Configuring Certificate Enrollment for a PKI

This module describes the different methods available for certificate enrollment and how to set up each method for a participating PKI peer. Certificate enrollment, which is the process of obtaining a certificate from a certification authority (CA), occurs between the end host that requests the certificate and the CA. Each peer that participates in the public key infrastructure (PKI) must enroll with a CA.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Prerequisites for PKI Certificate Enrollment

Before configuring peers for certificate enrollment, you should have the following items:

- A generated Rivest, Shamir, and Adelman (RSA) key pair to enroll and a PKI in which to enroll.
- An authenticated CA.
- Familiarity with the module "Cisco IOS PKI Overview: Understanding and Planning a PKI."
- Enable NTP on the device so that the PKI services such as auto enrollment and certificate rollover may function correctly.

Note As of Cisco IOS Release 12.3(7)T, all commands that begin with "**crypto ca**" have been changed to begin with "**crypto pki**." Although the router will still accept **crypto ca** commands, all output will be be displayed **crypto pki**.

Note For releases prior to 17.9.x, the trust point configuration accepts a blank or unspecified password. Starting with the 17.9.1 release and later, specifying a password is mandatory. Ensure that you specify a password for the trust point configuration, using strong type-6 encryption. If you do not specify a password, the system displays the following error message.

```
LDevID Trustpoint - Password

% Incomplete command.

?WARNING: Command has been added to the configuration using a type 0 password.
However, recommended to migrate to strong type-6 encryption
```

Note From Cisco IOS XE 17.9.x, subject-alt-name (SAN) is supported as part of Certificate signing request (CSR).

Information About Certificate Enrollment for a PKI

What Are CAs

A CA is an entity that issues digital certificates that other parties can use. It is an example of a trusted third party. CAs are characteristic of many PKI schemes.

A CA manages certificate requests and issues certificates to participating network devices. These services provide centralized key management for the participating devices to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

You can use the Cisco IOS certificate server or a CA provided by a third-party CA vendor.

Framework for Multiple CAs

A PKI can be set up in a hierarchical framework to support multiple CAs. At the top of the hierarchy is a root CA, which holds a self-signed certificate. The trust within the entire hierarchy is derived from the RSA key pair of the root CA. The subordinate CAs within the hierarchy can be enrolled with either the root CA or with another subordinate CA. Multiple tiers of CAs are configured by either the root CA or with another subordinate CA. Within a hierarchical PKI, all enrolled peers can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA.

When to Use Multiple CAs

Multiple CAs provide users with added flexibility and reliability. For example, subordinate CAs can be placed in branch offices while the root CA is at the office headquarters. Also, different granting policies can be implemented per CA, so you can set up one CA to automatically grant certificate requests while another CA within the hierarchy requires each certificate request to be manually granted.

Scenarios in which at least a two-tier CA is recommended are as follows:

- Large and very active networks in which a large number of certificates are revoked and reissued. A multiple tier CA helps to control the size of the certificate revocation lists (CRLs).
- When online enrollment protocols are used, the root CA can be kept offline except to issue subordinate CA certificates. This scenario provides added security for the root CA.

Authentication of the CA

The certificate of the CA must be authenticated before the device will be issued its own certificate and before certificate enrollment can occur. Authentication of the CA typically occurs only when you initially configure PKI support at your router. To authenticate the CA, issue the **crypto pki authenticate** command, which authenticates the CA to your router by obtaining the self-signed certificate of the CA that contains the public key of the CA.

Authentication via the fingerprint Command

Cisco IOS Release 12.3(12) and later releases allow you to issue the **fingerprint** command t o preenter a fingerprint that can be matched against the fingerprint of a CA certificate during authentication.

If a fingerprint is not preentered for a trustpoint, and if the authentication request is interactive, you must verify the fingerprint that is displayed during authentication of the CA certificate. If the authentication request is noninteractive, the certificate will be rejected without a preentered fingerprint.

Notes

- PKI does not support certificate with lifetime validity greater than the year 2099. So, It is recommended to choose a life time validity fewer than the value 2099.
- You do not need to explicitly configure the **crypto pki authenticate** command, if the **auto-trigger command** is configured. The **auto-trigger** command automatically triggers the authentication.
- If the authentication request is made using the command-line interface (CLI), the request is an interactive request. If the authentication request is made using HTTP or another management tool, the request is a noninteractive request.
- It is recommended you store the CA certificate in the filename format `<trustpoint-name>.ca`. That is, the file should have the same name as the trustpoint that you configure. Ensure that the file extension of the file is `.ca`.

Supported Certificate Enrollment Methods

Cisco IOS software supports the following methods to obtain a certificate from a CA:

- Simple Certificate Enrollment Protocol (SCEP)--A Cisco-developed enrollment protocol that uses HTTP to communicate with the CA or registration authority (RA). SCEP is the most commonly used method for sending and receiving requests and certificates.

Note To take advantage of automated certificate and key rollover functionality, you must be running a CA that supports rollover and SCEP must be used as your client enrollment method. If you are running a Cisco IOS CA, you must be running Cisco IOS Release 12.4(2)T or a later release for rollover support.

- PKCS12--The router imports certificates in PKCS12 format from an external server.
- IOS File System (IFS)--The router uses any file system that is supported by Cisco IOS software (such as TFTP, FTP, flash, and NVRAM) to send a certificate request and to receive the issued certificate. Users may enable IFS certificate enrollment when their CA does not support SCEP.

Note Prior to Cisco IOS Release 12.3(4)T, only the TFTP file system was supported within IFS.

- Manual cut-and-paste--The router displays the certificate request on the console terminal, allowing the user to enter the issued certificate on the console terminal. A user may manually cut-and-paste certificate requests and certificates when there is no network connection between the router and CA.
- Enrollment profiles-- Enrollment profiles are primarily used for EST or terminal based enrollment. In case that the CA server does not support SCEP, the recommended methods for enrollment are EST based enrollment or terminal based enrollment.
- Self-signed certificate enrollment for a trustpoint--The secure HTTP (HTTPS) server generates a self-signed certificate that is to be used during the secure socket layer (SSL) handshake, establishing a secure connection between the HTTPS server and the client. The self-signed certificate is then saved in the router's startup configuration (NVRAM). The saved, self-signed certificate can then be used for future SSL handshakes, eliminating the user intervention that was necessary to accept the certificate every time the router reloaded.

Note To take advantage of autoenrollment and autoreenrollment, do not use either TFTP or manual cut-and-paste enrollment as your enrollment method. Both TFTP and manual cut-and-paste enrollment methods are manual enrollment processes, requiring user input.

Cisco IOS Suite-B Support for Certificate Enrollment for a PKI

Suite-B requirements comprise of four user interface suites of cryptographic algorithms for use with IKE and IPSec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm.

Suite-B adds the following support for the certificate enrollment for a PKI:

- Elliptic Curve Digital Signature Algorithm (ECDSA) (256-bit and 384-bit curves) is used for the signature operation within X.509 certificates.
- PKI support for validation of for X.509 certificates using ECDSA signatures.
- PKI support for generating certificate requests using ECDSA signatures and for importing the issued certificates into IOS.

See the Configuring Security for VPNs with IPsec feature module for more detailed information about Cisco IOS Suite-B support.

Registration Authorities

A Cisco IOS certificate server can be configured to run in RA mode. An RA offloads authentication and authorization responsibilities from a CA. When the RA receives a SCEP or manual enrollment request, the administrator can either reject or grant it on the basis of local policy. If the request is granted, it will be forwarded to the issuing CA, and the CA can be configured to automatically generate the certificate and return it to the RA. The client can later retrieve the granted certificate from the RA.

Automatic Certificate Enrollment

Automatic certificate enrollment allows the CA client to automatically request a certificate from its CA sever. This automatic router request eliminates the need for operator intervention when the enrollment request is sent to the CA server. Automatic enrollment is performed on startup for any trustpoint CA that is configured and that does not have a valid client certificate. When the certificate expires, a new certificate is automatically requested.

Note When automatic enrollment is configured, clients automatically request client certificates. The CA server performs its own authorization checks; if these checks include a policy to automatically issue certificates, all clients will automatically receive certificates, which is not very secure. Thus, automatic certificate enrollment should be combined with additional authentication and authorization mechanisms (such as Secure Device Provisioning (SDP), leveraging existing certificates, and one-time passwords).

