CHAPTER 26

Configuring PKI Using the IPsec VPN SPA

This chapter provides information about configuring PKI-related features using the IPsec VPN SPA on the Catalyst 6500 Series switch. It includes the following sections:

Note The procedures in this chapter assume you have some familiarity with PKI configuration concepts. For detailed information about PKI configuration concepts and IPsec cryptographic operations and policies, refer to the following Cisco IOS documentation:

Cisco IOS Security Configuration Guide, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

Cisco IOS Security Command Reference, *Release 12.2*, at this URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

Note For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For additional information about the commands used in this chapter, see the the *Catalyst 6500 Series Cisco IOS Command Reference, Release 12.2SX* and the related Cisco IOS Release 12.2 software configuration guide and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xliv.

Tip To ensure a successful configuration of your VPN using the IPsec VPN SPA, read all of the configuration summaries and guidelines before you perform any configuration tasks.

Overview of PKI

Cisco IOS public key infrastructure (PKI) provides certificate management to support security protocols such as IP Security (IPsec), secure shell (SSH), and secure socket layer (SSL).

A PKI is composed of the following entities:

- Peers communicating on a secure network
- At least one certificate authority (CA) that grants and maintains certificates
- Digital certificates, which contain information such as the certificate validity period, peer identity information, encryption keys that are used for secure communications, and the signature of the issuing CA
- An optional registration authority (RA) to offload the CA by processing enrollment requests
- A distribution mechanism (such as Lightweight Directory Access Protocol (LDAP) or HTTP) for certificate revocation lists (CRLs)

PKI provides customers with a scalable, secure mechanism for distributing, managing, and revoking encryption and identity information in a secured data network. Every entity (a person or a device) participating in the secured communications is enrolled in the PKI , a process where the entity generates a Rivest, Shamir, and Adelman (RSA) key pair (one private key and one public key) and has their identity validated by a trusted entity (also known as a CA or trustpoint).

After each entity enrolls in a PKI, every peer (also known as an end host) in a PKI is granted a digital certificate that has been issued by a CA. When peers must negotiate a secured communication session, they exchange digital certificates. Based on the information in the certificate, a peer can validate the identity of another peer and establish an encrypted session with the public keys contained in the certificate.

Configuring PKI involves the following tasks:

- Deploying Rivest, Shamir, and Adelman (RSA) keys within a public key infrastructure (PKI). An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router; that is, the end host must generate a pair of RSA keys and exchange the public key with the certificate authority (CA) to obtain a certificate and enroll in a PKI.
- Configuring authorization and revocation of certificates within a PKI. After a certificate is validated as a properly signed certificate, it is authorized using methods such as certificate maps, PKI-AAA, or a certificate-based access control list (ACL). The revocation status is checked by the issuing certificate authority (CA) to ensure that the certificate has not been revoked.

- Configuring certificate enrollment, which is the process of obtaining a certificate from a certificate authority (CA). Certificate enrollment occurs between the end host requesting the certificate and the CA. Each peer that participates in the public key infrastructure (PKI) must enroll with a CA. Various methods are available for certificate enrollment.
- Storing public key infrastructure (PKI) credentials, such as Rivest, Shamir, and Adelman (RSA) keys and certificates. These credentials can be stored in the default location on the router, which is NVRAM, or other locations.

Configuring Multiple RSA Key Pairs

The multiple RSA key pair support feature allows you to configure a Catalyst 6500 Series switch to have multiple Rivest, Shamir, and Adelman (RSA) key pairs. The Cisco IOS software can maintain a different key pair for each identity certificate.

Before this feature, Cisco IOS public key infrastructure (PKI) configurations allowed either one general-purpose key pair or a set of special-purpose key pairs (an encryption and a signing key pair). The scenarios in which the key pairs were deployed often required configurations that required the switch to enroll with multiple certificate servers because each server has an independent policy and may also have different requirements regarding general-purpose versus special-purpose certificates or key length. With this feature, a user can configure different key pairs for each certification authority (CA) with which the switch enrolls and can match policy requirements for each CA without compromising the requirements specified by the other CAs, such as key length, key lifetime, and general-purpose versus special-usage keys.

Multiple RSA Key Pairs Configuration Guidelines and Restrictions

When configuring multiple RSA key pair support, follow these guidelines and restrictions:

- We recommend that Secure Socket Layer (SSL) or other PKI clients do not attempt to enroll with the same CA multiple times.
- Internet Key Exchange (IKE) will not work for any identity that is configured to use a named key pair. If an IKE peer requests a certificate from a PKI trustpoint that is using multiple key support, the initial portion of the exchange will work, that is, the correct certificate will be sent in the certificate response; however, the named keypair will not be used and the IKE negotiation will fail.
- Whenever you regenerate a key pair, you must always reenroll the certificate identities with that key pair.

To configure an RSA key pair, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto key generate rsa [usage-keys | general-keys] [modulus modulus-size] [key-pair-label]`	Generates RSA key pairs. - **usage-keys**—(Optional) Specifies that two special-usage key pairs should be generated, instead of one general-purpose key pair. - **general-keys**—(Optional) Specifies that the general-purpose key pair should be generated. - *key-pair-label*—(Optional) Specifies the name of the key pair that the switch will use. (If this argument is enabled, you must specify either **usage-keys** or **general-keys**.) - **modulus** *modulus-size*—(Optional) Specifies the modulus for generating the RSA keys. The range is 384 to 2048 bits, and the modulus must be a multiple of 64. The default is 1024.
Step 2	`Router(config)# crypto pki trustpoint name`	Declares the CA that the switch should use and enters ca-trustpoint configuration mode. - *name*—Name of the CA.
Step 3	`Router(ca-trustpoint)# rsakeypair key-label [key-size [encryption-key-size]]`	Specifies which key pair to associate with the certificate. - *key-label*—The name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. - *key-size*—(Optional) The size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the *encryption-key-size*.) - *encryption-key-size*—(Optional) The size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.)

Removing RSA Key Pair Settings

To delete a specified RSA key pair or all RSA key pairs that have been generated by your switch, enter the **crypto key zeroize rsa** command in global configuration mode as follows:

```
Router(config)# crypto key zeroize rsa [key-pair-label]
```

key-pair-label specifies the name of the key pair to be deleted. If the *key-pair-label* argument is used, you will delete only the specified RSA key pair. If no argument is used, you will delete all the RSA key pairs from your switch.

Verifying RSA Key Information

To verify RSA key information, use at least one of the privileged EXEC commands used in the examples.

To display your switch's RSA public keys, use the **show crypto key mypubkey rsa** command:

```
Router# show crypto key mypubkey rsa

% Key pair was generated at: 06:07:50 UTC Jan 13 1996

Key name: myswitch.example.com

 Usage: Encryption Key

 Key Data:

  00302017 4A7D385B 1234EF29 335FC973 2DD50A37 C4F4B0FD 9DADE748 429618D5

  18242BA3 2EDFBDD3 4296142A DDF7D3D8 08407685 2F2190A0 0B43F1BD 9A8A26DB

  07953829 791FCDE9 A98420F0 6A82045B 90288A26 DBC64468 7789F76E EE21
```

To display a list of all the RSA public keys stored on your switch (including the public keys of peers that have sent your switch their certificates during peer authentication for IPsec), or to display details of a particular RSA public key stored on your switch, use the **show crypto key pubkey-chain rsa** command:

```
Router# show crypto key pubkey-chain rsa

Codes: M - Manually Configured, C - Extracted from certificate

Code  Usage         IP-address     Name

M     Signature     10.0.0.1       myrouter.example.com

M     Encryption    10.0.0.1       myrouter.example.com

C     Signature     172.16.0.1     routerA.example.com

C     Encryption    172.16.0.1     routerA.example.com

C     General       192.168.10.3   routerB.domain1.com
```

For complete configuration information for Multiple RSA Key Pair Support, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t8/feature/guide/ftmltkey.html

For an RSA key pair configuration example, see the "Multiple RSA Key Pairs Configuration Example" section on page 26-54.

Configuring Protected Private Key Storage

The protected private key storage feature allows a user to encrypt and lock the RSA private keys that are used on a Catalyst 6500 Series switch, which prevents unauthorized use of the private keys.

Protected Private Key Storage Configuration Guidelines and Restrictions

When configuring protected private key storage, follow these guidelines and restrictions:

- An encrypted key is not effective after the switch boots up until you manually unlock the key (using the **crypto key unlock rsa** command). Depending on which key pairs are encrypted, this functionality may adversely affect applications such as IP Security (IPsec), Secure Shell (SSH) and Secure Socket Layer (SSL); that is, management of the switch over a secure channel may not be possible until the necessary key pair is unlocked.
- If a passphrase is lost, you must regenerate the key, enroll with the CA server again, and obtain a new certificate. A lost passphrase cannot be recovered.
- If you want to change a passphrase, you must decrypt the key with the current passphrase using the **crypto key decrypt rsa** command and encrypt the key once more to specify the new passphrase.

Configuring Private Keys

To encrypt, decrypt, lock, and unlock private keys, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto key encrypt [write] rsa [name key-name] passphrase passphrase`	Encrypts the RSA keys. After this command is entered, the switch can continue to use the key; the key remains unlocked. • **write**—(Optional) Switch configuration is immediately written to NVRAM. If the **write** keyword is not specified, the configuration must be manually written to NVRAM; otherwise, the encrypted key will be lost next time the switch is reloaded. • **name** *key-name*—(Optional) Name of the RSA key pair that is to be encrypted. If a key name is not specified, the default key name, switchname.domainname, is used. • **passphrase** *passphrase*—Passphrase that is used to encrypt the RSA key. To access the RSA key pair, the passphrase must be specified.
Step 2	`Router(config)# exit`	Exits global configuration mode.
Step 3	`Router# show crypto key mypubkey rsa`	(Optional) Shows that the private key is encrypted (protected) and unlocked.

	Command	Purpose
Step 4	`Router# crypto key lock rsa [name key-name] passphrase passphrase`	(Optional) Locks the encrypted private key on a running switch. • **name** *key-name*—(Optional) Name of the RSA key pair that is to be locked. If a key name is not specified, the default key name, switchname.domainname, is used. • **passphrase** *passphrase*—Passphrase that is used to lock the RSA key. To access the RSA key pair, the passphrase must be specified. **Note** After the key is locked, it cannot be used to authenticate the switch to a peer device. This behavior disables any IPsec or SSL connections that use the locked key. Any existing IPsec tunnels created on the basis of the locked key will be closed. If all RSA keys are locked, SSH will automatically be disabled.
Step 5	`Router# show crypto key mypubkey rsa`	(Optional) Shows that the private key is protected and locked. The output will also show failed connection attempts by applications such as IKE, SSH, and SSL.
Step 6	`Router# crypto key unlock rsa [name key-name] passphrase passphrase`	(Optional) Unlocks the private key. • **name** *key-name*—(Optional) Name of the RSA key pair that is to be unlocked. If a key name is not specified, the default key name, switchname.domainname, is used. • **passphrase** *passphrase*—Passphrase that is used to unlock the RSA key. To access the RSA key pair, the passphrase must be specified. **Note** After this command is entered, you can continue to establish IKE tunnels.

	Command	Purpose
Step 7	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 8	`Router(config)#` **`crypto key decrypt`** `[`**`write`**`]` **`rsa`** `[`**`name`** *`key-name`*`]` **`passphrase`** *`passphrase`*	(Optional) Deletes the encrypted key and leaves only the unencrypted key. • **write**—(Optional) Unencrypted key is immediately written to NVRAM. If the **write** keyword is not specified, the configuration must be manually written to NVRAM; otherwise, the key will remain encrypted the next time the switch is reloaded. • **name** *key-name*—(Optional) Name of the RSA key pair that is to be deleted. If a key name is not specified, the default key name, switchname.domainname, is used. • **passphrase** *passphrase*—Passphrase that is used to delete the RSA key. To access the RSA key pair, the passphrase must be specified.

Verifying the Protected and Locked Private Keys

To verify that the key is protected (encrypted) and locked, enter the **show crypto key mypubkey rsa** command:

```
Router# show crypto key mypubkey rsa

% Key pair was generated at:20:29:41 GMT Jun 20 2003
Key name:pki1-72a.cisco.com
Usage:General Purpose Key
*** The key is protected and LOCKED. ***
Key is exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D7808D C5FF14AC
0D2B55AC 5D199F2F 7CB4B355 C555E07B 6D0DECBE 4519B1F0 75B12D6F 902D6E9F
B6FDAD8D 654EF851 5701D5D7 EDA047ED 9A2A619D 5639DF18 EB020301 0001
```

For complete configuration information for protected private key storage, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gt_ppkey.html

For protected private key configuration examples, see the "Protected Private Key Storage Configuration Examples" section on page 26-54.

Configuring a Trustpoint CA

The **crypto pki trustpoint** command allows you to declare the certificate authority (CA) that your switch should use and to specify characteristics for the CA.

The **crypto pki trustpoint** command combines and replaces the functionality of the existing **crypto ca identity** command and the **crypto ca trusted-root** command. Although both of these existing commands allow you to declare the certification authority (CA) that your switch should use, only the **crypto ca identity** command supports enrollment (the requesting of a switch certificate from a CA).

Trustpoint CA Configuration Guidelines and Restrictions

When configuring a trustpoint CA, follow these guidelines and restrictions:

- After the trustpoint CA has been configured, you can obtain the certificate of the CA by using the **crypto pki authenticate** command or you can specify that certificates should not be stored locally but retrieved from a CA trustpoint by using the **crypto pki certificate query** command.
- Normally, certain certificates are stored locally in the switch's NVRAM, and each certificate uses a moderate amount of memory. To save NVRAM space, you can use the **crypto pki certificate query** command to put the switch into query mode, preventing certificates from being stored locally; instead, they are retrieved from a specified CA trustpoint when needed. This will save NVRAM space but could result in a slight performance impact.

