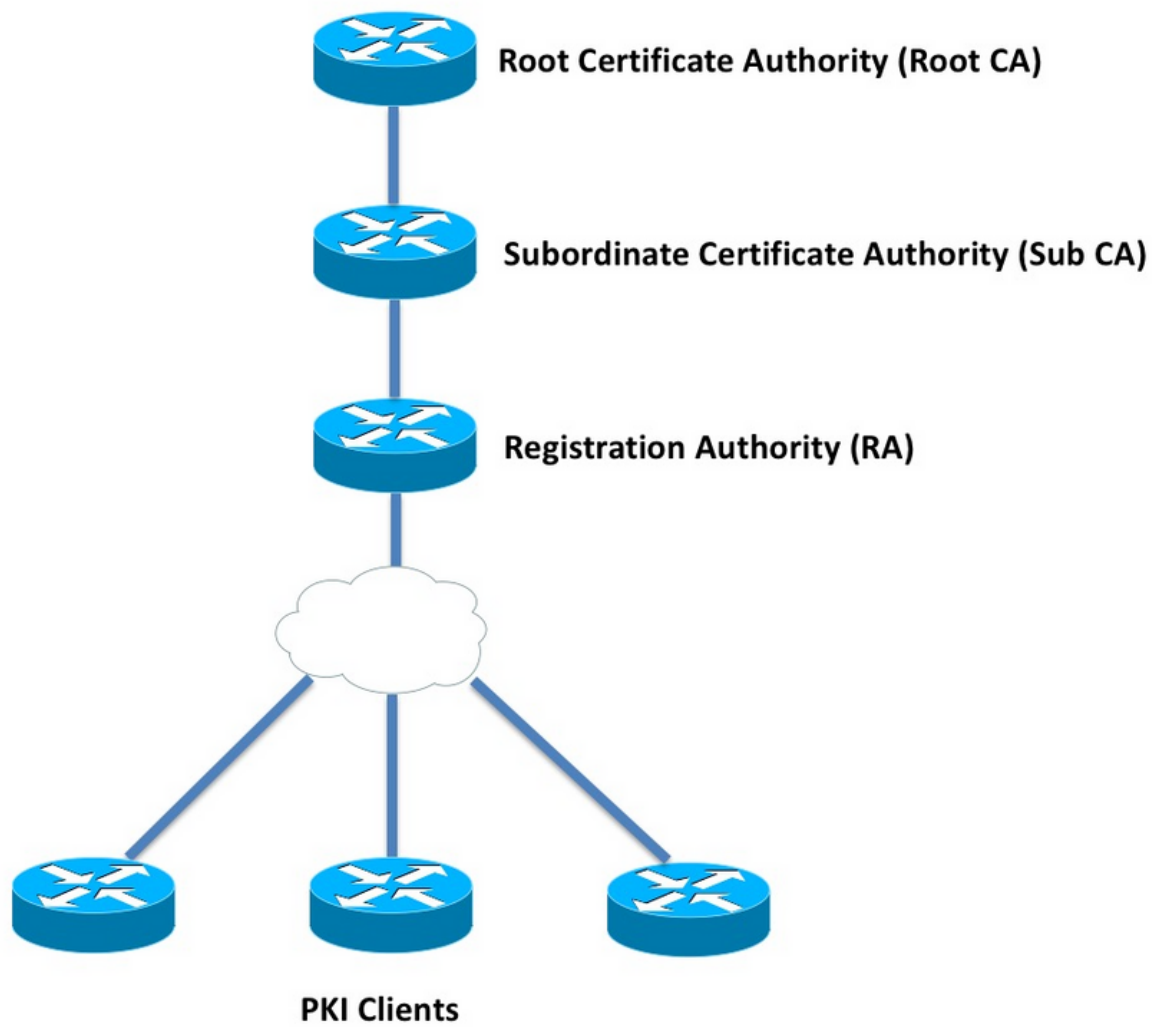
Guida alla distribuzione di IOS PKI: Progettazione e installazione iniziali

Sommario

Introduzione

In questo documento vengono descritte in dettaglio le funzionalità del server e del client PKI IOS. Vengono fornite considerazioni sulla progettazione iniziale e sulla distribuzione della PKI IOS.

Infrastruttura PKI



Autorità di certificazione

L'autorità di certificazione (CA), nota anche come server PKI in tutto il documento, è un'entità

attendibile che emette certificati. La PKI è basata sull'attendibilità e la gerarchia di attendibilità inizia dall'Autorità di certificazione (CA radice). Poiché la CA radice si trova nella parte superiore della gerarchia, dispone di un certificato autofirmato.

Autorità di certificazione subordinata

Nella gerarchia di attendibilità PKI tutte le autorità di certificazione sotto la radice sono note come Autorità di certificazione subordinate (CA subordinata). Evidentemente, un certificato CA secondaria viene rilasciato dalla CA, che si trova al livello superiore.

La PKI non impone alcun limite al numero di CA secondarie in una determinata gerarchia. Tuttavia, in una distribuzione aziendale con più di tre livelli di autorità di certificazione può risultare difficile da gestire.

Autorità registrazione

La PKI definisce un'autorità di certificazione speciale denominata Autorità di registrazione (RA, Registration Authority), responsabile dell'autorizzazione dei client PKI a effettuare la registrazione a una determinata CA secondaria o CA radice. L'Autorità di registrazione non rilascia certificati ai client PKI, ma decide a quale client PKI può o non può essere rilasciato un certificato dalla CA secondaria o dalla CA radice.

Il ruolo principale di un'Autorità di certificazione consiste nell'offload della convalida delle richieste di certificati client di base dall'Autorità di certificazione e nella protezione dell'Autorità di certificazione dall'esposizione diretta ai client. In questo modo, RA si trova tra i client PKI e la CA, proteggendo così la CA da qualsiasi tipo di attacchi Denial of Service.

Client PKI

Qualsiasi dispositivo che richieda un certificato basato su una coppia di chiavi pubblica-privata residente per provare la propria identità ad altri dispositivi è noto come client PKI.

Un client PKI deve essere in grado di generare o archiviare una coppia di chiavi pubblico-privato come RSA, DSA o ECDSA.

Un certificato è una prova dell'identità e della validità di una determinata chiave pubblica, a condizione che la chiave privata corrispondente esista nel dispositivo.

Server IOS PKI

Tabella 1. Evoluzione della funzionalità server PKI IOS

Funzionalità	IOS [ISR-G1, ISR-G2]	IOS-XE [ASR1K, ISR4K]
Server CA/PKI IOS	12.3(4)T	XE 3.14.0 / 15.5(1)S
Rollover del certificato del server PKI IOS	12.4(1)T	XE 3.14.0 / 15.5(1)S
IOS PKI HA	15.0(1)M	NA [Ridondanza inter-RP implicita disponibile]

IOS RA per CA di terze parti	15.1(3)T	XE 3.14.0 / 15.5(1)S

Prima di accedere alla configurazione del server PKI, l'amministratore deve comprendere questi concetti fondamentali.