Automated Client Certificate and Key Rollover

By default, the automatic certificate enrollment function requests a new client certificate and keys from the CS before the client's current certificate expires. Certificate and key rollover allows the certificate renewal rollover request to be made before the certificate expires by retaining the current key and certificate until the new, or rollover, certificate is available. After a specified amount of time, the rollover certificate and keys will become the active certificate and keys. The expired certificate and keys are immediately deleted upon rollover and removed from the certificate chain and CRL.

The setup for automatic rollover is twofold: CA clients must be automatically enrolled and the client's CAs must be automatically enrolled and have the **auto-rollover** command enabled. For more information on configuring your CA servers for automatic certificate rollover see the section "Automatic CA Certificate and Key Rollover" in the chapter "Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment " of the *Public Key Infrastructure Configuration Guide*.

An optional renewal percentage parameter can be used with the **auto-enroll** command to allow a new certificate to be requested when a specified percentage of the lifetime of the certificate has passed. For example, if the renewal percentage is configured as 90 and the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. In order for automatic rollover to occur, the renewal percentage must be less than 100.The specified percent value must not be less than 10. If a client certificate is issued for less than the configured validity period due to the impending expiration of the CA certificate, the rollover certificate will be issued for the balance of that period. A minimum of 10 percent of the configured validity period, with an absolute minimum of 3 minutes, is required to allow rollover enough time to function.

Tip If CA autoenrollment is not enabled, you may manually initiate rollover on an existing client with the **crypto pki enroll** command if the expiration time of the current client certificate is equal to or greater than the expiration time of the corresponding CA certificate. The client will initiate the rollover process, which occurs only if the server is configured for automated rollover and has an available rollover server certificate.

Note A key pair is also sent if configured by the **auto-enroll re-generate** command and keyword. It is recommended that a new key pair be issued for security reasons.

Certificate Enrollment Profiles

Certificate enrollment profiles allow users to specify certificate authentication, enrollment, and reenrollment parameters when prompted. The values for these parameters are referenced by two templates that make up the profile. One template contains parameters for the HTTP request that is sent to the CA server to obtain the certificate of the CA (also known as certificate authentication); the other template contains parameters for the HTTP request that is sent to the CA for certificate enrollment.

Configuring two templates enables users to specify different URLs or methods for certificate authentication and enrollment; for example, authentication (getting the certificate of the CA) can be performed via TFTP (using the **authentication url** command) and enrollment can be performed manually (using the **enrollment terminal** command).

Prior to Cisco IOS Release 12.3(11)T, certificate requests could be sent only in a PKCS10 format; however, an additional parameter was added to the profile, allowing users to specify the PKCS7 format for certificate renewal requests.

Note A single enrollment profile can have up to three separate sections for each task--certificate authentication, enrollment, and reenrollment.

Trust Points in IKEv2

Trust points, also known as trust anchors, are cryptographic entities that represent a trusted root or intermediate Certificate Authority (CA) whose certificates are implicitly trusted by the device. These trust points are used to validate the authenticity of peer certificates during the IKEv2 handshake.

Installed Trust Points:

These are certificates (Root CA, Intermediate CA, or end-entity) that have been loaded onto the device's certificate store. The device generally trusts these certificates for various PKI operations.

Trust Points Configured Under IKEv2 Profile:

For IKEv2, specific trust points must be explicitly referenced within the IKEv2 profile configuration. This tells the IKEv2 engine which CAs it should use as trusted anchors when validating the certificate presented by the peer.

Consider two devices, an Initiating Device and a Responding Device, attempting to establish an IKEv2 tunnel using certificates.

During the IKEv2 negotiation, the Responding Device typically sends a Certificate Request payload, indicating which Certificate Authorities (CAs) it trusts. These trusted CAs are derived from the trust points installed on the device.

The Initiating Device then selects an appropriate identity certificate from its own installed certificates that is issued by one of the CAs trusted by the Responding Device. It constructs and sends the necessary certificate chain (its identity certificate and its issuing CA certificates) to the Responding Device.

It's important to note that the certificate chain sent by the Initiating Device can be optimized. If the Initiating Device determines that the Responding Device already possesses certain intermediate CA certificates (e.g., if they are installed on the Responding Device), it might omit those intermediate certificates from the chain it sends, assuming the Responding Device can build the chain locally.

Below is an example scenario.

Trust Points Installed on Device A

crypto pki trustpoint <root-tp-1> (Contains root certificate 1) (Root CA 1)

crypto pki trustpoint <root-tp-2> (Contains root certificate 2) (Root CA 2)

crypto pki trustpoint <root-tp-3> (Contains root certificate 3) (Root CA 3)

crypto pki trustpoint <intermediate-tp-2> (Contains intermediate certificate issued by Root CA 2) (Intermediate CA 2)

crypto pki trustpoint <intermediate-tp-3> (Contains intermediate certificate issued by Root CA 3) (Intermediate CA 3)

crypto pki trustpoint <secondary-intermediate-tp-3> (Contains secondary intermediate certificate issued by Intermediate CA 3) (Secondary Intermediate CA 3) + ID certificate issued by Secondary Intermediate CA 3 (ID 3)

Trust Points Configured Under IKEv2 Profile on Device A

pki trustpoint <secondary-intermediate-tp-3> (Secondary Intermediate CA 3) + (ID 3)

pki trustpoint <root-tp-1> (Root CA 1)

pki trustpoint <root-tp-2> (Root CA 2)

pki trustpoint <root-tp-3> (Root CA 3)

Trust Points Installed on Device B

crypto pki trustpoint <root-tp-1> (Contains root certificate 1) (Root CA 1)

crypto pki trustpoint <root-tp-2> (Contains root certificate 2) (Root CA 2)

crypto pki trustpoint <root-tp-3> (Contains root certificate 3) (Root CA 3)

crypto pki trustpoint <intermediate-tp-1> (Contains intermediate certificate issued by Root CA 1) (Intermediate CA 1) + ID certificate issued by Intermediate CA 1 (ID 1)

Trust Points Configured Under IKEv2 Profile on Device B

pki trustpoint <intermediate-tp-1> (Intermediate CA 1) + (ID 1)

pki trustpoint <root-tp-1> (Root CA 1)

pki trustpoint <root-tp-2> (Root CA 2)

pki trustpoint <root-tp-3> (Root CA 3)

With the above config Device A is able to connect to Device B because the common trust anchors are picked from trust points installed on the box instead of trust points configured in IKEv2 profile. But note that verification of certificate happens against the trust points configured under IKEv2 profile and certificates sent by the peer.

During negotiation, Device B asks Device A to send an ID certificate trusted by either Root CA 1 or Root CA 2 or Root CA 3.

Device A finds out it has an ID cert issued by Secondary Intermediate CA 3 and Secondary Intermediate CA 3 in turn issued by Intermediate CA 3 and in turn Intermediate CA 3 issued by Root CA 3.

So, Device A sends 3 certificates to Device B.

1. ID 3
2. Secondary Intermediate CA 3
3. Intermediate CA 3

   Now, Device B sends these 3 certificate to PKI for verification. PKI verifies ID 3 using Secondary Intermediate CA 3 and Secondary Intermediate CA 3 using Intermediate CA 3 as they already came to PKI as a chain.

   Now PKI has to verify Intermediate CA 3, so it looks in to all the trust points configured under IKEv2 profile and it figures out suitable trust point based on issuer name. PKI finds out <root-tp-3> as suitable trust point as Root CA 3 is issuer of Intermediate CA 3 and verification completes.

   Scenario:

   As soon as a new trustpoint <intermediate-tp-3> which contains Intermediate CA 3 is installed on Device B, the Device A no more able to connect to Device B.

   Because, this time Device A sends 2 certificates to Device B instead of 3 certificates because Intermediate CA 3 is now common between Device A and Device B.

   1.ID 3

   2.Secondary Intermediate CA 3

   Now, Device 2 sends these 2 certificates to PKI for verification. PKI verifies

   ID 3 using Secondary Intermediate CA 3 as they already came to PKI as chain

   Now PKI has to verify Secondary Intermediate CA 3, so it looks into all the trust points configured under IKEv2 profile with issuer name. PKI couldn't find a suitable trust point under IKEv2 profile and verification failed.

   Solution :

   As soon as the trust point <intermediate-tp-3> which contains Intermediate CA 3 is configured under IKEv2 profile, PKI finds the suitable trust point with issuer match and verification is success and Device A is able to connect to Device B.

