configure URLs for CRL retrieval, using the **url** command. You can enter up to 5 URLs, ranked 1 through 5.

```
hostname/contexta(config-ca-crl)# url n URL
```

where *n* is the rank assigned to the URL. To remove a URL, use the **no url *n*** command.

**Step 6** Configure the retrieval method with the **protocol** command. The following keywords for this command determine the retrieval method.

- **http**—Specifies HTTP as the CRL retrieval method.
- **ldap**—Specifies LDAP as the CRL retrieval method.
- **scep**—Specifies SCEP as the CRL retrieval method.

**Step 7** Configure how long the security appliance caches CRLs for the current trustpoint. To specify the number of minutes the security appliance waits before considering a CRL stale, enter the following command.

```
hostname/contexta(config-ca-crl)# cache-time n
```

where *n* is the number of minutes. For example, to specify that CRLs should be cached for seven hours, enter the following command.

```
hostname/contexta(config-ca-crl)# cache-time 420
```

**Step 8** Configure whether the security appliance requires the NextUpdate field in CRLs. For more information about how the security appliance uses the NextUpdate field, see the [“About CRLs” section on page 41-3](#).

Do one of the following:

- To require the NextUpdate field, enter the **enforcenextupdate** command. This is the default setting.
- To allow the NextUpdate field to be absent in CRLs, enter the **no enforcenextupdate** command.

**Step 9** If you specified LDAP as the retrieval protocol, perform the following steps:

- Enter the following command to identify the LDAP server to the security appliance:

```
hostname/contexta(config-ca-crl)# ldap-defaults server
```

You can specify the server by DNS hostname or by IP address. You can also provide a port number if the server listens for LDAP queries on a port other than the default of 389. For example, the following command configures the security appliance to retrieve CRLs from an LDAP server whose hostname is ldap1.

```
hostname/contexta(config-ca-crl)# ldap-defaults ldap1
```



**Note** If you use a hostname rather than an IP address to specify the LDAP server, be sure you have configured the security appliance to use DNS. For information about configuring DNS, see the **dns** commands in the *Cisco Security Appliance Command Reference*.

- b. If LDAP server requires credentials to permit CRL retrieval, enter the following command:

```
hostname/contexta(config-ca-crl)# ldap-dn admin-DN password
```

For example:

```
hostname/contexta(config-ca-crl)# ldap-dn cn=admin,ou=devtest,o=engineering c001RunZ
```

- Step 10** To test CRL configuration for the current trustpoint, use the **crypto ca crl request** command. This command retrieves the current CRL from the CA represented by the trustpoint you specify.
- Step 11** Save the running configuration. Enter the **write memory** command.

## Exporting and Importing Trustpoints

You can export and import keypairs and issued certificates associated with a trustpoint configuration. The security appliance supports PKCS12 format for the export and import of trustpoints.

This section includes the following topics:

- [Exporting a Trustpoint Configuration, page 41-15](#)
- [Importing a Trustpoint Configuration, page 41-16](#)

### Exporting a Trustpoint Configuration

To export a trustpoint configuration with all associated keys and certificates in PKCS12 format, use the **crypto ca export** command. The security appliance displays the PKCS12 data in the terminal. You can copy the data. The trustpoint data is password protected; however, if you save the trustpoint data in a file, be sure the file is in a secure location.

The following example exports PKCS12 data for trustpoint Main using Wh0zits as the passphrase:

```
hostname (config)# crypto ca export Main pkcs12 Wh0zits
```

```
Exported pkcs12 follows:
```

```
[ PKCS12 data omitted ]
```

```
---End - This line not part of the pkcs12---
```

```
hostname (config)#
```

## Importing a Trustpoint Configuration

To import the keypairs and issued certificates associated with a trustpoint configuration, use the **crypto ca import pkcs12** command in global configuration mode. The security appliance prompts you to paste the text to the terminal in base-64 format.

The key pair imported with the trustpoint is assigned a label matching the name of the trustpoint you create. For example, if an exported trustpoint used an RSA key labeled <Default-RSA-Key>, creating trustpoint named Main by importing the PKCS12 creates a key pair named Main, not <Default-RSA-Key>.

**Note**

If a security appliance has trustpoints that share the same CA, only one of the trustpoints sharing the CA can be used to validate user certificates. The **crypto ca import pkcs12** command can create this situation. Use the **support-user-cert-validation** command to control which trustpoint sharing a CA is used for validation of user certificates issued by that CA.

The following example manually imports PKCS12 data to the trustpoint Main with the passphrase Wh0zits:

```
hostname (config)# crypto ca import Main pkcs12 Wh0zits

Enter the base 64 encoded pkcs12.
End with a blank line or the word "quit" on a line by itself:
[ PKCS12 data omitted ]
quit
INFO: Import PKCS12 operation completed successfully
hostname (config)#
```

## Configuring CA Certificate Map Rules

You can configure rules based on the Issuer and Subject fields of a certificate. Using the rules you create, you can map IPSec peer certificates to tunnel groups with the **tunnel-group-map** command. The security appliance supports one CA certificate map, which can contain many rules. For more information about using CA certificate map rules with tunnel groups, see the [“Creating a Certificate Group Matching Rule and Policy” section on page 29-10](#).

To configure a CA certificate map rule, perform the following steps:

- Step 1** Enter CA certificate map configuration mode for the rule you want to configure. To do so, enter the **crypto ca certificate map** command and specify the rule index number. The following example enters CA certificate map mode for the rule with index number 1.

```
hostname(config)# crypto ca certificate map 1
hostname(config-ca-cert-map)#
```

- Step 2** Use the **issuer-name** and **subject-name** commands to configure the rule. These commands specify tests that the security appliance can apply to values found in the Issuer or Subject fields of certificates. The tests can apply to specific attributes or to the whole of the Issuer or Subject fields. You can configure many tests per rule, and all the tests you specify with these commands must be true for a rule to match a certificate. Valid operators in the **issuer-name** and **subject-name** commands are as follows.

Operator	Meaning
eq	The field or attribute must be identical to the value given.
ne	The field or attribute cannot be identical to the value given.
co	Part or all of the field or attribute must match the value given.
nc	No part of the field or attribute can match the value given.

For more information about the **issuer-name** and **subject-name** commands, see the *Cisco Security Appliance Command Reference*.