To declare the CA that your switch should use and specify characteristics for the trustpoint CA, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoint** *name*	Declares the CA that your switch should use. Enabling this command puts you in ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	Router(ca-trustpoint)# **enrollment** [[**mode ra**] | [**retry period** *minutes*] | [**retry count** *number*] | [**url** *url*]]	Specifies enrollment parameters for your CA. • **mode ra**—(Optional) Specifies registration authority (RA) mode if your CA system provides a RA. RA mode is turned off until you enable the **mode ra** keyword. • *minutes*—(Optional) Specifies the wait period between certificate request retries. The default is 1 minute between retries. (Specify from 1 to 60 minutes.) • *number*—(Optional) Specifies the number of times a switch will resend a certificate request when it does not receive a response from the previous request. The default is 10 retries. (Specify from 1 to 100 retries.) • *url*—Specifies the URL of the CA where your switch should send certificate requests; for example, http://ca_server. *url* must be in the form http://CA_*name*, where CA_*name* is the CA's host Domain Name System (DNS) name or IP address.
	Router(ca-trustpoint)# **root tftp** *server-hostname filename*	Obtains the CA via TFTP. • *server-hostname*—Name for the server that will store the trustpoint CA • *filename*—Name for the file that will store the trustpoint CA.

	Command	Purpose
Step 3	`Router(ca-trustpoint)# enrollment http-proxy host-name port-num`	Obtains the CA via HTTP through the proxy server. • *host-name*—Name of the proxy server used to get the CA. • *port-num*—Port number used to access the CA. **Note** This command can be used in conjunction only with the **enrollment** command.
Step 4	`Router(ca-trustpoint)# primary name`	(Optional) Assigns a specified trustpoint as the primary trustpoint of the switch. • *name*—Name of the primary trustpoint of the switch.
Step 5	`Router(ca-trustpoint)# crl {query url | optional}`	(Optional) Queries the certificate revocation list (CRL) to ensure that the certificate of the peer has not been revoked. • *url* —Lightweight Directory Access Protocol (LDAP) URL published by the certificate authority (CA) server is specified to query the CRL; for example, ldap://another_server. • **optional**—CRL verification is optional. **Note** If the **query** *url* option is not enabled, the switch will check the certificate distribution point (CDP) that is embedded in the certificate.
Step 6	`Router(ca-trustpoint)# default command-name`	(Optional) Sets the value of ca-trustpoint configuration mode to its default. • *command-name*—pki-trustpoint configuration subcommand. Default is off.
Step 7	`Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and enters global configuration mode.
Step 8	`Router(config)# crypto pki authenticate name`	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1.
Step 9	`Router(config)# crypto pki trustpoint name`	Reenters ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 10	`Router(ca-trustpoint)# crypto pki certificate query`	(Optional) Turns on query mode per specified trustpoint, causing certificates not to be stored locally.

Verifying a Trustpoint CA

To verify information about your certificate, the certificate of the CA, and registration authority (RA) certificates, enter the **show crypto pki certificates** command:

```
Router# show crypto pki certificates
```

```
CA Certificate

  Status: Available

  Certificate Serial Number: 3051DF7123BEE31B8341DFE4B3A338E5F

  Key Usage: Not Set

RA Signature Certificate

  Status: Available

  Certificate Serial Number: 34BCF8A0

  Key Usage: Signature


RA KeyEncipher Certificate

  Status: Available

  Certificate Serial Number: 34BCF89F

  Key Usage: Encryption
```

To display the trustpoints that are configured in the switch, enter the **show crypto pki trustpoints** command:

```
Router# show crypto pki trustpoints

Trustpoint bo:
Subject Name:
CN = bomborra Certificate Manager
O = cisco.com
C = US
Serial Number:01
Certificate configured.
CEP URL:http://bomborra
CRL query url:ldap://bomborra
```

For complete configuration information for the trustpoint CA, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t8/feature/guide/fttrust.html

For a trustpoint CA configuration example, see the "Trustpoint CA Configuration Example" section on page 26-54.

Configuring Query Mode Definition Per Trustpoint

Certificates contain public key information and are signed by certificate authority (CA) as proof of identity. Normally, all certificates are stored locally in the switch's NVRAM, and each certificate uses a moderate amount of memory. The query mode definition per trustpoint feature allows you to define a query for a specific trustpoint so that the certificates associated with that specific trustpoint can be stored on a remote server.

This feature is especially useful for environments where multiple trustpoints are configured on a switch because it allows you more control over use of the trustpoint. Query mode can be activated on specific trustpoints rather than on all of the trustpoints on a switch.

Query Mode Definition Per Trustpoint Configuration Guidelines and Restrictions

When configuring query mode definition per trustpoint, follow these guidelines and restrictions:

- Normally, certain certificates are stored locally in the switch's NVRAM, and each certificate uses a moderate amount of memory. To save NVRAM space, you can use the **query certificate** command to prevent certificates from being stored locally; instead, they are retrieved from a remote server, such as a CA or LDAP server, during startup. This will save NVRAM space but could result in a slight performance impact.
- Certificates associated with a specified trustpoint will not be written into NVRAM and the certificate query will be attempted during the next reload of the switch.
- When the global **crypto pki certificate query** command is used, the query certificate will be added to all trustpoints on the switch. When the **no crypto pki certificate query** command is used, any previous query certificate configuration will be removed from all trustpoints and any query in progress will be halted and the feature disabled.

To configure a trustpoint CA and initiate query mode for the trustpoint, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoin**t *name*	Declares the CA that your switch should use. Enabling this command puts you in ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	Router(ca-trustpoint)# **enrollment** [[**mode ra**] | [**retry period** *minutes*] | [**retry count** *number*] | [**url** *url*]]	Specifies enrollment parameters for your CA. • **mode ra**—(Optional) Specifies registration authority (RA) mode if your CA system provides a RA. RA mode is turned off until you enable the **mode ra** keyword. • *minutes*—(Optional) Specifies the wait period between certificate request retries. The default is 1 minute between retries. (Specify from 1 to 60 minutes.) • *number*—(Optional) Specifies the number of times a switch will resend a certificate request when it does not receive a response from the previous request. The default is 10 retries. (Specify from 1 to 100 retries.) • *url*—Specifies the URL of the CA where your switch should send certificate requests; for example, http://ca_server. *url* must be in the form http://CA_*name*, where CA_*name* is the CA's host Domain Name System (DNS) name or IP address.

	Command	Purpose
Step 3	Router(ca-trustpoint)# **enrollment http-proxy** *host-name port-num*	(Optional) Obtains the CA via HTTP through the proxy server. • *host-name*—Name of the proxy server used to get the CA. • *port-num*—Port number used to access the CA. **Note** This command can be used in conjunction only with the **enrollment** command.
Step 4	Router(ca-trustpoint)# **crl query** *url*	(Optional) Specifies the URL for the CA server if the CA server supports query mode through LDAP. • *url* —Lightweight Directory Access Protocol (LDAP) URL published by the certificate authority (CA) server.
Step 5	Router(ca-trustpoint)# **query certificate**	Turns on query mode per specified trustpoint, causing certificates not to be stored locally and to be retrieved from a remote server.
Step 6	Router(ca-trustpoint)# **exit**	Exits ca-trustpoint configuration mode and enters global configuration mode.
Step 7	Router(config)# **crypto pki authenticate** *name*	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1.
Step 8	Router(config)# **crypto key generate rsa**	(Optional) Generates RSA key pairs.
Step 9	Router(config)# **crypto pki enroll** t*rustpoint-nam*e	(Optional) Obtains switch certificate. • *trustpoint-name*—Name of the CA. Enter the *name* value entered in Step 1.

Verifying Query Mode Definition Per Trustpoint CA

For query mode to operate correctly during the next reload, the certificates must be associated with the trustpoint. Use the **show crypto pki certificates** command to verify that each of the trustpoints has the needed certificates before storing the configuration and reloading the switch:

```
Router# show crypto pki certificates

Trustpoint yni:

  Issuing CA certificate pending:

    Subject Name:

     cn=nsca-r1 Cert Manager,ou=pki,o=cisco.com,c=US

    Fingerprint: C21514AC 12815946 09F635ED FBB6CF31

  Router certificate pending:

    Subject Name:

     hostname=trance.cisco.com,o=cisco.com
```

```
Next query attempt:

  52 seconds
```

For complete configuration information for Query Mode Definition Per Trustpoint, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gt_qerym.html

For a query mode definition per trustpoint configuration example, see the "Query Mode Definition Per Trustpoint Configuration Example" section on page 26-55.

Configuring a Local Certificate Storage Location

The Local Certificate Storage Location feature enables you to store public key infrastructure (PKI) credentials, such as Rivest, Shamir, and Adelman (RSA) keys and certificates in a specific location. An example of a certificate storage location includes NVRAM, which is the default location, and other local storage locations as supported by your platform, such as flash.

Note The Local Certificate Storage Location feature is supported in Cisco IOS Release 12.2(33)SXH and later releases.

Local Certificate Storage Location Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring a local certificate storage location:

- Before you can specify the local certificate storage location, your system should meet the following requirements:
  - A Cisco IOS Release 12.4(2)T PKI-enabled image or a later image
  - A platform that supports storing PKI credentials as separate files
  - A configuration that contains at least one certificate
  - An accessible local file system
- When storing certificates to a local storage location, the following restrictions are applicable:
  - Only local file systems may be used. An error message will be displayed if a remote file system is selected, and the command will not take effect.
  - A subdirectory may be specified if supported by the local file system. NVRAM does not support subdirectories.
  - Certificates are stored to NVRAM by default, however, some routers do not have the required amount of NVRAM to successfully store certificates. Introduced in Cisco IOS Release 12.4(2)T is the ability to specify where certificates are stored on a local file system.
  - During run time, you can specify what active local storage device you would like to use to store certificates.

Specifying a Local Storage Location for Certificates

To specify the local storage location for certificates, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki certificate storage** *location-name*	Specifies the local storage location for certificates. • *location-name*—Name of the storage location.
Step 2	Router (config)# **exit**	Exits global configuration mode.
Step 3	Router# **copy** *source-url destination-url*	(Optional) Saves the running configuration to the startup configuration. • *source-url*—The location URL (or alias) of the source file or directory to be copied. The source can be either local or remote, depending upon whether the file is being downloaded or uploaded. • *destination-url*—The destination URL (or alias) of the copied file or directory. The destination can be either local or remote, depending upon whether the file is being downloaded or uploaded. **Note** Settings will only take effect when the running configuration is saved to the startup configuration.

Verifying the Local Certificate Storage Location Configuration

To verify a local certificate storage location configuration, enter the **show crypto pki certificates storage** command.

The **show crypto pki certificates storage** command displays the current setting for the PKI certificate storage location.

The following example shows that certificates are stored in the certs subdirectory of disk0:

```
Router# show crypto pki certificates storage

Certificates will be stored in disk0:/certs/
```

For complete configuration information for local certificate storage location, refer to the *Cisco IOS Security Configuration Guide* or the following URL:

http://www.cisco.com/en/US/docs/ios/12_4t/12_4t2/st_pkist.html

For local certificate storage configuration examples, see the "Local Certificate Storage Location Configuration Example" section on page 26-55.

Configuring Direct HTTP Enroll with CA Servers (Reenroll Using Existing Certificates)

The direct HTTP enroll with CA servers feature allows users to bypass the registration authority (RA) when enrolling with a certification authority (CA) by configuring an enrollment profile. HTTP enrollment requests can be sent directly to the CA server.

The reenroll using existing certificates functionality allows a switch that is enrolled with a third-party vendor CA to use its existing certificate to enroll with the Cisco IOS certificate server so the enrollment request is automatically granted.

Direct HTTP Enroll with CA Servers Configuration Guidelines and Restrictions

When configuring direct HTTP enroll with CA servers, follow these guidelines and restrictions:

- The CA certificate and switch certificates must be returned in the privacy enhanced mail (PEM) format.
- If an enrollment profile is specified, an enrollment URL can not be specified in the trustpoint configuration.
- Because there is no standard for the HTTP commands used by various CAs, the user is required to enter the command that is appropriate to the CA that is being used.
- The newly created trustpoint can only be used one time (which occurs when the switch is enrolled with the Cisco IOS CA). After the initial enrollment is successfully completed, the credential information will be deleted from the enrollment profile.
- The Cisco IOS certificate server will automatically grant only the requests from clients who were already enrolled with the non-Cisco IOS CA. All other requests must be manually granted unless the server is set to be in auto grant mode (using the **grant automatic** command).
- To configure direct HTTP enroll with CA servers, you must perform the following steps:
  - Either configure a certificate enrollment profile for the client switch (see the "Configuring an Enrollment Profile for a Client Switch" section on page 26-17) or configure an enrollment profile for a client switch that is already enrolled with a third-party vendor (see the "Configuring an Enrollment Profile for a Client Switch Enrolled with a Third-Party Vendor CA" section on page 26-19).
  - Configure the CA certificate server to accept enrollment requests only from clients who are already enrolled with the third-party vendor CA trustpoint (see the "Configuring the CA to Accept Enrollment Requests from Clients of a Third-Party Vendor CA" section on page 26-20).

Configuring an Enrollment Profile for a Client Switch

To configure a certificate enrollment profile, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoint** *name*	Declares the trustpoint a given name and enters ca-trustpoint configuration mode. • *name*—Name of the CA trustpoint.
Step 2	Router(ca-trustpoint)# **enrollment profile** *label*	Specifies that an enrollment profile can be used for certificate authentication and enrollment. • *label*—Name for the enrollment profile.
Step 3	Router(ca-trustpoint)# **exit**	Exits ca-trustpoint configuration mode and enters global configuration mode.
Step 4	Router(config)# **crypto pki profile enrollment** *label*	Defines an enrollment profile and enters ca-profile-enroll configuration mode. • *label*—Name for the enrollment profile; the enrollment profile name must match the name specified in the **enrollment profile** command.
Step 5	Router(ca-profile-enroll)# **authentication url** *url*	(Optional) Specifies the URL of the CA server to which to send certificate authentication requests. • *url*—URL of the CA server to which your switch should send authentication requests. If using HTTP, the URL should read "http://CA_name," where CA_name is the host Domain Name System (DNS) name or IP address of the CA. If using TFTP, the URL should read "tftp://certserver/file_specification." (If the URL does not include a file specification, the fully qualified domain name (FQDN) of the switch will be used.
	Router(ca-profile-enroll)# **authentication terminal**	(Optional) Specifies manual cut-and-paste certificate authentication.
Step 6	Router(ca-profile-enroll)# **authentication command** *http-command*	(Optional) Sends the HTTP request to the CA for authentication. • *http-command*—HTTP request to be sent to the CA server. This command should be used after the **authentication url** command has been entered.
Step 7	Router(ca-profile-enroll)# **enrollment url** *url* or	Specifies the URL of the CA server to which to send certificate enrollment requests via HTTP or TFTP. • *url*—URL of the CA server.
	Router(ca-profile-enroll)# **enrollment terminal**	Specifies manual cut-and-paste certificate enrollment.