How to Configure Certificate Enrollment for a PKI

This section contains the following enrollment option procedures. If you configure enrollment or autoenrollment (the first task), you cannot configure manual certificate enrollment. Also, if you configure TFTP or manual

cut-and-paste certificate enrollment, you cannot configure autoenrollment, autoreenrollment, an enrollment profile, nor can you utilize the automated CA certificate rollover capability.

Configuring Certificate Enrollment or Autoenrollment

Perform this task to configure certificate enrollment or autoenrollment for clients participating in your PKI.

Before you begin

Before configuring automatic certificate enrollment requests, you should ensure that all necessary enrollment information is configured.

Prerequisites for Enabling Automated Client Certificate and Key Rollover

CA client support for certificate rollover is automatically enabled when using autoenrollment. For automatic CA certificate rollover to run successfully, the following prerequisites are applicable:

- Your network devices must support shadow PKI.
- Your clients must be running Cisco IOS Release 12.4(2)T or a later release.
- The client's CS must support automatic rollover. See the section "Automatic CA Certificate and Key Rollover" in the chapter "Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment " of the *Public Key Infrastructure Configuration Guide* for more information on CA server automatic rollover configuration.

Prerequisites for Specifying Autoenrollment Initial Key Generation Location

To specify the location of the autoenrollment initial key generation, you must be running Cisco IOS Release 12.4(11)T or a later release.

RSA Key Pair Restriction for Autoenrollment

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 mismatches.

Certificate renewal with regenerate option does not work with key label starting from zero ('0'), for example, '0test'. CLI allows configuring such name under trustpoint, and allows hostname starting from zero, but certificate regenerate will fail.

Restrictions for Automated Client Certificate and Key Rollover

In order for clients to run automatic CA certificate rollover successfully, the following restrictions are applicable:

- SCEP must be used to support rollover. Any device that enrolls with the PKI using an alternative to SCEP as the certificate management protocol or mechanism (such as enrollment profiles, manual enrollment, or TFTP enrollment) will not be able to take advantage of the rollover functionality provided by SCEP.
- If the configuration cannot be saved to the startup configuration after a shadow certificate is generated, rollover will not occur.
- Rollover with key regenerate does not work when keypair name starts from zero ('0') (for example, '0test'). When configuring **rsakeypair** *name* under a trustpoint, do not configure name starting from zero. When

keypair name is not configured and the default keypair is used, make sure the router hostname does not start from zero. If it does so, configure "**rsakeypair** *name* explicitly under the trustpoint with a different name.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Note For releases prior to 17.9.x, the trust point configuration accepts a blank or unspecified password. Starting with the 17.9.1 release and later, specifying a password is mandatory. Ensure that you specify a password for the trust point configuration, using strong type-6 encryption. If you do not specify a password, the system displays the following error message.

```
LDevID Trustpoint - Password

% Incomplete command.

?WARNING: Command has been added to the configuration using a type 0 password.
However, recommended to migrate to strong type-6 encryption
```

Note From Cisco IOS XE 17.9.x, subject-alt-name (SAN) is supported as part of Certificate signing request (CSR).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment** [**mode** | **retry period** *minutes* | **retry count** *number*] **url** *url* [**pem**]
5. **disable-scep**
6. **eckeypair** *label*
7. **subject-name** [*x.500-name*]
8. **vrf** *vrf-name*
9. **ip-address** {*ip-address* | *interface* | **none**}
10. serial-number [none]
11. **auto-enroll** [*percent*] [**regenerate**]
12. **auto-trigger**
13. **usage** *method1* [*method2* [*method3*]]
14. **password** *string*
15. **rsakeypair** *key-label key-size encryption-key-size*]]
16. **fingerprint** *ca-fingerprint*
17. **on** *devicename* **:**
18. **exit**

19. **crypto pki authenticate** *name*
20. **exit**
21. **copy system:running-config nvram:startup-config**
22. **show crypto pki certificates**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment** [**mode** | **retry period** *minutes* | **retry count** *number*] **url** *url* [**pem**] **Example:** `Router(ca-trustpoint)# enrollment url http://cat.example.com`	Specifies the URL of the CA on which your router should send certificate requests. • **mode** --Specifies RA mode if your CA system provides an RA. • **retry period** *minutes* --Specifies the wait period between certificate request retries. The default is 1 minute between retries. • **retry count** *number* -- Specifies the number of times a router will resend a certificate request when it does not receive a response from the previous request. (Specify from 1 to 100 retries.) • **url** *url* -- URL of the file system where your router should send certificate requests. An IPv6 address can be added in the URL enclosed in brackets. For example: http:// [2001:DB8:1:1::1]:80. • **pem** -- Adds privacy-enhanced mail (PEM) boundaries to the certificate request. **Note**

	Command or Action	Purpose
		An enrollment method other than TFTP or manual cut-and-paste must be configured to support autoenrollment.
Step 5	**disable-scep** **Example:** `Router(ca-trustpoint)# disable-scep`	(Optional) Utilizes and allows for the alternative method of enrolling HTTP file system instead of the default SCEP (Simple Certificate Enrollment Protocol) method.
Step 6	**eckeypair** *label* **Example:** `Router(ca-trustpoint)# eckeypair Router_1_Key`	(Optional) Configures the trustpoint to use an Elliptic Curve (EC) key on which certificate requests are generated using ECDSA signatures. The *label* argument specifies the EC key label that is configured using the **crypto key generate rsa** or **crypto key generate ec keysize** command in global configuration mode. See the Configuring Internet Key Exchange for IPsec VPNs feature module for more information. **Note** If an ECDSA signed certificate is imported without a trustpoint configuration, then the label defaults to the FQDN value.
Step 7	**subject-name** [*x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name cat`	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name* --If it is not specified, the fully qualified domain name (FQDN), which is the default subject name, will be used.
Step 8	**vrf** *vrf-name* **Example:** `Router(ca-trustpoint)# vrf myvrf`	(Optional) Specifies the the VRF instance in the public key infrastructure (PKI) trustpoint to be used for enrollment, certificate revocation list (CRL) retrieval, and online certificate status protocol (OCSP) status.
Step 9	**ip-address** {*ip-address* | *interface* | **none**} **Example:** `Router(ca-trustpoint)# ip address 192.168.1.66`	(Optional) Includes the IP address of the specified interface in the certificate request. • Issue the *ip-address* argument to specify either an IPv4 or IPv6 address. • Issue the *interface* argument to specify an interface on the router. • Issue the **none** keyword if no IP address should be included. **Note** If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.

	Command or Action	Purpose
Step 10	serial-number [none] **Example:** `Router(ca-trustpoint)# serial-number`	(Optional) Specifies the router serial number in the certificate request, unless the **none** keyword is issued. • Issue the **none** keyword to specify that a serial number will not be included in the certificate request.
Step 11	**auto-enroll** [*percent*] [**regenerate**] **Example:** `Router(ca-trustpoint)# auto-enroll regenerate`	(Optional) Enables autoenrollment, allowing the client to automatically request a rollover certificate from the CA. • If autoenrollment is not enabled, the client must be manually re-enrolled in your PKI upon certificate expiration. • By default, only t he Domain Name System (DNS) name of the router is included in the certificate. • Use the *percent* argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached. • Use the **regenerate** 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." **Note** It is recommended that a new key pair be generated for security reasons.
Step 12	**auto-trigger** **Example:** `Router(ca-trustpoint)# auto-trigger`	(Optional) Triggers automatic certificate authentication. **Note** You do not need to explicitly configure the **crypto pki authenticate** command, if the **auto-trigger** command is configured. The **auto-trigger** command automatically triggers the authentication.
Step 13	**usage** *method1* [*method2* [*method3*]] **Example:** `Router(ca-trustpoint)# usage ssl-client`	(Optional) Specifies the intended use for the certificate. • Available options are **ike**, **ssl-client**, and **ssl-server**; the default is **ike**.
Step 14	**password** *string* **Example:** `Router(ca-trustpoint)# password string1`	(Optional) Specifies the revocation password for the certificate. • If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint.