The following example specifies that any attribute within the Issuer field must contain the string ASC:

```
hostname(config-ca-cert-map)# issuer-name co asc
hostname(config-ca-cert-map)#
```

The following example specifies that within the Subject field an Organizational Unit attribute must exactly match the string Engineering.

```
hostname(config-ca-cert-map)# subject-name attr ou eq Engineering
hostname(config-ca-cert-map)#
```

Map rules appear in the output of the **show running-config** command.

```
crypto ca certificate map 1
  issuer-name co asc
  subject-name attr ou eq Engineering
```

- Step 3** When you have finished configuring the map rule, save your work. Enter the **write memory** command.

## The Local CA

The Local Certificate Authority (Local CA) performs the following tasks:

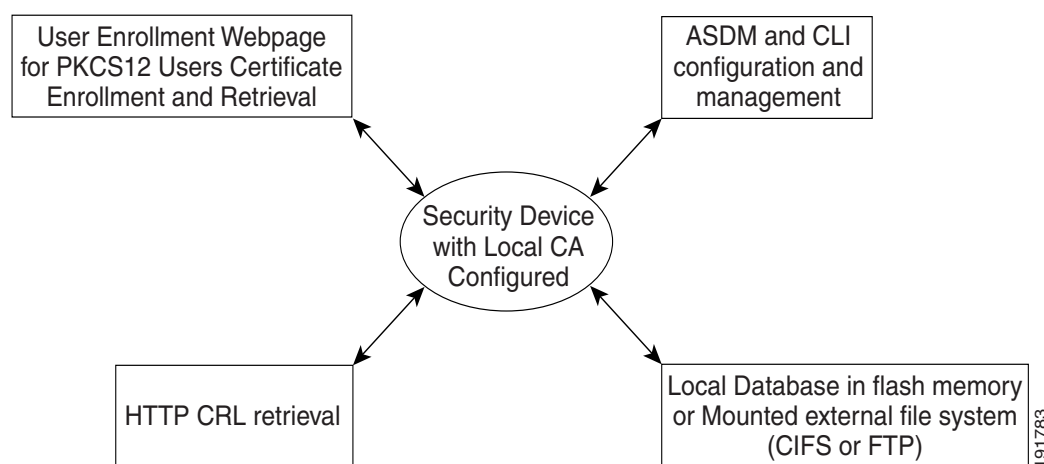
- Integrates basic certificate authority functionality on the security appliance.
- Deploys certificates.
- Provides secure revocation checking of issued certificates.
- Provides a certificate authority on the adaptive security appliance for use with SSL VPN connections, both browser- and client-based.
- Provides trusted digital certificates to users, without the need to rely on external certificate authorization.
- Provides a secure in-house authority for certificate authentication and offers straightforward user enrollment by means of a browser web page login.

**Note**

The Local CA feature is not supported if you use active/active failover or VPN load balancing. The Local CA cannot be subordinate to another CA; it can act only as the Root CA.

After you configure a Local CA server on the security appliance, users can enroll for a certificate by visiting a specified browser-based enrollment page and entering a username and a one-time password that is provided by the Local CA administrator to validate their eligibility for enrollment.

As shown in [Figure 41-1](#), the Local CA server, configurable from both the CLI and ASDM, resides on the security appliance and handles enrollment requests from web page users and CRL inquiries coming from other certificate validating devices and security appliances. Local CA database and configuration files are maintained either on the security appliance flash memory (default storage) or on a separate storage device.



**Figure 41-1**      *The Local Certificate Authority (CA)*

**Note**

Only one Local CA server can be resident on a security appliance at a time, and the Local CA cannot be configured as a subordinate to an external CA.

## Configuring the Local CA Server

This section describes how to configure the Local CA server on the security appliance and includes the following topics:

- [The Default Local CA Server, page 41-19](#)
- [Customizing the Local CA Server, page 41-20](#)
- [Certificate Characteristics, page 41-21](#)

## The Default Local CA Server

The default Local CA server requires only a few configuration commands to set up with the following characteristics. Once you use the **crypto ca server** command to access config-ca-server mode, all you must specify are CLI commands described in the following steps:

- 
- Step 1** Specify the SMTP (Simple Mail Transfer Protocol) from-address with the **smtp from-address** command. This command provides a valid e-mail address the Local CA uses as a from: address when sending e-mails that deliver one-time passwords for an enrollment invitation to users.
- Step 2** For an optional subject-name DN appended to each username on issued certificates, specify the subject-name DN with the **subject-name-default** command. The subject-name DN and the username combine to form the DN in all user certificates issued by the Local CA server. If you do not specify a subject-name DN, you must specify the exact subject name DN to be included in a user certificate each time you add a user to the user database.
- 

The following example shows the few CLI commands required to configure and enable the Local CA server when you are using the predefined default values for all required parameters.

```
hostname(config)# crypto ca server
hostname (config-ca-server) # smtp from-address SecurityAdmin@hostcorp.com
hostname (config-ca-server)# subject-name-default cn=engineer, o=asc Systems, c=US
hostname(config-ca-server)# no shutdown
```

All other required parameter values are the system defaults. [Table 41-1](#) lists the configurable characteristics of the Local CA server, their pre-defined default values, and the CLI commands that configure them.



### Note

**Issuer-name** and **keysize server** values cannot be changed after you enable the Local CA initially. Be sure to review all optional parameters carefully before you enable the configured Local CA.

**Table 41-1 Local CA Local CA Server Default Characteristics**

Local CA Server Characteristic	Default Value	CLI Configuration Command(s)
Storage Location for database and configuration	On-board flash memory in the directory LOCAL-CA-SERVER.	<b>mount</b> (global config mode) <b>database path</b>
Certificate Issuer Name	cn=FQDN	<b>issuer-name</b>
Enabled/disabled. <b>no-shutdown</b> enables the Local CA; <b>shutdown</b> disables it.	No Local CA Server configured.	<b>shutdown vs. no shutdown</b> (enables)
Access to config-ca-server mode and Local CA server configuration commands	No server enabled	<b>crypto ca server</b>
Issued certificate keypair size	1024 bits per key	<b>keysize</b>
Local CA Certificate key-pair size	1024 bits per key	<b>keysize server</b>
Length of time a user certificate, server certificate, or CRL is valid	User Certificate=1 yr.; Server Certificate=3 yrs.; CRL=6 hours	<b>lifetime</b>
Length of time a one-time password is valid	Expires in 72 hrs. (three days)	<b>otp-expiration</b>

