Configuring Authorization and Revocation of Certificates in a PKI

This module describes how to configure authorization and revocation of certificates in a public key infrastructure (PKI). It includes information on high-availability support for the certificate 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.

Prerequisites for Authorization and Revocation of Certificates

Plan Your PKI Strategy

Tip It is strongly recommended that you plan your entire PKI strategy before you begin to deploy actual certificates.

Authorization and revocation can occur only after you or a network administrator have completed the following tasks:

- Configured the certificate authority (CA).
- Enrolled peer devices with the CA.
- Identified and configured the protocol (such as IP Security [IPsec] or secure socket layer [SSL]) that is to be used for peer-to-peer communication.

You should decide which authorization and revocation strategy you are going to configure before enrolling peer devices because the peer device certificates might have to contain authorization and revocation-specific information.

"crypto ca" to "crypto pki" CLI Change

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

High Availability

For high availability, IPsec-secured Stream Control Transmission Protocol (SCTP) must be configured on both the active and the standby routers. For synchronization to work, the redundancy mode on the certificate servers must be set to ACTIVE/STANDBY after you configure SCTP.

Restrictions for Authorization and Revocation of Certificates

- PKI High Availability (HA) support of intra-chassis stateful switchover (SSO) redundancy is currently not supported on all switches running the Cisco IOS Release 12.2 S software. See Cisco bug CSCtb59872 for more information.
- Depending on your Cisco IOS release, Lightweight Directory Access Protocol (LDAP) is supported.

Information About Authorization and Revocation of Certificates

PKI Authorization

PKI authentication does not provide authorization. Current solutions for authorization are specific to the router that is being configured, although a centrally managed solution is often required.

There is not a standard mechanism by which certificates are defined as authorized for some tasks and not for others. This authorization information can be captured in the certificate itself if the application is aware of the certificate-based authorization information. But this solution does not provide a simple mechanism for real-time updates to the authorization information and forces each application to be aware of the specific authorization information embedded in the certificate.

When the certificate-based ACL mechanism is configured as part of the trustpoint authentication, the application is no longer responsible for determining this authorization information, and it is no longer possible to specify for which application the certificate is authorized. In some cases, the certificate-based ACL on the router gets so large that it cannot be managed. Additionally, it is beneficial to retrieve certificate-based ACL indications from an external server.

Current solutions to the real-time authorization problem involve specifying a new protocol and building a new server (with associated tasks, such as management and data distribution).

PKI and AAA Server Integration for Certificate Status

Integrating your PKI with an authentication, authorization, and accounting (AAA) server provides an alternative online certificate status solution that leverages the existing AAA infrastructure. Certificates can be listed in the AAA database with appropriate levels of authorization. For components that do not explicitly support PKI-AAA, a default label of "all" from the AAA server provides authorization. Likewise, a label of "none" from the AAA database indicates that the specified certificate is not valid. (The absence of any application label is equivalent, but "none" is included for completeness and clarity). If the application component does support PKI-AAA, the component may be specified directly; for example, the application component could be "ipsec," "ssl," or "osp." (ipsec=IP Security, ssl=Secure Sockets Layer, and osp=Open Settlement Protocol.)

Note Currently, no application component supports specification of the application label.

- There may be a time delay when accessing the AAA server. If the AAA server is not available, the authorization fails.

RADIUS or TACACS+ Choosing a AAA Server Protocol

The AAA server can be configured to work with either the RADIUS or TACACS+ protocol. When you are configuring the AAA server for the PKI integration, you must set the RADIUS or TACACS attributes that are required for authorization.

If the RADIUS protocol is used, the password that is configured for the username in the AAA server should be set to "cisco," which is acceptable because the certificate validation provides authentication and the AAA database is only being used for authorization. When the TACACS protocol is used, the password that is configured for the username in the AAA server is irrelevant because TACACS supports authorization without requiring authentication (the password is used for authentication).

In addition, if you are using TACACS, you must add a PKI service to the AAA server. The custom attribute "cert-application=all" is added under the PKI service for the particular user or usergroup to authorize the specific username.

Attribute-Value Pairs for PKI and AAA Server Integration

The table below lists the attribute-value (AV) pairs that are to be used when setting up PKI integration with a AAA server. (Note the values shown in the table are possible values.) The AV pairs must match the client configuration. If they do not match, the peer certificate is not authorized.

Note Users can sometimes have AV pairs that are different from those of every other user. As a result, a unique username is required for each user. The **all** parameter (within the **authorization username** command) specifies that the entire subject name of the certificate will be used as the authorization username.

Table 1: AV Pairs That Must Match

AV Pair	Value
cisco-avpair=pki:cert-application=all	Valid values are "all" and "none."

AV Pair	Value
cisco-avpair=pki:cert-trustpoint=msca	The value is a Cisco IOS command-line interface (CLI) configuration trustpoint label. **Note** The cert-trustpoint AV pair is normally optional. If it is specified, the Cisco IOS router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.
cisco-avpair=pki:cert-serial=16318DB7000100001671	The value is a certificate serial number. **Note** The cert-serial AV pair is normally optional. If it is specified, the Cisco IOS router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.
cisco-avpair=pki:cert-lifetime-end=1:00 jan 1, 2003	The cert-lifetime-end AV pair is available to artificially extend a certificate lifetime beyond the time period that is indicated in the certificate itself. If the cert-lifetime-end AV pair is used, the cert-trustpoint and cert-serial AV pairs must also be specified. The value must match the following form: hours:minutes month day, year. **Note** Only the first three characters of a month are used: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. If more than three characters are entered for the month, the remaining characters are ignored (for example Janxxxx).

CRLs or OCSP Server Choosing a Certificate Revocation Mechanism

After a certificate is validated as a properly signed certificate, a certificate revocation method is performed to ensure that the certificate has not been revoked by the issuing CA. Cisco IOS software supports two revocation mechanisms--certificate revocation lists (CRLs) and Online Certificate Status Protocol (OCSP). Cisco IOS software also supports AAA integration for certificate checking; however, additional authorization functionality is included. For more information on PKI and AAA certificate authorization and status check, see the PKI and AAA Server Integration for Certificate Status section.

The following sections explain how each revocation mechanism works:

What Is a CRL

A certificate revocation list (CRL) is a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when each certificate was issued and when it expires.

CAs publish new CRLs periodically or when a certificate for which the CA is responsible has been revoked. By default, a new CRL is downloaded after the currently cached CRL expires. An administrator may also configure the duration for which CRLs are cached in router memory or disable CRL caching completely. The CRL caching configuration applies to all CRLs associated with a trustpoint.

When the CRL expires, the router deletes it from its cache. A new CRL is downloaded when a certificate is presented for verification; however, if a newer version of the CRL that lists the certificate under examination is on the server but the router is still using the CRL in its cache, the router does not know that the certificate has been revoked. The certificate passes the revocation check even though it should have been denied.

When a CA issues a certificate, the CA can include in the certificate the CRL distribution point (CDP) for that certificate. Cisco IOS client devices use CDPs to locate and load the correct CRL. The Cisco IOS client supports multiple CDPs, but the Cisco IOS CA currently supports only one CDP; however, third-party vendor CAs may support multiple CDPs or different CDPs per certificate. If a CDP is not specified in the certificate, the client device uses the default Simple Certificate Enrollment Protocol (SCEP) method to retrieve the CRL. (The CDP location can be specified through the **cdp-url**command.)

When implementing CRLs, you should consider the following design considerations:

- CRL lifetimes and the security association (SA) and Internet Key Exchange (IKE) lifetimes.
- The CRL lifetime determines the length of time between CA-issued updates to the CRL. The default CRL lifetime value, which is 168 hours [1 week], can be changed through the **lifetime crl** command.
- The method of the CDP determines how the CRL is retrieved; some possible choices include HTTP, Lightweight Directory Access Protocol (LDAP), SCEP, or TFTP. HTTP, TFTP, and LDAP are the most commonly used methods. Although Cisco IOS software defaults to SCEP, an HTTP CDP is recommended for large installations using CRLs because HTTP can be made highly scalable.
- The location of the CDP determines from where the CRL is retrieved; for example, you can specify the server and file path from which to retrieve the CRL.

Note If Public Key Infrastructure (PKI) with Certificate Revocation Lists (CRLs) are used, and if the size of the PKI CRL file exceeds 200 KB (approximately) and above, a CPU HOG may be generated.

Querying All CDPs During Revocation Check

When a CDP server does not respond to a request, the Cisco IOS software reports an error, which may result in the peer's certificate being rejected. To prevent a possible certificate rejection and if there are multiple CDPs in a certificate, the Cisco IOS software will attempt to use the CDPs in the order in which they appear in the certificate. The router will attempt to retrieve a CRL using each CDP URL or directory specification. If an error occurs using a CDP, an attempt will be made using the next CDP.

Note Prior to Cisco IOS Release 12.3(7)T, the Cisco IOS software makes only one attempt to retrieve the CRL, even when the certificate contains more than one CDP.

Tip Although the Cisco IOS software will make every attempt to obtain the CRL from one of the indicated CDPs, it is recommended that you use an HTTP CDP server with high-speed redundant HTTP servers to avoid application timeouts because of slow CDP responses.