Origine di tempo autorevole

Uno dei fondamenti dell'infrastruttura PKI è il Tempo. L'orologio di sistema indica se un certificato è valido o meno. Pertanto, in IOS, l'orologio deve essere reso autorevole o affidabile. Senza una fonte di tempo autorevole, il server PKI potrebbe non funzionare come previsto ed è consigliabile rendere autorevole l'orologio su IOS utilizzando questi metodi:

NTP (Network Time Protocol)

La sincronizzazione dell'orologio di sistema con un server di riferimento orario è l'unico modo efficace per rendere attendibile l'orologio di sistema. Un router IOS può essere configurato come client NTP su un server NTP noto e stabile nella rete:

```
configure terminal
ntp server <NTP Server IP address>
ntp source <source interface name>
ntp update-calendar

!! optional, if the NTP Server requires the clients to authenticate themselves
ntp authenticate
ntp authentication-key 1 md5 <key>

!! optionally an access-list can be configured to restrict time-updates from a specifc NTP
server
access-list 1 permit <NTP Server IP address>
ntp access-group peer 1
```

IOS può anche essere configurato come server NTP, che contrassegnerà l'orologio di sistema locale come autorevole. In una distribuzione PKI su piccola scala, il server PKI può essere configurato come server NTP per i client PKI:

```
configure terminal
ntp master <stratum-number>

!! optionally, NTP authentication can be enforced
ntp authenticate
ntp authentication-key 1 md5 <key-1>
ntp authentication-key 2 md5 <key-2>
ntp authentication-key 2 md5 <key-2>
ntp trusted-key 1 - 3

!! optinally, an access-list can be configured to restrict NTP clients
!! first allow the local router to synchronize with the local time-server
access-list 1 permit 127.127.7.1
ntp access-group peer 1
```

```
!! define an access-list to which the local time-server will serve time-synchronization services
access-list 2 permit <NTP-Client-IP>
ntp access-group serve-only 2
```

Contrassegno dell'orologio hardware come attendibile

In IOS, l'orologio hardware può essere contrassegnato come autorevole utilizzando:

```
config terminal
clock calendar-valid
```

Questa impostazione può essere configurata insieme all'NTP e la ragione principale per fare ciò è mantenere l'orologio di sistema autorevole quando un router viene ricaricato, ad esempio a causa di un'interruzione dell'alimentazione, e i server NTP non sono raggiungibili. In questa fase, i timer PKI smetteranno di funzionare, con conseguenti errori di rinnovo/rollover dei certificati. **calendario orologio-valido** agisce come una salvaguardia in tali situazioni.

Durante la configurazione, è importante tenere presente che l'orologio di sistema non sarà sincronizzato se la batteria di sistema si esaurisce e che il PKI inizierà a considerare attendibile un orologio non sincronizzato. Tuttavia, è relativamente più sicuro configurarlo, piuttosto che non avere una fonte di tempo autorevole.

> **Nota: clock calendar-valid** command è stato aggiunto in IOS-XE versione XE 3.10.0 / 15.3(3)S in avanti.

Nome host e nome di dominio

Si consiglia di configurare un nome host e un nome di dominio in Cisco IOS come uno dei primi passaggi prima di configurare qualsiasi servizio correlato a PKI. Il nome host del router e il nome di dominio vengono utilizzati negli scenari seguenti:

- Il nome della coppia di chiavi RSA predefinita viene derivato dalla combinazione del nome host e del nome di dominio
- Durante la registrazione di un certificato, il nome soggetto predefinito è costituito dall'attributo hostname e da un nome non strutturato, ovvero hostname e domain-name uniti.

Per quanto riguarda il server PKI, il nome host e il nome di dominio non vengono utilizzati:

- Il nome della coppia di chiavi predefinita sarà uguale al nome del server PKI
- Il nome soggetto predefinito è costituito da CN, che è lo stesso nome del server PKI.

In generale, è consigliabile configurare un nome host e un nome di dominio appropriati.

```
config terminal
hostname <string>
ip domain name <domain>
```

Server HTTP

Il server PKI IOS è abilitato solo se è abilitato il server HTTP. È importante notare che, se il server PKI è disabilitato a causa della disabilitazione del server HTTP, può continuare a concedere richieste offline [tramite terminale]. Per elaborare le richieste SCEP e inviare le risposte SCEP è necessaria la funzionalità del server HTTP.

Il server HTTP IOS è abilitato tramite:

```
ip http server
```

La porta predefinita del server HTTP può inoltre essere modificata da 80 a qualsiasi numero di porta valido utilizzando:

```
ip http port 8080
```

HTTP Max-connection

Durante la distribuzione di IOS come server PKI tramite SCEP, uno dei colli di bottiglia è rappresentato dal numero massimo di connessioni HTTP simultanee e dalla media delle connessioni HTTP al minuto.
Al momento, il numero massimo di connessioni simultanee su un server HTTP IOS è limitato a 5 per impostazione predefinita e può essere aumentato a 16, il che è altamente consigliato in una distribuzione di media scala:

```
ip http max-connections 16
```

Le seguenti installazioni di IOS consentono un massimo di connessioni HTTP simultanee fino a 1000:

- UniversalK9 IOS con licenza Cuck9

La CLI viene modificata automaticamente in modo da accettare un argomento numerico compreso tra 1 e 1000

```
ip http max-connections 1000
```

Il server HTTP IOS consente 80 connessioni al minuto [580] nel caso di versioni IOS in cui il numero massimo di sessioni simultanee HTTP può essere aumentato a 1000] e quando questo limite viene raggiunto in un minuto, il listener HTTP IOS avvia la limitazione delle connessioni HTTP in ingresso arrestando il listener per 15 secondi. In questo modo le richieste di connessione client vengono ignorate a causa del **raggiungimento del limite della coda di connessione** TCP. Ulteriori informazioni sono disponibili qui

coppia di chiavi RSA

La coppia di chiavi RSA per la funzionalità del server PKI su IOS può essere generata automaticamente o manualmente.
Durante la configurazione di un server PKI, IOS crea automaticamente un trust point con lo stesso

nome del server PKI per archiviare il certificato del server PKI.

Generazione manuale della coppia di chiavi RSA del server PKI:

Passaggio 1. Creare una coppia di chiavi RSA con lo stesso nome del server PKI:

```
crypto key generate rsa general-keys label <LABEL> modulus 2048
```

Passaggio 2. Prima di abilitare il server PKI, modificare il trust tra server PKI:

```
crypto pki trustpoint <PKI-SERVER-Name>
  rsakeypair <LABEL>
```

> **Nota:** Il valore del modulo di coppia di chiavi RSA indicato nel trust point del server PKI non viene preso in considerazione fino a quando non viene utilizzato IOS versione 15.4(3)M4. Si tratta di un'avvertenza nota. Il modulo chiave predefinito è 1024 bit.

Coppia di chiavi RSA del server PKI che genera automaticamente:

Quando si abilita il server PKI, IOS genera automaticamente una coppia di chiavi RSA con lo stesso nome del server PKI e le dimensioni del modulo di chiavi sono pari a 1024 bit.

A partire dalla versione 15.4(3)M5 di IOS, questa configurazione crea una coppia di chiavi RSA con <LABEL> come nome e la forza della chiave sarà basata sul modulo <MOD> definito.

```
crypto pki trustpoint <PKI-SERVER-Name>
rsakeypair <LABEL> <MOD>
```

Spoiler

Il server PKI IOS CSCuu73408 deve consentire dimensioni della chiave non predefinite per il certificato di rollover.

Il server PKI IOS CSCuu73408 deve consentire dimensioni della chiave non predefinite per il certificato di rollover.

L'attuale standard di settore prevede l'utilizzo di almeno 2048 bit di coppia di chiavi RSA.