	Command or Action	Purpose
		Note When SCEP is used, this password can be used to authorize the certificate request--often via a one-time password or similar mechanism.
Step 15	**rsakeypair** *key-label key-size encryption-key-size*]] **Example:** `Router(ca-trustpoint)# rsakeypair key-label 2048 2048`	(Optional) Specifies which key pair to associate with the certificate. • A key pair with the *key-label* argument will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued. • Specify the *key-size* argument for generating the key, and specify the *encryption-key-size* argument to request separate encryption, signature keys, and certificates. The key-size and encryption-key-size must be the same size. Length of less than 2048 is not recommended. **Note** If this command is not enabled, the FQDN key pair is used.
Step 16	**fingerprint** *ca-fingerprint* **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication. **Note** If the fingerprint is not provided and authentication of the CA certificate is interactive, the fingerprint will be displayed for verification.
Step 17	**on** *devicename* **:** **Example:** `Router(ca-trustpoint)# on usbtoken0:`	(Optional) Specifies that RSA keys will be created on the specified device upon autoenrollment initial key generation. • Devices that may be specified include NVRAM, local disks, and Universal Serial Bus (USB) tokens. USB tokens may be used as cryptographic devices in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication to be performed on the token.
Step 18	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 19	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate mytp`	Retrieves the CA certificate and authenticates it. Check the certificate fingerprint if prompted. **Note** This command is optional if the CA certificate is already loaded into the configuration.
Step 20	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Step 21	**copy system:running-config nvram:startup-config**
	Example:
	`Router#`
	`copy system:running-config nvram:startup-config`
		(Optional) Copies the running configuration to the NVRAM startup configuration.
		Note Autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.

Step 22	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, including any rollover certificates.

Configuring Manual Certificate Enrollment

Manual certificate enrollment can be set up via TFTP or the manual cut-and-paste method. Both options can be used if your CA does not support SCEP or if a network connection between the router and CA is not possible. Perform one of the following tasks to set up manual certificate enrollment:

PEM-Formatted Files for Certificate Enrollment Request

Using PEM-formatted files for certificate requests can be helpful for customers who are using terminal or profile-based enrollment to request certificates from their CA server. Customers using PEM-formatted files can directly use existing certificates on their routers.

Restrictions for Manual Certificate Enrollment

SCEP Restriction

We do not recommend switching URLs if SCEP is used; that is, if the enrollment URL is “http://myca,” do not change the enrollment URL after getting the CA certificate and before enrolling the certificate. A user can switch between TFTP and manual cut-and-paste.

Key Regeneration Restriction

Do not regenerate the keys manually using the **crypto key generate** command; key regeneration will occur when the **crypto pki enroll**command is issued if the **regenerate** keyword is specified.

Configuring Cut-and-Paste Certificate Enrollment

Perform this task to configure cut-and-paste certificate enrollment. This task helps you to configure manual certificate enrollment via the cut-and-paste method for peers participating in your PKI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment terminal pem**
5. **fingerprint** *ca-fingerprint*
6. **exit**
7. **crypto pki authenticate** *name*
8. crypto pki enroll name
9. crypto pki import name certificate
10. **exit**
11. **show crypto pki certificates**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment terminal pem** **Example:**	Specifies the manual cut-and-paste certificate enrollment method.

	Command or Action	Purpose
	`Router(ca-trustpoint)# enrollment terminal`	• The certificate request will be displayed on the console terminal so that it may be manually copied (or cut). • **pem** --Configures the trustpoint to generate PEM-formatted certificate requests to the console terminal.
Step 5	**fingerprint** *ca-fingerprint* **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication. **Note** If the fingerprint is not provided, it will be displayed for verification.
Step 6	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 7	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate mytp`	Retrieves the CA certificate and authenticates it.
Step 8	crypto pki enroll name **Example:** `Router(config)# crypto pki enroll mytp`	Generates certificate request and displays the request for copying and pasting into the certificate server. • You are prompted for enrollment information, such as whether to include the router FQDN and IP address in the certificate request. You are also given the choice about displaying the certificate request to the console terminal. • The base-64 encoded certificate with or without PEM headers as requested is displayed.
Step 9	crypto pki import name certificate **Example:** `Router(config)# crypto pki import mytp certificate`	Imports a certificate manually at the console terminal (pasting). • The base-64 encoded certificate is accepted from the console terminal and inserted into the internal certificate database. **Note** You must enter this command twice if usage keys, a signature key, and an encryption key are used. The first time the command is entered, one of the certificates is pasted into the router. The second time the command is

	Command or Action	Purpose
		entered, the other certificate is pasted into the router. It does not matter which certificate is pasted first. **Note** Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If this applies to the certificate authority you are using, import the general purpose certificate. The router will not use one of the two key pairs generated.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

Configuring TFTP Certificate Enrollment

Perform this task to configure TFTP certificate enrollment. This task helps you to configure manual certificate enrollment using a TFTP server.

Before you begin

- You must know the correct URL to use if you are configuring certificate enrollment via TFTP.
- The router must be able to write a file to the TFTP server for the **crypto pki enroll** command.
- If you are using a file specification with the **enrollment** command, the file must contain the CA certificate either in binary format or be base-64 encoded.
- You must know if your CA ignores key usage information in a certificate request and issues only a general purpose usage certificate.

Caution Some TFTP servers require that the file must exist on the server before it can be written. Most TFTP servers require files that can be written over. This requirement may pose a risk because any router or other device may write or overwrite the certificate request; thus, the replacement certificate request will not be used by the CA administrator, who must first check the enrollment request fingerprint before granting the certificate request.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **crypto pki trustpoint** *name*
4. **enrollment** [**mode**] [**retry period** minutes] [**retry count** number] **url** url [**pem**]
5. **fingerprint** *ca-fingerprint*
6. **exit**
7. **crypto pki authenticate** *name*
8. crypto pki enroll name
9. crypto pki import name certificate
10. **exit**
11. **show crypto pki certificates**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment** [**mode**] [**retry period** minutes] [**retry count** number] **url** url [**pem**] **Example:** `Router(ca-trustpoint)# enrollment url tftp://certserver/file_specification`	Specifies TFTP as the enrollment method to send the enrollment request and to retrieve the CA certificate and router certificate and any optional parameters. **Note** For TFTP enrollment, the URL must be configured as a TFTP URL, tftp://example_tftp_url. • An optional file specification filename may be included in the TFTP URL. If the file specification is not included, the FQDN will be used. If the file specification is included, the router will append the extension ".ca" to the specified filename.

	Command or Action	Purpose
Step 5	**fingerprint** *ca-fingerprint* **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies the fingerprint of the CA certificate received via an out-of-band method from the CA administrator. **Note** If the fingerprint is not provided, it will be displayed for verification.
Step 6	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 7	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate mytp`	Retrieves the CA certificate and authenticates it from the specified TFTP server.
Step 8	crypto pki enroll name **Example:** `Router(config)# crypto pki enroll mytp`	Generates certificate request and writes the request out to the TFTP server. • You are prompted for enrollment information, such as whether to include the router FQDN and IP address in the certificate request. You are queried about whether to display the certificate request to the console terminal. • The filename to be written is appended with the extension ".req". For usage keys, a signature key and an encryption key, two requests are generated and sent. The usage key request filenames are appended with the extensions "-sign.req" and "-encr.req", respectively.
Step 9	crypto pki import name certificate **Example:** `Router(config)# crypto pki import mytp certificate`	Imports a certificate via TFTP at the console terminal, which retrieves the granted certificate. • The router will attempt to retrieve the granted certificate via TFTP using the same filename used to send the request, except the extension is changed from ".req" to ".crt". For usage key certificates, the extensions "-sign.crt" and "-encr.crt" are used. • The router will parse the received files, verify the certificates, and insert the certificates into the internal certificate database on the router. **Note** Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If your CA ignores the usage key information

	Command or Action	Purpose
		in the certificate request, only import the general purpose certificate. The router will not use one of the two key pairs generated.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

Certifying a URL Link for Secure Communication with a Trend Micro Server

Perform this task to certify a link used in URL filtering that allows secure communication with a Trend Micro Server.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the *Next Generation Encryption* (NGE) white paper.