Local CA Server Characteristic	Default Value	CLI Configuration Command(s)
Certificate Revocation List (CRL) Distribution Point (CDP), the location of the CRL on the Local CA security appliance or on an external server	For a local CRL, the same as security appliance, <code>http://hostname.domain/+CSCOCA+/asa_ca.crl</code>	<code>cdp-url</code>
* E-mail address issuing Local CA e-mail notices	<b>Required.</b> You must supply an e-mail address as the default, <code>admin@FQDN</code> , might not be an actual address.	<code>smtp from-address</code>
Subject line in Local CA e-mail notices	“Certificate Enrollment Invitation”	<code>smtp subject</code>
* subject-name DN default to append to a username on issued certificates	<b>Optional. No default.</b> Supply a subject-name default value.	<code>subject-name-default</code>
Days before expiration reminders are sent.	14 days prior to expiration	<b>renewal-reminder</b>
Post-enrollment/renewal period an issued certificate file is available for re-use.	24 hours	<b>enrollment-retrieval</b>

\*Indicates values without defaults that you must configure.

Once the **crypto ca server** command executes, the Local CA is generated. A self-signed certificate is created and associated with that Local CA on the security appliance when you execute the **no shutdown** command. The self-signed certificate key usage extension has key encryption, key signature, CRL signing, and certificate signing ability.

You can debug the configured default Local CA server with the **debug crypto ca server** command, which displays debug messages during configuration and test. This command is detailed further on in the section, [Enabling the Local CA Server](#).



#### Note

Once the self-signed Local CA certificate is generated, to modify its characteristics you must delete the existing Local CA server and completely recreate it.

## Customizing the Local CA Server

This section describes configuring and enabling the Local CA server. Enabling it for the first time generates the server certificate and keypair, which automatically produces a CA. To begin configuring the Local CA server you must be in `config-ca-server` mode.

Once you execute the **crypto ca server** command to enter `config-ca-server` mode, you can begin to configure the various parameters of the Local CA server on the security appliance. Typically, to configure a customized Local CA server on a security appliance, you would perform the following steps:

- Step 1** Enter the **crypto ca server** command to access the Local CA Server Configuration mode CLI command set, which allows you to configure and manage a Local CA. An example follows:

```
hostname (config)# crypto ca server
hostname (config-ca-server)#
```

- Step 2** As with the default Local CA server, you must specify the parameters that do not have defaults, specifically the issuer-name command. An example follows:

```
hostname (config-ca-server)# issuer-name
CN=xx5520,CN=30.132.0.25,ou=DevTest,ou=QA,O=ASC Systems
hostname (config-ca-server)#
```

- Step 3** To customize the text that appears in the subject field of all e-mails sent from the Local CA server, use the **smtp subject** *subject-line* command as follows:

```
hostname (config-ca-server) # smtp subject Priority E-Mail: Enclosed Confidential
Information is Required for Enrollment
hostname (config-ca-server)#
```

- Step 4** To specify the e-mail address that is to be used as the From: field of all e-mails generated by the Local CA server, use the **smtp from-address** command as follows:

```
hostname (config-ca-server) # smtp from-address SecurityAdmin@hostcorp.com
hostname (config-ca-server)#
```

- Step 5** To specify an optional subject-name DN to be appended to a username on issued certificates, use the **subject-name-default** command. The default subject-name DN becomes part of the username in all user certificates issued by the Local CA server. For example, if the username is maryjane@ASC.com and you set the subject-name default to cn=engineer, o=ASC Systems, c=US, the subject-name DN in the certificate would be cn=maryjane@ASC.com, cn=Engineer, o=ASC Systems, c=US.



**Note**

If you do not specify a subject-name-default to serve as a standard subject-name default, you must specify a DN each time you add a user.

The permitted DN attribute keywords are listed in the following table:

Subject-name-default Keywords	
CN= Common Name	C = Country
SN = Surname	OU = Organization Unit
T = Title	EA = E-mail Address
O = Organization Name	ST = State/Province
L = Locality	

An example follows:

```
hostname (config-ca-server) # subject-name-default cn=engineer, o=ABC Systems, c=US
hostname (config-ca-server)#
```

Note that there are additional Local CA server commands that allow you to customize your server further. These commands are described in the following sections.

## Certificate Characteristics

Configurable Local CA certificate characteristics include the following:

- The name of the certificate issuer as it appears on all user certificates
- The lifetime of the Local CA certificates (server and user) and the CRL
- The length of the public and private keypair associated with Local CA and user certificates.

## Issuer Name

The certificate issuer name that is configured is both the subject-name and issuer-name of the self-signed Local CA certificate, as well as the issuer-name in all client certificates that are issued and in the issued CRL. The default issuer name in the Local CA is *hostname.domainname*. Use the **issuer-name** command to specify the Local CA certificate subject-name as shown in the following example:

```
hostname(config-ca-server)# issuer-name CN=xx5520,CN=30.132.0.25,ou=DevTest,ou=QA,O=ABC Systems
hostname(config-ca-server)#
```



### Note

The **issuer-name** value cannot be changed after the initial enabling of the Local CA.

## CA Certificate Lifetime

You can specify the lifetime, the period of validity for the Local CA certificate, all issued user certificates, or the CRL with the **lifetime** command. This command determines the expiration date included in the certificate; the default lifetime of a Local CA certificate is three years.



### Note

Make sure that you limit the validity period of the certificate to less than the recommended end date of 03:14:08 UTC, January 19, 2038.

Use the **lifetime ca-certificate** command to set the number of days that you want the Local CA server certificate to remain valid as shown in the following example of configuring a Local CA certificate to last for one year:

```
hostname (config)# crypto ca server
hostname (config-ca-server)# lifetime ca-certificate 365
hostname (config-ca-server)
```

To reset the Local CA certificate lifetime to the default of three years during configuration, use the **no lifetime ca-certificate command**. You can use the same command (or its **no** form) to specify (or reset) the valid lifetime of user certificates (**lifetime certificate...**) and the CRL (**lifetime crl...**).

The Local CA Server automatically generates a replacement CA certificate 30 days prior to the CA certificate expiration, allowing the replacement certificate to be exported and imported onto any other devices for certificate validation of user certificates issued by the Local CA certificate after expiration of the current Local CA certificate. The pre-expiration Syslog message:

```
%ASA-1-717049: Local CA Server certificate is due to expire in <days> days and a replacement certificate is available for export.
```



### Note

When notified of this automatic rollover, the administrator must take action to ensure the new Local CA certificate is imported to all necessary devices prior to expiration.