	Command	Purpose
Step 8	`Router(ca-profile-enroll)# enrollment command http-command`	(Optional) Specifies the HTTP command to be sent to the CA for enrollment. • *http-command*—HTTP command to be sent to the CA server.
Step 9	`Router(ca-profile-enroll)# parameter number {value value | prompt string}`	(Optional) Specifies parameters for an enrollment profile. • *number*—User parameters. Valid values range from 1 to 8. • *value*—To be used if the parameter has a constant value. • *string*—To be used if the parameter is supplied after the **crypto pki authenticate** command or the **crypto pki enroll** command has been entered. **Note** The value of the *string* argument does not have an effect on the value that is used by the switch. This command can be used multiple times to specify multiple values.
Step 10	`Router(ca-profile-enroll config)# exit`	Exits ca-profile-enroll configuration mode and enters global configuration mode.
Step 11	`Router(config)# exit`	Exits global configuration mode and enters Privileged EXEC mode.
Step 12	`Router# show crypto pki certificates`	(Optional) Verifies information about your certificate, the certificate of the CA, and RA certificates.
Step 13	`Router# show crypto pki trustpoints`	(Optional) Displays the trustpoints that are configured in the switch.

In configuring the direct HTTP enrollment profile, you can use the **parameter** command within an enrollment profile to provide predefined or console-input parameters to the **authentication command** command or the **enrollment command** command.

When you enter the **parameter** *number* command, a macroinstruction (macro) is created and named $P*number* (for example, $P1). If the **value** keyword is specified, the *value* argument is assigned to the macro. If the prompt keyword is specified, when the switch executes the **authentication command** command or the **enrollment command** command, the console will display the *string* argument as a prompt for user input. You can then enter a value to be assigned to the macro.

In addition to user-defined macros, three predefined macros are available:

- $REQ—The Certification Request Standard (PKCS #10) message to request certification of a key
- $FQDN—The FQDN of the switch
- $HOST—The hostname of the switch

This example shows how to use one predefined and three user-defined macros:

```
Router(config)# crypto ca profile enrollment E
Router(ca-profile-enroll)# authentication url http://ca-server.example.com
Router(ca-profile-enroll)# authentication command GET $P1
Router(ca-profile-enroll)# enrollment url
Router(ca-profile-enroll)# enrollment command ^C
POST action=getServerCert
&pkcs10Request=$REQ
&reference_number=$P2
&authcode=P3
&retrievedAs=rawDER
^C
Router(ca-profile-enroll)# parameter 1 value cacert.crt
Router(ca-profile-enroll)# parameter 2 prompt Enter the Reference Number:
Router(ca-profile-enroll)# parameter 3 prompt Enter the Auth Code:
```

Before the HTTP authentication and enrollment commands are posted by the switch to the CA, the console will prompt for any required user input, and macro values will be substituted for the macro names in the posted commands.

For an example of how to configure an enrollment profile for direct HTTP enrollment with a CA server, see the "Enrollment Profile for a Client Switch Configuration Example" section on page 26-55.

Configuring an Enrollment Profile for a Client Switch Enrolled with a Third-Party Vendor CA

When a client switch is already enrolled with a third-party vendor CA, but you want to reenroll that switch with a Cisco IOS certificate server, perform the following procedures. Note that some prerequisite steps are required before beginning the configuration.

Prerequisites

Before configuring a certificate enrollment profile for the client switch enrolled with a third-party vendor, you should have already performed the following tasks at the client switch:

- Defined a trustpoint that points to a third-party vendor CA.
- Authenticated and enrolled the client switch with the third-party vendor CA.

To configure a certificate enrollment profile for a client switch that is already enrolled with a third-party vendor CA so that the switch can reenroll with a Cisco IOS certificate server, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# `**`crypto pki trustpoint`** *`name`*	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the Cisco IOS CA that is to be used.
Step 2	`Router(ca-trustpoint)# `**`enrollment profile`** *`label`*	Specifies that an enrollment profile is to be used for certificate reenrollment. • *label*—Name for the enrollment profile.
Step 3	`Router(ca-trustpoint)# `**`exit`**	Exits ca-trustpoint configuration mode and enters global configuration mode.

	Command	Purpose
Step 4	`Router(config)# crypto pki profile enrollment label`	Defines an enrollment profile and enters ca-profile-enroll configuration mode. • *label*—Name for the enrollment profile; the enrollment profile name must match the name specified in the **enrollment profile** command in Step 2.
Step 5	`Router(ca-profile-enroll)# enrollment url url`	Specifies the URL of the CA server to which to send certificate enrollment requests via HTTP. • *url*—The enrollment URL should point to the Cisco IOS CA.
Step 6	`Router(ca-profile-enroll)# enrollment credential label`	Specifies the non-Cisco IOS CA trustpoint that is to be enrolled with the Cisco IOS CA. • *label*—Name of the CA trustpoint of another vendor.
Step 7	`Router(ca-profile-enroll)# exit`	Exits ca-profile-enroll configuration mode and enters global configuration mode.
Step 8	`Router(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 9	`Router# show crypto pki certificates`	(Optional) Verifies information about your certificate, the certificate of the CA, and RA certificates
Step 10	`Router# show crypto pki trustpoints`	(Optional) Displays the trustpoints that are configured in the switch.

For an example of how to configure a certificate enrollment profile for a client switch that is already enrolled with a third-party vendor CA, see the "Enrollment Profile for a Client Switch Already Enrolled with a Third-Party Vendor CA Example" section on page 26-56.

Configuring the CA to Accept Enrollment Requests from Clients of a Third-Party Vendor CA

To configure the CA certificate server to accept enrollment requests only from clients who are already enrolled with the third-party vendor CA trustpoint, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip http server`	Enables the HTTP server on your system.
Step 2	`Router(config)# crypto pki server cs-label`	Enables the certificate server and enters certificate server configuration mode. • *cs-label*—The *cs-label* argument must match the name that was specified by the **crypto pki trustpoint** command for the client switch.

	Command	Purpose
Step 3	`Router(cs-server)# database url root-url`	Specifies the location where all database entries for the certificate server will be stored. • *root-url*—Root URL. **Note** If this command is not specified, all database entries will be written to NVRAM.
Step 4	`Router(cs-server)# database level {minimal | names | complete}`	Controls what type of data is stored in the certificate enrollment database. • **minimal**—Enough information is stored only to continue issuing new certificates without conflict; the default value. • **names**—In addition to the information given in the minimal level, the serial number and subject name of each certificate. • **complete**—In addition to the information given in the minimal and names levels, each issued certificate is written to the database. **Note** The **complete** keyword produces a large amount of information; if it is specified, you should also specify an external TFTP server in which to store the data using the **database url** command.
Step 5	`Router(cs-server)# issuer-name DN-string`	Sets the CA issuer name to the specified DN-string. • *DN-string*—The default value is as follows: **issuer-name CN=*cs***-*label*.
Step 6	`Router(cs-server)# grant auto trustpoint label`	Enables the certificate server to automatically grant only the requests from clients that are already enrolled with the specified non-Cisco IOS CA trustpoint. • *label*—Name of the CA trustpoint of another vendor. **Note** The *label* argument should match the trustpoint that was specified for the client switch's enrollment profile (using the **enrollment credential** command).
Step 7	`Router(cs-server)# lifetime {ca-certificate | certificate} time`	(Optional) Specifies the lifetime, in days, of a CA certificate or a certificate. • *time*—Valid values range from 1 day to 1825 days. The default CA certificate lifetime is 3 years; the default certificate lifetime is 1 year. The maximum certificate lifetime is 1 month less than the lifetime of the CA certificate.

	Command	Purpose
Step 8	Router(cs-server)# **lifetime crl** *time*	(Optional) Defines the lifetime, in hours, of the Certificate Revocation List (CRL) that is used by the certificate server. • *time*—Maximum lifetime value is 336 hours (2 weeks). The default value is 168 hours (1 week).
Step 9	Router(cs-server)# **cdp-url** *url*	(Optional) Defines a Certificate Distribution Point (CDP) to be used in the certificates that are issued by the certificate server. • *url*—URL must be an HTTP URL.
Step 10	Router(cs-server)# **shutdown**	Disables a certificate server without removing the configuration. You should enter this command only after you have completely configured your certificate server.
Step 11	Router(cs-server)# **exit**	Exits certificate server configuration mode.
Step 12	Router(config)# **exit**	Exits global configuration mode.
Step 13	Router# **show crypto pki server**	(Optional) Displays the current state and configuration of the certificate server.

For complete configuration information for direct HTTP enroll with CA servers, including the “reenroll using existing certificates” functionality, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t4/feature/guide/gthttpca.html

For direct HTTP enroll with CA servers configuration examples, see the “Direct HTTP Enrollment with CA Servers Configuration Examples” section on page 26-55.

Configuring Manual Certificate Enrollment (TFTP and Cut-and-Paste)

The manual certificate enrollment (TFTP and cut-and-paste) feature allows users to generate a certificate request and accept certification authority (CA) certificates as well as the switch’s certificates; these tasks are accomplished by a TFTP server or manual cut-and-paste operations. You might want to utilize TFTP or manual cut-and-paste enrollment in the following situations:

- The CA does not support Simple Certificate Enrollment Protocol (SCEP) (which is the most commonly used method for sending and receiving requests and certificates).
- A network connection between the switch and CA is not possible (which is how a switch running Cisco IOS software obtains its certificate).

Manual Certificate Enrollment (TFTP and Cut-and-Paste) Configuration Guidelines and Restrictions

When configuring nanualcertificate enrollment (TFTP and cut-and-paste), follow these guidelines and restrictions:

- You can switch between TFTP and cut-and-paste; for example, you can paste the CA certificate using the **enrollment terminal** command, then enter **no enrollment terminal** and **enrollment url tftp://certserver/file_specification** to switch to TFTP to send or receive requests and switch certificates. However, Cisco does not recommend switching URLs if SCEP is used; that is, if the enrollment URL is http://, do not change the enrollment URL between fetching the CA certificate and enrolling the certificate.

Configuring Manual Enrollment Using TFTP

Before configuring manual enrollment using TFTP, you must meet the following prerequisites:

- You must know the correct URL to use if you are configuring certificate enrollment using TFTP.
- The switch must be able to write a file to the TFTP server for the **crypto pki enroll** command.
- Some TFTP servers require that the file exist on the server before it may be written.
- Most TFTP servers require that the file be writeable by anyone. This requirement may pose a risk because any switch or other device may write or overwrite the certificate request; thus, the switch will not be able to use the certificate once it is granted by the CA because the request was modified.

To declare the trustpoint CA that your switch should use and configure that trustpoint CA for manual enrollment using TFTP, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	`Router(ca-trustpoint)# enrollment [mode] [retry period minutes] [retry count number] url url`	Specifies the enrollment parameters of your CA. • **mode**—Specifies registration authority (RA) mode if your CA system provides a RA. • *minutes*—Specifies the wait period between certificate request retries. The default is 1 minute between retries. • *number*—Specifies the number of times a switch will resend a certificate request when it does not receive a response from the previous request. (Specify from 1 to 100 retries.) • *url*—Specifies the URL of the CA where your switch should send certificate requests. If you are using SCEP for enrollment, the URL must be in the form http://CA_name, where CA_name is the CA's host Domain Name System (DNS) name or IP address. If you are using TFTP for enrollment, the URL must be in the form tftp://certserver/file_specification.

	Command	Purpose
Step 3	`Router(ca-trustpoint)# `**`crypto pki authenticate`** *`name`*	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1.
Step 4	`Router(ca-trustpoint)# `**`exit`**	Exits ca-trustpoint configuration mode and returns to global configuration.
Step 5	`Router(config)# `**`crypto pki enroll`** *`name`*	Obtains your switch's certificates from the CA. • *name*—Name of the CA. Enter the *name* value entered in Step 1.
Step 6	`Router(config)# `**`crypto pki import`** *`name`* **`certificate`**	Imports a certificate using TFTP. • *name*—Name of the CA. Enter the *name* value entered in Step 1.

Configuring Certificate Enrollment Using Cut-and-Paste

To declare the trustpoint CA that your switch should use and configure that trustpoint CA for manual enrollment using cut-and-paste, perform this task:

	Command	Purpose
Step 1	`Router(config)# `**`crypto pki trustpoint`** *`name`*	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	`Router(ca-trustpoint)# `**`enrollment terminal`**	Specifies manual cut-and-paste certificate enrollment.
Step 3	`Router(ca-trustpoint)# `**`crypto pki authenticate`** *`name`*	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Specifies the name of the CA. Enter the *name* value entered in Step 1.
Step 4	`Router(ca-trustpoint)# `**`exit`**	Exits ca-trustpoint configuration mode and returns to global configuration.

	Command	Purpose
Step 5	`Router(config)# crypto pki enroll name`	Obtains your switch's certificates from the CA. • *name*—Specifies the name of the CA. Enter the *name* value entered in Step 1.
Step 6	`Router(config)# crypto pki import name certificate`	Imports a certificate manually at the terminal. • *name*—Specifies the name of the CA. Enter the *name* value entered in Step 1. **Note** You must enter the **crypto pki import** command twice if usage keys (signature and encryption keys) are used. The first time the command is entered, one of the certificates is pasted into the switch; the second time the command is entered, the other certificate is pasted into the switch. (It does not matter which certificate is pasted first.)