SUMMARY STEPS

1. **enable**
2. **clock set** *hh* **:** *mm* **:** *ss date month year*
3. **configure terminal**
4. **clock timezone** *zone hours-offset* [*minutes-offset*]
5. **ip http server**
6. **hostname** *name*
7. **ip domain-name** *name*
8. **crypto key generate rsa general-keys modulus** *modulus-size*
9. **crypto pki trustpoint** *name*
10. **enrollment terminal**
11. **crypto ca authenticate** *name*
12. Copy the following block of text containing the base 64 encoded CA certificate and paste it at the prompt.
13. Enter **yes** to accept this certificate.
14. **serial-number**
15. **revocation-check none**
16. **end**
17. **trm register**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clock set** *hh* **:** *mm* **:** *ss date month year* **Example:** `Router# clock set 23:22:00 22 Dec 2009`	Sets the clock on the router.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**clock timezone** *zone hours-offset* [*minutes-offset*] **Example:** `Router(config)# clock timezone PST -08`	Sets the time zone. • The *zone* argument is the name of the time zone (typically a standard acronym). The *hours-offset* argument is the number of hours the time zone is different from Universal Time Coordinated (UTC). The *minutes-offset* argument is the number of minutes the time zone is different from UTC. **Note** The *minutes-offset*argument of the **clock timezone** command is available for those cases where a local time zone is a percentage of an hour different from UTC or Greenwich Mean Time (GMT). For example, the time zone for some sections of Atlantic Canada (AST) is UTC-3.5. In this case, the necessary command would be **clock timezone AST -3 30**.
Step 5	**ip http server** **Example:** `Router(config)# ip http server`	Enables the HTTP server.
Step 6	**hostname** *name* **Example:** `Router(config)# hostname hostname1`	Configures the hostname of the router.

	Command or Action	Purpose
Step 7	**ip domain-name** *name* **Example:** `Router(config)# ip domain-name example.com`	Defines the domain name for the router.
Step 8	**crypto key generate rsa general-keys modulus** *modulus-size* **Example:** `Router(config)# crypto key generate rsa general-keys modulus general`	Generates the crypto keys. • The **general-keys** keyword specifies that a general purpose key pair is generated, which is the default. • The **modulus** keyword and *modulus-size* argument specify the IP size of the key modulus. By default, the modulus of a CA key is 1024 bits. When generating RSA keys, you will be prompted to enter a modulus length. A longer modulus could offer stronger security but takes longer to generate and to use. A length of less than 2048 is not recommended. **Note** The name for the general keys that are generated are based on the domain name that is configured in Step 7. For example, the keys will be called "example.com."
Step 9	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the CA that your router should use and enters ca-trustpoint configuration mode. **Note** Effective with Cisco IOS Release 12.3(8)T, the **crypto pki trustpoint** command replaced the **crypto ca trustpoint** command.
Step 10	**enrollment terminal** **Example:** `Router(ca-trustpoint)# enrollment terminal`	Specifies the manual cut-and-paste certificate enrollment method. • The certificate request will be displayed on the console terminal so that you may manually copy (or cut).

Step 11	**crypto ca authenticate** *name*
	Example:
	`Router(ca-trustpoint)# crypto ca authenticate mytp`
		Takes the name of the CA as the argument and authenticates it.
		• The following command output displays:
		`Enter the base 64 encoded CA certificate.`
		`End with a blank line or the word "quit" on a line by itself.`

Step 12	Copy the following block of text containing the base 64 encoded CA certificate and paste it at the prompt.	`MIIDIDCCAomgAwIBAgIENd70zzANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJV` `UzEQMA4GA1UEChMHRXF1aWZheDEtMCsGA1UECxMkRXF1aWZheCBTZWN1cmUgQ2Vy`

	Command or Action	Purpose

		dGlmaWNhdGUgQXV0aG9yaXR5MB4XDTk4MDgyMjE2NDE1MVoXDTE4MDgyMjE2NDE1
		MVowTjELMAkGA1UEBhMCVVMxEDAOBgNVBAoTB0VxdWlmYXgxLTArBgNVBAsTJEVx
		dWlmYXggU2VjdXJlIEN1cnRpZmljYXRlIEF1dGhvcml0eTCBnzANBgkqhkiG9w0B
		AQEFAAOBjQAwgYkCgYEAwV2xWGcIYu6gmi0fCG2RFGiYCh7+2gRvE4RiIcPRfM6f
		BeC4AfBONOziipUEZKzxa1NfBbPLZ4C/QgKO/t0BCezhABRP/PvwDN1Dulsr4R+A
		cJkVV5MW8Q+XarfCaCMczE1ZMKxRHjuvK9buY0V7xdlfUNLjUA86iOe/FP3gx7kC
		AwEAAaOCAQkwggEFMHAGA1UdHwRpMGcwZaBjoGGkXzBdMQswCQYDVQQGEwJVUzEQ
		MA4GA1UEChMHRXF1aWZheDEtMCsGA1UECxMkRXF1aWZheCBITZWN1cmUgQ2VydGlm
		aWNhdGUgQXV0aG9yaXR5MQ0wCwYDVQQDEwRDUkwxMBoGA1UdEAQTMBGBDzIwMTgw
		ODIyMTY0MTUxWjALBgNVHQ8EBAMCAQYwHwYDVR0jBBgwFoAUSOZo+SvSspXXR9gj
		IBBPM5iQn9QwHQYDVR0OBBYEFEjmaPkr0rKV10fYIyAQTzOYkJ/UMAwGA1UdEwQF
		MAMBAf8wGgYJKoZIhvZ9B0EABA0wCxsFVjMuMGMDAgbAMA0GCSqGSIb3DQEBBQUA
		A4GBAFjOKer89961zgK5F7WF0bnj4JXMJTENAKaSbn+2kmOeUJXRmm/kEd5jhW6Y
		7qj/WsjTVbJmcVfewCHrPSqnI0kBBIZCe/zuf6IWUrVnZ9NA2zsmWLIodz2uFHdh
		1voqZiegDfqnc1zqcPGUIWVEX/r87yloqaKHee9570+sB3c4
		The following command output displays:
		`Certificate has the following attributes:`
		`Fingerprint MD5: 67CB9DC0 13248A82 9BB2171E D11BECD4`
		`Fingerprint SHA1: D23209AD 23D31423 2174E40D 7F9D6213 9786633A`

Step 13	Enter **yes** to accept this certificate.	`% Do you accept this certificate? [yes/no]: yes`
		The following command output displays:
		`Trustpoint CA certificate accepted.`
		`% Certificate successfully imported`

	Command or Action	Purpose
Step 14	**serial-number** **Example:** `hostname1(ca-trustpoint)# serial-number`	Specifies the router serial number in the certificate request.
Step 15	**revocation-check none** **Example:** `hostname1(ca-trustpoint)# revocation-check none` **Example:**	Specifies that certificate checking is ignored.
Step 16	**end** **Example:** `hostname1(ca-trustpoint)# end`	Exits ca-trustpoint configuration mode and returns to privileged EXEC mode.
Step 17	**trm register** **Example:** `hostname1# trm register`	Manually starts the Trend Micro Server registration process.

Configuring a Persistent Self-Signed Certificate for Enrollment via SSL

This section contains the following tasks:

Note These tasks are optional because if you enable the HTTPS server, it generates a self-signed certificate automatically using default values.

Persistent Self-Signed Certificates Overview

The SSL protocol can be used to establish a secure connection between an HTTPS server and a client (web browser). During the SSL handshake, 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 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 may present an opportunity for an attacker to substitute an unauthorized certificate when you are being asked to accept the certificate. Persistent self-signed certificates overcome all these limitations by saving a certificate in the router's startup configuration.

Restrictions

- You can configure only one trustpoint for a persistent self-signed certificate.
- The maximum lifetime of a self-signed certificate is 00:00:00 GMT Jan 1, 2030.

Note Do not change the IP domain name or the hostname of the router after creating the self-signed certificate. Changing either name triggers the regeneration of the self-signed certificate and overrides the configured trustpoint. WebVPN ties the SSL trustpoint name to the WebVPN gateway configuration. If a new self-signed certificate is triggered, then the new trustpoint name does not match the WebVPN configuration, causing the WebVPN connections to fail.

Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the *Next Generation Encryption* (NGE) white paper.