What Is OCSP

OCSP is an online mechanism that is used to determine certificate validity and provides the following flexibility as a revocation mechanism:

- OCSP can provide real-time certificate status checking.
- OCSP allows the network administrator to specify a central OCSP server, which can service all devices within a network.
- OCSP also allows the network administrator the flexibility to specify multiple OCSP servers, either per client certificate or per group of client certificates.
- OCSP server validation is usually based on the root CA certificate or a valid subordinate CA certificate, but may also be configured so that external CA certificates or self-signed certificates may be used. Using external CA certificates or self-signed certificates allows the OCSP servers certificate to be issued and validated from an alternative PKI hierarchy.

A network administrator can configure an OCSP server to collect and update CRLs from different CA servers. The devices within the network can rely on the OCSP server to check the certificate status without retrieving and caching each CRL for every peer. When peers have to check the revocation status of a certificate, they send a query to the OCSP server that includes the serial number of the certificate in question and an optional unique identifier for the OCSP request, or a nonce. The OCSP server holds a copy of the CRL to determine if the CA has listed the certificate as being revoked; the server then responds to the peer including the nonce. If the nonce in the response from the OCSP server does not match the original nonce sent by the peer, the response is considered invalid and certificate verification fails. The dialog between the OCSP server and the peer consumes less bandwidth than most CRL downloads.

If the OCSP server is using a CRL, CRL time limitations will be applicable; that is, a CRL that is still valid might be used by the OCSP server although a new CRL has been issued by the CRL containing additional certificate revocation information. Because fewer devices are downloading the CRL information on a regular basis, you can decrease the CRL lifetime value or configure the OCSP server not to cache the CRL. For more information, check your OCSP server documentation.

When to Use an OCSP Server

OCSP may be more appropriate than CRLs if your PKI has any of the following characteristics:

- Real-time certificate revocation status is necessary. CRLs are updated only periodically and the latest CRL may not always be cached by the client device. For example, if a client does not yet have the latest CRL cached and a newly revoked certificate is being checked, that revoked certificate will successfully pass the revocation check.
- There are a large number of revoked certificates or multiple CRLs. Caching a large CRL consumes large portions of Cisco IOS memory and may reduce resources available to other processes.
- CRLs expire frequently, causing the CDP to handle a larger load of CRLs.

Note As of Cisco IOS Release 12.4(9)T or later, an administrator may configure CRL caching, either by disabling CRL caching completely or setting a maximum lifetime for a cached CRL per trustpoint.

When to Use Certificate-Based ACLs for Authorization or Revocation

Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action.

Because certificate-based ACLs are configured on the device, they do not scale well for large numbers of ACLs; however, certificate-based ACLs do provide very granular control of specific device behavior. Certificate-based ACLs are also leveraged by additional features to help determine when PKI components such as revocation, authorization, or a trustpoint should be used. They provide a general mechanism allowing users to select a specific certificate or a group of certificates that are being validated for either authorization or additional processing.

Certificate-based ACLs specify one or more fields within the certificate and an acceptable value for each specified field. You can specify which fields within a certificate should be checked and which values those fields may or may not have.

There are six logical tests for comparing the field with the value--equal, not equal, contains, does not contain, less than, and greater than or equal. If more than one field is specified within a single certificate-based ACL, the tests of all of the fields within the ACL must succeed to match the ACL. The same field may be specified multiple times within the same ACL. More than one ACL may be specified, and ACL will be processed in turn until a match is found or all of the ACLs have been processed.

Ignore Revocation Checks Using a Certificate-Based ACL

Certificate-based ACLs can be configured to instruct your router to ignore the revocation check and expired certificates of a valid peer. Thus, a certificate that meets the specified criteria can be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. You can also use a certificate-based ACL to ignore the revocation check when the communication with a AAA server is protected with a certificate.

Ignoring Revocation Lists

To allow a trustpoint to enforce CRLs except for specific certificates, enter the **match certificate**command with the **skip revocation-check** keyword. This type of enforcement is most useful in a hub-and-spoke configuration in which you also want to allow direct spoke-to-spoke connections. In pure hub-and-spoke configurations, all spokes connect only to the hub, so CRL checking is necessary only on the hub. For one spoke to communicate directly with another spoke, the **match certificate**command with the **skip revocation-check** keyword can be used for neighboring peer certificates instead of requiring a CRL on each spoke.

Ignoring Expired Certificates

To configure your router to ignore expired certificates, enter the **match certificate** command with the **allow expired-certificate** keyword. This command has the following purposes:

- If the certificate of a peer has expired, this command may be used to “allow” the expired certificate until the peer can obtain a new certificate.

- If your router clock has not yet been set to the correct time, the certificate of a peer will appear to be not yet valid until the clock is set. This command may be used to allow the certificate of the peer even though your router clock is not set.

Note If Network Time Protocol (NTP) is available only via the IPSec connection (usually via the hub in a hub-and-spoke configuration), the router clock can never be set. The tunnel to the hub cannot be "brought up" because the certificate of the hub is not yet valid.

- "Expired" is a generic term for a certificate that is expired or that is not yet valid. The certificate has a start and end time. An expired certificate, for purposes of the ACL, is one for which the current time of the router is outside the start and end times specified in the certificate.

Skipping the AAA Check of the Certificate

If the communication with an AAA server is protected with a certificate, and you want to skip the AAA check of the certificate, use the **match certificate** command with the **skip authorization-check** keyword. For example, if a virtual private network (VPN) tunnel is configured so that all AAA traffic goes over that tunnel, and the tunnel is protected with a certificate, you can use the **match certificate** command with the **skip authorization-check** keyword to skip the certificate check so that the tunnel can be established.

The **match certificate**command and the **skip authorization-check** keyword should be configured after PKI integration with an AAA server is configured.

Note If the AAA server is available only via an IPSec connection, the AAA server cannot be contacted until after the IPSec connection is established. The IPSec connection cannot be "brought up" because the certificate of the AAA server is not yet valid.

PKI Certificate Chain Validation

A certificate chain establishes a sequence of trusted certificates --from a peer certificate to the root CA certificate. Within a PKI hierarchy, all enrolled peers can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA. Each CA corresponds to a trustpoint.

When a certificate chain is received from a peer, the default processing of a certificate chain path continues until the first trusted certificate, or trustpoint, is reached. In Cisco IOS Release 12.4(6)T and later releases, an administrator may configure the level to which a certificate chain is processed on all certificates including subordinate CA certificates.

Configuring the level to which a certificate chain is processed allows for the reauthentication of trusted certificates, the extension of a trusted certificate chain, and the completion of a certificate chain that contains a gap.

Reauthentication of Trusted Certificates

The default behavior is for the router to remove any trusted certificates from the certificate chain sent by the peer before the chain is validated. An administrator may configure certificate chain path processing so that

the router does not remove CA certificates that are already trusted before chain validation, so that all certificates in the chain are re-authenticated for the current session.

Extending the Trusted Certificate Chain

The default behavior is for the router to use its trusted certificates to extend the certificate chain if there are any missing certificates in the certificate chain sent by the peer. The router will validate only certificates in the chain sent by the peer. An administrator may configure certificate chain path processing so that the certificates in the peer's certificate chain and the router's trusted certificates are validated to a specified point.

Completing Gaps in a Certificate Chain

An administrator may configure certificate chain processing so that if there is a gap in the configured Cisco IOS trustpoint hierarchy, certificates sent by the peer can be used to complete the set of certificates to be validated.

Note If the trustpoint is configured to require parent validation and the peer does not provide the full certificate chain, the gap cannot be completed and the certificate chain is rejected and invalid.

Note It is a configuration error if the trustpoint is configured to require parent validation and there is no parent trustpoint configured. The resulting certificate chain gap cannot be completed and the subordinate CA certificate cannot be validated. The certificate chain is invalid.

How to Configure Authorization and Revocation of Certificates for Your PKI

Configuring PKI Integration with a AAA Server

Perform this task to generate a AAA username from the certificate presented by the peer and specify which fields within a certificate should be used to build the AAA database username.

Note The following restrictions should be considered when using the **all** keyword as the subject name for the **authorization username** command:

- Some AAA servers limit the length of the username (for example, to 64 characters). As a result, the entire certificate subject name cannot be longer than the limitation of the server.
- Some AAA servers limit the available character set that may be used for the username (for example, a space [ ] and an equal sign [=] may not be acceptable). You cannot use the **all** keyword for a AAA server having such a character-set limitation.
- The **subject-name** command in the trustpoint configuration may not always be the final AAA subject name. If the fully qualified domain name (FQDN), serial number, or IP address of the router are included in a certificate request, the subject name field of the issued certificate will also have these components. To turn off the components, use the **fqdn**, **serial-number**, and **ip-address** commands with the **none** keyword.
- CA servers sometimes change the requested subject name field when they issue a certificate. For example, CA servers of some vendors switch the relative distinguished names (RDNs) in the requested subject names to the following order: CN, OU, O, L, ST, and C. However, another CA server might append the configured LDAP directory root (for example, O=cisco.com) to the end of the requested subject name.
- Depending on the tools you choose for displaying a certificate, the printed order of the RDNs in the subject name could be different. Cisco IOS software always displays the least significant RDN first, but other software, such as Open Source Secure Socket Layer (OpenSSL), does the opposite. Therefore, if you are configuring a AAA server with a full distinguished name (DN) (subject name) as the corresponding username, ensure that the Cisco IOS software style (that is, with the least significant RDN first) is used.

or

radius-server host *hostname* [**key** *string*]