Considerazione timer di rollover automatico

Al momento, il server PKI IOS non genera un certificato di rollover per impostazione predefinita e deve essere abilitato in modo esplicito nel server PKI utilizzando il comando **auto-rollover <days-before-expendy>**. Ulteriori informazioni sul rollover dei certificati sono illustrate in

Questo comando specifica quanti giorni prima della scadenza del certificato del server PKI/CA se IOS crea un certificato CA di rollover. Il certificato CA di rollover viene attivato alla scadenza del certificato CA attivo corrente. Il valore predefinito è 30 giorni. Questo valore deve essere

impostato su un valore ragionevole in base alla durata del certificato CA e ciò influisce sulla configurazione del timer di registrazione automatica nel client PKI.

> **Nota:** Il timer di rollover automatico deve essere sempre attivato prima della registrazione automatica del timer sul client durante il rollover dei certificati CA e client [noto come]

Considerazioni su CRL

L'infrastruttura PKI di IOS supporta due modalità di distribuzione di CRL:

Pubblica CRL su un server HTTP

Èpossibile configurare il server PKI IOS in modo che pubblichi il file CRL in una posizione specifica in un server HTTP utilizzando questo comando nel server PKI:

```
crypto pki server <PKI-SERVER-Name>
 database crl publish <URL>
```

Èinoltre possibile configurare il server PKI in modo da incorporare il percorso CRL in tutti i certificati del client PKI utilizzando questo comando nel server PKI:

```
crypto pki server <PKI-SERVER-Name>
cdp-url <CRL file location>
```

Metodo GetCRL SCEP

Il server PKI IOS memorizza automaticamente il file CRL nella posizione del database specifica, che per impostazione predefinita è nvram, ed è consigliabile conservare una copia su un server SCP/FTP/TFTP utilizzando questo comando nel server PKI:

```
crypto pki server <PKI-SERVER-Name>
database url <URL>
or
database crl <URL>
```

Per impostazione predefinita, il server PKI IOS non incorpora il percorso CDP nei certificati del client PKI. Se i client di Infrastruttura a chiave pubblica di IOS sono configurati per eseguire il controllo di revoca, ma il certificato da convalidare non include un CDP incorporato e il trust point CA di convalida è configurato con il percorso CA (utilizzando http://<CA-Server-IP o FQDN>), per impostazione predefinita IOS torna al metodo GetCRL basato su SCEP.
SCEP GetCRL esegue il recupero CRL eseguendo HTTP GET su questo URL:

```
http:://<CA-Server-IP/FQDN>/cgi-bin/pkiclient.exe?operation=GetCRL
```

> **Nota:** Nella CLI di IOS, prima di immettere **?**, premere **Ctrl + V** key-sequence.

Il server PKI IOS può anche incorporare questo URL come percorso CDP. Il vantaggio è duplice:

- Garantisce che tutti i client PKI basati su SCEP non IOS possano eseguire il recupero CRL.
- Senza un CDP incorporato, i messaggi di richiesta IOS SCEP GetCRL vengono firmati (utilizzando un certificato autofirmato temporaneo) come definito nella bozza SCEP. Tuttavia, non è necessario firmare le richieste di recupero CRL e, incorporando l'URL CDP per il metodo GetCRL, è possibile evitare di firmare le richieste CRL.

Durata del CRL

La durata CRL del server PKI IOS può essere controllata utilizzando questo comando in Server PKI:

```
crypto pki server <PKI-SERVER-Name>
 lifetime crl <0 - 360>
```

Il valore è espresso in ore. Per impostazione predefinita, la durata del CRL è di 6 ore. A seconda della frequenza di revoca dei certificati, l'ottimizzazione della durata della CRL su un valore ottimale aumenta le prestazioni di recupero della CRL nella rete.

Considerazioni sul database

Il server PKI IOS utilizza nvram come posizione predefinita del database ed è consigliabile utilizzare un server FTP o TFTP o SCP come posizione del database. Per impostazione predefinita, il server IOS PKI crea due file:

- <Server-Name>.ser - Contiene l'ultimo numero di serie emesso dalla CA in formato esadecimale. Il file è in formato testo normale e contiene le seguenti informazioni:
  db_version = 1
  last_serial = 0x4
- <Nome-Server>.crl - File CRL con codifica DER pubblicato dalla CA

Il server IOS PKI memorizza le informazioni nel database a 3 livelli configurabili:

- Minimo: questo è il livello predefinito e a questo livello non viene creato alcun file nel database e pertanto sul server CA non sono disponibili informazioni relative ai certificati client concessi in passato.
- Nomi: a questo livello, il server IOS PKI crea un file denominato <Numero-Serie>.cnm per ogni certificato client emesso, in cui il nome <Numero-Serie> si riferisce al numero di serie del certificato client emesso. Questo file cnm contiene il nome del soggetto e la data di scadenza del certificato client.
- Completo - A questo livello, il server IOS PKI crea due file per ogni certificato client emesso:
- <Numero di serie>.cnm

- <Numero-Serie>.crt

in questo caso, il file crt è il file del certificato client, codificato in DER.

Questi punti sono importanti:

- Prima di emettere un certificato client, il server PKI IOS fa riferimento a <Nome-server>.ser per determinare e derivare il numero di serie del certificato.
- Se il livello Database è impostato su Nomi o Completo, è necessario scrivere nel database <Numero-Serie>.cnm e <Numero-Serie>.crt prima di inviare il certificato concesso/emesso al client
- Se l'URL del database è impostato su Nomi o Completo, l'URL del database deve disporre di spazio sufficiente per il salvataggio dei file. Si consiglia pertanto di configurare un file server esterno [FTP o TFTP o SCP] come URL del database.
- Se l'URL del database esterno è configurato, è assolutamente necessario verificare che il file server sia raggiungibile durante il processo di concessione dei certificati. In caso contrario, il server CA verrebbe contrassegnato come disabilitato. È inoltre necessario un intervento manuale per riportare online il server CA.

Archivio database

Durante la distribuzione di un server PKI, è importante considerare gli scenari di errore ed essere preparati in caso di errore hardware. Esistono due modi per raggiungere questo obiettivo:

1. Ridondanza
   In questo caso, due dispositivi o unità di elaborazione agiscono come standby attivo per fornire ridondanza.
   Èpossibile ottenere un'elevata disponibilità del server PKI IOS utilizzando due router ISR abilitati per HSRP [ISR G1 e ISR G2], come spiegato in

   I sistemi basati su IOS XE [ISR4K e ASR1k] non dispongono dell'opzione di ridondanza del dispositivo. Tuttavia, in ASR1k la ridondanza Inter-RP è disponibile per impostazione predefinita.
2. Archiviazione dei file e della coppia di chiavi del server CA
   IOS fornisce una funzionalità per archiviare la coppia di chiavi del server PKI e il certificato. L'archiviazione può essere eseguita utilizzando due tipi di file:
   PEM - IOS crea file in formato PEM per archiviare la chiave pubblica RSA, la chiave privata RSA crittografata, il certificato del server CA. Coppia di chiavi e certificati di rollover archiviati automaticamentePKCS12 - IOS crea un singolo file PKCS12 contenente il certificato del server CA e la corrispondente chiave privata RSA crittografata utilizzando una password.