Verifying the Manual Certificate Enrollment Configuration

To verify information about your certificate, the certificate of the CA, and RA certificates, enter the **show crypto pki certificates** command:

```
Router# show crypto pki certificates

Certificate
  Status:Available
  Certificate Serial Number:14DECE05000000000C48
  Certificate Usage:Encryption

  Issuer:
    CN = msca-root
    O = Cisco Systems
    C = U

  Subject:
    Name:Switch.cisco.com
    OID.1.2.840.113549.1.9.2 = Switch.cisco.com

      CRL Distribution Point:
    http://msca-root/CertEnroll/msca-root.crl

      Validity Date:
    start date:18:16:45 PDT Jun 7 2008
    end   date:18:26:45 PDT Jun 7 2009
    renew date:16:00:00 PST Dec 31 1969

      Associated Trustpoints:MS

   Certificate
    Status:Available
    Certificate Serial Number:14DEC2E9000000000C47
    Certificate Usage:Signature

      Issuer:
    CN = msca-root
    O = Cisco Systems
    C = US
   Subject:
  Name:Switch.cisco.com
  OID.1.2.840.113549.1.9.2 = Switch.cisco.com

   CRL Distribution Point:
     http://msca-root/CertEnroll/msca-root.crl

   Validity Date:
  start date:18:16:42 PDT Jun 7 2008
  end   date:18:26:42 PDT Jun 7 2009
  renew date:16:00:00 PST Dec 31 1969

   Associated Trustpoints:MS

 CA Certificate
  Status:Available
  Certificate Serial Number:3AC0A65E9547C2874AAF2468A942D5EE
  Certificate Usage:Signature

   Issuer:
  CN = msca-root
  O = Cisco Systems
  C = US

   Subject:
  CN = msca-root
  O = Cisco Systems
  C = US

   CRL Distribution Point:
     http://msca-root/CertEnroll/msca-root.crl

Validity Date:
  start date:16:46:01 PST Feb 13 2008
  end   date:16:54:48 PST Feb 13 2013

Associated Trustpoints:MS
```

To display the trustpoints that are configured in the switch, enter the **show crypto pki trustpoints** command:

```
Router# show crypto pki trustpoints

Trustpoint bo:

    Subject Name:

    CN = ACSWireless Certificate Manager

     O = cisco.com

     C = US

          Serial Number:01

    Certificate configured.

    CEP URL:http://ACSWireless

    CRL query url:ldap://ACSWireless
```

For complete configuration information for manual certificate enrollment (TFTP and cut-and-paste), refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ftmancrt.html

For manual certificate enrollment configuration examples, see the "Manual Certificate Enrollment Configuration Examples" section on page 26-57.

Configuring Certificate Autoenrollment

The certificate autoenrollment feature allows you to configure your switch to automatically request a certificate from the certification authority (CA) that is using the parameters in the configuration. Thus, operator intervention is no longer required at the time the enrollment request is sent to the CA server.

Automatic enrollment will be performed on startup for any trustpoint CA that is configured and does not have a valid certificate. When the certificate expires that is issued by a trustpoint CA that has been configured for autoenrollment, a new certificate is requested. Although this feature does not provide seamless certificate renewal, it does provide unattended recovery from expiration.

Before the certificate autoenrollment feature, certificate enrollment required complicated, interactive commands that had to be executed on every switch. This feature allows you to preload all of the necessary information into the configuration and cause each switch to obtain certificates automatically when it is booted. Autoenrollment also checks for expired switch certificates.

Note Before submitting an automatic enrollment request, all necessary enrollment information must be configured.

To configure autoenrollment with a CA on startup, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the name of the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	`Router(ca-trustpoint)# enrollment url url`	Specifies the URL of the CA on which your switch should send certificate requests; for example, http://ca_server. • *url*—Must be in the form of http://CA_name, where CA_name is the name of the CA's host Domain Name System or the IP address.
Step 3	`Router(ca-trustpoint)# subject-name [x.500-name]`	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name*—(Optional) If the *x-500-name* argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.
Step 4	`Router(ca-trustpoint)# ip-address {interface | none}`	Includes the IP address of the specified interface in the certificate request. • *interface*—IP address of the interface. • **none**—Specify this keyword if no IP address should be included. If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.
Step 5	`Router(ca-trustpoint)# serial-number [none]`	Specifies the switch serial number in the certificate request, unless the **none** keyword is specified. • **none**—(Optional) Specify this keyword if no serial number should be included.
Step 6	`Router(ca-trustpoint)# auto-enroll [regenerate]`	Enables autoenrollment. This command allows you to automatically request a switch certificate from the CA. By default, only the DNS name of the switch is included in the certificate. • **regenerate**—(Optional) Specify this keyword to generate a new key for the certificate even if a named key already exists.

	Command	Purpose
Step 7	`Router(ca-trustpoint)# password string`	(Optional) Specifies the revocation password for the certificate. • *string*—Text of the password. **Note** If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint.
Step 8	`Router(ca-trustpoint)# rsakeypair key-label [key-size [encryption-key-size]]`	Specifies which key pair to associate with the certificate. • *key-label*—Name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) Size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the *encryption-key-size*.) • *encryption-key-size*—(Optional) Size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.) If this command is not enabled, the FQDN key pair is used.

Preloading Root CAs

After enabling automatic enrollment, you must authenticate the CA to establish a chain of trust. This can be done by implementing one of the following methods:

- Obtaining the CA Certificate, page 26-29
- Adding the Certificate of the CA, page 26-30

Obtaining the CA Certificate

To obtain the certificate of the CA, enter the **crypto pki authenticate** command in global configuration mode.

```
Router(config)# crypto pki authenticate name
```

name specifies the name of the CA.

Adding the Certificate of the CA

To add the certificate of the CA, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki certificate chain name`	Enters certificate chain configuration mode, which allows you to add or delete specified certificates. • *name*—Name of the CA.
Step 2	`Router(config-cert-chain)# certificate certificate-serial-number`	Manually adds or deletes certificates. • *certificate-serial-number*—Serial number of the CA to add.

Verifying CA Information

To display information about your certificates, the certificates of the CA, and registration authority (RA) certificates, enter the **show crypto pki certificates** command:

```
Router# show crypto pki certificates

Certificate

  Subject Name

    Name: myrouter.example.com
    IP Address: 10.0.0.1

  Status: Available

  Certificate Serial Number: 428125BDA34196003F6C78316CD8FA95

  Key Usage: Signature


Certificate

  Subject Name

    Name: myswitch.example.com
    IP Address: 10.0.0.1

  Status: Available

  Certificate Serial Number: AB352356AFCD0395E333CCFD7CD33897

  Key Usage: Encryption


CA Certificate

  Status: Available

  Certificate Serial Number: 3051DF7123BEE31B8341DFE4B3A338E5F

  Key Usage: Not Set
```

To display the trustpoints configured in the switch, enter the **show crypto pki trustpoints** command:

```
Router# show crypto pki trustpoints

Trustpoint bo:

    Subject Name:

    CN = bomborra Certificate Manager

     O = cisco.com

     C = US

    Serial Number:01

    Certificate configured.

    CEP URL:http://bomborra

    CRL query url:ldap://bomborra
```

For complete configuration information for Certificate Autoenrollment, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t8/feature/guide/ftautoen.html

For a certificate autoenrollment configuration example, see the "Certificate Autoenrollment Configuration Example" section on page 26-60.

Configuring Key Rollover for Certificate Renewal

Automatic certificate enrollment was introduced to allow the switch to automatically request a certificate from the certification authority (CA) server. By default, the automatic enrollment feature requests a new certificate when the old certificate expires. Connectivity can be lost while the request is being serviced because the existing certificate and key pairs are deleted immediately after the new key is generated. The new key does not have a certificate to match it until the process is complete, and incoming Internet Key Exchange (IKE) connections cannot be established until the new certificate is issued. The key rollover for certificate renewal feature allows the certificate renewal request to be made before the certificate expires and retains the old key and certificate until the new certificate is available.

Key rollover can also be used with a manual certificate enrollment request. Using the same method as key rollover with certificate autoenrollment, a new key pair is created with a temporary name, and the old certificate and key pair are retained until a new certificate is received from the CA. When the new certificate is received, the old certificate and key pair are discarded and the new key pair is renamed with the name of the original key pair. Do not regenerate the keys manually; key rollover will occur when you enter the **crypto pki enroll** command.

Key Rollover for Certificate Renewal Configuration Guidelines and Restrictions

When configuring key rollover for certificate renewal, follow these guidelines and restrictions:

- Trustpoints configured to generate a new key pair using the **regenerate** command or the **regenerate** keyword of the **auto-enroll** command must not share key pairs with other trustpoints. To give each trustpoint its own key pair, use the **rsakeypair** command in ca-trustpoint configuration mode. Sharing key pairs among regenerating trustpoints is not supported and will cause loss of service on some of the trustpoints because of key and certificate mismatch.

Configuring Automatic Certificate Enrollment with Key Rollover

To configure key rollover with automatic certificate enrollment, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the name of the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	`Router(ca-trustpoint)# enrollment url url`	Specifies the URL of the CA on which your switch should send certificate requests; for example, http://ca_server. • *url*—Must be in the form of http://CA_name, where CA_name is the name of the CA's host Domain Name System or the IP address.
Step 3	`Router(ca-trustpoint)# subject-name [x.500-name]`	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name*—(Optional) If the *x-500-name* argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.
Step 4	`Router(ca-trustpoint)# ip-address {interface | none}`	Includes the IP address of the specified interface in the certificate request. • *interface*—IP address of the interface. • **none**—Specify this keyword if no IP address should be included. If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.
Step 5	`Router(ca-trustpoint)# serial-number [none]`	Specifies the switch serial number in the certificate request, unless the **none** keyword is specified. • **none**—(Optional) Specify this keyword if no serial number should be included.

	Command	Purpose
Step 6	Router(ca-trustpoint)# **auto-enroll** [*percent*][**regenerate**]	Enables autoenrollment. This command allows you to automatically request a switch certificate from the CA. By default, only the DNS name of the switch is included in the certificate. • *percent*—Use the *percent* argument to specify that a new certificate will be requested after the percent lifetime of the current certificate is reached. • **regenerate**—Specify this keyword to generate a new key for the certificate even if a named key already exists. **Note** If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: ! RSA key pair associated with trustpoint is exportable.
Step 7	Router(ca-trustpoint)# **password** *string*	(Optional) Specifies the revocation password for the certificate. • *string*—Text of the password. **Note** If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint.
Step 8	Router(ca-trustpoint)# **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]	Specifies which key pair to associate with the certificate. • *key-label*—Name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) Size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the e*ncryption-key-size*.) • *encryption-key-size*—(Optional) Size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.) **Note** If this command is not enabled, the FQDN key pair is used.
Step 9	Router(ca-trustpoint)# **exit**	Exits ca-trustpoint configuration mode and returns to global configuration mode.

	Command	Purpose
Step 10	`Router(config)# crypto pki authenticate name`	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1. Check the certificate fingerprint if prompted. **Note** This command is optional if the CA certificate is already loaded into the configuration.
Step 11	`Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 12	`Router# copy system:running-config nvram:startup-config`	(Optional) Copies the running configuration to the NVRAM startup configuration.

Configuring Manual Certificate Enrollment with Key Rollover

Note Do not regenerate the keys manually using the **crypto key generate** command; key rollover will occur when the **crypto pki enroll** command is entered.

To configure key rollover with manual certificate enrollment, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the name of the CA that your switch should use and enters ca-trustpoint configuration mode. • name—Name of the CA.
Step 2	`Router(ca-trustpoint)# enrollment url url`	Specifies the URL of the CA on which your switch should send certificate requests; for example, http://ca_server. • *url*—Must be in the form of http://CA_name,_ where CA_name is the name of the CA's host Domain Name System or the IP address.
Step 3	`Router(ca-trustpoint)# subject-name [x.500-name]`	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name*—If the *x-500-name* argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.

	Command	Purpose
Step 4	Router(ca-trustpoint)# **ip-address** {*interface* | **none**}	Includes the IP address of the specified interface in the certificate request. • *interface*—IP address of the interface. • **none**—Specify this keyword if no IP address should be included. If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.
Step 5	Router(ca-trustpoint)# **serial-number** [**none**]	Specifies the switch serial number in the certificate request, unless the **none** keyword is specified. • **none**—Specify this keyword if no serial number should be included.
Step 6	Router(ca-trustpoint)# **regenerate**	Enables key rollover with certificate enrollment when the **crypto pki enroll** command is entered. **Note** This command generates a new key for the certificate even if a named key already exists. Do not use the **crypto key generate** command with the key rollover feature. If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: ! RSA key pair associated with trustpoint is exportable.
Step 7	Router(ca-trustpoint)# **password** *string*	(Optional) Specifies the revocation password for the certificate. • *string*—Text of the password. **Note** If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint.

	Command	Purpose
Step 8	Router(ca-trustpoint)# **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]	Specifies which key pair to associate with the certificate. • *key-label*—Name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) Size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the *encryption-key-size*.) • *encryption-key-size*—(Optional) Size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.) **Note** If this command is not enabled, the FQDN key pair is used.
Step 9	Router(ca-trustpoint)# **exit**	Exits ca-trustpoint configuration mode and enters global configuration mode.
Step 10	Router(config)# **crypto pki authenticate** *name*	Authenticates the CA (by obtaining the certificate of the CA.) • *name*—Name of the CA. Enter the *name* value entered in Step 1. Check the certificate fingerprint if prompted. **Note** This command is optional if the CA certificate is already loaded into the configuration.
Step 11	Router(config)# **crypto pki enroll** *name*	Requests certificates for all of your RSA key pairs. • *name*—Name of the CA. This command causes your switch to request as many certificates as there are RSA key pairs, so you need perform this command only once, even if you have special-usage RSA key pairs. When the **regenerate** configuration command is configured, this command will perform key rollover. **Note** This command requires you to create a challenge password that is not saved with the configuration. This password is required if your certificate needs to be revoked, so you must remember this password.
Step 12	Router(config)# **exit**	Exits global configuration mode.