## User Certificate Lifetime

To set the number of days that you want user certificates to remain valid, use the **lifetime certificate** command as shown in the following example of configuring all user certificates to be valid for two months:

```
hostname (config)# crypto ca server
hostname (config-ca-server)# lifetime certificate 60
hostname (config-ca-server)#
```

Prior to a user certificate expiring, the Local CA server automatically initiates certificate renewal processing by granting that user enrollment privileges a number of days ahead of the certificate expiration, renewal-reminder setting, and by delivering an e-mail with the enrollment username and OTP for renewal of the certificate.

**Note**

Make sure that you limit the validity period of the certificate to less than the recommended end date of 03:14:08 UTC, January 19, 2038.

## CRL Lifetime

The Local CA updates and reissues the CRL every time a user certificate is revoked or unrevoked, but if there are no revocation changes, the CRL is reissued automatically once every CRL lifetime, the period of time you specify with the **lifetime crl** command during Local CA configuration. If you do not specify a CRL lifetime, the default time period is six hours.

Use the **lifetime crl** command to set the number of hours that you want the certificate revocation list to remain valid as shown in the following example:

```
hostname (config)# crypto ca server
hostname (config-ca-server)# lifetime crl 10
hostname (config-ca-server)#
```

To force the issuance of a CRL at any time, you can use the **crypto ca server crl issue** command, which immediately updates and regenerates a current CRL to overwrite the existing CRL. This command can force the issuance of a CRL in any circumstances, such as a corrupt or destroyed CRL file.

This command displays a message indicating that the CRL is updated. An example follows:

```
hostname(config)# crypto ca server crl issue
A new CRL has been issued.
hostname(config)#
```

Note that it should never be necessary to use this command unless the CRL file is removed by mistake or is corrupted and needs to be regenerated from scratch.

## Server Keysize

The Local CA server keypair size can be configured independently of the user-issued certificate keypair size. The **keysize server** command is used to configure the size of the Local CA's own keypair. The **keysize** command specifies the size of the public and private keys generated at user-certificate enrollment. The **keysize server** command is illustrated in the following example:

```
hostname(config)# crypto ca server
hostname(config-ca-server)# keysize server 2048
hostname(config-ca-server)#
```

For both the **keysize** command and the **keysize server** command, key-pair size options are 512, 768, 1024, 2048 bits, and both commands have default values of 1024 bits.

**Note**

The Local CA keysize cannot be changed once the Local CA is enabled without deleting the Local CA and reconfiguring a new Local CA. This would invalidate all issued certificates.

## Defining Storage for Local CA Files

The security appliance accesses and implements user information, issued certificates, revocation lists, and so forth using a Local CA database. That database resides in local flash memory by default or can be configured to be on an off-box file system that is mounted and accessible to the security appliance.

### Default Flash Memory Data Storage

By default, the Local CA server database is stored in flash memory, a nonvolatile storage space that stores the configuration and database files when the security appliance is powered down.

There are no limits on the number of users that can be in the Local CA user database; however, if flash memory storage issues arise, syslog messages are generated to alert the administrator to take action, and the Local CA could be disabled until the storage problems are solved. Flash memory can store a database with 3500 users or less, but a database of more than 3500 users requires off-box storage.

### Setting up External Local CA File Storage

Storage for Local CA files on a server external to the security appliance requires an already mounted file system of file type CIFS or FTP that is username- and password-protected to secure the stored information. With the file system mounted, you then can establish a path to the server and specify the file or folder name for the Local CA to use for file storage and retrieval.

Configure the file system path with the **database path** command. To return Local CA file storage to the security appliance flash memory, use the **no database path** command.

To specify external off-box storage for the Local CA, perform the following steps:

- 
- Step 1** Enter the **mount** command with a file system label and type in global configuration mode. This lets the security appliance access the configuration mode for the specific file system type. An example that mounts a CIFS file system follows:

```
hostname(config)# mount mydata type cifs
hostname(config-mount-cifs)# mount mydata type cifs
server 99.1.1.99 share myshare
domain frqa.ASC.com
username user6
password *****
status enable
hostname(config-mount-cifs)#
```

- Step 2** Use the **database path** command to specify the location of mydata, the pre-mounted CIFS file system to be used for the Local CA server database.

```
hostname(config)# crypto ca server
hostname(config-ca-server)# database path mydata:newuser
hostname(config-ca-server)#
```



**Note**

Only the user who mounts a file system can un-mount it with the **no mount** command.

---

## CRL Storage

The Certificate Revocation List (CRL) exists for other devices to validate the revocation of certificates issued by the Local CA. In addition, the Local CA tracks all issued certificates and status within its own certificate database. Revocation checking is done when a validating party needs to validate a user certificate by retrieving the revocation status from an external server, which might be the CA that issued the certificate or a server designated by the CA.

If you do not configure a specific location for the CDP, the default location URL is `http://hostname.domain/+CSCOCA+/asa_ca.crl`. To establish a specific location for the Local CA's automatically generated CRL, use the **cdp-url** command to specify the certificate revocation list distribution point (CDP) to be included in all issued certificates. An example follows:

```
hostname(config)# crypto ca server
hostname(config-ca-server)# cdp-url http://99.1.1.99/pathname/myca.crl
```

The Local CA updates and reissues the CRL every time a user certificate is revoked or unrevoked. If there are no revocation changes, the CRL is reissued once every CRL lifetime, the period of time you specify with the **lifetime** command during Local CA configuration. An example follows:

If you do not specify a CRL lifetime, the default time period is six hours.

```
hostname(config)# crypto ca server
hostname (config-ca-server)# lifetime crl 72
```

If the **cdp-url** command is set to serve the CRL directly from the Local CA security appliance, use the **publish-crl** CLI command to open a port on an interface to make the CRL accessible from that interface. The **publish-crl** command is detailed in the following section.