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authorization network** *listname* [*method*]
5. **crypto pki trustpoint** *name*
6. **enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]
7. revocation-check method
8. **exit**
9. **authorization username subjectname** *subjectname*
10. **authorization list** *listname*
11. **tacacs-server host** hostname [**key** string]

DETAILED STEPS

	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	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa authorization network** *listname* [*method*] **Example:** `Router (config)# aaa authorization network maxaaa group tacacs+`	Sets the parameters that restrict user access to a network. • *method* --Can be **group radius**, **group tacacs+**, or **group group-name**.
Step 5	**crypto pki trustpoint** *name* **Example:** `Route (config)# crypto pki trustpoint msca`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

Step 6	**enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]
	Example:
	`Router (ca-trustpoint)# enrollment url http://caserver.myexample.com`
	- or-
	`Router (ca-trustpoint)# enrollment url http://[2001:DB8:1:1::1]:80`
		Specifies the following enrollment parameters of the CA:
		• (Optional) The **mode** keyword specifies the registration authority (RA) mode, if your CA system provides an RA. By default, RA mode is disabled.
		• (Optional) The **retry period** keyword and *minutes* argument specifies the period, in minutes, in which the router waits before sending the CA another certificate request. Valid values are from 1 to 60. The default is 1.
		• (Optional) The **retry count** keyword and *number* argument specifies the number of times a router will resend a certificate request when it does not receive a response from the previous request. Valid values are from 1 to 100. The default is 10.
		• The *url* argument is the URL of the CA to which your router should send certificate requests.

	Command or Action	Purpose
		Note With the introduction of Cisco IOS Release 15.2(1)T, an IPv6 address can be added to the **http:** enrolment method. For example: http://[ipv6-address]:80. The IPv6 address must be enclosed in brackets in the URL. See the Command Reference document for more information on the other enrollment methods that can be used. • (Optional) The **pem** keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.
Step 7	revocation-check method **Example:** `Router (ca-trustpoint)# revocation-check crl`	(Optional) Checks the revocation status of a certificate.
Step 8	**exit** **Example:** `Router (ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 9	**authorization username subjectname** *subjectname* **Example:** `Router (config)# authorization username subjectname serialnumber`	Sets parameters for the different certificate fields that are used to build the AAA username. The *subjectname* argument can be any of the following: • **all** --Entire distinguished name (subject name) of the certificate. • **commonname** --Certification common name. • **country** --Certificate country. • **email** --Certificate e-mail. • **ipaddress** --Certificate IP address. • **locality** --Certificate locality. • **organization** --Certificate organization. • **organizationalunit** --Certificate organizational unit. • **postalcode** --Certificate postal code. • **serialnumber** --Certificate serial number. • **state** --Certificate state field. • **streetaddress** --Certificate street address. • **title** --Certificate title.

	Command or Action	Purpose
		• **unstructuredname** --Certificate unstructured name.
Step 10	**authorization list** *listname* **Example:** `Route (config)# authorization list maxaaa`	Specifies the AAA authorization list.

Step 11	**tacacs-server host** hostname [**key** string]
	Example:
	`Router(config)# tacacs-server host 192.0.2.2 key a_secret_key`
	Example:
	`radius-server host hostname [key string]`
	Example:
	`Router(config)# radius-server host 192.0.2.1 key another_secret_key`
		Specifies a TACACS+ host.
		or
		Specifies a RADIUS host.

Troubleshooting Tips

To display debug messages for the trace of interaction (message type) between the CA and the router, use the **debug crypto pki transactions**command. (See the sample output, which shows a successful PKI integration with AAA server exchange and a failed PKI integration with AAA server exchange.)

Successful Exchange

```
Router# debug crypto pki transactions
Apr 22 23:15:03.695: CRYPTO_PKI: Found a issuer match
Apr 22 23:15:03.955: CRYPTO_PKI: cert revocation status unknown.
Apr 22 23:15:03.955: CRYPTO_PKI: Certificate validated without revocation check
```

Each line that shows “CRYPTO_PKI_AAA” indicates the state of the AAA authorization checks. Each of the AAA AV pairs is indicated, and then the results of the authorization check are shown.

```
Apr 22 23:15:04.019: CRYPTO_PKI_AAA: checking AAA authorization (ipsecca_script_aaalist,
PKIAAA-L, <all>)
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-serial" = "15DE")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: authorization passed
Apr 22 23:12:30.327: CRYPTO_PKI: Found a issuer match
```

Failed Exchange

```
Router# debug crypto pki transactions
Apr 22 23:11:13.703: CRYPTO_PKI_AAA: checking AAA authorization =
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-application" = “all”)
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint"= “CA1”)
```

```
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-serial" = “233D”)
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: parsed cert-lifetime-end as: 21:30:00
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: timezone specific extended
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end is expired
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end check failed.
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: authorization failed
```

In the above failed exchange, the certificate has expired.

Configuring a Revocation Mechanism for PKI Certificate Status Checking

Perform this task to set up a CRL as the certificate revocation mechanism--CRLs or OCSP--that is used to check the status of certificates in a PKI.

The revocation-check Command

Use the **revocation-check** command to specify at least one method (OCSP, CRL, or skip the revocation check) that is to be used to ensure that the certificate of a peer has not been revoked. For multiple methods, the order in which the methods are applied is determined by the order specified via this command.

If your router does not have the applicable CRL and is unable to obtain one or if the OCSP server returns an error, your router will reject the peer’s certificate--unless you include the **none** keyword in your configuration. If the **none** keyword is configured, a revocation check will not be performed and the certificate will always be accepted.

Note When the revocation check is changed to 'none' under a trustpoint, the CRL cache associated with the CA certificate of the trustpoint will be cleared.

Nonces and Peer Communications with OCSP Servers

When using OCSP, nonces, unique identifiers for OCSP requests, are sent by default during peer communications with your OCSP server. The use of nonces offers a more secure and reliable communication channel between the peer and OCSP server.

If your OCSP server does not support nonces, you may disable the sending of nonces. For more information, check your OCSP server documentation.

Before you begin

- Before issuing any client certificates, the appropriate settings on the server (such as setting the CDP) should be configured.
- When configuring an OCSP server to return the revocation status for a CA server, the OCSP server must be configured with an OCSP response signing certificate that is issued by that CA server. Ensure that the signing certificate is in the correct format, or the router will not accept the OCSP response. See your OCSP manual for additional information.

Note

- OCSP transports messages over HTTP, so there may be a time delay when you access the OCSP server.
- If the OCSP server depends on normal CRL processing to check revocation status, the same time delay that affects CRLs will also apply to OCSP.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. crypto pki trustpoint *name*
4. **ocsp url** *url*
5. **revocation-check** *method1* [*method2 method3*]]
6. **ocsp disable-nonce**
7. **exit**
8. **exit**
9. **show crypto pki certificates**
10. **show crypto pki trustpoints** [**status** | *label* [**status**]]

DETAILED STEPS

	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 hazel`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

Step 4	**ocsp url** *url*
	Example:
	`Router(ca-trustpoint)# ocsp url http://ocsp-server`
	- or -
	`Router(ca-trustpoint)# ocsp url http://10.10.10.1:80`
		The *url* argument specifies the URL of an OCSP server so that the trustpoint can check the certificate status. This URL overrides the URL of the OCSP server (if one exists) in the Authority Info Access (AIA) extension of the certificate. All certificates associated with a configured trustpoint are checked by the OCSP server. The URL can be a hostname, IPv4 address, or an IPv6 address.

	Command or Action	Purpose
	- or - `Router(ca-trustpoint)# ocsp url http://[2001DB8:1:1::2]:80`	
Step 5	**revocation-check** *method1* [*method2 method3*]] **Example:** `Router(ca-trustpoint)# revocation-check ocsp none`	Checks the revocation status of a certificate. • **crl** --Certificate checking is performed by a CRL. This is the default option. • **none** --Certificate checking is ignored. • **ocsp** --Certificate checking is performed by an OCSP server. If a second and third method are specified, each method will be used only if the previous method returns an error, such as a server being down.
Step 6	**ocsp disable-nonce** **Example:** `Router(ca-trustpoint)# ocsp disable-nonce`	(Optional) Specifies that a nonce, or an OCSP request unique identifier, will not be sent during peer communications with the OCSP server.
Step 7	**exit** **Example:** `Router(ca-trustpoint)# exit`	Returns to global configuration mode.
Step 8	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 9	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates.
Step 10	**show crypto pki trustpoints** [**status** | *label* [**status**]] **Example:** `Router# show crypto pki trustpoints`	Displays information about the trustpoint configured in router.

Configuring Certificate Authorization and Revocation Settings

Perform this task to specify a certificate-based ACL, to ignore revocation checks or expired certificates, to manually override the default CDP location, to manually override the OCSP server setting, to configure CRL caching, or to set session acceptance or rejection based on a certificate serial number, as appropriate.

Configuring Certificate-Based ACLs to Ignore Revocation Checks

To configure your router to use certificate-based ACLs to ignore revocation checks and expired certificates, perform the following steps:

- Identify an existing trustpoint or create a new trustpoint to be used when verifying the certificate of the peer. Authenticate the trustpoint if it has not already been authenticated. The router may enroll with this trustpoint if you want. Do not set optional CRLs for the trustpoint if you plan to use the **match certificate** command and **skip revocation-check** keyword.
- Determine the unique characteristics of the certificates that should not have their CRL checked and of the expired certificates that should be allowed.
- Define a certificate map to match the characteristics identified in the prior step.
- You can add the **match certificate** command and **skip revocation-check** keyword and the **match certificate command** and **allow expired-certificate** keyword to the trustpoint that was created or identified in the first step.