For complete configuration information for key rollover for certificate renewal, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gtkyroll.html

For key rollover configuration examples, see the "Key Rollover for Certificate Renewal Configuration Examples" section on page 26-60.

Configuring PKI: Query Multiple Servers During Certificate Revocation Check

Before an X.509 certificate presented by a peer is validated, the certificate revocation list (CRL) is checked to make sure that the certificate has not been revoked by the issuing certification authority (CA). The certificate usually contains a certificate distribution point (CDP) in the form of a URL. Cisco IOS software uses the CDP to locate and retrieve the CRL.

Previous versions of Cisco IOS software make only one attempt to retrieve the CRL, even when the certificate contains more than one CDP. If the CDP server does not respond, the Cisco IOS software reports an error, which may result in the peer's certificate being rejected.

The PKI:query multiple servers during certificate revocation check feature provides the ability for Cisco IOS software to make multiple attempts to retrieve the CRL by trying all of the available CDPs in a certificate. This allows operations to continue when a particular server is not available. In addition, the ability to override the CDPs in a certificate with a manually configured CDP is also provided. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.

To manually override the existing CDPs for a certificate with a URL or directory specification, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto pki trustpoin`**`t` *`name`*	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	`Router(ca-trustpoint)#` **`match certificate`** *`certificate-map-label`* **`override cdp`** `{`**`url`** `|` **`directory`**`}` *`string`*	Manually overrides the existing CDP entries for a certificate with a URL or directory specification. • *certificate-map-label*—A user-specified label that must match the label argument specified in a previously defined **crypto pki certificate map** command. • **url**—Specifies that the certificate's CDPs will be overridden with an HTTP or LDAP URL. • **directory**—Specifies that the certificate's CDPs will be overridden with an LDAP directory specification. • *string*—The URL or directory specification. Some applications may time out before all CDPs have been tried and will report an error message. This will not affect the switch, and the Cisco IOS software will continue attempting to retrieve a CRL until all CDPs have been tried.

For complete configuration information for the PKI: Query Multiple Servers During Certificate Revocation Check feature, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gtcertrc.html

For a query multiple servers configuration example, see the "PKI: Query Multiple Servers During Certificate Revocation Check (CDP Override) Configuration Example" section on page 26-61.

Configuring the Online Certificate Status Protocol

The Online Certificate Status Protocol (OCSP) feature allows users to enable OCSP instead of certificate revocation lists (CRLs) to check certificate status. Unlike CRLs, which provide only periodic certificate status, OCSP can provide timely information regarding the status of a certificate.

OCSP Configuration Guidelines and Restrictions

When configuring OCSP, follow these guidelines and restrictions:

- OCSP transports messages over HTTP, so there may be a time delay when you access the OCSP server. If the OCSP server is unavailable, certificate verification will fail.

- The increased certificate size may cause a problem for low-end switches when certificates are stored on NVRAM. Before you add the Authority Info Access (AIA) extension to a certificate, make sure that the increased size will not cause deployment problems.
- An OCSP server usually operates in either push or poll mode. You can configure a CA server to push revocation information to an OCSP server or configure an OCSP server to periodically download (poll) a CRL from the CA server. To ensure that timely certificate revocation status is obtained, you should carefully consider the push and poll interval.
- When configuring an OCSP server to return the revocation status for a CA server, the OCSP server must be configured with an OCSP response signing certificate that is issued by that CA server. Ensure that the signing certificate is in the correct format, or the switch will not accept the OCSP response. Refer to your OCSP manual for additional information.

To configure your switch for OCSP to check certificate status, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto pki trustpoint name`	Declares the CA that your switch should use and puts you in ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	`Router(ca-trustpoint)# ocsp url url`	(Optional) Specifies the URL of an OCSP server so that the trustpoint can check the certificate status. This URL will override the URL of the OCSP server (if one exists) in the Authority Info Access (AIA) extension of the certificate. • *url* —Specifies the HTTP URL to be used.
Step 3	`Router(ca-trustpoint)# revocation-check method1 [method2[method3]]`	Checks the revocation status of a certificate. • *method1* [*method2*[*method3*]]—Specifies the method used by the switch to check the revocation status of the certificate. Available methods are as follows: – **crl**—Certificate checking is performed by a CRL. This is the default option. – **none**—Certificate checking is ignored. – **ocsp**—Certificate checking is performed by an OCSP server. If a second and third method are specified, each method will be used only if the previous method returns an error, such as a server being down.

Verifying the OCSP Configuration

To display information about your certificate and the CA certificate, enter the **show crypto pki certificates** command:

```
Router# show crypto pki certificates

Certificate
```

```
Status: Available
Version: 3
Certificate Serial Number: 18C1EE03000000004CBD
Certificate Usage: General Purpose

Issuer:
  cn=msca-root
  ou=pki msca-root
  o=cisco
  l=santa cruz2
  st=CA
  c=US
  ea=user@example.com

Subject:
  Name: myrouter.example.com
  hostname=myrouter.example.com

CRL Distribution Points:
  http://msca-root/CertEnroll/msca-root.crl

Validity Date:
  start date: 19:50:40 GMT Oct 5 2004
  end   date: 20:00:40 GMT Oct 12 2004

  Subject Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (360 bit)

  Signature Algorithm: SHA1 with RSA Encryption
  Fingerprint MD5: 2B5F53E6 E3E892E6 3A9D3706 01261F10
  Fingerprint SHA1: 315D127C 3AD34010 40CE7F3A 988BBDA5 CD528824

  X509v3 extensions:
  X509v3 Key Usage: A0000000
  Digital Signature
  Key Encipherment
  X509v3 Subject Key ID: D156E92F 46739CBA DFE66D2D 3559483E B41ECCF4
  X509v3 Authority Key ID: 37F3CC61 AF5E7C0B 434AB364 CF9FA0C1 B17C50D9
  Authority Info Access:

  Associated Trustpoints: msca-root
  Key Label: myrouter.example.com

 CA Certificate

  Status: Available
  Version: 3
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
  Certificate Usage: Signature

  Issuer:
  cn=msca-root
  ou=pki msca-root
  o=cisco
  l=santa cruz2
  st=CA
  c=US
  ea=user@example.com

  Subject:
  cn=msca-root
  ou=pki msca-root
  o=cisco
```

```
    l=santa cruz2
    st=CA
    c=US
    ea=user@example.com

    CRL Distribution Points:
    http://msca-root.example.com/CertEnroll/msca-root.crl

    Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017

    Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (512 bit)

    Signature Algorithm: SHA1 with RSA Encryption
    Fingerprint MD5: 84E470A2 38176CB1 AA0476B9 C0B4F478
    Fingerprint SHA1: 0F57170C 654A5D7D 10973553 EFB0F94F 2FAF9837

    X509v3 extensions:
    X509v3 Key Usage: C6000000
    Digital Signature
    Non Repudiation
    Key Cert Sign
    CRL Signature

  X509v3 Subject Key ID: 37F3CC61 AF5E7C0B 434AB364 CF9FA0C1 B17C50D9
  X509v3 Basic Constraints:
    CA: TRUE

  Authority Info Access:
  Associated Trustpoints: msca-root
```

To display the trustpoints and configured trustpoint subcommands that are configured in the switch, enter the **show crypto pki trustpoints** command:

```
Router# show crypto pki trustpoints

Trustpoint bo:
    Subject Name:
    CN = bomborra Certificate Manager
    O = cisco.com
    C = US
    Serial Number:01
    Certificate configured.
    CEP URL:http://bomborra
    CRL query url:ldap://bomborra
```

For complete configuration information for OCSP, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t2/feature/guide/gt_ocsp.html

For OCSP configuration examples, see the "Online Certificate Status Protocol Configuration Examples" section on page 26-61.

Configuring Certificate Security Attribute-Based Access Control

Under the IPsec protocol, certificate authority (CA) interoperability permits Cisco IOS devices and a CA to communicate so that the Cisco IOS device can obtain and use digital certificates from the CA. Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action. The certificate security attribute-based access control feature adds fields to the certificate to create a certificate-based ACL.

Certificate Security Attribute-Based Access Control Configuration Guidelines and Restrictions

When configuring certificate security attribute-based access control, follow these guidelines and restrictions:

- The certificate-based ACL specifies one or more fields within the certificate and an acceptable value for each specified field. You can specify which fields within a certificate should be checked and which values those fields may or may not have. There are six logical tests for comparing the field with the value: equal, not equal, contains, does not contain, less than, and greater than or equal.
- If more than one field is specified within a single certificate-based ACL, the tests of all of the fields within the ACL must succeed to match the ACL.
- The same field can be specified multiple times within the same ACL.
- More than one ACL can be specified. Each ACL will be processed in turn until a match is found or all of the ACLs have been processed.
- Memory is required to hold the ACLs as they are created and as they are loaded from the configuration file. The amount of memory depends on which fields within the certificate are being checked and how many ACLs have been defined. Certificate-based ACL support requires one or more compare operations when the fields in a certificate are being checked. Only the fields specified by the ACL are checked. The compare operations are a small part of certificate validation and will not have a noticeable effect on switch performance when validating certificates.

To configure Certificate Security Attribute-Based Access Control, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki certificate map** *label sequence-number*	Starts ca-certificate-map mode and defines certificate-based ACLs by assigning a label for the ACL that will also be referenced within the **crypto pki trustpoint** command. • *label*—An arbitrary string that identifies the ACL. • *sequence-number*—A sequence number that orders ACLs with the same label.
Step 2	Router(ca-certificate-map)# *field-name match-criteria match-value*	In ca-certificate-map mode, you specify one or more certificate fields together with their matching criteria and the value to match. • *field-name*—Specifies one of the following case-insensitive name strings or a date: – **subject-name** – **issuer-name** – **unstructured-subject-name** – **alt-subject-name** – **name** – **valid-start** – **expires-on** **Note** Date field format is *dd mm yyyy hh*:*mm*:*ss* or *mmm dd yyyy hh*:*mm*:*ss*. • *match-criteria*—Specifies one of the following logical operators: – **eq**—Equal (valid for name and date fields) – **ne**—Not equal (valid for name and date fields) – **co**—Contains (valid only for name fields) – **nc**—Does not contain (valid only for name fields) – **lt** —Less than (valid only for date fields) – **ge** —Greater than or equal (valid only for date fields) • *match-value*—Specifies the name or date to test with the logical operator assigned by *match-criteria*. For example: Router(ca-certificate-map)# **subject-name co Cisco**
Step 3	Router(ca-certificate-map)# **exit**	Exits ca-certificate-map mode.

	Command	Purpose
Step 4	`Router(config)# `**`crypto pki trustpoint`** *`name`*	Starts ca-trustpoint configuration mode and creates a name for the CA. • *name*—Specifies a name for the CA.
Step 5	`Router(ca-trustpoint)# `**`match certificate`** *`certificate-map-label`*	Associates the certificate-based ACL defined with the **crypto pki certificate map** command to the trustpoint. • *certificate-map-label*—Specifies the label argument specified in the previously defined **crypto pki certificate map** command in Step 1.
Step 6	`Router(ca-trustpoint)# `**`exit`**	Exits ca-trustpoint configuration mode.

Verifying Certificate-Based ACLs

To verify the certificate-based ACL configuration, enter the **show crypto pki certificates** command. The following example shows the components of the certificates (CA and switch certificate) installed on the switch when the switch has both authenticated and enrolled with a trustpoint:

```
Router# show crypto pki certificates

CA Certificate
   Status: Available
   Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
   Certificate Usage: Signature

   Issuer:
    CN = new-user
    OU = pki new-user
    O = cisco
    L = santa cruz2
    ST = CA
    C = US
    EA = user@cysco.net

    Subject:
    CN = new-user
    OU = pki new-user
    O = cisco
    L = santa cruz2
    ST = CA
    C = US
    EA = user@cysco.net

    CRL Distribution Point:
    http://new-user.cysco.net/CertEnroll/new-user.crl

    Validity Date:
    start date: 14:19:29 PST Oct 31 2002
    end date: 14:27:27 PST Oct 31 2017

    Associated Trustpoints: MS


Certificate
  Status: Available
  Certificate Serial Number: 193E28D20000000009F7
Certificate Usage: Signature

Issuer:
   CN = new-user
   OU = pki new-user
   O = cisco
   L = santa cruz2
   ST = CA
  C = US
   EA = user@cysco.net

Subject:
  Name: User1.Cysco.Net
  OID.1.2.840.113549.1.9.2 = User1.Cysco.Net

CRL Distribution Point:
  http://new-user.cysco.net/CertEnroll/new-user.crl

Validity Date:
  start date: 12:40:14 PST Feb 26 2003
  end   date: 12:50:14 PST Mar 5 2003
  renew date: 16:00:00 PST Dec 31 1969

Associated Trustpoints: MS
```

For complete configuration information for Certificate Security Attribute-Based Access Control, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ftcrtacl.html

For a certificate-based ACL example, see the "Certificate Security Attribute-Based Access Control Configuration Example" section on page 26-62.

Configuring PKI AAA Authorization Using the Entire Subject Name

When using public key infrastructure (PKI) and authentication, authorization, and accounting (AAA) functionality, users sometimes have attribute-value (AV) pairs that are different from those of every other user. As a result, a unique username is required for each user. The PKI AAA authorization using the entire subject name feature provides users with the ability to query the AAA server using the entire subject name from the certificate as a unique AAA username.

PKI AAA Authorization Using the Entire Subject Name Configuration Guidelines and Restrictions

When configuring PKI AAA authorization using the entire subject name, follow these guidelines and restrictions:

- Some AAA servers limit the length of the username (for example, to 64 characters). As a result, the entire certificate subject name cannot be longer than the limitation of the server.
- Some AAA servers limit the available character set that may be used for the username (for example, a space [ ] and an equal sign [=] may not be acceptable). This feature will not work for the AAA server having such a character-set limitation.