## CRL Downloading

To make the CRL available for HTTP download on a given interface or port, use the **publish-crl** command in config-ca-server mode. The specified interface and port are used to listen for incoming requests for the CRL. Interface options are:

inside	name of interface GigabitEthernet0/1
management	name of interface Management0/0
outside	name of interface GigabitEthernet0/0

The optional port option can be any port number in a range of 1-65535, and TCP port 80 is the HTTP default port number. For example, to specify port 70 for outside access to the CRL, use the following command:

```
hostname (config)# crypto ca server
hostname (config-ca-server)#publish-crl outside 70
hostname (config-ca-server)#
```

The CDP URL can be configured to utilize the IP address of an interface, and the path of the CDP URL and the file name can be configured also. For example, the CDP URL could be configured to be:

```
http://10.10.10.100/user8/my_crl_file
```

In this case only the interface with that IP address configured listens for CRL requests, and when a request comes in, the security appliance matches the path /user8/my\_crl\_file to the configured CDP URL. When the path matches, the security appliance returns the CRL file stored in storage. Note that the protocol must be http, so the prefix is http://.

**Note**

If you do not specify a **publish-crl** command, the CRL is not accessible from the CDP location because the **publish-crl** command is required in order to open an interface for downloading the CRL file.

## Enrolling Local CA Users

Each user who wishes to be enrolled as a Local CA user must be added to the Local CA server user database. User enrollment is initiated by the Local CA administrator who adds new users to the database with the **crypto ca server user-db add** command.

Next, the administrator issues a **crypto ca server user-db allow...** command, and, if email-OTP is specified, the Local CA Server e-mails a one-time-password and username to the new user to enable enrollment. The e-mail, an automatically generated message, contains the enrollment URL of the security appliance. Figure 41-2 shows a sample e-mail to a new user.

---

Date: 12/22/06  
To: wuser6@wuser.com  
From: Wuseradmin  
Subject: Certificate Enrollment Invitation

You have been granted access to enroll for a certificate.

The credentials below can be used to obtain your certificate.  
Username: [wuser6@wuser.com](mailto:wuser6@wuser.com)  
One-time Password: C93BBB733CD80C74  
Enrollment is allowed until: 15:54:31 UTC Thu Dec 27 2006

NOTE: The one-time password is also used as the passphrase to unlock the certificate file.

Please visit the following site to obtain your certificate:  
<https://wu5520-FO.frdevtestad.local/+CSCOCA+/enroll.html>  
You may be asked to verify the fingerprint/thumbprint of the CA certificate during installation of the certificates. The fingerprint/thumbprint should be:  
MD5: 76DD1439 AC94FDBC 74A0A89F CB815ACC  
SHA1: 58754FFD 9F19F9FD B13B4B02 15B3E4BE B70B5A83

---

**Figure 41-2**      **Sample Local CA Enrollment E-mail**

When a user enrolls successfully, a PKCS12 file is created, which contains a keypair and a certificate issued to the user, along with the Local CA certificate. The user must browse to the enrollment interface and enter a valid username and one-time password. Once the Local CA authenticates the user's credentials within the enrollment time frame, the user is permitted to download the newly generated certificate, which is included in a PKCS12 file.

The PKCS12 file contents are protected by a passphrase, the One-Time-Password (OTP). The OTP can be handled manually, or this file can be e-mailed to the user by the Local CA to download once the administrator allows enrollment.

The file is saved to storage temporarily as username.p12. This file contains the user certificate, the keypair, and the Local CA certificate. To install these certificates on the user's PC, the user is prompted for the passphrase (one-time password) for the file, the same one-time password used to authenticate the user to the Local CA.

With the file in storage, the user can return within the enrollment-retrieval time period to retrieve the file a second or subsequent times as needed. When the time period expires, the file is removed from storage automatically and is no longer available for downloading.

## Setting Up Enrollment Parameters

For a secure enrollment process, the Local CA automatically generates one-time passwords (OTPs), which are e-mailed to enrolling users at the e-mail address the administrator configures. OTPs can be handled manually but are e-mailed if configured with an e-mail address when the user is added to the database. In order to complete enrollment and receive a certificate, the user must enter the OTP in the enrollment interface along with a username in order to complete enrollment.

Each unique OTP has a configurable window of time in which it can be used to retrieve a certificate. If the OTP expiration period expires before the user retrieves the PKCS12 enrollment file that contains the user certificate, enrollment is not permitted. The **otp expiration** command defines the amount of time the OTP is valid for user enrollment.

The **enrollment-retrieval** command specifies the time in hours that an enrolled user can retrieve a certificate. An example of setting up enrollment parameters follows:

- Step 1** Enter the **crypto ca server** command to access the Local CA Server Configuration mode. An example follows:

```
hostname(config)# crypto ca server
hostname (config-ca-server)#
```

- Step 2** Specify the number of hours (24) that an issued One-Time Password (OTP) for the local Certificate Authority (CA) enrollment page is valid with the **otp expiration** command. This time period begins when the user is allowed to enrol. The default expiration time of 72 hours can be changed to 24 as follows:

```
hostname(config-ca-server)# otp expiration 24
hostname(config-ca-server)#
```



### Note

The user OTP for enrolling for a certificate with the enrollment interface page is also used as the password to unlock the PKCS12 file containing that user's issued certificate and keypair.

- Step 3** Specify the number of hours an already-enrolled user can retrieve a PKCS12 enrollment file with the **enrollment-retrieval** command. This time period begins when the user is successfully enrolled. This command modifies the default 24-hours retrieval period to any value between one and 720 hours. Note that enrollment retrieval period is independent of the OTP expiration period. The following example sets the retrieval time to 120 hours (five days).

```
hostname(config)# crypto ca server
hostname(config-ca-server)# enrollment-retrieval 120
hostname(config-ca-server)#
```

After the enrollment-retrieval time expires, the user certificate and keypair are no longer available; the only way for the user to receive a certificate is for the administrator to reinitialize certificate enrollment by allowing the user to enroll again.

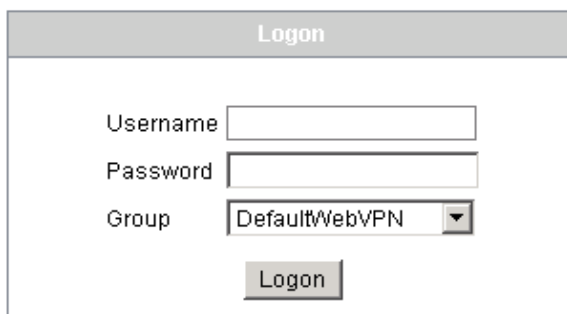
For the CLI commands that let you display and view the database entries, refer to the section [Displaying Local CA Server Information](#) further on in this chapter.

## Enrollment Requirements

End-users enroll for a certificate by visiting the Local CA Enrollment Interface webpage and entering a username and one-time password. Enrolling as a user on the Local CA server initially requires valid user credentials, which typically are a username and a password.