Perform the following task to configure a trustpoint and specify self-signed certificate parameters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** name
4. **enrollment selfsigned**
5. **subject-name** [*x.500-name*]
6. **rsakeypair** *key-label* [key-size [encryption-key-size]]
7. **crypto pki enroll** name
8. **end**
9. **show crypto pki certificates** [*trustpoint-name*[**verbose**]]
10. **show crypto pki trustpoints** [**status** | *label* [**status**]]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** name **Example:** `Router(config)# crypto pki trustpoint local`	Declares the CA that your router should use and enters ca-trustpoint configuration mode. **Note** Effective with Cisco IOS Release 12.3(8)T, the **crypto pki trustpoint** command replaced the **crypto ca trustpoint** command.
Step 4	**enrollment selfsigned** **Example:** `Router(ca-trustpoint)# enrollment selfsigned`	Specifies self-signed enrollment.
Step 5	**subject-name** [*x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name`	(Optional) Specifies the requested subject name to be used in the certificate request. • If no value for the *x-500-name* argument is specified, the FQDN, which is the default subject name, is used.
Step 6	**rsakeypair** *key-label* [key-size [encryption-key-size]] **Example:** `Router(ca-trustpoint)# rsakeypair examplekey 2048`	(Optional) Specifies which key pair to associate with the certificate. • The value for the *key-label* argument will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued. • Specify a value for the *key-size* argument for generating the key, and specify a value for the *encryption-key-size* argument to request separate encryption, signature keys, and certificates. The key-size and encryption-key-size must be the same size. Length of less than 2048 is no recommended. **Note** If this command is not enabled, the FQDN key pair is used.
Step 7	**crypto pki enroll** name **Example:**	Tells the router to generate the persistent self-signed certificate.

	Command or Action	Purpose
	`Router(config)# crypto pki enroll local`	
Step 8	**end** **Example:** `Router(ca-trustpoint)# end`	(Optional) Exits ca-trustpoint configuration mode. • Enter this command a second time to exit global configuration mode.
Step 9	**show crypto pki certificates** [*trustpoint-name*[**verbose**]] **Example:** `Router# show crypto pki certificates local verbose`	Displays information about your certificate, the certification authority certificate, and any registration authority certificates.
Step 10	**show crypto pki trustpoints** [**status** | *label* [**status**]] **Example:** `Router# show crypto pki trustpoints status`	Displays the trustpoints that are configured in the router.

Enabling the HTTPS Server

Perform the following task to enable the HTTPS server.

Before you begin

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 the server is enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http secure-server**
4. **end**
5. **copy system:running-config nvram: startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip http secure-server** **Example:** `Router(config)# ip http secure-server`	Enables the HTTPS web server. **Note** A key pair (modulus 1024) and a self-signed certificate are automatically generated.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode.
Step 5	**copy system:running-config nvram: startup-config** **Example:** `Router# copy system:running-config nvram: startup-config`	Saves the self-signed certificate and the HTTPS server in enabled mode.

Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment

Perform this task to configure a certificate enrollment profile for enrollment or reenrollment. This task helps you to configure an enrollment profile for certificate enrollment or reenrollment of a router with a Cisco IOS CA that is already enrolled with a third-party vendor CA.

Enable a router 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. To enable this functionality, you must issue the **enrollment credential** command. Also, you cannot configure manual certificate enrollment.

Before you begin

Perform the following tasks at the client router before configuring a certificate enrollment profile for the client router that is already enrolled with a third-party vendor CA so that the router can reenroll with a Cisco IOS certificate server:

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

Note

- To use certificate profiles, your network must have an HTTP interface to the CA.
- If an enrollment profile is specified, an enrollment URL may not be specified in the trustpoint configuration. Although both commands are supported, only one command can be used at a time in a trustpoint.
- 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.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. enrollment profile label
5. **exit**
6. **crypto pki profile enrollment** *label*
7. Do one of the following:
   - **authentication url** *url*
   - **authentication terminal**
8. **authentication command**
9. Do one of the following:
   - **enrollment url** *url*
   -
   - **enrollment terminal**
10. **enrollment credential** *label*
11. **enrollment command**
12. **parameter** *number* {**value** *value* | **prompt** *string*}
13. **exit**
14. **show crypto pki certificates**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint Entrust`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	enrollment profile label **Example:** `Router(ca-trustpoint)# enrollment profile E`	Specifies that an enrollment profile is to be used for certificate authentication and enrollment.
Step 5	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 6	**crypto pki profile enrollment** *label* **Example:** `Router(config)# crypto pki profile enrollment E`	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 7	Do one of the following:
	• **authentication url** *url*
	• **authentication terminal**
	Example:
	`Router(ca-profile-enroll)# authentication url http://entrust:81`
	Example:
	`Router(ca-profile-enroll)# authentication terminal`
		Specifies the URL of the CA server to which to send certificate authentication requests.
		• *url* --URL of the CA server to which your router should send authentication requests. If you are using HTTP, the URL should read "http://CA_name," where CA_name is the host DNS name or IP address of the CA. If you are using TFTP, the URL should read "tftp://certserver/file_specification." (If the URL does not include a file specification, the FQDN of the router will be used.)
		Specifies manual cut-and-paste certificate authentication.

Step 8	**authentication command** **Example:** `Router(ca-profile-enroll)# authentication command`	(Optional) Specifies the HTTP command that is sent to the CA for authentication.
Step 9	Do one of the following: • **enrollment url** *url* •	Specifies the URL of the CA server to which to send certificate enrollment requests via HTTP or TFTP. Specifies manual cut-and-paste certificate enrollment.

	Command or Action	Purpose

	• **enrollment terminal**
	Example:
	`Router(ca-profile-enroll)# enrollment url http://entrust:81/cda-cgi/clientcgi.exe`
	Example:
	Example:
	`Router(ca-profile-enroll)# enrollment terminal`	

Step 10	**enrollment credential** *label* **Example:** `Router(ca-profile-enroll)# enrollment credential Entrust`	(Optional) Specifies the third-party vendor CA trustpoint that is to be enrolled with the Cisco IOS CA. **Note** This command cannot be issued if manual certificate enrollment is being used.
Step 11	**enrollment command** **Example:** `Router(ca-profile-enroll)# enrollment command`	(Optional) Specifies the HTTP command that is sent to the CA for enrollment.
Step 12	**parameter** *number* {**value** *value* | **prompt** *string*} **Example:** `Router(ca-profile-enroll)# parameter 1 value aaaa-bbbb-cccc`	(Optional) Specifies parameters for an enrollment profile. • This command can be used multiple times to specify multiple values.
Step 13	**exit** **Example:** `Router(ca-profile-enroll)# exit`	(Optional) Exits ca-profile-enroll configuration mode. • Enter this command a second time to exit global configuration mode.
Step 14	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

What to Do Next

If you configured the router to reenroll with a Cisco IOS CA, you should configure the Cisco IOS certificate server to accept enrollment requests only from clients already enrolled with the specified third-party vendor CA trustpoint to take advantage of this functionality. For more information, see the module " Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment."

Configuring Certificate Enrollment in a Two-Tier PKI Environment

The feature enables sub-CAs to issue certificates to their clients when a root CA is offline. The root certificate can be imported through the CLI first, and then it is used to validate the issuing sub CA certificate configured under the trustpoint.

Note Enable revocation checking as per your environment before performing the following tasks.

For importing the ROOT-CA through terminal, perform the following steps:

```
enable
!
configure terminal
!
crypto pki trustpoint ROOT-CA
revocation-check none
enrollment terminal
!
crypto pki authenticate ROOT-CA
!
exit
```

For authenticating SUB-CA without specifying or accepting the fingerprint.

```
enable
!
configure terminal
!
crypto pki trustpoint SUB-CA
revocation-check none
enrollment url url
chain-validation continue ROOT-CA
exit
!
crypto pki authenticate SUB-CA
exit
```

Configuring Certificate Renewal by Enabling Multiple Trustpoints

Starting from the Cisco IOS XE 17.4.1 release, you can enable the registration authority to use multiple trustpoints to validate router credentials for initial certificate enrollment and certificate renewal. This enhancement enables automated validation of multiple trustpoints while maintaining zero-touch certificate enrollment through the SCEP enrollment protocol.

When you enroll a router for the first time, an SCEP request is initiated and this request is signed by using the SUDI credentials. The request is then sent to a registration authority which validates the SUDI certificate through a local trustpoint. The local trustpoint validates the router SCEP credentials. If the validation is successful, the registration authority uses the SUDI certificate to decrypt the signature and validate the hash. After the hash validation is also successful, the registration authority forwards the SCEP request to the certificate authority (CA). The CA then signs the request and sends the certificate back to the registration authority which in turn forwards the certificate to the router. At this point, the SCEP enrollment is complete.