- The **subject-name** command in the trustpoint configuration might not always be the final AAA subject name. If the fully qualified domain name (FQDN), serial number, or IP address of the switch are included in a certificate request, the subject name field of the issued certificate will also have these components. To turn off the components, use the **fqdn**, **serial-number**, and **ip-address** commands with the **none** keyword.
- Certificate authority (CA) servers sometimes change the requested subject name field when they issue a certificate. For example, CA servers of some vendors switch the relative distinguished names (RDNs) in the requested subject names to the following order: CN, OU, O, L, ST, and C. However, another CA server might append the configured Lightweight Directory Access Protocol (LDAP) directory root (for example, O=cisco.com) to the end of the requested subject name.
- Depending on the tools you choose for displaying a certificate, the printed order of the RDNs in the subject name could be different. Cisco IOS software always displays the least significant RDN first, but other software, such as Open Source Secure Socket Layer (OpenSSL), does the opposite. Therefore, if you are configuring the AAA server with a full DN (subject name) as the corresponding username, ensure that the Cisco IOS software style (that is, with the least-significant RDN first) is used.

To configure the entire certificate subject name for PKI authentication, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables the AAA access control model.
Step 2	`Router config)# aaa authorization network listname [method]`	Sets the parameters that restrict user access to a network. • *listname*—Character string used to name the list of authorization methods. • *method*—(Optional) Specifies an authorization method to be used for authorization. The *method* argument can be **group radius**, **group tacacs+**, or **group** *group-name*.
Step 3	`Router(config)# crypto pki trustpoint name`	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 4	`Router(ca-trustpoint)# enrollment url url`	Specifies the enrollment parameters of your CA. • *url*—The *url* argument is the URL of the CA to which your switch should send certificate requests.
Step 5	`Router(ca-trustpoint)# revocation-check method`	(Optional) Checks the revocation status of a certificate. • *method*—Method used by the switch to check the revocation status. Available methods are **ocsp**, **none**, and **crl**.
Step 6	`Router(ca-trustpoint)# exit`	Exits ca-truspoint configuration mode and enters global configuration mode.
Step 7	`Router(config)# authorization list {listname}`	Specifies the AAA authorization list. • *listname*—Name of the list.

	Command	Purpose
Step 8	`Router(config)#` **`authorization username subjectname all`**	Sets parameters for the different certificate fields that are used to build the AAA username. • **all**—Specifies that the entire subject name of the certificate will be used as the authorization username.
Step 9	`Router(config)#` **`tacacs-server host`** *`hostname`* `[`**`key`** *`string`*`]` or	Specifies a TACACS+ host. • *hostname*—Name of the host. • **key** *string*—(Optional) Character string specifying authentication and encryption key.
	`Router(config)#` **`radius-server host`** *`hostname`* `[`**`key`** *`string`*`]`	Specifies a RADIUS host.

For complete configuration information for the PKI AAA authorization using the entire subject name feature, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t11/feature/guide/gt_dnall.html

For a PKI AAA Authorization Using the Entire Subject Name configuration example, see the "PKI AAA Authorization Using the Entire Subject Name Configuration Example" section on page 26-62.

Configuring Source Interface Selection for Outgoing Traffic with Certificate Authority

The source interface selection for outgoing traffic with certificate authority feature allows you to specify that the address of an interface be used as the source address for all outgoing TCP connections associated with that trustpoint when a designated trustpoint has been configured.

To configure the interface that you want to use as the source address for all outgoing TCP connections associated with a trustpoint, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`crypto pki trustpoin`**`t` *`name`*	Declares the CA that your switch should use and enters ca-trustpoint configuration mode. • *name*—Name for the trustpoint CA.
Step 2	`Router(ca-trustpoint)#` **`enrollment url`** *`url`*	Specifies the enrollment parameters of your CA. • *url*—Specifies the URL of the CA where your switch should send certificate requests; for example, http://ca_server. *url* must be in the form http://CA_*name*, where CA_*name* is the CA's host Domain Name System (DNS) name or IP address.
Step 3	`Router(ca-trustpoint)#` **`source interface`** *`interface-address`*	Specifies the interface to be used as the source address for all outgoing TCP connections associated with that trustpoint. • *interface-address*—Interface address.

	Command	Purpose
Step 4	Router(config)# **interface** *type slot*/[subslot]/*port*	Configures an interface type and enters interface configuration mode. • *type*—Type of interface being configured. • *slot*/[subslot]/ *port*—Number of the slot, subslot (optional), and port to be configured.
Step 5	Router(config-if)# **description** *string*	Adds a description to an interface configuration. • *string*—Descriptive string.
Step 6	Router(config-if)# **ip address** *ip-address mask*	Sets a primary or secondary IP address for an interface. • *ip-address*—IP address. • *mask*—Subnet mask.
Step 7	Router(config-if)# **interface** *type slot*/[*subslot*]/*port*	Configures an interface type. • *type*—Type of interface being configured. • *slot*/[*subslot*]/ *port*—Number of the slot, subslot (optional), and port to be configured.
Step 8	Router(config-if)# **description** *string*	Adds a description to an interface configuration. • *string*—Descriptive string.
Step 9	Router(config-if)# **ip address** *ip-address mask* [*secondary*]	Sets a primary or secondary IP address for an interface. • *ip-address*—IP address. • *mask*—Subnet mask. • [*secondary*]—(Optional) Secondary address.
Step 10	Router(config-if)# **crypto map** *map-name*	Applies a previously defined crypto map set to the interface. • *map-name*—Name that identifies the crypto map set.

For complete configuration information for source interface selection for outgoing traffic with certificate authority, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ft_asish.html

For a source interface selection configuration example, see the "Source Interface Selection for Outgoing Traffic with Certificate Authority Configuration Example" section on page 26-63.

Configuring Persistent Self-Signed Certificates

The persistent self-signed certificates feature saves a certificate generated by a Secure HTTP (HTTPS) server for the Secure Sockets Layer (SSL) handshake in a router's startup configuration.

Note The persistent self-signed certificates feature is supported in Cisco IOS Release 12.2(33)SXH and later releases.

Cisco IOS software has an HTTPS server that allows access to web-based management pages using a secure SSL connection. SSL requires the server to have an X.509 certificate that is sent to the client (web browser) during the SSL handshake to establish a secure connection between the server and the client.

The client expects the SSL server's certificate to be verifiable using a certificate the client already possesses.

If Cisco IOS software does not have a certificate that the HTTPS server can use, the server generates a self-signed certificate by calling a public key infrastructure (PKI) application programming interface (API). When the client receives this self-signed certificate and is unable to verify it, intervention is needed. The client asks you if the certificate should be accepted and saved for future use. If you accept the certificate, the SSL handshake continues.

Future SSL handshakes between the same client and the server use the same certificate. However, if the router is reloaded, the self-signed certificate is lost. The HTTPS server must then create a new self-signed certificate. This new self-signed certificate does not match the previous certificate, so you are once again asked to accept it.

Requesting acceptance of the router's certificate each time that the router reloads can be annoying and may present an opportunity for an attacker to substitute an unauthorized certificate during the time that you are being asked to accept the certificate.

The persistent self-signed certificates feature overcomes all these limitations by saving a certificate in the router's startup configuration, resulting in the following benefits:

- Having a persistent self-signed certificate stored in the router's startup configuration (NVRAM) lessens the opportunity for an attacker to substitute an unauthorized certificate because the browser is able to compare the certificate offered by the router with the previously saved certificate and warn you if the certificate has changed.
- Having a persistent self-signed certificate stored in the router's startup configuration eliminates the user intervention that is necessary to accept the certificate every time that the router reloads.
- Because user intervention is no longer necessary to accept the certificate, the secure connection process is faster.

Persistent Self-Signed Certificates Configuration Guidelines and Restrictions

When configuring persistent self-signed certificates, follow these guidelines and restrictions:

- You must load an image that supports SSL.
- You can configure only one trustpoint for a persistent self-signed certificate.

Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters

Note This section is optional because if you enable the Secure HTTP (HTTPS) server, it generates a self-signed certificate automatically using default values. To specify parameters, you must create a trustpoint and configure it. To use default values, delete any existing self-signed trustpoints. Deleting all self-signed trustpoints causes the HTTPS server to generate a persistent self-signed certificate using default values as soon as it is enabled.

To configure a trustpoint and specify self-signed certificate parameters, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto pki trustpoint** *name*	Declares the certificate authority (CA) that your router should use and enters ca-trustpoint configuration mode. • *name*—Name of the CA.
Step 2	Router(ca-trustpoint)# **enrollment selfsigned**	Specifies self-signed enrollment.
Step 3	Router(ca-trustpoint)# **subject-name** [*x.500-name*]	(Optional) Specifies the requested subject name to be used in the certificate request. • *x.500-name*—(Optional) If the x.500-name argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.
Step 4	Router(ca-trustpoint)# **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]	(Optional) Specifies which key pair to associate with the certificate. • *key-label*—Name of the key pair, which is generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command is configured. • *key-size*—(Optional) Size of the desired RSA key. If not specified, the existing key size is used. (The specified size must be the same as the *encryption-key-size*.) • *encryption-key-size*—(Optional) Size of the second key, which is used to request separate encryption, signature keys, and certificates. (The specified size must be the same as the *key-size*.) **Note** If this command is not enabled, the FQDN key pair is used.
Step 5	Router(ca-trustpoint)# **crypto pki enroll** *trustpoint-name*	Tells the router to generate the persistent self-signed certificate. • *trustpoint-name*—Name of the CA.
Step 6	Router(ca-trustpoint)# **end**	(Optional) Exits ca-trustpoint configuration mode.

Enabling the HTTPS Server

To enable the HTTPS server, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip http secure-server`	Enables the secure HTTP web server. **Note** A key pair (modulus 1024) and a certificate are generated.
Step 2	`Router(config)# end`	Exits global configuration mode.

Note You must enter a **write memory** command to save the configuration. This command also saves the self-signed certificate and the HTTPS server in enabled mode.

Verifying the Persistent Self-Signed Certificate Configuration

To verify that a self-signed certificate and a trustpoint have been created, use the **show crypto pki certificates**, **show crypto mypubkey rsa,** and the **show crypto pki trustpoints** commands.

The **show crypto pki certificates** command displays information about your certificate, the CA certificate, and any registration authority certificates:

```
Router# show crypto pki certificates

Router Self-Signed Certificate
   Status: Available
   Certificate Serial Number: 01
   Certificate Usage: General Purpose
   Issuer:
    cn=IOS-Self-Signed-Certificate-3326000105
   Subject:
    Name: IOS-Self-Signed-Certificate-3326000105
    cn=IOS-Self-Signed-Certificate-3326000105
   Validity Date:
    start date: 19:14:14 GMT Dec 21 2004
    end   date: 00:00:00 GMT Jan 1 2020
   Associated Trustpoints: TP-self-signed-3326000105
```

Note The number 3326000105 above is the router's serial number and varies depending on the router's actual serial number.

The **show crypto mypubkey rsa** command displays information about the key pair corresponding to the self-signed certificate:

```
Router# show crypto mypubkey rsa

% Key pair was generated at: 19:14:10 GMT Dec 21 2004
Key name: TP-self-signed-3326000105
  Usage: General Purpose Key
  Key is not exportable.
  Key Data:
    30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00B88F70
    6BC78B6D 67D6CFF3 135C1D91 8F360292 CA44A032 5AC1A8FD 095E4865 F8C95A2B
    BFD1C2B7 E64A3804 9BBD7326 207BD456 19BAB78B D075E78E 00D2560C B09289AE
    6DECB8B0 6672FB3A 5CDAEE92 9D4C4F71 F3BCB269 214F6293 4BA8FABF 9486BCFC
    2B941BCA 550999A7 2EFE12A5 6B7B669A 2D88AB77 39B38E0E AA23CB8C B7020301 0001
% Key pair was generated at: 19:14:13 GMT Dec 21 2004
Key name: TP-self-signed-3326000105.server
  Usage: Encryption Key
  Key is not exportable.
  Key Data:
    307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00C5680E 89777B42
    463E5783 FE96EA9E F446DC7B 70499AF3 EA266651 56EE29F4 5B003D93 2FC9F81D
    8A46E12F 3FBAC2F3 046ED9DD C5F27C20 1BBA6B9B 08F16E45 C34D6337 F863D605
    34E30F0E B4921BC5 DAC9EBBA 50C54AA0 BF551BDD 88453F50 61020301 0001
```

Note The second key pair with the name TP-self-signed-3326000105.server is the SSH key pair and is generated once any key pair is created on the router and SSH starts up.

The **show crypto pki trustpoints** command displays the trustpoints that are configured in the router:

```
Router# show crypto pki trustpoints

Trustpoint local:
    Subject Name:
    serialNumber=C63EBBE9+ipaddress=10.3.0.18+hostname=test.cisco.com
        Serial Number: 01
    Persistent self-signed certificate trust point
```

For complete configuration information for persistent self-signed certificates, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t14/feature/guide/gtpsscer.html

For persistent self-signed certificates configuration examples, see the "Persistent Self-Signed Certificates Configuration Examples" section on page 26-63.

Configuration Examples

This section provides examples of the following configurations:

- Multiple RSA Key Pairs Configuration Example, page 26-54
- Protected Private Key Storage Configuration Examples, page 26-54
- Trustpoint CA Configuration Example, page 26-54
- Query Mode Definition Per Trustpoint Configuration Example, page 26-55
- Local Certificate Storage Location Configuration Example, page 26-55
- Direct HTTP Enrollment with CA Servers Configuration Examples, page 26-55
- Manual Certificate Enrollment Configuration Examples, page 26-57
- Certificate Autoenrollment Configuration Example, page 26-60
- Key Rollover for Certificate Renewal Configuration Examples, page 26-60
- PKI: Query Multiple Servers During Certificate Revocation Check (CDP Override) Configuration Example, page 26-61
- Online Certificate Status Protocol Configuration Examples, page 26-61
- Certificate Security Attribute-Based Access Control Configuration Example, page 26-62

Note The following examples use commands at the level of Cisco IOS Release 12.2(33)SXH.

As of Cisco IOS Release 12.2(33)SXH, the **crypto engine subslot** command used in previous releases has been replaced with the **crypto engine slot** command (of the form **crypto engine slot** *slot* {**inside** | **outside**}). The **crypto engine subslot** command is no longer supported. When upgrading, ensure that this command has been modified in your start-up configuration to avoid extended maintenance time.