Èpossibile abilitare l'archiviazione del database utilizzando questo comando nel server PKI:

```
crypto pki server <PKI-SERVER-Name>
 database archive {pkcs12 | pem} password <password>
```

Èinoltre possibile archiviare i file archiviati in un server separato, possibilmente utilizzando un protocollo di protezione (SCP, Secure Protocol) utilizzando il comando seguente nel server PKI:

```
crypto pki server <PKI-SERVER-Name>
```

```
 database url {p12 | pem} <URL>
```
Di tutti i file del database, ad eccezione dei file archiviati e del file Ser, tutti gli altri file sono in testo non crittografato e non rappresentano una minaccia reale in caso di perdita, pertanto possono essere archiviati su un server separato senza incorrere in un sovraccarico durante la scrittura dei file, ad esempio un server TFTP.

IOS come CA secondaria

Per impostazione predefinita, il server IOS PKI svolge il ruolo di CA radice. Per configurare un server PKI subordinato (CA secondaria), attivare questo comando nella sezione Configurazione server PKI (prima di attivare il server PKI):

```
crypto pki server <Sub-PKI-SERVER-Name>
 mode sub-cs
```
In questo modo configurare l'URL della CA radice nel trust point del server PKI:

```
crypto pki trustpoint <Sub-PKI-SERVER-Name>
 enrollment url <Root-CA URL>
```
L'attivazione di questo server PKI attiva gli eventi seguenti:

- Il trust point del server PKI viene autenticato per installare il certificato CA radice.
- Dopo l'autenticazione della CA radice, IOS genera un CSR per la CA subordinata [vincolo di base x509 contenente il flag CA:TRUE] e lo invia alla CA radice

Indipendentemente dalla modalità di concessione configurata nella CA radice, IOS inserisce le richieste di certificati CA (o RA) nella coda in sospeso. Un amministratore deve concedere manualmente i certificati CA.
Per visualizzare la richiesta di certificato in sospeso e l'ID della richiesta:

```
show crypto pki server <Server-Name> requests
```

Per concedere la richiesta:

```
crypto pki server <Server-Name> grant <request-id>
```

- Utilizzando questo comando, la successiva operazione SCEP POLL (GetCertInitial) scarica il certificato CA secondaria e lo installa sul router, abilitando il server PKI subordinato

IOS come RA

Èpossibile configurare il server PKI IO come Autorità registrazione per una determinata CA subordinata o radice. Per configurare il server PKI come Autorità di registrazione, abilitare innanzitutto questo comando nella sezione Configurazione server PKI (prima di abilitare il server PKI):

```
crypto pki server <RA-SERVER-Name>
 mode ra
```
In seguito, configurare l'URL della CA nel trust point del server PKI. Indica la CA protetta dall'Autorità registrazione:

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

Un'autorità di registrazione non rilascia certificati, pertanto la configurazione **del nome dell'autorità emittente** nell'Autorità registrazione non è necessaria e non è valida anche se configurata. Il nome del soggetto di un'Autorità registrazione è configurato nel trust point dell'Autorità registrazione utilizzando il comando **subject-name**. È importante configurare **OU = ioscs RA** come parte del nome del soggetto affinché la CA IOS identifichi l'RA IOS, ossia identifichi le richieste di certificato autorizzate dall'RA IOS.

IOS può fungere da autorità di registrazione per le CA di terze parti, ad esempio le CA Microsoft, e per mantenere la compatibilità è necessario abilitare l'Autorità di registrazione IOS utilizzando questo comando nella sezione di configurazione del server PKI (prima di abilitare il server PKI):

```
mode ra transparent
```

Nella modalità predefinita RA, IOS firma le richieste client [PKCS#10] utilizzando il certificato RA. Questa operazione indica al server IOS PKI che la richiesta di certificato è stata autorizzata da un Autorità registrazione integrità.

Con la modalità RA trasparente, IOS inoltra le richieste del client nel formato originale senza introdurre il certificato RA e ciò è compatibile con Microsoft CA come esempio noto.

Client IOS PKI

Una delle entità di configurazione più importanti nel client PKI IOS è un trust. I parametri di configurazione del trust point sono illustrati in dettaglio in questa sezione.

Origine di tempo autorevole

Come accennato in precedenza, anche il client PKI deve disporre di una fonte di tempo autorevole. Il client PKI IOS può essere configurato come client NTP utilizzando la seguente configurazione:

```
configure terminal
ntp server <NTP Server IP address>
ntp source <source interface name>
ntp update-calendar

!! optional, if the NTP Server requires the clients to authenticate themselves
ntp authenticate
ntp authentication-key 1 md5 <key>

!! Optionally an access-list can be configured to restrict time-updates from a specifc NTP
server
access-list 1 permit <NTP Server IP address>
ntp access-group peer 1
```

Nome host e nome di dominio

In generale, è consigliabile configurare un nome host e un nome di dominio sul router:

```
configure terminal
hostname <string>
ip domain name <domain>
```

Coppia di chiavi RSA

Nel client PKI IOS, la coppia di chiavi RSA per una determinata registrazione del trust point può essere generata automaticamente o manualmente.

Il processo di generazione automatica delle chiavi RSA prevede quanto segue:

- Per impostazione predefinita, IOS crea una coppia di chiavi RSA a 512 bit
- Il nome della coppia di chiavi generato automaticamente è hostname.domain-name, che è il nome host del dispositivo combinato con il nome di dominio del dispositivo
- La coppia di chiavi generata automaticamente non è contrassegnata come esportabile.

Il processo di generazione automatica delle chiavi RSA prevede quanto segue:

- In alternativa, è possibile generare manualmente una coppia di chiavi RSA per scopi generici di livello adeguato utilizzando:
- `crypto key generate rsa general-keys label <LABEL> modulus < MOD> [exportable]`
  Label - nome della coppia di chiavi RSA
  MOD: modulo di chiave RSA o forza in bit tra 360 e 4096, che è tradizionalmente 512, 1024, 2048 o 4096.
  Il vantaggio della generazione manuale della coppia di chiavi RSA è la capacità di contrassegnare la coppia di chiavi come esportabile, il che a sua volta consente al certificato di identità di essere completamente esportato, che può quindi essere ripristinato su un altro dispositivo. Tuttavia, si dovrebbero comprendere le implicazioni di questa azione in termini di sicurezza.
- Una coppia di chiavi RSA viene collegata a un trust point prima dell'iscrizione tramite questo comando

```
crypto pki trustpoint MGMT
rsakeypair <LABEL> [<MOD> <MOD>]
```

In questo caso, se esiste già una coppia di chiavi RSA denominata **<LABEL>**, viene selezionata durante la registrazione del trust.

Se non esiste una coppia di chiavi RSA denominata **<LABEL>**, durante la registrazione viene eseguita una delle azioni seguenti:

- Se non viene passato alcun argomento **<MOD>**, viene generata una coppia di chiavi di 512 bit denominata **<LABEL>**.

- se viene passato un argomento **<MOD>**, viene generata una coppia di chiavi generiche **<MOD>** bit denominata **<LABEL>**

- se vengono passati due argomenti **<MOD>**, vengono generate una coppia di chiavi di crittografia **<MOD>** bit e una coppia di chiavi di crittografia **<MOD>** bit, entrambe denominate **<LABEL>**

Trustpoint

Un trust point è un contenitore astratto che contiene un certificato in IOS. Un singolo trust point è in grado di archiviare due certificati attivi in un determinato momento:

- Un certificato CA - Il caricamento di un certificato CA in un determinato trust point è noto come processo di autenticazione trust point.
- Un certificato ID rilasciato dalla CA - Caricamento o importazione di un certificato ID in un determinato trust point è noto come processo di registrazione del trust point.