Note Certificate maps are checked even if the peer's public key is cached. For example, when the public key is cached by the peer, and a certificate map is added to the trustpoint to ban a certificate, the certificate map is effective. This prevents a client with the banned certificate, which was once connected in the past, from reconnecting.

Manually Overriding CDPs in a Certificate

Users can override the CDPs in a certificate with a manually configured CDP. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.

Manually Overriding the OCSP Server Setting in a Certificate

Administrators can override the OCSP server setting specified in the Authority Information Access (AIA) field of the client certificate or set by the issuing the **ocsp url** command. One or more OCSP servers may be manually specified, either per client certificate or per group of client certificates by the **match certificate override ocsp** command. The **match certificate override ocsp**command overrides the client certificate AIA field or the **ocsp url**command setting if a client certificate is successfully matched to a certificate map during the revocation check.

Note Only one OCSP server can be specified per client certificate.

Configuring CRL Cache Control

By default, a new CRL will be downloaded after the currently cached CRL expires. Administrators can either configure the maximum amount of time in minutes a CRL remains in the cache by issuing the **crl cache delete-after** command or disable CRL caching by issuing the **crl cache none** command. Only the **crl-cache**

delete-aftercommand or the **crl-cache none** command may be specified. If both commands are entered for a trustpoint, the last command executed will take effect and a message will be displayed.

Neither the **crl-cache none** command nor the **crl-cache delete-after** command affects the currently cached CRL. If you configure the **crl-cache none** command, all CRLs downloaded after this command is issued will not be cached. If you configure the **crl-cache delete-after** command, the configured lifetime will only affect CRLs downloaded after this command is issued.

This functionality is useful is when a CA issues CRLs with no expiration date or with expiration dates days or weeks ahead.

Configuring Certificate Serial Number Session Control

A certificate serial number can be specified to allow a certificate validation request to be accepted or rejected by the trustpoint for a session. A session may be rejected, depending on certificate serial number session control, even if a certificate is still valid. Certificate serial number session control may be configured by using either a certificate map with the **serial-number** field or an AAA attribute, with the **cert-serial-not** command.

Using certificate maps for session control allows an administrator to specify a single certificate serial number. Using the AAA attribute allows an administrator to specify one or more certificate serial numbers for session control.

Before you begin

- The trustpoint should be defined and authenticated before attaching certificate maps to the trustpoint.
- The certificate map must be configured before the CDP override feature can be enabled or the **serial-number** command is issued.
- The PKI and AAA server integration must be successfully completed to use AAA attributes as described in “PKI and AAA Server Integration for Certificate Status.”

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. crypto pki certificate map label sequence-number
4. *field-name match-criteria match-value*
5. **exit**
6. **crypto pki trustpoint** *name*
7. Do one of the following:
   - **crl-cache none**
   - **crl-cache delete-after** *time*
8. **match certificate** *certificate-map-label* [**allow expired-certificate** | **skip revocation-check** | **skip authorization-check**
9. **match certificate** *certificate-map-label* **override cdp** {**url** | **directory**} *string*
10. **match certificate** *certificate-map-label* **override ocsp** [**trustpoint** *trustpoint-label*] *sequence-number* **url** *ocsp-url*
11. **exit**
12. **aaa new-model**

13. **aaa attribute list** *list-name*
14. **attribute type** {*name*} {*value*}
15. **exit**
16. **exit**
17. **show crypto pki certificates**

DETAILED STEPS

	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 certificate map label sequence-number **Example:** `Router(config)# crypto pki certificate map Group 10`	Defines values in a certificate that should be matched or not matched and enters ca-certificate-map configuration mode.
Step 4	*field-name match-criteria match-value* **Example:** `Router(ca-certificate-map)# subject-name co MyExample`	Specifies one or more certificate fields together with their matching criteria and the value to match. The *field-name* is one of the following case-insensitive name strings or a date: • **alt-subject-name** • **expires-on** • **issuer-name** • **name** • **serial-number** • **subject-name** • **unstructured-subject-name** • **valid-start** **Note** Date field format is dd mm yyyy hh:mm:ss or mmm dd yyyy hh:mm:ss. The *match-criteria* is one of the following logical operators:

	Command or Action	Purpose
		• **co** --contains (valid only for name fields and serial number field) • **eq** --equal (valid for name, serial number, and date fields) • **ge** --greater than or equal (valid only for date fields) • **lt** --less than (valid only for date fields) • **nc** --does not contain (valid only for name fields and serial number field) • **ne** --not equal (valid for name, serial number, and date fields) The *match-value*is the name or date to test with the logical operator assigned by match-criteria. **Note** Use this command only when setting up a certificate-based ACL--not when setting up a certificate-based ACL to ignore revocation checks or expired certificates.
Step 5	**exit** **Example:** `Router(ca-certificate-map)# exit`	Returns to global configuration mode.
Step 6	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint Access2`	Declares the trustpoint, given name and enters ca-trustpoint configuration mode.

Step 7	Do one of the following:
	• **crl-cache none**
	• **crl-cache delete-after** *time*
	Example:
	`Router(ca-trustpoint)# crl-cache none`
	Example:
	`Router(ca-trustpoint)# crl-cache delete-after 20`
		(Optional) Disables CRL caching completely for all CRLs associated with the trustpoint.
		The **crl-cache none** command does not affect any currently cached CRLs. All CRLs downloaded after this command is configured will not be cached.
		(Optional) Specifies the maximum time CRLs will remain in the cache for all CRLs associated with the trustpoint.
		• *time* --The amount of time in minutes before the CRL is deleted.
		
		The **crl-cache delete-after** command does not affect any currently cached CRLs. The configured lifetime will only affect CRLs downloaded after this command is configured.

	Command or Action	Purpose
Step 8	**match certificate** *certificate-map-label* [**allow expired-certificate** | **skip revocation-check** | **skip authorization-check** **Example:** `Router(ca-trustpoint)# match certificate Group skip revocation-check`	(Optional) Associates the certificate-based ACL (that was defined via the **crypto pki certificate map** command) to a trustpoint. • *certificate-map-label* --Must match the *label* argument specified via the **crypto pki certificate map** command. • **allow expired-certificate** --Ignores expired certificates. • **skip revocation-check** --Allows a trustpoint to enforce CRLs except for specific certificates. • **skip authorization-check** --Skips the AAA check of a certificate when PKI integration with an AAA server is configured.
Step 9	**match certificate** *certificate-map-label* **override cdp** {**url** | **directory**} *string* **Example:** `Router(ca-trustpoint)# match certificate Group1 override cdp url http://server.cisco.com`	(Optional) Manually overrides the existing CDP entries for a certificate with a URL or directory specification. • *certificate-map-label* --A user-specified label that must match the *label* argument specified in a previously defined **crypto pki certificate map** command. • **url** --Specifies that the certificate's CDPs will be overridden with an HTTP or LDAP URL. • **directory** --Specifies that the certificate's CDPs will be overridden with an LDAP directory specification. • *string* --The URL or directory specification. **Note** Some applications may time out before all CDPs have been tried and will report an error message. The error message will not affect the router, and the Cisco IOS software will continue attempting to retrieve a CRL until all CDPs have been tried.
Step 10	**match certificate** *certificate-map-label* **override ocsp** [**trustpoint** *trustpoint-label*] *sequence-number* **url** *ocsp-url* **Example:** `Router(ca-trustpoint)# match certificate mycertmapname override ocsp trustpoint mytp 15 url http://192.0.2.2`	(Optional) Specifies an OCSP server, either per client certificate or per group of client certificates, and may be issued more than once to specify additional OCSP servers and client certificate settings including alternative PKI hierarchies. • *certificate-map-label* --The name of an existing certificate map. • **trustpoint** --The trustpoint to be used when validating the OCSP server certificate.

	Command or Action	Purpose
		• *sequence-number* --The order the **match certificate override ocsp** command statements apply to the certificate being verified. Matches are performed from the lowest sequence number to the highest sequence number. If more than one command is issued with the same sequence number, it overwrites the previous OCSP server override setting. • **url** --The URL of the OCSP server. When the certificate matches a configured certificate map, the AIA field of the client certificate and any previously issued **ocsp url** command settings are overwritten with the specified OCSP server. If no map-based match occurs, one of the following two cases will continue to apply to the client certificate. • If OCSP is specified as the revocation method, the AIA field value will continue to apply to the client certificate. • If the **ocsp url** configuration exists, the **ocsp url** configuration settings will continue to apply to the client certificates.
Step 11	**exit** **Example:** `Router(ca-trustpoint)# exit`	Returns to global configuration mode.
Step 12	**aaa new-model** **Example:** `Router(config)# aaa new-model`	(Optional) Enables the AAA access control model.
Step 13	**aaa attribute list** *list-name* **Example:** `Router(config)# aaa attribute list crl`	(Optional) Defines an AAA attribute list locally on a router and enters config-attr-list configuration mode.
Step 14	**attribute type** {*name*} {*value*} **Example:** `Router(config-attr-list)# attribute type cert-serial-not 6C4A`	(Optional) Defines an AAA attribute type that is to be added to an AAA attribute list locally on a router. To configure certificate serial number session control, an administrator may specify a specific certificate in the *value* field to be accepted or rejected based on its serial number where *name* is set to **cert-serial-not**. If the serial number of the certificate matches the serial number specified by the attribute type setting, the certificate will be rejected.