When a user enrolls, the Local CA generates the user certificate and provides a link so the user can install the certificate on the client machine. The user's private keypair is generated by the Local CA and is issued to the user as part of the PKCS12 file. The PKCS12 file includes a keypair and the certificate issued to the user and the Local CA certificate.

The Local CA WebVPN login screen is provided in the following figure:



## Starting and Stopping the Local CA Server

When you complete Local CA Server configuration, to activate it, use the **no shutdown** command. To disable enrollment and/or to modify the configuration, use the **shutdown** command.

### Enabling the Local CA Server

Initially, you need to specify a passphrase to create and protect the archive of the CA certificate and keypair that are generated. The passphrase unlocks the PKCS12 archive in case the CA certificate or keypair are lost.

Once you enable the Local CA server, with the **no shutdown** command, it generates the Local CA server certificate, keypair and necessary database files, and also archives the Local CA server certificate and keypair to storage in a PKCS12 file. After the initial startup, you can issue **no shutdown** and **shutdown** commands that enable and disable the Local CA without being prompted for the passphrase.



#### Note

Once you enable the Local CA Server, be sure to save the configuration to ensure that the Local CA certificate and keypair are not lost after a reboot.

At initial startup, you are prompted for the passphrase in the CLI as illustrated in the example that follows. To enable the Local CA server on a security appliance, perform the following steps:

**Step 1** Create a password (7-character min.) in order to encode and archive a PKCS12 file containing the Local CA certificate and keypair that is to be generated.

**Step 2** Enter the following command to enable the Local CA server on the security appliance. The command requires an 8-65 alphanumeric character password:

```
hostname(config)# crypto ca server
hostname(config-ca-server)# no shutdown
hostname(config-ca-server)#

hostname(config-ca-server)# no shutdown

% Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key
% or type Return to exit

Password: caserver

Re-enter password: caserver

Keypair generation process begin. Please wait...

hostname(config-ca-server)#
```

Re-enabling the same Local CA Server with the **no shutdown** command and disabling it with the **shutdown** command do not require the passphrase.

## Debugging the Local CA Server

To debug the newly configured Local CA Server, use the **debug crypto ca server** command in global configuration mode. This command displays debug messages when you configure and enable the Local CA server. By default, the **debug crypto ca server** command performs level 1 debug functions; levels 1-255 are available.



### Note

Debug commands might slow down traffic on busy networks. Levels 5 and higher are reserved for raw data dumps and should be avoided during normal debugging because of excessive debug output.

## Disabling the Local CA Server

When you disable the Local CA server with the **shutdown** command, the configuration and all associated files remain in storage. Webpage enrollment is disabled, but you can change or reconfigure the Local CA Server during shutdown and then restart it with the **no shutdown** command.

To disable the Local CA server on a security appliance, perform the following:

```
asal(config-ca-server)#
asal(config-ca-server)# shutdown
INFO: Local CA Server has been shutdown.
asal(config-ca-server)#
```

## Managing the Local CA User Database

The Local CA server keeps track of user certificates, so the administrator can revoke or restore privileges as needed. This section describes how to add, allow for enrollment, remove, and manage users in the Local CA database with CLI commands. These operations are all initiated with the **crypto ca server user-db** (*function*) command in Privileged Exec mode. The functions are summarized in [Table 41-2 Crypto CA Server User Commands](#) and described in the following subsections.

Note that users must be added to the database with the **crypto ca server user-db add** command, but it is the **crypto ca server user-db allow** command that grants each user enrollment privileges.

**Table 41-2** *Crypto CA Server User Commands*

Command	Description
<b>crypto ca server user-db add</b>	Adds a user to the Local CA server user database. If a DN string contains a comma, enclose the value string with double quotes (for example, O="Company, Inc.>").
<b>crypto ca server user-db allow</b>	Permits a specific user or subset of users in the Local CA server database to enroll and generates OTPs for users.
<b>crypto ca server user-db remove</b>	Removes a user from the Local CA server user database by username.
<b>crypto ca server user-db email-otp</b>	E-mails the one-time password to a specific user or to a subset of users in the Local CA server database.
<b>crypto ca server user-db show-otp</b>	Displays the one-time password for a specific user or a subset of users in the Local CA server database.

## Adding and Enrolling Users

Both the **crypto ca server user-db add** command and the **crypto ca server user-db allow** command are used to add and allow new Local CA users. To add a user who is eligible for enrollment to the Local CA database, perform the following steps:

**Step 1** Add a new user with the following CLI commands:

```
hostname (config)#
hostname (config-ca-server)# crypto ca server user-db add username [dn dn] [email emailad-
dress]
hostname (config-ca-server)#
```

where the options are as follows:

- username**—A string from 4-64 characters, the simple user name for the user being added. The username can be an e-mail address, which then is used to contact the user as necessary for enrollment invitations
- dn**— distinguished name, a global, authoritative name of an entry in the OSI Directory (X.500), for example, cn=maryjane@ASC.com, cn=Engineer, o=ASC Systems, c=US. For details, see [Customizing the Local CA Server](#)
- e-mail-address**—The e-mail address of the new user where OTPs and notices are to be sent.

**Step 2** Provide user privileges to an added user with the following command:

```
hostname (config)#
```

```
hostname(config-ca-server)# crypto ca server user-db allow user6
hostname(config-ca-server)#
```

- Step 3** Notify a user in the Local CA database to enroll and download a user certificate with the **crypto ca server user-db email-otp** command, which automatically e-mails the one-time password to that user.

```
hostname(config)#
hostname(config-ca-server)# crypto ca server user-db email-otp username
hostname(config-ca-server)#
```

If the user specifies the e-mail address in the **crypto ca server user-db add** command, it is to send the e-mail as part of the **crypto ca server user-db allow** command or after using the **crypto ca server user-db email-otp** command. When an administrator wants to be able to notify a user by means of e-mail, the e-mail address must be specified as the username or the e-mail field when adding the user.

Once a user is added with a valid e-mail address, the administrator has choice of **crypto ca server user-db allow username email-otp**, or **crypto ca server user-db allow username** and **crypto ca server user-db email-otp username**.