Multiple RSA Key Pairs Configuration Example

The following example is a sample trustpoint configuration that specifies the RSA key pair exampleCAkeys:

```
Router(config)# crypto key generate rsa general-keys label exampleCAkeys
Router(config)# crypto pki trustpoint exampleCAkeys
Router(ca-trustpoint)# enrollment url http://exampleCAkeys/certsrv/mscep/mscep.dll
Router(ca-trustpoint)# rsakeypair exampleCAkeys 1024 1024
```

Protected Private Key Storage Configuration Examples

This section contains the following configuration examples:

Encrypted Key Configuration Example

The following example shows how to encrypt the pki1-72a.cisco.com RSA key:

```
Router(config)# crypto key encrypt rsa name pki1-72a.cisco.com passphrase cisco1234
Router(config)# exit
```

Locked Key Configuration Example

The following example shows how to lock the pki1-72a.cisco.com key:

```
Router# crypto key lock rsa name pki1-72a.cisco.com passphrase cisco1234
```

Trustpoint CA Configuration Example

The following example shows how to declare the CA named kahului and specify characteristics for the trustpoint CA:

```
Router(config)# crypto pki trustpoint kahului
Router(ca-trustpoint)# enrollment url http://kahului
Router(ca-trustpoint)# crl query ldap://kahului
```

Query Mode Definition Per Trustpoint Configuration Example

The following configuration example shows a trustpoint CA that uses query mode:

```
Router(config)# crypto pki trustpoint trustpoint1
Router(ca-trustpoint)# enrollment url http://ca-server1
Router(ca-trustpoint)# crl query http://ca-server1
Router(ca-trustpoint)# query certificate
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate trustpoint1
Router(config)# crypto key generate rsa
Router(config)# crypto pki enroll trustpoint1
```

Local Certificate Storage Location Configuration Example

The following example shows how to store certificates to the certs subdirectory. Note that the certs subdirectory does not exist and is automatically created.

```
Router(config)# crypto pki certificate storage disk0:/certs
Requested directory does not exist -- created
Certificates will be stored in disk0:/certs/
Router(config)# end
Router# write
*May 27 02:09:00:%SYS-5-CONFIG_I:Configured from console by consolemem
Building configuration...
[OK]
Router# directory disk0:/certs
Directory of disk0:/certs/
14  -rw-        707  May 27 2005 02:09:02 +00:00  ioscaroot#7401CA.cer
15  -rw-        863  May 27 2005 02:09:02 +00:00  msca-root#826E.cer
16  -rw-        759  May 27 2005 02:09:02 +00:00  msca-root#1BA8CA.cer
17  -rw-        863  May 27 2005 02:09:02 +00:00  msca-root#75B8.cer
18  -rw-       1149  May 27 2005 02:09:02 +00:00  storagename#6500CA.cer
19  -rw-        863  May 27 2005 02:09:02 +00:00  msca-root#83EE.cer
47894528 bytes total (20934656 bytes free)

! The certificate files are now on disk0/certs:
```

Direct HTTP Enrollment with CA Servers Configuration Examples

This section provides the following configuration examples:

- Enrollment Profile for a Client Switch Configuration Example, page 26-55
- Enrollment Profile for a Client Switch Already Enrolled with a Third-Party Vendor CA Example, page 26-56
- Certificate Server Automatically Accepting Enrollment Requests Only from the Client Switch Configuration Example, page 26-56

Enrollment Profile for a Client Switch Configuration Example

The following example show how to configure an enrollment profile for direct HTTP enrollment with a CA server:

```
Router(config)# crypto pki trustpoint Entrust
Router(ca-trustpoint)# enrollment profile E
Router(ca-trustpoint)# exit
Router(config)# crypto pki profile enrollment E
Router(ca-profile-enroll)# authentication url  http://entrust:81
Router(ca-profile-enroll)# authentication command  GET /certs/cacert.der
Router(ca-profile-enroll)# enrollment url  http://entrust:81/cda-cgi/clientcgi.exe
Router(ca-profile-enroll)# enrollment command  POST reference_number=$P2&authcode=$P1
&retrievedAs=rawDER&action=getServerCert&pkcs10Request=$REQ
Router(ca-profile-enroll)# parameter 1 value aaaa-bbbb-cccc
Router(ca-profile-enroll)# parameter 2 value 5001

Router(config)# crypto ca profile enrollment E
Router(ca-profile-enroll)# authentication url http://ca-server.example.com
Router(ca-profile-enroll)# authentication command GET $P1
Router(ca-profile-enroll)# enrollment url
Router(ca-profile-enroll)# enrollment command ^C
POST action=getServerCert
&pkcs10Request=$REQ
&reference_number=$P2
&authcode=P3
&retrievedAs=rawDER
^C
Router(ca-profile-enroll)# parameter 1 value cacert.crt
Router(ca-profile-enroll)# parameter 2 prompt Enter the Reference Number:
Router(ca-profile-enroll)# parameter 3 prompt Enter the Auth Code:
```

Enrollment Profile for a Client Switch Already Enrolled with a Third-Party Vendor CA Example

The following example shows how to configure the following tasks on the client switch:

- Define the msca-root trustpoint that points to the third-party vendor CA and enroll and authenticate the client with the third-party vendor CA.
- Define cs trustpoint for the Cisco IOS CA.
- Define enrollment profile "cs1," which points to Cisco IOS CA, and mention (via the enrollment credential command) that msca-root is being initially enrolled with the Cisco IOS CA.

```
! Define trustpoint "msca-root" for non-Cisco IOS CA.
Router(config)# crypto pki trustpoint msca-root
Router(ca-trustpoint)# enrollment mode ra
Router(ca-trustpoint)# enrollment url http://msca-root:80/certsrv/mscep/mscep.dll
Router(ca-trustpoint)# ip-address FastEthernet2/0
Router(ca-trustpoint)# revocation-check crl

! Configure trustpoint "cs" for Cisco IOS CA.
Router(config)# crypto pki trustpoint cs
Router(ca-trustpoint)# enrollment profile cs1
Router(ca-trustpoint)# revocation-check crl

! Define enrollment profile "cs1."
Router(config)# crypto pki profile enrollment cs1
Router(ca-profile-enroll)# enrollment url  http://cs:80
Router(ca-profile-enroll)# enrollment credential  msca-root
```

Certificate Server Automatically Accepting Enrollment Requests Only from the Client Switch Configuration Example

The following example shows how to configure the certificate server, and enter the **grant auto trustpoint** command to instruct the certificate server to accept enrollment requests only from clients who are already enrolled with msca-root trustpoint:

```
Router(config)# crypto pki server cs
Router(cs-server)# database level minimum
Router(cs-server)# database url nvram:
Router(cs-server)# issuer-name CN=cs
Router(cs-server)# grant auto trustpoint msca-root

Router(config)# crypto pki trustpoint cs
Router(ca-trustpoint)# revocation-check crl
Router(ca-trustpoint)# rsakeypair cs

Router(ca-trustpoint)# crypto pki trustpoint msca-root
Router(ca-trustpoint)# enrollment mode ra
Router(ca-trustpoint)# enrollment url http://msca-root:80/certsrv/mscep/mscep.dll
Router(ca-trustpoint)# revocation-check crl
```

Manual Certificate Enrollment Configuration Examples

This section provides the following manual certificate enrollment configuration examples:

- Manual Certificate Enrollment Using TFTP Configuration Example, page 26-57
- Manual Certificate Enrollment Using Cut-and-Paste Configuration Example, page 26-57

Manual Certificate Enrollment Using TFTP Configuration Example

The following example shows the configuration of manual certificate enrollment using TFTP:

```
Router(config)# crypto pki trustpoint MS
Router(ca-trustpoint)# enrollment url tftp://CA-Server/TFTPfiles/switch1
Router(ca-trustpoint)# crypto pki authenticate MS
Router(ca-trustpoint)# exit
Router(config)# crypto pki enroll MS
Router(config)# crypto pki import MS certificate
```

Manual Certificate Enrollment Using Cut-and-Paste Configuration Example

The following example shows how to configure manual cut-and-paste certificate enrollment. In this example, the name of the trustpoint CA is MS, and the **crypto pki import** command is entered twice because usage keys (signature and encryption keys) are used.

```
Router(config)# crypto pki trustpoint MS
Router(ca-trustpoint)# enrollment terminal
Router(ca-trustpoint)# crypto pki authenticate MS


Enter the base 64 encoded CA certificate.

End with a blank line or the word "quit" on a line by itself


-----BEGIN CERTIFICATE-----

MIICNDCCAd6gAwIBAgIQOsCmXpVHwodKryRoqULV7jANBgkqhkiG9w0BAQUFADA5
MQswCQYDVQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJ
bXNjYS1yb290MB4XDTAyMDIxNDAwNDYwMVoXDTA3MDIxNDAwNTQ0OFowOTELMAkG
A1UEBhMCVVMxFjAUBgNVBAoTDUNpc2NvIFN5c3RlbXMxEjAQBgNVBAMTCW1zY2Et
cm9vdDBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQCix8nIGFg+wvy3BjFbVi25wYoG
K2N0HWWHpqxFuFhqyBnIC0OshIn9CtrdN3JvUNHr0NIKocEwNKUGYmPwWGTfAgMB
AAGjgcEwgb4wCwYDVR0PBAQDAgHGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYE
FKIacsl6dKAfuNDVQymlSp7esf8jMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9t
```

```
c2NhLXJvb3QvQ2VydEVucm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8v
XFxtc2NhLXJvb3RcQ2VydEVucm9sbFxtc2NhLXJvb3QuY3JsMBAGCSsGAQQBgjcV
AQQDAgEAMA0GCSqGSIb3DQEBBQUAA0EAeuZkZMX9qkoLHfETYTpVWjZPQbBmwNRA
oJDSdYdtL3BcI/uLL5q7EmODyGfLyMGxuhQYx5r/40aSQgLCqBq+yg==

-----END CERTIFICATE-----


Certificate has the following attributes:

Fingerprint:D6C12961 CD78808A 4E02193C 0790082A

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

Trustpoint CA certificate accepted.

% Certificate successfully imported


Router(config)#

Router(config)# crypto pki enroll MS

% Start certificate enrollment..



% The subject name in the certificate will be:Router.cisco.com

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

% Include an IP address in the subject name? [no]:n

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

Signature key certificate request -

Certificate Request follows:

MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTYW5kQmFnZ2VyLmNpc2NvLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAxdhXFDiWAn/hIZs9zfOtssKA
daoWYu0ms9Fe/Pew01dh14vXdxgacstOs2Pr5wk6jLOPxpvxOJPWyQM6ipLmyVxv
ojhyLTrVohrh6Dnqcvk+G/5ohss9o9RxvONwx042pQchFnx9EkMuZC7evwRxJEqR
mBHXBZ8GmP3jYQsjS8MCAwEAAaAhMB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/
BAQDAgeAMA0GCSqGSIb3DQEBBAUAA4GBAMT6WtyFw95POY7UtF+YIYHiVRUf4SCq
hRIAGrljUePLo9iTqyPU1Pnt8JnIZ5P5BHU3MfgP8sqodaWub6mubkzaohJ1qD06
O87fnLCNid5Tov5jKogFHIki2EGGZxBosUw9lJlenQdNdDPbJc5LIWdfDvciA6jO
Nl8rOtKnt8Q+

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

Redisplay enrollment request? [yes/no]:

Encryption key certificate request -

Certificate Request follows:

MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTYW5kQmFnZ2VyLmNpc2NvLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAwG60QojpDbzbKnyj8FyTiOcv
THkDP7XD4vLT1XaJ409z0gSIoGnIcdFtXhVlBWtpq3/O9zYFXr1tH+BMCRQi3Lts
0IpxYa3D9iFPqev7SPXpsAIsY8a6FMq7TiwLObqiQjLKL4cbuV0Frjl0Yuv5A/Z+
kqMOm7c+pWNWFdLe9lsCAwEAAaAhMB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/
BAQDAgUgMA0GCSqGSIb3DQEBBAUAA4GBACF7feURj/fJMojPBlR6fa9BrlMJx+2F
H91YM/CIiz2n4mHTeWTWKhLoT8wUfa9NGOk7yi+nF/F7035twLfq6n2bSCTW4aem
```

```
8jLMMaeFxwkrV/ceQKrucmNC1uVx+fBy9rhnKx8j60XE25tnp1U08r6om/pBQABU
eNPFhozcaQ/2

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


Redisplay enrollment request? [yes/no]:

n

Router(config)#crypto pki import MS certificate


Enter the base 64 encoded certificate.

End with a blank line or the word "quit" on a line by itself


MIIDajCCAxSgAwIBAgIKFN7C6QAAAAAMRzANBgkqhkiG9w0BAQUFADA5MQswCQYD
VQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1y
b290MB4XDTAyMDYwODAxMTY0MloXDTAzMDYwODAxMjY0MlowJTEjMCEGCSqGSIb3
DQEJAhMUU2FuZEJhZ2dlci5jaXNjby5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAMXYVxQ4lgJ/4SGbPc3zrbLCgHWqFmLtJrPRXvz3sNNXYdeL13cYGnLL
TrNj6+cJOoyzj8ab8TiT1skDOoqS5slcb6I4ci061aIa4eg56nL5Phv+aIbLPaPU
cbzjcMdONqUHIRZ8fRJDLmQu3r8EcSRKkZgR1wWfBpj942ELI0vDAgMBAAGjggHM
MIIByDALBgNVHQ8EBAMCB4AwHQYDVR0OBBYEFL8Quz8dyz4EGIeKx9A8UMNHLE4s
MHAGA1UdIwRpMGeAFKIacsl6dKAfuNDVQymlSp7esf8joT2kOzA5MQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1yb290
ghA6wKZelUfCh0qvJGipQtXuMCIGA1UdEQEB/wQYMBaCFFNhbmRCYWdnZXIuY2lz
Y28uY29tMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9tc2NhLXJvb3QvQ2VydEVu
cm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8vXFxtc2NhLXJvb3RcQ2Vy
dEVucm9sbFxtc2NhLXJvb3QuY3JsMIGUBggrBgEFBQcBAQSBhzCBhDA/BggrBgEF
BQcwAoYzaHR0cDovL21zY2Etcm9vdC9DZXJ0RW5yb2xsL21zY2Etcm9vdF9tc2Nh
LXJvb3QuY3J0MEEGCCsGAQUFBzAChjVmaWxlOi8vXFxtc2NhLXJvb3RcQ2VydEVu
cm9sbFxtc2NhLXJvb3RfbXNjYS1yb290LmNydDANBgkqhkiG9w0BAQUFAANBAJo2
r6sHPGBdTQX2EDoJpR/A2UHXxRYqVSHkFKZw0z31r5JzUM0oPNUETV7mnZlYNVRZ
CSEX/G8boi3WOjz9wZo=

% Router Certificate successfully imported


Router(config)#

Router(config)# crypto pki import MS certificate


Enter the base 64 encoded certificate.

End with a blank line or the word "quit" on a line by itself


MIIDajCCAxSgAwIBAgIKFN7OBQAAAAAMSDANBgkqhkiG9w0BAQUFADA5MQswCQYD
VQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1y
b290MB4XDTAyMDYwODAxMTY0NVoXDTAzMDYwODAxMjY0NVowJTEjMCEGCSqGSIb3
DQEJAhMUU2FuZEJhZ2dlci5jaXNjby5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAMButEKI6Q282yp8o/Bck4jnL0x5Az+1w+Ly09V2ieNPc9IEiKBpyHHR
bV4VZQVraat/zvc2BV69bR/gTAkUIty7bNCKcWGtw/YhT6nr+0j16bACLGPGuhTK
u04sCzm6okIyyi+HG7ldBa45dGLr+QP2fpKjDpu3PqVjVhXS3vZbAgMBAAGjggHM
MIIByDALBgNVHQ8EBAMCBSAwHQYDVR0OBBYEFPDO29oRdlEUSgBMg6jZR+YFRWlj
MHAGA1UdIwRpMGeAFKIacsl6dKAfuNDVQymlSp7esf8joT2kOzA5MQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1yb290
ghA6wKZelUfCh0qvJGipQtXuMCIGA1UdEQEB/wQYMBaCFFNhbmRCYWdnZXIuY2lz
Y28uY29tMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9tc2NhLXJvb3QvQ2VydEVu
cm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8vXFxtc2NhLXJvb3RcQ2Vy
```

```
dEVucm9sbFxtc2NhLXJvb3QuY3JsMIGUBggrBgEFBQcBAQSBhzCBhDA/BggrBgEF
BQcwAoYzaHR0cDovL21zY2Etcm9vdC9DZXJ0RW5yb2xsL21zY2Etcm9vdF9tc2Nh
LXJvb3QuY3J0MEEGCCsGAQUFBzAChjVmaWxlOi8vXFxtc2NhLXJvb3RcQ2VydEVu
cm9sbFxtc2NhLXJvb3RfbXNjYS1yb290LmNydDANBgkqhkiG9w0BAQUFAANBAHaU
hyCwLirUghNxCmLzXRG7C3W1j0kSX7a4fX9OxKR/Z2SoMjdMNPPyApuh8SoT2zBP
ZKjZU2WjcZG/nZF4W5k=

% Router Certificate successfully imported
```