	Command or Action	Purpose
		For a full list of available AAA attribute types, execute the **show aaa attributes** command.

Step 15	**exit**
	Example:
	`Router(ca-trustpoint)# exit`
	Example:
	`Router(config-attr-list)# exit`
		Returns to global configuration mode.

Step 16	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 17	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays the components of the certificates installed on the router if the CA certificate has been authenticated.

Example

The following is a sample certificate. The OCSP-related extensions are shown using exclamation points.

```
Certificate:
        Data:
            Version: v3
            Serial Number:0x14
            Signature Algorithm:SHAwithRSA - 1.2.840.113549.1.1.4
            Issuer:CN=CA server,OU=PKI,O=Cisco Systems
            Validity:
                Not Before:Thursday, August 8, 2002 4:38:05 PM PST
                Not After:Tuesday, August 7, 2003 4:38:05 PM PST
            Subject:CN=OCSP server,OU=PKI,O=Cisco Systems
            Subject Public Key Info:
                Algorithm:RSA - 1.2.840.113549.1.1.1
                Public Key:
                    Exponent:65537
                    Public Key Modulus:(2048 bits) :
                        <snip>
            Extensions:
                Identifier:Subject Key Identifier - 2.5.29.14
                    Critical:no
                    Key Identifier:
                        <snip>
                Identifier:Authority Key Identifier - 2.5.29.35
                    Critical:no
                    Key Identifier:
                        <snip>
!                Identifier:OCSP NoCheck:- 1.3.6.1.5.5.7.48.1.5
                     Critical:no
```

```
                Identifier:Extended Key Usage:- 2.5.29.37
                     Critical:no
                     Extended Key Usage:
                     OCSPSigning
!
                Identifier:CRL Distribution Points - 2.5.29.31
                    Critical:no
                    Number of Points:1
                    Point 0
                        Distribution Point:
[URIName:ldap://CA-server/CN=CA server,OU=PKI,O=Cisco Systems]
        Signature:
            Algorithm:SHAwithRSA - 1.2.840.113549.1.1.4
            Signature:
            <snip>
```

The following example shows an excerpt of the running configuration output when adding a **match certificate override ocsp** command to the beginning of an existing sequence:

```
match certificate map3 override ocsp 5 url http://192.0.2.3/
show running-configuration
.
.
.
        match certificate map3 override ocsp 5 url http://192.0.2.3/
        match certificate map1 override ocsp 10 url http://192.0.2.1/
        match certificate map2 override ocsp 15 url http://192.0.2.2/
```

The following example shows an excerpt of the running configuration output when an existing **match certificate override ocsp** command is replaced and a trustpoint is specified to use an alternative PKI hierarchy:

```
match certificate map4 override ocsp trustpoint tp4 10 url http://192.0.2.4/newvalue
show running-configuration
.
.
.
        match certificate map3 override ocsp trustpoint tp3 5 url http://192.0.2.3/
        match certificate map1 override ocsp trustpoint tp1 10 url http://192.0.2.1/
        match certificate map4 override ocsp trustpoint tp4 10 url
http://192.0.2.4/newvalue
        match certificate map2 override ocsp trustpoint tp2 15 url http://192.0.2.2/
```

Troubleshooting Tips

If you ignored revocation check or expired certificates, you should carefully check your configuration. Verify that the certificate map properly matches either the certificate or certificates that should be allowed or the AAA checks that should be skipped. In a controlled environment, try modifying the certificate map and determine what is not working as expected.

Configuring Certificate Chain Validation

Perform this task to configure the processing level for the certificate chain path of your peer certificates.

Before you begin

- The device must be enrolled in your PKI hierarchy.

- The appropriate key pair must be associated with the certificate.

Note

- A trustpoint associated with the root CA cannot be configured to be validated to the next level.

The **chain-validation** command is configured with the **continue** keyword for the trustpoint associated with the root CA, an error message will be displayed and the chain validation will revert to the default **chain-validation**command setting.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. crypto pki trustpoint *name*
4. **chain-validation** [{**stop** | **continue**} [*parent-trustpoint*]]
5. **exit**

DETAILED STEPS

	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 ca-sub1`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**chain-validation** [{**stop** | **continue**} [*parent-trustpoint*]] **Example:** `Router(ca-trustpoint)# chain-validation continue ca-sub1`	Configures the level to which a certificate chain is processed on all certificates including subordinate CA certificates. • Use the **stop**keyword to specify that the certificate is already trusted. This is the default setting. • Use the **continue** keyword to specify that the that the subordinate CA certificate associated with the trustpoint must be validated. • The *parent-trustpoint* argument specifies the name of the parent trustpoint the certificate must be validated against.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(ca-trustpoint)# exit`	Returns to global configuration mode

Configuring CRL Autodownload

Perform this step to configure the certificate revocation list (CRL) autodownload.

Improper configuration of this feature can enable excessive CRL downloads for CRLs already cached by the device thereby halting validations because the CRL download and CRL validation cannot be executed in parallel. If a CRL is already downloaded, the downloaded CRL can be used for certificate validation without downloading additional CRLs.

If you configure the **crl-cache none** command, you cannot auto download a CRL for a trustpoint. To download the CRL, execute the **no crl cache none** command to remove the CRL cache from trustpoint. Similarly, when a CRL download is configured, you cannot enable the **crl-cache none** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki crl download url** *url* [**source-interface** *interface-name* | **vrf** *vrf-name*]
4. **crypto pki crl download trustpoint** *trustpoint-label*
5. **crypto pki crl download schedule time** *day hh*:*ss*
6. **crypto pki crl download schedule prepublish** *minutes*
7. **crypto pki crl download schedule retries** *number* **crypto pki crl download schedule retries interval** *minutes*
8. **end**
9. **crypto pki crl refresh-cache**
10. **show crypto pki crl download**
11. **show crypto pki timers**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto pki crl download url** *url* [**source-interface** *interface-name* | **vrf** *vrf-name*] **Example:** `Device(config)# crypto pki crl download url www.abc.com source-interface GigabitEthernet 1`	Specifies that the CRL auto download must fetch the CRL through the source interface or the VRF or both.
Step 4	**crypto pki crl download trustpoint** *trustpoint-label* **Example:** `Device(config)# crypto pki crl download trustpoint trp1`	Specifies that the CRL auto download must fetch the CRL distribution point (CDP) from the device certificate associated with that trustpoint.
Step 5	**crypto pki crl download schedule time** *day hh*:*ss* **Example:** `Device(config)# crypto pki crl download schedule time Monday 00:00`	Specifies the day and time when the CRL auto download must be triggered. • *time*—Indicates the exact time of the day to download the CRL, if no CRL is found. Must be specified in hour and minute format (*mm*:*ss*).
Step 6	**crypto pki crl download schedule prepublish** *minutes* **Example:** `Device(config)# crypto pki crl download schedule prepublish 720`	Time interval, in minutes, to download the CRL before the CRL expires. the default value is 0.
Step 7	**crypto pki crl download schedule retries** *number* **crypto pki crl download schedule retries interval** *minutes* **Example:** `Device(config)# crypto pki crl download schedule retries 15 interval 15 crypto pki crl download schedule retries 15 interval 15`	Specifies the time interval, in minutes, for a device to retry downloading a CRL from a CDP location if previous download attempts fail. The default number of retries is 5. • **interval** *minutes*—Time interval between retry attempts, in minutes. The default retry interval is 30 minutes.
Step 8	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 9	**crypto pki crl refresh-cache** **Example:** `Device# crypto pki crl refresh-cache`	Refreshes the CRL entries in the cache.
Step 10	**show crypto pki crl download** **Example:** `Device# show crypto pki crl download`	Displays auto download configurations.
Step 11	**show crypto pki timers** **Example:** `Device(config)# show crypto pki timers`	Displays information about the timers set for Cisco IOS for public key infrastructure.

Example

The following is a sample output from the **show crypto pki crl download** command.

```
Device# show crypto pki crl download

CRL Issuer Name:
    cn=ios
    LastUpdate: 10:38:23 IST Sep 18 2013
    NextUpdate: 16:38:23 IST Sep 18 2013

    Valid after expiry till: 16:58:23 IST Sep 18 2013

    CRL Downloaded at 12:38:23 IST Sep 18 2013

    Retrieved from CRL Distribution Point:
      ** CDP Not Published - Retrieved via SCEP

 CRL DER is 213 bytes
 CRL is stored in parsed CRL cache

CRL prepublish timer interval: 10

Parsed CRL cache current size is 213 bytes
Parsed CRL cache maximum size is 65536 bytes
```

- The field Valid after expiry till: indicates the duration for which the CRL is valid after expiry when crl cache extend is configured.
- The field CRL Downloaded at denotes the time when the CRL is downloaded.

The following is a sample output from the **show crypto pki timer** command.

```
Device# show crypto pki timers

PKI Timers
  |       13:42.564
  |       13:42.564  SESSION CLEANUP
  |       11:44.111
  |       11:44.111  CRL UPDATE cn=IOS-CA
  |       21:44.111  CRL EXPIRE cn=IOS-CA
  |    7:59:56.917  STATIC CRL DOWNLOAD
CS Timers
  |        1:44.071
  |        1:44.071  CS DB CLEANUP
  |       11:43.999  CS SHADOW CERT GENERATION
  |       21:43.883  CS CERT EXPIRE
```

The field CRL UPDATE denotes the updated timer based on the prepublish time.