Una configurazione di trust point è nota come criterio di attendibilità e definisce quanto segue:

- Quale certificato CA è caricato nel trust point?
- A quale CA si registra il trust point?
- In che modo IOS registra il trust point?
- In che modo viene convalidato un certificato rilasciato dalla CA specificata [caricata nel trust point]?

Di seguito sono illustrati i componenti principali di un trust point.

Modalità di registrazione

La modalità di registrazione del trust point, che definisce anche la modalità di autenticazione del trust point, può essere eseguita tramite tre metodi principali:

1. Terminal Enrollment (Registrazione terminale): metodo manuale per eseguire l'autenticazione del trust point e la registrazione dei certificati utilizzando il comando copy-paste (copia e incolla) nel terminale CLI.
2. Iscrizione SCEP - Autenticazione e registrazione di un trust point tramite SCEP su HTTP.
3. Profilo di registrazione: i metodi di autenticazione e di registrazione vengono definiti separatamente. Oltre ai metodi di registrazione di terminal e SCEP, i profili di registrazione offrono un'opzione per specificare i comandi HTTP/TFTP per eseguire il recupero dei file dal server, definito mediante un URL di autenticazione o di registrazione nel profilo.

Interfaccia di origine e VRF

L'autenticazione e la registrazione dei punti di accesso tramite HTTP (SCEP) o TFTP (profilo di registrazione) utilizzano il file system IOS per eseguire operazioni di I/O dei file. Questi scambi di pacchetti possono provenire da un'interfaccia di origine specifica e da un VRF.

Nel caso della configurazione di un trust point classico, questa funzionalità è abilitata utilizzando l'**interfaccia di origine** e i sottocomandi **vrf** del trust point.

In caso di profili di iscrizione, **interfaccia di origine** e **iscrizione | i** comandi **url di autenticazione <http/tftp://Server-location> vrf <vrf-name>** offrono la stessa funzionalità.

Esempio di configurazione:

```
vrf definition MGMT
 rd 1:1
 address-family ipv4
 exit-address-family

crypto pki trustpoint MGMT
```

```
 source interface Ethernet0/0
 vrf MGMT
```

o

```
crypto pki profile enrollment MGMT-Prof
 enrollment url http://10.1.1.1:80 vrf MGMT
 source-interface Ethernet0/0
crypto pki trustpoint MGMT
 enrollment profile MGMT-Prof
```

Registrazione e rinnovo automatici dei certificati

Èpossibile configurare il client PKI IOS per eseguire la registrazione e il rinnovo automatici utilizzando questo comando nella sezione del trust PKI:

```
crypto pki trustpoint MGMT
auto-enroll <percentage> <regenerate>
```

In questo caso, il comando **auto-enroll <percentuale> [regenerate]** indica che IOS deve eseguire il rinnovo del certificato esattamente all'80% della durata del certificato corrente.

La parola chiave **regenerate** indica che IOS deve rigenerare la coppia di chiavi RSA, nota come coppia di chiavi shadow, durante ogni operazione di rinnovo del certificato.

Questo è il comportamento di iscrizione automatica:

- Quando la **registrazione automatica** è configurata, se il trust point è autenticato, IOS esegue un'iscrizione automatica al server che si trova nell'URL indicato come parte del comando **enrollment url** nella sezione del trust PKI o nel profilo di registrazione.
- Quando un trust point viene registrato con un server PKI o una CA, nel client PKI viene inizializzato un timer di rinnovo o un timer SHADOW in base alla percentuale di **registrazione automatica** del certificato di identità corrente installato nel trust point. Questo timer è visibile nel comando **show crypto pki timer**. Ulteriori informazioni sulle funzioni del timer *consultare*
- Il supporto per le funzionalità di rinnovo viene fornito dal server PKI. Ulteriori informazioni in
  Il client PKI IOS esegue due tipi di rinnovo:
  Rinnovo implicito: Se il server PKI non invia "Rinnovo" come funzionalità supportata, IOS esegue un'iscrizione iniziale alla percentuale di registrazione automatica definita. Ad esempio, IOS utilizza un certificato autofirmato per firmare la richiesta di rinnovo.Rinnovo esplicito: Quando il server PKI supporta la funzionalità di rinnovo del certificato client PKI, annuncia "Rinnovo" come funzionalità supportata. IOS prende in considerazione questa funzionalità durante il rinnovo del certificato, ovvero utilizza il certificato di identità attivo corrente per firmare la richiesta del certificato di rinnovo.

Prestare attenzione durante la configurazione della percentuale di registrazione automatica. In qualsiasi client PKI specificato nella distribuzione, se si verifica una condizione in cui il certificato di identità scade contemporaneamente al certificato CA emittente, il valore di registrazione automatica deve sempre attivare l'operazione di rinnovo [shadow] dopo che la CA ha creato il certificato di rollover. Fare riferimento alla sezione **Dipendenze timer PKI** *in*

Controllo revoca certificato

Un trust point PKI autenticato, ovvero un trust point PKI contenente un certificato CA, è in grado di eseguire la convalida del certificato durante una negoziazione IKE o SSL, in cui il certificato peer è sottoposto a una convalida completa del certificato. Uno dei metodi di convalida consiste nel controllare lo stato di revoca dei certificati peer utilizzando uno dei due metodi seguenti:

- CRL (Certificate Revocation List): file contenente i numeri di serie dei certificati revocati da una determinata CA. Il file è firmato utilizzando il certificato CA emittente. Il metodo CRL prevede il download del file CRL tramite HTTP o LDAP.

- Protocollo di stato del certificato online (OCSP) - IOS stabilisce un canale di comunicazione con un'entità chiamata Risponditore OCSP, che è un server designato dalla CA emittente. Un client come IOS invia una richiesta contenente il numero di serie del certificato in fase di convalida. Il risponditore OCSP risponde con lo stato di revoca del numero di serie specificato. Il canale di comunicazione può essere stabilito utilizzando qualsiasi protocollo di applicazione/trasporto supportato, che in genere è HTTP.

Il controllo di revoca può essere definito utilizzando questi comandi nella sezione del trust point PKI:

```
crypto pki trustpoint MGMT
 revocation-check crl ocsp none
```

Per impostazione predefinita, un trust point è configurato per eseguire il controllo delle revoche utilizzando crl.

I metodi possono essere riordinati e il controllo dello stato di revoca viene eseguito nell'ordine definito. Il metodo "none" ignora il controllo di revoca.

Cache CRL

Con la verifica della revoca basata su CRL, ogni convalida del certificato può attivare un nuovo download del file CRL. Inoltre, quando il file CRL diventa più grande o il punto di distribuzione CRL (CDP) è più lontano, il download del file durante ogni processo di convalida impedisce le prestazioni del protocollo in base alla convalida del certificato. Pertanto, la memorizzazione nella cache CRL viene eseguita per migliorare le prestazioni e la memorizzazione nella cache della CRL prende in considerazione la validità della CRL.

La validità del CRL viene definita utilizzando due parametri temporali: **LastUpdate**, l'ultima volta in cui il CRL è stato pubblicato dalla CA di emissione, e **NextUpdate**, l'ultima volta in cui una nuova versione del file CRL viene pubblicata dalla CA di emissione.