Certificate Autoenrollment Configuration Example

The following example shows how to configure the switch to autoenroll with a CA on start-up:

```
Router(config)# crypto pki trustpoint frog
Router(ca-trustpoint)# enrollment url http://frog.phoobin.com/
Router(ca-trustpoint)# subject-name OU=Spiral Dept., O=tiedye.com
Router(ca-trustpoint)# ip-address ethernet-0
Router(ca-trustpoint)# auto-enroll regenerate
Router(ca-trustpoint)# password revokeme
Router(ca-trustpoint)# rsa-key frog 2048
!
Router(config)# crypto pki certificate chain frog
Router(config-cert-chain)# certificate ca 0B
30820293 3082023D A0030201 0202010B 300D0609 2A864886 F70D0101 04050030
79310B30 09060355 04061302 5553310B 30090603 55040813 02434131 15301306
0355040A 130C4369 73636F20 53797374 656D3120 301E0603 55040B13 17737562
6F726420 746F206B 6168756C 75692049 50495355 31243022 06035504 03131B79
6E692D75 31302043 65727469 66696361 7465204D 616E6167 6572301E 170D3030
30373134 32303536 32355A17 0D303130 37313430 31323834 335A3032 310E300C
06035504 0A130543 6973636F 3120301E 06092A86 4886F70D 01090216 11706B69
2D343562 2E636973 636F2E63 6F6D305C 300D0609 2A864886 F70D0101 01050003
4B003048 024100B3 0512A201 3B4243E1 378A9703 8AC5E3CE F77AF987 B5A422C4
15E947F6 70997393 70CF34D6 63A86B9C 4347A81A 0551FC02 ABA62360 01EF7DD2
6C136AEB 3C6C3902 03010001 A381F630 81F3300B 0603551D 0F040403 02052030
1C060355 1D110415 30138211 706B692D 3435622E 63697363 6F2E636F 6D301D06
03551D0E 04160414 247D9558 169B9A21 23D289CC 2DDA2A9A 4F77C616 301F0603
551D2304 18301680 14BD742C E892E819 1D551D91 683F6DB2 D8847A6C 73308185
0603551D 1F047E30 7C307AA0 3CA03AA4 38303631 0E300C06 0355040A 13054369
73636F31 24302206 03550403 131B796E 692D7531 30204365 72746966 69636174
65204D61 6E616765 72A23AA4 38303631 0E300C06 0355040A 13054369 73636F31
24302206 03550403 131B796E 692D7531 30204365 72746966 69636174 65204D61
6E616765 72300D06 092A8648 86F70D01 01040500 03410015 BC7CECF9 696697DF
E887007F 7A8DA24F 1ED5A785 C5C60452 47860061 0C18093D 08958A77 5737246B
0A25550A 25910E27 8B8B428E 32F8D948 3DD1784F 954C70
quit
```

Key Rollover for Certificate Renewal Configuration Examples

This section contains the following examples:

- Certificate Autoenrollment with Key Rollover Configuration Example, page 26-61
- Manual Certificate Enrollment with Key Rollover Configuration Example, page 26-61

Certificate Autoenrollment with Key Rollover Configuration Example

The following example shows how to configure the switch to autoenroll with the CA named trustme1 on startup. In this example, the **regenerate** keyword is specified, so a new key will be generated for the certificate. The renewal percentage is configured as 90 so if the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. The changes made to the running configuration are saved to the NVRAM startup configuration because autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.

```
Router(config)# crypto pki trustpoint trustme1
Router(ca-trustpoint)# enrollment url http://trustme1.company.com/
Router(ca-trustpoint)# subject-name OU=Spiral Dept., O=tiedye.com
Router(ca-trustpoint)# ip-address ethernet0
Router(ca-trustpoint)# serial-number none
Router(ca-trustpoint)# auto-enroll 90 regenerate
Router(ca-trustpoint)# password revokeme
Router(ca-trustpoint)# rsakeypair trustme1 2048
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate trustme1
```

Manual Certificate Enrollment with Key Rollover Configuration Example

The following example shows how to configure key rollover to regenerate new keys with a manual certificate enrollment from the CA named trustme2:

```
Router(config)# crypto pki trustpoint trustme2
Router(ca-trustpoint)# enrollment url http://trustme2.company.com/
Router(ca-trustpoint)# subject-name OU=Spiral Dept., O=tiedye.com
Router(ca-trustpoint)# ip-address ethernet0
Router(ca-trustpoint)# serial-number none
Router(ca-trustpoint)# regenerate
Router(ca-trustpoint)# password revokeme
Router(ca-trustpoint)# rsakeypair trustme2 2048
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate trustme2
Router(config)# crypto pki enroll trustme2
Router(config)# exit
```

PKI: Query Multiple Servers During Certificate Revocation Check (CDP Override) Configuration Example

The following example uses the **match certificate override cdp** command to override the CDPs for the certificate map named Group1 defined in a **crypto pki certificate map** command:

```
Router(config)# crypto pki certificate map Group1 10
Router(ca-certificate-map)# subject-name co ou=WAN
Router(ca-certificate-map)# subject-name co o=Cisco
Router(config)# crypto pki trustpoint pki
Router(ca-trustpoint)# match certificate Group1 override cdp url http://server.cisco.com
```

Online Certificate Status Protocol Configuration Examples

This section provides the following configuration examples:

- OCSP Server Configuration Example, page 26-62
- CRL Then OCSP Server Configuration Example, page 26-62

- Specific OCSP Server Configuration Example, page 26-62

OCSP Server Configuration Example

The following example shows how to configure the switch to use the OCSP server that is specified in the AIA extension of the certificate:

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check ocsp
```

CRL Then OCSP Server Configuration Example

The following example shows how to configure the switch to download the CRL from the certificate distribution point (CDP); if the CRL is unavailable, the OCSP server that is specified in the AIA extension of the certificate will be used. If both options fail, certificate verification will also fail.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check crl ocsp
```

Specific OCSP Server Configuration Example

The following example shows how to configure your switch to use the OCSP server at the HTTP URL http://myocspserver:81. If the server is down, revocation check will be ignored.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
```

Certificate Security Attribute-Based Access Control Configuration Example

The following example shows how to configure a certificate-based ACL:

```
Router(config)# crypto pki certificate map Group 10
Router(ca-certificate-map)# subject-name co Cisco
Router(config-cert-map)# exit
Router(config)# crypto pki trustpoint Access
Router(ca-trustpoint)# match certificate Group
Router(ca-trustpoint)# exit
```

PKI AAA Authorization Using the Entire Subject Name Configuration Example

The following example shows that the entire subject name of the certificate is to be used for PKI AAA authorization:

```
Router(config)# aaa new-model
Router(config)# aaa authorization network tac-o group tacacs+

Router(config)# crypto pki trustpoint test
Router(ca-trustpoint)# enrollment url http://caserver:80
Router(ca-trustpoint)# revocation-check crl
Router(ca-trustpoint)# exit
Router(ca-trustpoint)# authorization list tac-o
Router(ca-trustpoint)# authorization username subjectname all

Router(config)# tacacs-server host 20.2.2.2 key a_secret_key
```

Source Interface Selection for Outgoing Traffic with Certificate Authority Configuration Example

In the following example, the switch is located in a branch office. The switch uses IP Security (IPsec) to communicate with the main office. Ethernet 1 is the outside interface that connects to the Internet Service Provider (ISP). Ethernet 0 is the interface connected to the LAN of the branch office. To access the CA server located in the main office, the switch must send its IP datagrams out interface Ethernet 1 (address 10.2.2.205) using the IPsec tunnel. Address 10.2.2.205 is assigned by the ISP. Address 10.2.2.205 is not a part of the branch office or main office.

The CA cannot access any address outside the company because of a firewall. The CA sees a message coming from 10.2.2.205 and cannot respond (that is, the CA does not know that the switch is located in a branch office at address 10.1.1.1, which it is able to reach).

Adding the **source interface** command tells the switch to use address 10.1.1.1 as the source address of the IP datagram that it sends to the CA. The CA is able to respond to 10.1.1.1.

This example is configured using the **source interface** command and the interface addresses as described above.

```
Router(config)# crypto pki trustpoint ms-ca
Router(ca-trustpoint)# enrollment url http://ms-ca:80/certsrv/mscep/mscep.dll
Router(ca-trustpoint)# source interface ethernet0

Router(config)# interface ethernet 0
Router(config-if)# description inside interface
Router(config-if)# ip address 10.1.1.1 255.255.255.0

Router(config)# interface ethernet 1
Router(config-if)# description outside interface
Router(config-if)# ip address 10.2.2.205 255.255.255.0
Router(config-if)# crypto map main-office
```

Persistent Self-Signed Certificates Configuration Examples

The following examples show how to configure a persistent self-signed certificate:

- Trustpoint and Self-Signed Certificate Configuration Example, page 26-63
- Enabling the HTTPS Server Configuration Example, page 26-64

Trustpoint and Self-Signed Certificate Configuration Example

The following example shows how to configure a trustpoint and a self-signed certificate. In this example, a trustpoint named local is declared, its enrollment is requested, and a self-signed certificate with an IP address is generated.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# crypto pki trustpoint local
Router(ca-trustpoint)# enrollment selfsigned
Router(ca-trustpoint)# end
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# crypto pki enroll local
Nov 29 20:51:13.067: %SSH-5-ENABLED: SSH 1.99 has been enabled
Nov 29 20:51:13.267: %CRYPTO-6-AUTOGEN: Generated new 512 bit key pair
```

```
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: yes
Enter Interface name or IP Address[]: ethernet 0
Generate Self Signed Router Certificate? [yes/no]: yes
Router Self Signed Certificate successfully created
```

Note A switch can have only one self-signed certificate. If you attempt to enroll a trustpoint configured for a self-signed certificate and one already exists, you receive a notification and are asked if you want to replace it. If so, a new self-signed certificate is generated to replace the existing one.

Enabling the HTTPS Server Configuration Example

In the following example, the HTTPS server is enabled and a default trustpoint is generated because one was not previously configured:

```
Router(config)# ip http secure-server

% Generating 1024 bit RSA keys ...[OK]

*Dec 21 19:14:15.421:%PKI-4-NOAUTOSAVE:Configuration was modified. Issue "write memory"
to save new certificate

Router(config)#
```

Note You must save the configuration to NVRAM if you want to keep the self-signed certificate and have the HTTPS server enabled following switch reloads.

The following message also appears:

```
*Dec 21 19:14:10.441:%SSH-5-ENABLED:SSH 1.99 has been enabled

Router(config)#
```

Note Creation of the key pair used with the self-signed certificate causes the Secure Shell (SSH) server to start. This behavior cannot be suppressed. You may want to modify your access control lists (ACLs) to permit or deny SSH access to the router.