Configuration Examples for Setting Up Authorization and Revocation of Certificates

Configuring and Verifying PKI AAA Authorization Examples

This section provides configuration examples of PKI AAA authorizations:

Router Configuration Example

The following **show running-config**command output shows the working configuration of a router that is set up to authorize VPN connections using the PKI Integration with AAA Server feature:

```
Router# show running-config
Building configuration...
!
version 12.3
!
hostname router7200router7200
!
aaa new-model
!
!
aaa authentication login default group tacacs+
aaa authentication login no_tacacs enable
aaa authentication ppp default group tacacs+
aaa authorization exec ACSLab group tacacs+
aaa authorization network ACSLab group tacacs+
aaa accounting exec ACSLab start-stop group tacacs+
aaa accounting network default start-stop group ACSLab
aaa session-id common
!
ip domain name example.com
!
crypto pki trustpoint EM-CERT-SERV
 enrollment url http://192.0.2.33:80
 serial-number
 crl optional
 rsakeypair STOREVPN 2048
 auto-enroll
 authorization list ACSLab
!
crypto pki certificate chain EM-CERT-SERV
 certificate 04
  30820214 3082017D A0030201 02020104 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 454D2D43 4552542D 53455256 301E170D 30343031
  31393232 30323535 5A170D30 35303131 38323230 3235355A 3030312E 300E0603
  55040513 07314437 45424434 301C0609 2A864886 F70D0109 02160F37 3230302D
  312E6772 696C2E63 6F6D3081 9F300D06 092A8648 86F70D01 01010500 03818D00
  30818902 818100BD F3B837AA D925F391 2B64DA14 9C2EA031 5A7203C4 92F8D6A8
  7D2357A6 BCC8596F A38A9B10 47435626 D59A8F2A 123195BB BE5A1E74 B1AA5AE0
  5CA162FF 8C3ACA4F B3EE9F27 8B031642 B618AE1B 40F2E3B4 F996BEFE 382C7283
  3792A369 236F8561 8748AA3F BC41F012 B859BD9C DB4F75EE 3CEE2829 704BD68F
  FD904043 0F555702 03010001 A3573055 30250603 551D1F04 1E301C30 1AA018A0
  16861468 7474703A 2F2F3633 2E323437 2E313037 2E393330 0B060355 1D0F0404
  030205A0 301F0603 551D2304 18301680 1420FC4B CF0B1C56 F5BD4C06 0AFD4E67
  341AE612 D1300D06 092A8648 86F70D01 01040500 03818100 79E97018 FB955108
  12F42A56 2A6384BC AC8E22FE F1D6187F DA5D6737 C0E241AC AAAEC75D 3C743F59
  08DEEFF2 0E813A73 D79E0FA9 D62DC20D 8E2798CD 2C1DC3EC 3B2505A1 3897330C
  15A60D5A 8A13F06D 51043D37 E56E45DF A65F43D7 4E836093 9689784D C45FD61D
  EC1F160C 1ABC8D03 49FB11B1 DA0BED6C 463E1090 F34C59E4
  quit
 certificate ca 01
  30820207 30820170 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 454D2D43 4552542D 53455256 301E170D 30333132
  31363231 34373432 5A170D30 36313231 35323134 3734325A 30173115 30130603
  55040313 0C454D2D 43455254 2D534552 5630819F 300D0609 2A864886 F70D0101
  01050003 818D0030 81890281 8100C14D 833641CF D784F516 DA6B50C0 7B3CB3C9
  589223AB 99A7DC14 04F74EF2 AAEEE8F5 E3BFAE97 F2F980F7 D889E6A1 2C726C69
  54A29870 7E7363FF 3CD1F991 F5A37CFF 3FFDD3D0 9E486C44 A2E34595 C2D078BB
  E9DE981E B733B868 AA8916C0 A8048607 D34B83C0 64BDC101 161FC103 13C06500
```

```
  22D6EE75 7D6CF133 7F1B515F 32830203 010001A3 63306130 0F060355 1D130101
  FF040530 030101FF 300E0603 551D0F01 01FF0404 03020186 301D0603 551D0E04
  16041420 FC4BCF0B 1C56F5BD 4C060AFD 4E67341A E612D130 1F060355 1D230418
  30168014 20FC4BCF 0B1C56F5 BD4C060A FD4E6734 1AE612D1 300D0609 2A864886
  F70D0101 04050003 81810085 D2E386F5 4107116B AD3AC990 CBE84063 5FB2A6B5
  BD572026 528E92ED 02F3A0AE 1803F2AE AA4C0ED2 0F59F18D 7B50264F 30442C41
  0AF19C4E 70BD3CB5 0ADD8DE8 8EF636BD 24410DF4 DB62DAFC 67DA6E58 3879AA3E
  12AFB1C3 2E27CB27 EC74E1FC AEE2F5CF AA80B439 615AA8D5 6D6DEDC3 7F9C2C79
  3963E363 F2989FB9 795BA8
  quit
!
!
crypto isakmp policy 10
 encr aes
 group 14
!
!
crypto ipsec transform-set ISC_TS_1 esp-aes esp-sha-hmac
!
crypto ipsec profile ISC_IPSEC_PROFILE_2
 set security-association lifetime kilobytes 530000000
 set security-association lifetime seconds 14400
 set transform-set ISC_TS_1
!
!
controller ISA 1/1
!
!
interface Tunnel0
 description MGRE Interface provisioned by ISC
 bandwidth 10000
 ip address 192.0.2.172 255.255.255.0
 no ip redirects
 ip mtu 1408
 ip nhrp map multicast dynamic
 ip nhrp network-id 101
 ip nhrp holdtime 500
 ip nhrp server-only
 no ip split-horizon eigrp 101
 tunnel source FastEthernet2/1
 tunnel mode gre multipoint
 tunnel key 101
 tunnel protection ipsec profile ISC_IPSEC_PROFILE_2
!
interface FastEthernet2/0
 ip address 192.0.2.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet2/1
 ip address 192.0.2.2 255.255.255.0
 duplex auto
 speed auto
!
!
tacacs-server host 192.0.2.55 single-connection
tacacs-server directed-request
tacacs-server key company lab
!
ntp master 1
!
end
```

Debug of a Successful PKI AAA Authorization Example

The following **show debugging** command output shows a successful authorization using the PKI Integration with AAA Server feature:

```
Router# show debugging
General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
Crypto PKI Trans debugging is on
Router#
May 28 19:36:11.117: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:36:12.789: CRYPTO_PKI: Found a issuer match
May 28 19:36:12.805: CRYPTO_PKI: cert revocation status unknown.
May 28 19:36:12.805: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:36:12.813: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com,
 <all>)
May 28 19:36:12.813: AAA/BIND(00000042): Bind i/f
May 28 19:36:12.813: AAA/AUTHOR (0x42): Pick method list 'ACSLab'
May 28 19:36:12.813: TPLUS: Queuing AAA Authorization request 66 for processing
May 28 19:36:12.813: TPLUS: processing authorization request id 66
May 28 19:36:12.813: TPLUS: Protocol set to None .....Skipping
May 28 19:36:12.813: TPLUS: Sending AV service=pki
May 28 19:36:12.813: TPLUS: Authorization request created for 66(POD5.example.com)
May 28 19:36:12.813: TPLUS: Using server 192.0.2.55
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT/203A4628: Started 5 sec timeout
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT: wrote entire 46 bytes request
May 28 19:36:12.813: TPLUS: Would block while reading pak header
May 28 19:36:12.817: TPLUS(00000042)/0/READ: read entire 12 header bytes (expect 27 bytes)
May 28 19:36:12.817: TPLUS(00000042)/0/READ: read entire 39 bytes response
May 28 19:36:12.817: TPLUS(00000042)/0/203A4628: Processing the reply packet
May 28 19:36:12.817: TPLUS: Processed AV cert-application=all
May 28 19:36:12.817: TPLUS: received authorization response for 66: PASS
May 28 19:36:12.817: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
May 28 19:36:12.817: CRYPTO_PKI_AAA: authorization passed
Router#
Router#
May 28 19:36:18.681: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 101: Neighbor 192.0.2.171 (Tunnel0) is
 up: new adjacency
Router#
Router# show crypto isakmp sa
dst             src             state          conn-id slot
192.0.2.22     192.0.2.102   QM_IDLE             84    0
```

Debugs of a Failed PKI AAA Authorization Example

The following **show debugging** command output shows that the router is not authorized to connect using VPN. The messages are typical of those that you might see in such a situation.