IOS memorizza nella cache ogni CRL scaricato per tutto il periodo di validità del CRL. Tuttavia, in determinate circostanze, ad esempio se il CDP non è raggiungibile temporaneamente, può essere necessario conservare il CRL nella cache per un periodo di tempo esteso. In IOS un CRL memorizzato nella cache può essere mantenuto per un massimo di 24 ore dopo la scadenza della validità del CRL e può essere configurato utilizzando questo comando nella sezione del trust point PKI:

```
crypto pki trustpoint MGMT
 crl cache extend <0 - 1440>
!! here the value is in minutes
```

In alcune circostanze, ad esempio in un'autorità di certificazione emittente che revoca i certificati

entro il periodo di validità dell'elenco di revoche di certificati, è possibile configurare IOS in modo che la cache venga eliminata con maggiore frequenza. Se si elimina il CRL in modo prematuro, IOS è costretto a scaricare il CRL con maggiore frequenza per mantenere aggiornata la cache del CRL. Questa opzione di configurazione è disponibile nella sezione del trust point PKI:

```
crypto pki trustpoint MGMT
 crl cache delete-after <1-43200>
!! here the value is in minutes
```

Infine, è possibile configurare IOS in modo che il file CRL non venga memorizzato nella cache utilizzando questo comando nella sezione del trust PKI:

```
crypto pki trustpoint MGMT
 crl cache none
```

Configurazione consigliata

Di seguito è riportata una tipica distribuzione CA con Root-CA e una configurazione Sub-CA. L'esempio include anche una configurazione Sub-CA protetta da un Autorità registrazione integrità.

Con una coppia di chiavi RSA da 2048 bit su tutto il sistema, questo esempio consiglia una configurazione in cui:
Root-CA ha una durata di 8 anni
La durata della Sub-CA è di 3 anni
I certificati client vengono rilasciati per un anno, configurati per richiedere automaticamente il rinnovo di un certificato.

CA RADICE - Configurazione

```
crypto pki server ROOTCA
database level complete
database archive pkcs12 password p12password
issuer-name CN=RootCA,OU=TAC,O=Cisco
lifetime crl 120
lifetime certificate 1095
lifetime ca-certificate 2920
grant auto rollover ca-cert
auto-rollover 85
database url ftp:://10.1.1.1/CA/ROOT/
database url crl ftp:://10.1.1.1/CA/ROOT/
database url crl publish ftp:://10.1.1.1/WWW/CRL/ROOT/
cdp-url http://10.1.1.1/WWW/CRL/ROOT/ROOTCA.crl
```

SUBCA senza RA - Configurazione

```
crypto pki server SUBCA
database level complete
database archive pkcs12 password p12password
issuer-name CN=SubCA,OU=TAC,O=Cisco
lifetime crl 12
lifetime certificate 365
grant auto SUBCA
```

```
auto-rollover 85
database url ftp:://10.1.1.1/CA/SUB/
database url crl ftp:://10.1.1.1/CA/SUB/
database url crl publish ftp:://10.1.1.1/WWW/CRL/SUB/
cdp-url http://10.1.1.1/WWW/CRL/SUB/SUBCA.crl
mode sub-cs


crypto pki trustpoint SUBCA
revocation-check crl
rsakeypair SUBCA 2048
enrollment url http://172.16.1.1
```

SUBCA con RA - Configurazione

```
crypto pki server SUBCA
database level complete
database archive pkcs12 password p12password
issuer-name CN=SubCA,OU=TAC,O=Cisco
lifetime crl 12
lifetime certificate 365
grant ra-auto
grant auto rollover ra-cert
auto-rollover 85
 database url ftp:://10.1.1.1/CA/SUB/
database url crl ftp:://10.1.1.1/CA/SUB/
database url crl publish ftp:://10.1.1.1/WWW/CRL/SUB/
cdp-url http://10.1.1.1/WWW/CRL/SUB/SUBCA.crl
mode sub-cs

crypto pki trustpoint SUBCA
revocation-check crl
rsakeypair SUBCA 2048
enrollment url http://172.16.1.1
```

RA per SUBCA - Configurazione

```
crypto pki server RA-FOR-SUBCA
database level complete
database archive pkcs12 password p12password
mode ra
grant auto RA-FOR-SUBCA
auto-rollover 85
database url ftp:://10.1.1.1/CA/RA4SUB/


crypto pki trustpoint RA-FOR-SUBCA
enrollment url http://172.16.1.2:80
password ChallengePW123
subject-name CN=RA,OU=ioscs RA,OU=TAC,O=Cisco
revocation-check crl
rsakeypair RA 2048
```

Registrazione certificato

Iscrizione manuale

La registrazione manuale implica la generazione offline di CSR nel client PKI, che viene copiata

manualmente nella CA. L'amministratore firma manualmente la richiesta, che viene quindi importata nel client.

Client PKI

Configurazione client PKI:

```
crypto pki trustpoint MGMT
enrollment terminal
serial-number
ip-address none
password ChallengePW123
subject-name CN=PKI-Client,OU=MGMT,OU=TAC,O=Cisco
revocation-check crl
rsakeypair PKI-Key
```

Passaggio 1. Eseguire innanzitutto l'autenticazione del trust point (questa operazione può essere eseguita anche dopo il passaggio 2).

```
crypto pki authenticate MGMT
!! paste the CA, in this case the SUBCA, certificate in pem format and enter “quit” at the end
in a line by itself]

PKI-Client-1(config)# crypto pki authenticate MGMT

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

-----BEGIN CERTIFICATE-----
MIIDODCCAiCgAwIBAgIBAjANBgkqhkiG9w0BAQUFADAvMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ8wDQYDVQQDEwZSb290Q0EwHhcNMTUxMDE4MjA0MjI3
WhcNMTgxMDE3MjA0MjI3WjAuMQ4wDAYDVQQKEwVDaXNjbzEMMAoGA1UECxMDVEFD
MQ4wDAYDVQQDEwVTdWJDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AJ7hKmBfDo/GOQAEYY/1ptpg28DejUE0ZlDorDkADP2vKfRI0kalSnOs2PIe01ip
7pHFurFVUx/p8teMCkmvnbrSBfyUrWo9YfQeGOELb4d3dSW4jGakm6M8lNRkO7HP
s+IVVTuJSeUZxov6DPa92Y/6HLayX15Iq8ZL+KwmA9oS5NeTi1tBbrcc3Hq8W2Ay
879nDDOqDOsQMQqKtc7E/IA7SBjowImra6FUxzgJ5ye5MymRfRYAH+c4qZJxwHTc
/tSmjiOJlM7X5dtehU/XPEEEbs78peXO9FyzAbhOtCRBVTnhc8WWijq84xu8Oej7
LbXGBKIHSP0uDe32CV0noEUCAwEAAaNgMF4wDwYDVR0TAQH/BAUwAwEB/zALBgNV
HQ8EBAMCAYYwHwYDVR0jBBgwFoAU+oNBdIj9mjpieQ2Z7v79JhKnL68wHQYDVR0O
BBYEFFOv8xtHRojMdJ65oQ2PFBeD5oHiMA0GCSqGSIb3DQEBBQUAA4IBAQAZ/W3P
Wqs4vuQ2jCnVE0v1PVQe/VNS54P/fprQRelceawiBCHA3D0SRgHqUWJUIqBLv4sD
QBegmyTmS76C8YC/jN7VbI30hf6R4qP7CWu8Ef9sWPRC/+Oy6e8AinrK+sVd2dp/
LLDMVoBhS2bQFLWiyRvC9FgyczXRdF+rhKTKeEVXGs7C/Yk/9z+/00rVmSGZAS+v
aPpZWjoC3459t51t8Y3iE6GtjBvmyxBwWt01/5gCu6Mszi7X/kXdmqgNfT5bBBnv
yjWE2ZS8NsH4hwDZpmDJqx4qhrH6bw3iUm+pK9fceZ/HTYasxtcr4NUvvxwXc60y
Wrtlpq3g2XfG+qFB
-----END CERTIFICATE-----
quit
Trustpoint 'MGMT' is a subordinate CA and holds a non self signed cert
Certificate has the following attributes:
      Fingerprint MD5: DBE6AFAC 9E1C3697 01C5466B 78E0DFE3
     Fingerprint SHA1: EAD41B32 BB37BC11 6E0FBC13 41701BFE 200DC46E

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
% Certificate successfully imported
```