In the case of a certificate renewal, when the same process is followed, the renewal fails. This is because the registration authority cannot validate the renewal request since the router uses the current certificate as the

credentials. Since the registration authority can use only one trustpoint to validate the router identity, the certificate renewal fails.

To overcome this challenge, you can now configure the registration authority to use multiple trustpoints to validate the router credentials. In this manner, the initial enrollment as well as the renewal works seamlessly.

To configure multiple trustpoints, use the **grant auto <tp-list>** command. You can configure from upto 5 trustpoints by using this command. For example:

```
grant auto tp-list <tp1 tp2>
grant auto tp-list <tp1 tp2 tp3>
grant auto tp-list <tp1 tp2 tp3 tp4>
grant auto tp-list <tp1 tp2 tp3 tp4 tp5>
```

After you configure the trustpoints, the registration authority validates the certificates that are received by using one of the configured trustpoints. The validation starts from the first trustpoint. If the validation is successful, the certificate is renewed. Else, the authority validates using the next available trustpoint.

Sample Configuration

```
crypto pki server FANRSACA
no database archive
grant auto <tp-list> ACT2_SUDI_CA <CA_TRUSTPOINT>
hash sha256
mode ra transparent
!
crypto pki trustpoint FANRSACA
enrollment url http://10.4.1.117:8080/ejbca/publicweb/apply/scep/FANRSACA
serial-number none
fqdn none
ip-address none
subject-name serialNumber=PID:ISR4451-X/K9 SN:FOC23231CRY, CN=ISR4k-1-ra
revocation-check none
rsakeypair FANRSACA_Key 4096
!
crypto pki trustpoint ACT2_SUDI_CA
enrollment profile ACT2_SUDI_CA
revocation-check none
!
crypto pki trustpool policy
revocation-check none
```

Note **Grant auto trustpoint** and **grant auto tp-list** are mutually exclusive. You cannot run the **grant auto tp-list** command if you have already configured grant auto trustpoint.

Configuring PKI Server to Include EKU Attributes

Perform this task to configure PKI Server to Include EKU Attributes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **crypto pki server** *cs-label*

5. **eku request** *attribute*
6. **exit**
7. **exit**
8. **show crypto pki counters**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **a.** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip http server** **Example:** `Device(config)# ip http server`	Enables the HTTP server on your system.
Step 4	**crypto pki server** *cs-label* **Example:** `Device(config)# crypto pki server server-pki`	Defines a label for the certificate server and enters certificate server configuration mode. **Note** If you manually generated an RSA key pair, the *cs-label* argument must match the name of the key pair.
Step 5	**eku request** *attribute* **Example:** `Device(cs-server)# eku request ssh-server`	Requests to include specified eku *attribute* in the certificate. The *attribute* argument can be one of the following: • client-auth • code-signing • email-protection • ipsec-end-system • ipsec-tunnel • ipsec-user • ocsp-signing • server-auth • time-stamping • ssh-server

	Command or Action	Purpose
		• ssh-client
Step 6	**exit** **Example:** `Device(cs-server)# exit`	Exits cs-server configuration mode and returns to global configuration mode.
Step 7	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 8	**show crypto pki counters** **Example:** `Device# show crypto pki counters`	(Optional) Displays the PKI counters of the device.

Example

The following is sample output from the **show crypto pki counters**.

```
Device# show crypto pki counters

PKI Sessions Started: 0
PKI Sessions Ended: 0
PKI Sessions Active: 0
Successful Validations: 0
Failed Validations: 0
Bypassed Validations: 0
Pending Validations: 0
CRLs checked: 0
CRL - fetch attempts: 0
CRL - failed attempts: 0
CRL - rejected busy fetching: 0
OCSP - fetch requests: 0
OCSP - received responses: 0
OCSP - failed attempts: 0
OCSP - staple requests: 0
AAA authorizations: 0
```

Configuration Examples for PKI Certificate Enrollment Requests

Configuring Certificate Enrollment or Autoenrollment Example

The following example shows the configuration for the “mytp-A” certificate server and its associated trustpoint, where RSA keys generated by the initial autoenrollment for the trustpoint will be stored on a USB token, “usbtoken0”:

```
crypto pki server mytp-A
   database level complete
   issuer-name CN=company, L=city, C=country
   grant auto
! Specifies that certificate requests will be granted automatically.
!
crypto pki trustpoint mytp-A
   revocation-check none
   rsakeypair myTP-A
   storage usbtoken0:
! Specifies that keys will be stored on usbtoken0:.
   on usbtoken0:
```

! Specifies that keys generated on initial auto enroll will be generated on and stored o**n !** usbtoken0:

Configuring Autoenrollment Example

The following example shows how to configure the router to automatically enroll with a CA on startup, enabling automatic rollover, and how to specify all necessary enrollment information in the configuration:

```
crypto pki trustpoint trustpt1
 enrollment url http://trustpt1.example.com//
 subject-name OU=Spiral Dept., O=example.com
 ip-address ethernet-0
 serial-number none
 usage ike
 auto-enroll regenerate
 password password1
 rsa-key trustpt1 2048
!
crypto pki certificate chain trustpt1
certificate pki 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
```

Note In this example, keys are neither regenerated nor rolled over.

Configuring Certificate Autoenrollment with Key Regeneration Example

The following example shows how to configure the router to automatically enroll with the CA named "trustme1" on startup and enable automatic rollover. The **regenerate** keyword is issued, so a new key will be generated for the certificate and reissued when the automatic rollover process is initiated. 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.

```
crypto pki trustpoint trustme1
 enrollment url http://trustme1.example.com/
 subject-name OU=Spiral Dept., O=example.com
 ip-address ethernet0
 serial-number none
 auto-enroll 90 regenerate
 password password1
 rsakeypair trustme1 2048
 exit
crypto pki authenticate trustme1
copy system:running-config nvram:startup-config
```

Configuring Cut-and-Paste Certificate Enrollment Example

The following example shows how to configure certificate enrollment using the manual cut-and-paste enrollment method:

```
Router(config)#
crypto pki trustpoint TP
Router(ca-trustpoint)#
enrollment terminal
Router(ca-trustpoint)#
crypto pki authenticate TP
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)#
crypto pki enroll TP
% Start certificate enrollment..
```

```
% The subject name in the certificate will be:
Router.example.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+
!
!
!
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
!
!
!
Redisplay enrollment request? [yes/no]:
n
Router(config)#
crypto pki import TP 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)#
```

```
crypto pki import TP cert
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
```

You can verify that the certificate was successfully imported by issuing the **show crypto pki certificates** command:

```
Router# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number: 14DECE05000000000C48
  Certificate Usage: Encryption
  Issuer:
    CN = TPCA-root
     O = Company
     C = US
  Subject:
    Name: Router.example.com
    OID.1.2.840.113549.1.9.2 = Router.example.com
  CRL Distribution Point:
    http://tpca-root/CertEnroll/tpca-root.crl
  Validity Date:
    start date: 18:16:45 PDT Jun 7 2002
    end   date: 18:26:45 PDT Jun 7 2003
    renew date: 16:00:00 PST Dec 31 1969
  Associated Trustpoints: TP
Certificate
  Status: Available
  Certificate Serial Number: 14DEC2E9000000000C47
  Certificate Usage: Signature
  Issuer:
    CN = tpca-root
     O = company
     C = US
  Subject:
    Name: Router.example.com
    OID.1.2.840.113549.1.9.2 = Router.example.com
  CRL Distribution Point:
    http://tpca-root/CertEnroll/tpca-root.crl
  Validity Date:
    start date: 18:16:42 PDT Jun 7 2002
    end   date: 18:26:42 PDT Jun 7 2003
    renew date: 16:00:00 PST Dec 31 1969
```

```
  Associated Trustpoints: TP
CA Certificate
  Status: Available
  Certificate Serial Number: 3AC0A65E9547C2874AAF2468A942D5EE
  Certificate Usage: Signature
  Issuer:
    CN = tpca-root
     O = Company
     C = US
  Subject:
    CN = tpca-root
     O = company
     C = US
  CRL Distribution Point:
    http://tpca-root/CertEnroll/tpca-root.crl
  Validity Date:
    start date: 16:46:01 PST Feb 13 2002
    end   date: 16:54:48 PST Feb 13 2007
  Associated Trustpoints: TP
```

Configuring Manual Certificate Enrollment with Key Regeneration Example

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

```
crypto pki trustpoint trustme2
 enrollment url http://trustme2.example.com/
 subject-name OU=Spiral Dept., O=example.com
 ip-address ethernet0
 serial-number none
 regenerate
 password password1
 rsakeypair trustme2 2048
 exit
crypto pki authenticate trustme2
crypto pki enroll trustme2
```

Creating and Verifying a Persistent Self-Signed Certificate Example

The following example shows how to declare and enroll a trustpoint named “local” and generate a self-signed certificate with an IP address:

```
crypto pki trustpoint local
 enrollment selfsigned
 end
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
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 router 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 Example

The following example shows how to enable the HTTPS server and generate a default trustpoint because one was not previously configured:

```
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
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 need to save the configuration to NVRAM if you want to keep the self-signed certificate and have the HTTPS server enabled following router reloads.