In this example, the peer username was configured as not authorized, by moving the username to a Cisco Secure ACS group called VPN_Router_Disabled in Cisco Secure ACS. The router, router7200.example.com, has been configured to check with a Cisco Secure ACS AAA server prior to establishing a VPN connection to any peer.

```
Router# show debugging
General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
```

```
  AAA Authorization debugging is on
Cryptographic Subsystem:
  Crypto PKI Trans debugging is on

Router#
May 28 19:48:29.837: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:48:31.509: CRYPTO_PKI: Found a issuer match
May 28 19:48:31.525: CRYPTO_PKI: cert revocation status unknown.
May 28 19:48:31.525: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:48:31.533: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com,
 <all>)
May 28 19:48:31.533: AAA/BIND(00000044): Bind i/f
May 28 19:48:31.533: AAA/AUTHOR (0x44): Pick method list 'ACSLab'
May 28 19:48:31.533: TPLUS: Queuing AAA Authorization request 68 for processing
May 28 19:48:31.533: TPLUS: processing authorization request id 68
May 28 19:48:31.533: TPLUS: Protocol set to None .....Skipping
May 28 19:48:31.533: TPLUS: Sending AV service=pki
May 28 19:48:31.533: TPLUS: Authorization request created for 68(POD5.example.com)
May 28 19:48:31.533: TPLUS: Using server 192.0.2.55
May 28 19:48:31.533: TPLUS(00000044)/0/NB_WAIT/203A4C50: Started 5 sec timeout
May 28 19:48:31.533: TPLUS(00000044)/0/NB_WAIT: wrote entire 46 bytes request
May 28 19:48:31.533: TPLUS: Would block while reading pak header
May 28 19:48:31.537: TPLUS(00000044)/0/READ: read entire 12 header bytes (expect 6 bytes)
May 28 19:48:31.537: TPLUS(00000044)/0/READ: read entire 18 bytes response
May 28 19:48:31.537: TPLUS(00000044)/0/203A4C50: Processing the reply packet
May 28 19:48:31.537: TPLUS: received authorization response for 68: FAIL
May 28 19:48:31.537: CRYPTO_PKI_AAA: authorization declined by AAA, or AAA server not found.
May 28 19:48:31.537: CRYPTO_PKI_AAA: No cert-application attribute found. Failing.
May 28 19:48:31.537: CRYPTO_PKI_AAA: authorization failed
May 28 19:48:31.537: CRYPTO_PKI: AAA authorization for list 'ACSLab', and user
'POD5.example.com' failed.
May 28 19:48:31.537: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.0.2.162 is
bad: certificate invalid
May 28 19:48:39.821: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:48:41.481: CRYPTO_PKI: Found a issuer match
May 28 19:48:41.501: CRYPTO_PKI: cert revocation status unknown.
May 28 19:48:41.501: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:48:41.505: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com,
 <all>)
May 28 19:48:41.505: AAA/BIND(00000045): Bind i/f
May 28 19:48:41.505: AAA/AUTHOR (0x45): Pick method list 'ACSLab'
May 28 19:48:41.505: TPLUS: Queuing AAA Authorization request 69 for processing
May 28 19:48:41.505: TPLUS: processing authorization request id 69
May 28 19:48:41.505: TPLUS: Protocol set to None .....Skipping
May 28 19:48:41.505: TPLUS: Sending AV service=pki
May 28 19:48:41.505: TPLUS: Authorization request created for 69(POD5.example.com)
May 28 19:48:41.505: TPLUS: Using server 198.168.244.55
May 28 19:48:41.509: TPLUS(00000045)/0/IDLE/63B22834: got immediate connect on new 0
May 28 19:48:41.509: TPLUS(00000045)/0/WRITE/63B22834: Started 5 sec timeout
May 28 19:48:41.509: TPLUS(00000045)/0/WRITE: wrote entire 46 bytes request
May 28 19:48:41.509: TPLUS(00000045)/0/READ: read entire 12 header bytes (expect 6 bytes)
May 28 19:48:41.509: TPLUS(00000045)/0/READ: read entire 18 bytes response
May 28 19:48:41.509: TPLUS(00000045)/0/63B22834: Processing the reply packet
May 28 19:48:41.509: TPLUS: received authorization response for 69: FAIL
May 28 19:48:41.509: CRYPTO_PKI_AAA: authorization declined by AAA, or AAA server not found.
May 28 19:48:41.509: CRYPTO_PKI_AAA: No cert-application attribute found. Failing.
May 28 19:48:41.509: CRYPTO_PKI_AAA: authorization failed
May 28 19:48:41.509: CRYPTO_PKI: AAA authorization for list 'ACSLab', and user
'POD5.example.com' failed.
May 28 19:48:41.509: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.0.2.162 is
bad: certificate invalid
Router#
Router# show crypto iskmp sa
```

```
dst             src             state          conn-id slot
192.0.2.2        192.0.2.102   MM_KEY_EXCH         95    0
```

Configuring a Revocation Mechanism Examples

This section contains the following configuration examples that can be used when specifying a revocation mechanism for your PKI:

Configuring an OCSP Server Example

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

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

Specifying a CRL and Then an OCSP Server Example

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

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

Specifying an OCSP Server Example

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

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

Disabling Nonces in Communications with the OCSP Server Example

The following example shows communications when a nonce, or a unique identifier for the OCSP request, is disabled for communications with the OCSP server:

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

Configuring a Hub Router at a Central Site for Certificate Revocation Checks Example

The following example shows a hub router at a central site that is providing connectivity for several branch offices to the central site.

The branch offices are also able to communicate directly with each other using additional IPSec tunnels between the branch offices.

The CA publishes CRLs on an HTTP server at the central site. The central site checks CRLs for each peer when setting up an IPSec tunnel with that peer.

The example does not show the IPSec configuration--only the PKI-related configuration is shown.

Home Office Hub Configuration

```
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
```

Central Site Hub Router

```
Router# show crypto ca certificate
Certificate
  Status: Available
  Certificate Serial Number: 2F62BE14000000000CA0
  Certificate Usage: General Purpose
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    Name: Central VPN Gateway
    cn=Central VPN Gateway
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 00:43:26 GMT Sep 26 2003
    end   date: 00:53:26 GMT Sep 26 2004
    renew date: 00:00:00 GMT Jan 1 1970
  Associated Trustpoints: VPN-GW
CA Certificate
  Status: Available
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
  Certificate Usage: Signature
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    cn=Central Certificate Authority
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017
  Associated Trustpoints: VPN-GW
```

Trustpoint on the Branch Office Router

```
crypto pki trustpoint home-office
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
```

```
ip-address none
 subject-name o=Home Office Inc,cn=Branch 1
 revocation-check crl
```

A certificate map is entered on the branch office router.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
branch1(config)# crypto pki certificate map central-site 10
branch1(ca-certificate-map)#
```

The output from the **show certificate** command on the central site hub router shows that the certificate was issued by the following:

```
cn=Central Certificate Authority
o=Home Office Inc
```

These two lines are combined into one line using a comma (,) to separate them, and the original lines are added as the first criteria for a match.

```
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home Office
 Inc
!The above line wrapped but should be shown on one line with the line above it.
```

The same combination is done for the subject name from the certificate on the central site router (note that the line that begins with “Name:” is not part of the subject name and must be ignored when creating the certificate map criteria). This is the subject name to be used in the certificate map.

cn=Central VPN Gateway

o=Home Office Inc

```
Router (ca-certificate-map)# subject-name eq cn=central vpn gateway, o=home office inc
```

Now the certificate map is added to the trustpoint that was configured earlier.

```
Router (ca-certificate-map)# crypto pki trustpoint home-office
Router (ca-trustpoint)# match certificate central-site skip revocation-check
Router (ca-trustpoint)# exit
Router (config)# exit
```

The configuration is checked (most of configuration is not shown).

```
Router# write term
!Many lines left out
.
.
.
crypto pki trustpoint home-office
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Branch 1
 revocation-check crl
 match certificate central-site skip revocation-check
!
!
crypto pki certificate map central-site 10
 issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
```

```
 subject-name eq cn = central vpn gateway, o = home office inc
!many lines left out
```

Note that the issuer-name and subject-name lines have been reformatted to make them consistent for later matching with the certificate of the peer.

If the branch office is checking the AAA, the trustpoint will have lines similar to the following:

```
crypto pki trustpoint home-office
 auth list allow_list
 auth user subj commonname
```

After the certificate map has been defined as was done above, the following command is added to the trustpoint to skip AAA checking for the central site hub.

```
match certificate central-site skip authorization-check
```

In both cases, the branch site router has to establish an IPSec tunnel to the central site to check CRLs or to contact the AAA server. However, without the **match certificate**command and **central-site skip authorization-check (argument and keyword)**, the branch office cannot establish the tunnel until it has checked the CRL or the AAA server. (The tunnel will not be established unless the **match certificate**command and **central-site skip authorization-check** argument and keyword are used.)

The **match certificate** command and **allow expired-certificate** keyword would be used at the central site if the router at a branch site had an expired certificate and it had to establish a tunnel to the central site to renew its certificate.

Trustpoint on the Central Site Router

```
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
```

Trustpoint on the Branch 1 Site Router

```
Router# show crypto ca certificate
Certificate
  Status: Available
  Certificate Serial Number: 2F62BE14000000000CA0
  Certificate Usage: General Purpose
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    Name: Branch 1 Site
    cn=Branch 1 Site
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 00:43:26 GMT Sep 26 2003
    end   date: 00:53:26 GMT Oct 3 2003
    renew date: 00:00:00 GMT Jan 1 1970
  Associated Trustpoints: home-office
CA Certificate
```

```
  Status: Available
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
  Certificate Usage: Signature
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    cn=Central Certificate Authority
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017
  Associated Trustpoints: home-office
```

A certificate map is entered on the central site router.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router (config)# crypto pki certificate map branch1 10
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home Office
 Inc
!The above line wrapped but should be part of the line above it.
Router (ca-certificate-map)# subject-name eq cn=Brahcn 1 Site,o=home office inc
```

The certificate map is added to the trustpoint.

```
Router (ca-certificate-map)# crypto pki trustpoint VPN-GW
Router (ca-trustpoint)# match certificate branch1 allow expired-certificate
Router (ca-trustpoint)# exit
Router (config) #exit
```

The configuration should be checked (most of the configuration is not shown).

```
Router# write term
!many lines left out
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
 match certificate branch1 allow expired-certificate
!
!
crypto pki certificate map central-site 10
 issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
 subject-name eq cn = central vpn gateway, o = home office inc
! many lines left out
```

The **match certificate**command and **branch1 allow expired-certificate** (argument and keyword) and the certificate map should be removed as soon as the branch router has a new certificate.