Passaggio 2. Generare la richiesta di firma del certificato e portare il CSR alla CA per ottenere il certificato concesso:

```
PKI-Client-1(config)# crypto pki enroll MGMT
% Start certificate enrollment ..

% The subject name in the certificate will include: CN=PKI-Client,OU=MGMT,OU=TAC,O=Cisco
% The subject name in the certificate will include: PKI-Client-1.cisco.com
% The serial number in the certificate will be: 104Certificate Request follows:

MIIC2zCCAcMCAQAwdTEOMAwGA1UEChMFQ2lzY28xDDAKBgNVBAsTA1RBQzENMAsG
A1UECxMETUdNVDETMBEGA1UEAxMKUEtJLUNsaWVudDExMAoGA1UEBRMDMTA0MCMG
CSqGSIb3DQEJAhYWUEtJLUNsaWVudC0xLmNpc2NvLmNvbTCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBANwa7g+DJxG57sMg020w1Fdv9+mIZ6R4livbt7vo
AbW8jpzQlMv4lV3r6ulTJumhBvV7xI+1ZijXP0EqqQZLNboYv37UTJgm83DGO57I
8RTn9DfDQpHiqvhtNuC5S3SCC/hvCxFXnfNXqC3dkfuVkVWoJiLZY87R6j44jUq0
tTL5d8t6lz2L0BeekzKJlOs73gONx0VgQyI/WjDiEwL0xF4DNHURaYyOxBWJc7/B
psDCf7376mb7XXz0LB++E8SvvM/Li6+yQzYv1Lagr0b8C4uE+tCDxG5OniNDiS82
JXsVd43vKRFW85W2ssrElgkuWAvS017XlwK+UDX21dtFdfUCAwEAAaAhMB8GCSqG
SIb3DQEJDjESMBAwDgYDVR0PAQH/BAQDAgWgMA0GCSqGSIb3DQEBBQUAA4IBAQA+
UqkqUZZar9TdmB8I7AHku5m79l42o8cuhwOccehxE6jmzh9P+Ttb9Me7l7L8Y2iR
yYyJHsL7m6tjK2+Gllg7RJdoxG8l8aMZS1ruXOBqFBrmo7OSz1nfXpiTyh88jyca
Hw/8G8uaYuQbZIj53BWmQGRpm7J//ktn0D4W3Euh9HttMuYYX7BOct05BLqqiCCw
n+kKHZxzGXy7JSZpUlDtvPPnnuqWK7iVoy3vtV6GoFOrxRoo05QVFehS0/m4NFQI
mXA0eTEgujSaQi4iWte/UxruO/3p/eHr67MtZXLRl0YDFgaQd7vD7fCsDx5pquKV
jNEUT6FNHdsnqrAKqodO

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

Redisplay enrollment request? [yes/no]: no
```

Passaggio 3. Importare il certificato concesso tramite il terminale:

```
PKI-Client-1(config)# crypto pki import MGMT certificate

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

-----BEGIN CERTIFICATE-----
MIIDcDCCAligAwIBAgIBAzANBgkqhkiG9w0BAQQFADAuMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ4wDAYDVQQDEwVTdWJDQTAeFw0xNTEwMTkyMDM1MDZa
Fw0xNjEwMTgyMDM1MDZaMHUxDjAMBgNVBAoTBUNpc2NvMQwwCgYDVQQLEwNUQUMx
DTALBgNVBAsTBE1HTVQxEzARBgNVBAMTClBLSS1DbGllbnQxMTAKBgNVBAUTAzEw
NDAjBgkqhkiG9w0BCQIWFlBLSS1DbGllbnQtMS5jaXNjby5jb20wggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQDcGu4PgycRue7DINNtMNRXb/fpiGekeJYr
27e76AG1vI6c0JTL+JVd6+rpUybpoQb1e8SPtWYo1z9BKqkGSzW6GL9+1EyYJvNw
xjueyPEU5/Q3w0KR4qr4bTbguUt0ggv4bwsRV53zV6gt3ZH7lZFVqCYi2WPO0eo+
OI1KtLUy+XfLepc9i9AXnpMyiZTrO94DjcdFYEMiP1ow4hMC9MReAzR1EWmMjsQV
iXO/wabAwn+9++pm+1189CwfvhPEr7zPy4uvskM2L9S2oK9G/AuLhPrQg8RuTp4j
Q4kvNiV7FXeN7ykRVvOVtrLKxJYJLlgL0tNe15cCvlA19tXbRXX1AgMBAAGjUjBQ
MA4GA1UdDwEB/wQEAwIFoDAfBgNVHSMEGDAWgBRTr/MbR0aIzHSeuaENjxQXg+aB
4jAdBgNVHQ4EFgQUK+9/lrlL+TyYxvsgxzPwwrhmS5UwDQYJKoZIhvcNAQEEBQAD
ggEBAIrLrzFLnm9z7ula1uRh03r6dSCFy9XkOk6ZaHfksbENoDmkcgIwKoAsSF9E
rQmA9W5qXVU7PEsqOmcu8zEv7uuiqM4D4nDP69HsyToPjxVcoG7PSyKJYnXRgkVa
IYyMaSaRKWlhb2uWj3XPLzS0/ZBOGAG9rMBVzaqLfLAZgnQUVJvwsNofe+ASojk9
mCRsEHD8WVuAzcnwYKXx3j3x/T7jbB3ibPfbYKQqlS12XFHhJoK+HfSA2fyZBFLF
syN/B2Ow0bvc71YlYOQuYwz3XOMIHD6vARTO4f0ZIQti2dy1kHc+5lIdhLsn/bA5
yUo7WxnAE8LOoYIf9iU9q0mqkMU=
```

```
-----END CERTIFICATE-----
quit
% Router Certificate successfully imported
```