Alternatively, you could specify the email address in step 2, and omit the **crypto ca server user-db email-otp** command. To view the one-time-password issued, use the **crypto ca server user-db show-otp** command. You can use a separate **show-otp** command in order to communicate the OTP to the user by other means

Once a user enrolls within the time limit with the correct OTP, the Local CA Server generates a keypair for the user and a user certificate based on the public key from the keypair generated and the subject-name DN specified with the DN field when the user is added or the subject-name-default setting if not specified. The enrollment time limit is set with the **otp-expiration** command, and the expiration date for the user certificate is specified during configuration with the **lifetime certificate** command.

## Renewing Users

Renewing a user certificate is similar to the initial enrollment process. Each user certificate has an expiration date, and Local CA automatically reminds the user by e-mail to renew before the time period runs out. If a certificate expires, it becomes invalid. Renewal notices and the times they are e-mailed to users are variable and can be configured by the administrator during Local CA server configuration.

To specify the timing of renewal notices, use the **renewal-reminder** command to specify the number of days (1-90) prior to Local CA certificate expiration that an initial reminder to re-enroll is sent to certificate owners.

```
hostname(config)# crypto ca server
hostname(config-ca-server)# renewal-reminder 7
hostname(config-ca-server)#
```

There are three reminders in all, and an automatic e-mail goes out to the certificate owner for each of the three reminders, provided an e-mail address is specified in the user database. If no e-mail address exists for the user, a syslog message alerts you of the renewal requirement.

The security appliance automatically grants certificate renewal privileges to any user who holds a valid certificate that is about to expire provided the user still is in the user database. Therefore, if an administrator does not want to allow a user to renew automatically, the user must be removed from the database prior to the renewal time period.

## Revoking Certificates and Removing or Restoring Users

Any time that user is to have a valid certificate revoked, use the **crypto ca server revoke** command to mark the certificate as revoked in the certificate database on the CA server and in the CRL, which is automatically reissued. To revoke a user certificate, enter the certificate serial number in hex format as shown in the following example, which revokes the certificate with the serial number 782ea09f:

```
hostname(config-ca-server)## crypto ca server revoke 782ea09f
Certificate with the serial number 0x782ea09f has been revoked. A new CRL has been issued.
hostname(config-ca-server)#
```

Note that the CRL is regenerated automatically after the specified certificate is revoked.

To restore a user and unvoke a previously revoked certificate issued by the Local CA server, use the **crypto ca server unrevoke** command.

If you delete a user from the user database by username with the **crypto ca server user-db remove** command, you are prompted to permit revocation of any valid certificates issued to the user.

## Revocation Checking

The Local CA maintains a current Certification Revocation List (CRL) with serial numbers of all revoked user certificates. This list is available to external devices and can be retrieved directly from the Local CA if it is configured as such with the **cdp-url** and the **publish-crl** CLI commands. When you revoke (or unrevoke) any current certificate, by certificate serial number, the CRL reflect these changes.

## Displaying Local CA Server Information

There are various ways to display and print the Local CA server configuration and user information as described in the following subsections. The following table summarizes the Local CA Server CLI commands that display configuration and database information.

Command	Display
show crypto ca server	Local CA configuration and status
show crypto ca server cert-db	User certificate(s)
show crypto ca server certificate	Local CA certificate
show crypto ca server crl	Certificate Revocation List
show crypto ca server user-db	Users and their status
show crypto ca server user-db allowed	Users eligible to enroll.
show crypto ca server user-db enrolled	Enrolled users with valid certificate
show crypto ca server user-db expired	Users with an expired certificate.
show crypto ca server user-db on-hold	Users without certificate not permitted to enroll

## Display Local CA Configuration

To display the characteristics of the configured Local CA, use the **show crypto ca server** command in Privileged EXEC mode. The following is a sample **show crypto ca server** display.

```
Certificate Server LOCAL-CA-SERVER:
  Status: enabled
  State: enabled
  Server's configuration is locked (enter "shutdown" to unlock it)
  Issuer name: CN=wz5520-1-16
  CA certificate fingerprint/thumbprint: (MD5)
    76dd1439 ac94fdb7 74a0a89f cb815acc
  CA certificate fingerprint/thumbprint: (SHA1)
    58754ffd 9f19f9fd b13b4b02 15b3e4be b70b5a83
  Last certificate issued serial number: 0x6
  CA certificate expiration timer: 14:25:11 UTC Jan 16 2008
  CRL NextUpdate timer: 16:09:55 UTC Jan 24 2007
  Current primary storage dir: flash:
```

## Display Certificate Database

To display a list with all of the certificates issued by the Local CA, use the **show crypto ca server cert-db command** in Privileged EXEC mode. The following is a sample **show crypto ca server cert-db command** display showing just two of the user certificates in the database.

```
Username: emily1
Renewal allowed until: Not Allowed
Number of times user notified: 0
PKCS12 file stored until: 12:45:52 UTC Fri Jan 4 2008
Certificates Issued:
serial:    0x71
issued:    12:45:52 UTC Thu Jan 3 2008
expired:    12:17:37 UTC Sun Dec 31 2017
status:    Not Revoked
Username: fred1
Renewal allowed until: Not Allowed
Number of times user notified: 0
PKCS12 file stored until: 12:27:59 UTC Fri Jan 4 2008
Certificates Issued:
serial:    0x2
issued:    12:27:59 UTC Thu Jan 3 2008
expired:    12:17:37 UTC Sun Dec 31 2017
status:    Not Revoked
<--- More --->
```

## Display the Local CA Certificate

To display the certificate of the Local CA on the console use the **show crypto ca server certificate** command in Privileged EXEC mode. The certificate displays in base 64 format and can be cut-and-pasted as an import into other devices that need the local CA certificate. A sample display follows:

The base64 encoded local CA certificate follows:

```
MIIXlwIBAzCCF1EGCSqGSIb3DQEHAaCCF0IEghc+MIIXOjCCFzYGCsqGSIb3DQEHbqCCFycwghc+jAgEAMIIXHAYJKo
ZiIhvcNAQcBMBsGCiqGSIb3DQEMAQMwDQIjph4SxJyTgCAQGAghbw3v4bFy+GGG2dJnB4OLphsUM+IG3SDOiDwZG9
n1SvtMieoxd7Hxknxbum06JDrujWktHBIqkrm+td34qlNEliGeP2YC94/NQ2z+4kS+uZzwcRh11KEZTS1E4L0fSaC3
uMTxJq2NUHYWmoc8pi4CieLj3h7VVMY6qbx2AC8I+q57+QG5vG515Hi5imwtYfaWwPEdPQxaWZPrzoG1J8BFqdPa1j
BGhAzzuSmElm3j/2dQ3Atro1G9nIsRHgV39fcBgwz4fEabHG7/Vanb+fj81d5n1OiJjDYYbP86tVbZ2yOVZR6aKFVI
0b2AfCr6Pbwfc9U8Z/af3BCyM2sN2xPJrXva94CaYrqyotZdAkSYA5KWScyEcgdqmuBeGDKOncTknfgy0XM+fG5rb3
qAXylGkjyFI5Bm9Do6RUROoG1DSrQrKeq/hj...
```