The following message also appears:

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

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. You can use the **ip ssh rsa keypair-name** *unexisting-key-pair-name* command to disable the SSH server.

Verifying the Self-Signed Certificate Configuration Example

The following example displays information about the self-signed certificate that you just created:

```
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 is the router's serial number and varies depending on the router's actual serial number.

The following example displays information about the key pair corresponding to the self-signed certificate:

```
Router# show crypto key 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 when any key pair is created on the router and SSH starts up.

The following example displays information about the trustpoint named "local":

```
Router# show crypto pki trustpoints
Trustpoint local:
    Subject Name:
    serialNumber=C63EBBE9+ipaddress=10.3.0.18+hostname=test.example.com
          Serial Number: 01
    Persistent self-signed certificate trust point
```

Configuring Direct HTTP Enrollment Example

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

```
crypto pki trustpoint Entrust
 enrollment profile E
 serial
crypto pki profile enrollment E
 authentication url http://entrust:81
 authentication command GET /certs/cacert.der
 enrollment url http://entrust:81/cda-cgi/clientcgi.exe
 enrollment command POST reference_number=$P2&authcode=$P1
 &retrievedAs=rawDER&action=getServerCert&pkcs10Request=$REQ
```

```
 parameter 1 value aaaa-bbbb-cccc
 parameter 2 value 5001
```

Configuring Certificate Enrollment in a Two-Tier PKI Environment Example

Example of importing the ROOT-CA via terminal.

```
(config)#crypto pki trustpoint ROOT-CA
(ca-trustpoint)#revocation-check none
(ca-trustpoint)#enrollment terminal


(config)#crypto pki authenticate ROOT-CA

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
MIIDdTCCAl2gAwIBAgIQIfTArEE1yKZPXHaAVgDk5jANBgkqhkiG9w0BAQsFADBN
MRMwEQYKCZImiZPyLGQBGRYDY29tMRgwFgYKCZImiZPyLGQBGRYIdnBuLWVhc3Qx
HDAaBgNVBAMTE3Zwbi1lYXN0LXphY2ttY2ktQ0EwHhcNMTgxMjIwMDAwNjMyWhcN
MjgxMjIwMDAxNjMyWjBNMRMwEQYKCZImiZPyLGQBGRYDY29tMRgwFgYKCZImiZPy
LGQBGRYIdnBuLWVhc3QxHDAaBgNVBAMTE3Zwbi1lYXN0LXphY2ttY2ktQ0EwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC9Gdns9lU2HHc+XYhrmZKg6+Xo
5kNflu6mMgCfZ7ZiAKxZ03whJWZqNC7JRZQ+LkIJAcBUSf2mSJWRp+HVgI6k4Zf7
bMgIBq629HT8XmFLrr3lfh1lfL7WqI1Uez7/PEzjsw09y/m/WiSnrlgR3+PvyDbH
E86A6JnmtTNIs4qawUe72BlnEzwwRaFNi7VQz7GQw3CUo+RX9wtFYjABTyTUM/BA
MP47pI8CVh1jHVHqHcbqpyd97j1/8n1d/NCmcHKIg2hnKEO1Hx8oK7QIHe1rkryl
+r0ol2fS3CGgY000+FINs3qw4h8H8xfmsc5cs8lJCIbZGJhMTXq6u4Ecp+N1AgMB
AAGjUTBPMAsGA1UdDwQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBTb
zvfa7aNZspz3GwJCvKDIKO8KFTAQBgkrBgEEAYI3FQEEAwIBADANBgkqhkiG9w0B
AQsFAAOCAQEAgTIPTauHsPp7h1v/iFXkbVV1aG7O8/IaJG0sCr0f9/nsfM9HO0Jm
LP+twy5KkFa7I6u4vMlMlfNyujS60Fqnw3m8UJCy2SkYVwlGrBddN+BQbnkZ460M
sYfaynFBsvsbmmaLEqUQ3t9cmNCskXoda+FffyFTwAUBFzV66BGKpn6Y7oyIghF5
NLjjgWPVmRy7RKM4IKe9J0+oEmnugwtdfHgiFdX+d6qPovjbApj2j6N4+Cv6qHDO
/c+wUXRxz08eFNOqHNJipk70OXMrUh4UaWMnM/CYA9E1sjjSAWhBl4ii/+fiaILw
xgof+2mmIzafzFZz+eVf5kgwpV07GlZlng==
-----END CERTIFICATE-----
quit
Certificate has the following attributes:
       Fingerprint MD5: 99182E1E 96FB0595 DF86BFCE 3C781CF5
      Fingerprint SHA1: 6E55B878 9AA3B603 D689AC25 F027615E 0C88E6E4

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

Authenticating SUB-CA without having to specify or accept the fingerprint.

(config)#crypto pki trustpoint SUB-CA
(ca-trustpoint)#enrollment url  http://<SUBCA_IP/FQDN>:80/certsrv/mscep/mscep.dll
(ca-trustpoint)#chain-validation continue ROOT-CA
(ca-trustpoint)#revocation-check none

(ca-trustpoint)#crypto pki authenticate SUB-CA
Certificate has the following attributes:
       Fingerprint MD5: 5C38CB0A 050AAE87 84A08A75 5F7084B8
      Fingerprint SHA1: EB829470 B8B9E26E 4457F346 7A3E957C C623C6F9
Certificate validated - Signed by existing trustpoint CA certificate.

Trustpoint CA certificate accepted.
```

Additional References

Related Documents

Related Topic	Document Title
USB token RSA operations: Benefits of using USB tokens	"Storing PKI Credentials" module in the Cisco IOS Security Configuration Guide: Secure Connectivity
USB token RSA operations: Certificate server configuration	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" chapter in the Cisco IOS Security Configuration Guide: Secure Connectivity See the "Generating a Certificate Server RSA Key Pair" section, the "Configuring a Certificate Server Trustpoint" section, and related examples.
Overview of PKI, including RSA keys, certificate enrollment, and CAs	" Cisco IOS PKI Overview: Understanding and Planning a PKI " module in the Cisco IOS Security Configuration Guide: Secure Connectivity
Secure Device Provisioning: functionality overview and configuration tasks	" Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI " module in the Cisco IOS Security Configuration Guide: Secure Connectivity
RSA key generation and deployment	" Deploying RSA Keys Within a PKI " module in the Cisco IOS Security Configuration Guide: Secure Connectivity
Cisco IOS certificate server overview information and configuration tasks	" Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment " module in the Cisco IOS Security Configuration Guide: Secure Connectivity
Setting up and using a USB token	" Storing PKI Credentials " module in the Cisco IOS Security Configuration Guide: Secure Connectivity
Cisco IOS security commands	*Cisco IOS Security Command Reference*
Suite-B ESP transforms	Configuring Security for VPNs with IPsec feature module.
Suite-B SHA-2 family (HMAC variant) and Elliptic Curve (EC) key pair configuration.	Configuring Internet Key Exchange for IPsec VPNs feature module.
Suite-B Integrity algorithm type transform configuration.	Configuring Internet Key Exchange Version 2 (IKEv2) feature module.
Suite-B Elliptic Curve Digital Signature Algorithm (ECDSA) signature (ECDSA-sig) authentication method configuration for IKEv2.	Configuring Internet Key Exchange Version 2 (IKEv2) feature module.

Related Topic	Document Title
Suite-B Elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation	Configuring Internet Key Exchange for IPsec VPNs and Configuring Internet Key Exchange Version 2 (IKEv2) feature modules.
Recommended cryptographic algorithms	*Next Generation Encryption*

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.