Configuring Certificate Authorization and Revocation Settings Examples

This section contains the following configuration examples that can be used when specifying a CRL cache control setting or certificate serial number session control:

Configuring CRL Cache Control

The following example shows how to disable CRL caching for all CRLs associated with the CA1 trustpoint:

```
crypto pki trustpoint CA1
 enrollment url http://CA1:80
 ip-address FastEthernet0/0
 crl query ldap://ldap_CA1
 revocation-check crl
 crl-cache none
```

The current CRL is still cached immediately after executing the example configuration shown above:

Router# **show crypto pki crls**

```
CRL Issuer Name:
    cn=name Cert Manager,ou=pki,o=example.com,c=US
    LastUpdate: 18:57:42 GMT Nov 26 2005
    NextUpdate: 22:57:42 GMT Nov 26 2005
    Retrieved from CRL Distribution Point:
      ldap://ldap.example.com/CN=name Cert Manager,O=example.com
```

When the current CRL expires, a new CRL is then downloaded to the router at the next update. The **crl-cache none**command takes effect and all CRLs for the trustpoint are no longer cached; caching is disabled. You can verify that no CRL is cached by executing the **show crypto pki crls** command. No output will be shown because there are no CRLs cached.

The following example shows how to configure the maximum lifetime of 2 minutes for all CRLs associated with the CA1 trustpoint:

```
crypto pki trustpoint CA1
 enrollment url http://CA1:80
 ip-address FastEthernet0/0
 crl query ldap://ldap_CA1
 revocation-check crl
 crl-cache delete-after 2
```

The current CRL is still cached immediately after executing the example configuration above for setting the maximum lifetime of a CRL:

Router# **show crypto pki crls**

```
CRL Issuer Name:
    cn=name Cert Manager,ou=pki,o=example.com,c=US
    LastUpdate: 18:57:42 GMT Nov 26 2005
    NextUpdate: 22:57:42 GMT Nov 26 2005
    Retrieved from CRL Distribution Point:
      ldap://ldap.example.com/CN=name Cert Manager,O=example.com
When the current CRL expires, a new CRL is downloaded to the router at the next update and
 the crl-cache delete-after
command takes effect. This newly cached CRL and all subsequent CRLs will be deleted after
a maximum lifetime of 2 minutes.
You can verify that the CRL will be cached for 2 minutes by executing the show crypto pki
crls
 command. Note that the NextUpdate time is 2 minutes after the LastUpdate time.
```

Router# **show crypto pki crls**

```
CRL Issuer Name:
    cn=name Cert Manager,ou=pki,o=example.com,c=US
```

```
    LastUpdate: 22:57:42 GMT Nov 26 2005

    NextUpdate: 22:59:42 GMT Nov 26 2005
    Retrieved from CRL Distribution Point:
```

ldap://ldap.example.com/CN=name Cert Manager,O=example.com

Configuring Certificate Serial Number Session Control

The following example shows the configuration of certificate serial number session control using a certificate map for the CA1 trustpoint:

```
crypto pki trustpoint CA1
 enrollment url http://CA1
 chain-validation stop
 crl query ldap://ldap_server
 revocation-check crl
 match certificate crl
!
crypto pki certificate map crl 10
 serial-number co 279d
```

Note If the *match-criteria* value is set to **eq** (equal) instead of **co** (contains), the serial number must match the certificate map serial number exactly, including any spaces.

The following example shows the configuration of certificate serial number session control using AAA attributes. In this case, all valid certificates will be accepted if the certificate does not have the serial number "4ACA."

```
crypto pki trustpoint CA1
 enrollment url http://CA1
 ip-address FastEthernet0/0
 crl query ldap://ldap_CA1
 revocation-check crl
 aaa new-model
!
aaa attribute list crl
attribute-type aaa-cert-serial-not 4ACA
```

The server log shows that the certificate with the serial number "4ACA" was rejected. The certificate rejection is shown using exclamation points.

```
.
.
.
Dec 3 04:24:39.051: CRYPTO_PKI: Trust-Point CA1 picked up
Dec 3 04:24:39.051: CRYPTO_PKI: locked trustpoint CA1, refcount is 1
Dec 3 04:24:39.051: CRYPTO_PKI: unlocked trustpoint CA1, refcount is 0
Dec 3 04:24:39.051: CRYPTO_PKI: locked trustpoint CA1, refcount is 1
Dec 3 04:24:39.135: CRYPTO_PKI: validation path has 1 certs
Dec 3 04:24:39.135: CRYPTO_PKI: Found a issuer match
Dec 3 04:24:39.135: CRYPTO_PKI: Using CA1 to validate certificate
Dec 3 04:24:39.135: CRYPTO_PKI: Certificate validated without revocation check
Dec 3 04:24:39.135: CRYPTO_PKI: Selected AAA username: 'PKIAAA'
Dec 3 04:24:39.135: CRYPTO_PKI: Anticipate checking AAA list:'CRL'
Dec 3 04:24:39.135: CRYPTO_PKI_AAA: checking AAA authorization (CRL, PKIAAA-L1, <all>)
Dec 3 04:24:39.135: CRYPTO_PKI_AAA: pre-authorization chain validation status (0x4)
```

```
Dec 3 04:24:39.135: AAA/BIND(00000021): Bind i/f
Dec 3 04:24:39.135: AAA/AUTHOR (0x21): Pick method list 'CRL'
.
.
.
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
!
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-serial-not" = "4ACA")
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: cert-serial doesn't match ("4ACA" != "4ACA")
!
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: post-authorization chain validation status (0x7)
!
Dec 3 04:24:39.175: CRYPTO_PKI: AAA authorization for list 'CRL', and user 'PKIAAA' failed.
Dec 3 04:24:39.175: CRYPTO_PKI: chain cert was anchored to trustpoint CA1, and chain
validation result was: CRYPTO_PKI_CERT_NOT_AUTHORIZED
!
Dec 3 04:24:39.175: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.0.2.43 is bad:
 certificate invalid
Dec 3 04:24:39.175: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Main mode failed with peer
at 192.0.2.43
.
.
.
```

Configuring Certificate Chain Validation Examples

This section contains the following configuration examples that can be used to specify the level of certificate chain processing for your device certificates:

Configuring Certificate Chain Validation from Peer to Root CA

In the following configuration example, all of the certificates will be validated--the peer, SubCA11, SubCA1, and RootCA certificates.

```
crypto pki trustpoint RootCA
 enrollment terminal
 chain-validation stop
 revocation-check none
 rsakeypair RootCA
crypto pki trustpoint SubCA1
 enrollment terminal
 chain-validation continue RootCA
 revocation-check none
 rsakeypair SubCA1
crypto pki trustpoint SubCA11
 enrollment terminal
 chain-validation continue SubCA1
 revocation-check none
 rsakeypair SubCA11
```

Configuring Certificate Chain Validation from Peer to Subordinate CA

In the following configuration example, the following certificates will be validated--the peer and SubCA1 certificates.

```
crypto pki trustpoint RootCA
 enrollment terminal
 chain-validation stop
```

```
 revocation-check none
 rsakeypair RootCA
crypto pki trustpoint SubCA1
 enrollment terminal
 chain-validation continue RootCA
 revocation-check none
 rsakeypair SubCA1
crypto pki trustpoint SubCA11
 enrollment terminal
 chain-validation continue SubCA1
 revocation-check none
 rsakeypair SubCA11
```

Configuring Certificate Chain Validation Through a Gap

In the following configuration example, SubCA1 is not in the configured Cisco IOS hierarchy but is expected to have been supplied in the certificate chain presented by the peer.

If the peer supplies the SubCA1 certificate in the presented certificate chain, the following certificates will be validated--the peer, SubCA11, and SubCA1 certificates.

If the peer does not supply the SubCA1 certificate in the presented certificate chain, the chain validation will fail.

```
crypto pki trustpoint RootCA
 enrollment terminal
 chain-validation stop
 revocation-check none
 rsakeypair RootCA
crypto pki trustpoint SubCA11
 enrollment terminal
 chain-validation continue RootCA
 revocation-check none
 rsakeypair SubCA11
```

Additional References

Related Documents

Related Topic	Document Title
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
Overview of PKI, including RSA keys, certificate enrollment, and CAs	“Cisco IOS PKI Overview: Understanding and Planning a PKI” module
RSA key generation and deployment	“Deploying RSA Keys Within a PKI” module
Certificate enrollment: supported methods, enrollment profiles, configuration tasks	“Configuring Certificate Enrollment for a PKI” module
Cisco IOS certificate server overview information and configuration tasks	“Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment ” module
Recommended cryptographic algorithms	*Next Generation Encryption*

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 2: 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.