END OF CERTIFICATE

## Display the CRL

To display the Local CA CRL, use the **show crypto ca server crl** command as follows:

```
hostname(config)# show crypto ca server crl
```

Certificate Revocation List:

```
Issuer: cn=xx5520-1-3-2007-1
This Update: 13:32:53 UTC Jan 4 2008
Next Update: 13:32:53 UTC Feb 3 2008
Number of CRL entries: 2
CRL size: 270 bytes
```

Revoked Certificates:

```
Serial Number: 0x6f
Revocation Date: 12:30:01 UTC Jan 4 2008
Serial Number: 0x47
Revocation Date: 13:32:48 UTC Jan 4 2008
```

```
hostname(config)#
```

## Display the User Database

To display users in the CA server user database, use the **show crypto ca server user-db** command. This command can be used with qualifiers to reduce number of records displayed. Qualifiers are:

- **allowed** show only users currently allowed to enroll.
- **enrolled** Show only users that are enrolled and hold a valid certificate
- **expired** Show only users holding expired certificates.
- **on-hold** List only users without a certificate and not currently allowed to enroll.

The following example shows the resulting display (edited) for the entire database with no qualifiers.

```
hostname (config)# show crypto ca server user-db
username: wilma24
email: wilma24@xxrown.com
```

```

dn:          CN=mycn,OU=Sales,O=ASC.com,L=Franklin,ST=Mass,C=US
allowed:     12:29:08 UTC Sun Jan 6 2008
notified:    1
.
.
.
username:    wilma98
email:       wilma98@xxrown.com
dn:          CN=mycn,OU=Sales,O=ASC.com,L=Franklin,ST=Mass,C=US
allowed:     12:29:18 UTC Sun Jan 6 2008
notified:    1

username:    wilma99
email:       wilma99@xxrown.com
dn:          CN=mycn,OU=Sales,O=ASC.com,L=Franklin,ST=Mass,C=US
allowed:     12:29:18 UTC Sun Jan 6 2008
notified:    1
hostname(config)#

```

The following example shows the display of the **show crypto ca server user-db** command when the **on-hold** qualifier is used yielding just one user on-hold:

```

hostname (config)# show crypto ca server user-db on-hold
username: wilma101
email:      <None>
dn:         <None>
allowed:    <not allowed>
notified:   0
hostname (config)#

```

## Local CA Server Maintenance and Backup Procedures

The stored Local CA Server configuration, users, issued certificates, CRL, etc. reside in the database in flash memory, or in file-system storage, depending on how you configure storage. The following subsections describe database maintenance procedures.

### Maintaining the Local CA User Database

Each time the security appliance configuration is saved, all user information in the Local CA Server database is saved automatically (with the **write memory** command) to the file specified by the **database path** command when you set up file storage external to the security appliance. For example, if you set up file storage using the following command:

```

hostname(config)# crypto ca server
hostname(config-ca-server)# database path mydata:newuser
hostname(config-ca-server)#

```

User database information is saved from the security appliance to *mydata/newuser* every time you save the security appliance configuration.

**Note**

For flash memory database storage, the user information is saved automatically to the default location for the start-up configuration.

## Maintaining the Local CA Certificate Database

The certificate database file, LOCAL-CA-SERVER.cdb, is to be saved anytime there is a change in the database.

- LOCAL-CA-SERVER.p12 is the archive of the Local CA certificate and keypair generated when the Local CA server is initially enabled with the **no shutdown** command.
- LOCAL-CA-SERVER.crl file is the actual CRL.
- LOCAL-CA-SERVER.ser file is used to keep track of the issued certificate serial numbers

The Local CA files can be seen on the flash memory or in external storage as follows:

```
hostname(config-ca-server)# dir LOCAL* //
```

Directory of disk0:/LOCAL\*

75	-rwx	32	13:07:49 Jan 20 2007	LOCAL-CA-SERVER.ser
77	-rwx	229	13:07:49 Jan 20 2007	LOCAL-CA-SERVER.cdb
69	-rwx	0	01:09:28 Jan 20 2007	LOCAL-CA-SERVER.udb
81	-rwx	232	19:09:10 Jan 20 2007	LOCAL-CA-SERVER.crl
72	-rwx	1603	01:09:28 Jan 20 2007	LOCAL-CA-SERVER.p12

127119360 bytes total (79693824 bytes free)  
hostname (config-ca-server)#

## Local CA Certificate Rollover

Thirty days prior to the expiration of the Local CA certificate, a rollover replacement certificate is generated, and a syslog message informs the administrator that it is time for Local CA rollover. The new Local CA certificate must be imported onto all necessary devices prior to the expiration of the current certificate. If the administrator does not respond by installing the rollover certificate as the new Local CA certificate, validations can begin to fail.

The Local CA certificate rolls over automatically upon expiration using the same keypair. The rollover certificate is available for export in base64 format and can be displayed using the **crypto ca server certificate** command, which displays both the current and the rollover certificates. This command shows information about the rollover certificate when available, including the thumbprint of the rollover certificate for verification of the new certificate during import on other devices.

## Archiving the Local CA Server Certificate and Keypair

For backup purposes, you can use FTP or TFTP to copy the Local CA Server certificate and keypair and all files from the security appliance. An example follows:

```
hostname#  
hostname# copy LOCAL-CA-SERVER_0001.p12 tftp://90.1.1.22/user6/
```

**Note**

---

Back up all Local CA files as often as possible.

---

## Deleting the Local CA Server

**Note**

---

Deleting the Local CA Server removes the configuration from the security appliance. Once deleted, the configuration is unrecoverable.

---

To delete the existing Local CA server, whether it is enabled or disabled, you must issue a **no crypto ca server** command or a **clear config crypto ca server** command in Global Configuration mode, and then delete the associated database and configuration files (all files with the wildcard name, LOCAL-CA-SERVER.\*).