Server PKI

Passaggio 1. Esportare innanzitutto il certificato CA emittente dalla CA, che in questo caso è un certificato CA SECONDARIA. Questa opzione viene importata nel passaggio 1 precedente nel client PKI, ad esempio l'autenticazione Trustpoint.

```
SUBCA(config)# crypto pki export SUBCA pem terminal
% CA certificate: !! Root-CA certificate
-----BEGIN CERTIFICATE-----
MIIDPDCCAiSgAwIBAgIBATANBgkqhkiG9w0BAQQFADAvMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ8wDQYDVQQDEwZSb290Q0EwHhcNMTUxMDE4MjAwOTIx
WhcNMjMxMDE2MjAwOTIxWjAvMQ4wDAYDVQQKEwVDaXNjbzEMMAoGA1UECxMDVEFD
MQ8wDQYDVQQDEwZSb290Q0EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
AQCajfMy8gU3ZXQfKgP/wYKLB0cuywzYcDaSoNVlEvUZOWgUltCGP4CiCXyw0U0U
Zmy0rusibMV7mtkTX5muaPC0XfT98rswPiZV0qvEYpHF2YodPOUoqR3FeKj/tDbI
IikcLrfj87aeMJjCrWD888wfTN9Hw9x2QVDoSxLbzTLticXdXxwS5wxlM16GspmT
WL4fg1JRWgjRqMmOcpf716Or88XJ2N2HeWxxVFIwYQf3thHR6DgTdCgJ1uqjVE6q
1LQ1g8k81mvuCZX0uLZiTMJ69xo+Ot/RpeeE2RShxK5rh56ObQq4MT4lbIPKqIxU
lbKzWdh10NiYwjgTNwTs9GGvAgMBAAGjYzBhMA8GA1UdEwEB/wQFMAMBAf8wDgYD
VR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaAFPqDQXSI/Zo6YnkNme7+/SYSpy+vMB0G
A1UdDgQWBBT6g0F0iP2aOmJ5DZnu/v0mEqcvrzANBgkqhkiG9w0BAQQFAAOCAQEA
VKwqI9vpmoRh9QoOJGtOA3qEgV4eCfXdMuYxmmo0sdaBYBfQm2RhZeQ1X90vVBso
G4Wx6cJVSXCtkqZTm1IoMtya+gdhLbKqZmxc+I5/js88SrbrBIm4zj+sOoySV9kW
THEEmZjdTCWXo2wNCr23gGdnb4RqZ0FTOfoZO/2Xnpcbvhz2/K7wlDRJ5k1wrsRW
RRwsQEh4LYMFIg0aBs4gmRLZ8ytwrvvrhQTVrAA/MeomUEPhcIYESg1AlWxoCYZU
0iqKfDa9+4wEJ+PMGDhM2UV0fuP0rWitKWxecSVbo54z3VHYwwCbz2jCs8XGE61S
+XlxCZKFVdlVaMmuaZTdFg==
-----END CERTIFICATE-----

% General Purpose Certificate: !! SUBCA certificate
-----BEGIN CERTIFICATE-----
MIIDODCCAiCgAwIBAgIBAjANBgkqhkiG9w0BAQUFADAvMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ8wDQYDVQQDEwZSb290Q0EwHhcNMTUxMDE4MjA0MjI3
WhcNMTgxMDE3MjA0MjI3WjAuMQ4wDAYDVQQKEwVDaXNjbzEMMAoGA1UECxMDVEFD
MQ4wDAYDVQQDEwVTdWJDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AJ7hKmBfDo/GOQAEYY/1ptpg28DejUE0ZlDorDkADP2vKfRI0kalSnOs2PIe01ip
7pHFurFVUx/p8teMCkmvnbrSBfyUrWo9YfQeGOELb4d3dSW4jGakm6M8lNRkO7HP
s+IVVTuJSeUZxov6DPa92Y/6HLayX15Iq8ZL+KwmA9oS5NeTi1tBbrcc3Hq8W2Ay
879nDDOqDOsQMQqKtc7E/IA7SBjowImra6FUxzgJ5ye5MymRfRYAH+c4qZJxwHTc
/tSmjiOJlM7X5dtehU/XPEEEbs78peXO9FyzAbhOtCRBVTnhc8WWijq84xu8Oej7
LbXGBKIHSP0uDe32CV0noEUCAwEAAaNgMF4wDwYDVR0TAQH/BAUwAwEB/zALBgNV
HQ8EBAMCAYYwHwYDVR0jBBgwFoAU+oNBdIj9mjpieQ2Z7v79JhKnL68wHQYDVR0O
BBYEFFOv8xtHRojMdJ65oQ2PFBeD5oHiMA0GCSqGSIb3DQEBBQUAA4IBAQAZ/W3P
Wqs4vuQ2jCnVE0v1PVQe/VNS54P/fprQRelceawiBCHA3D0SRgHqUWJUIqBLv4sD
QBegmyTmS76C8YC/jN7VbI30hf6R4qP7CWu8Ef9sWPRC/+Oy6e8AinrK+sVd2dp/
LLDMVoBhS2bQFLWiyRvC9FgyczXRdF+rhKTKeEVXGs7C/Yk/9z+/00rVmSGZAS+v
aPpZWjoC3459t51t8Y3iE6GtjBvmyxBwWt01/5gCu6Mszi7X/kXdmqgNfT5bBBnv
yjWE2ZS8NsH4hwDZpmDJqx4qhrH6bw3iUm+pK9fceZ/HTYasxtcr4NUvvxwXc60y
Wrtlpq3g2XfG+qFB
-----END CERTIFICATE-----
```

Passaggio 2. Dopo il passaggio 2 sul client PKI, estrarre il CSR dal client e fornirlo per la firma sulla SUBCA utilizzando questo comando:

```
crypto pki server SUBCA request pkcs10 terminal pem
```

Questo comando suggerisce che la SUBCA accetti una richiesta di firma del certificato dal terminale e, una volta concessa, i dati del certificato vengono stampati in formato PEM.

```
SUBCA# crypto pki server SUBCA request pkcs10 terminal pem
PKCS10 request in base64 or pem

% Enter Base64 encoded or PEM formatted PKCS10 enrollment request.
% End with a blank line or "quit" on a line by itself.
MIIC2zCCAcMCAQAwdTEOMAwGA1UEChMFQ2lzY28xDDAKBgNVBAsTA1RBQzENMAsG
A1UECxMETUdNVDETMBEGA1UEAxMKUEtJLUNsaWVudDExMAoGA1UEBRMDMTA0MCMG
CSqGSIb3DQEJAhYWUEtJLUNsaWVudC0xLmNpc2NvLmNvbTCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBANwa7g+DJxG57sMg020w1Fdv9+mIZ6R4livbt7vo
AbW8jpzQlMv4lV3r6ulTJumhBvV7xI+1ZijXP0EqqQZLNboYv37UTJgm83DGO57I
8RTn9DfDQpHiqvhtNuC5S3SCC/hvCxFXnfNXqC3dkfuVkVWoJiLZY87R6j44jUq0
tTL5d8t6lz2L0BeekzKJlOs73gONx0VgQyI/WjDiEwL0xF4DNHURaYyOxBWJc7/B
psDCf7376mb7XXz0LB++E8SvvM/Li6+yQzYv1Lagr0b8C4uE+tCDxG5OniNDiS82
JXsVd43vKRFW85W2ssrElgkuWAvS017XlwK+UDX21dtFdfUCAwEAAaAhMB8GCSqG
SIb3DQEJDjESMBAwDgYDVR0PAQH/BAQDAgWgMA0GCSqGSIb3DQEBBQUAA4IBAQA+
UqkqUZZar9TdmB8I7AHku5m79l42o8cuhwOccehxE6jmzh9P+Ttb9Me7l7L8Y2iR
yYyJHsL7m6tjK2+Gllg7RJdoxG8l8aMZS1ruXOBqFBrmo7OSz1nfXpiTyh88jyca
Hw/8G8uaYuQbZIj53BWmQGRpm7J//ktn0D4W3Euh9HttMuYYX7BOct05BLqqiCCw
n+kKHZxzGXy7JSZpUlDtvPPnnuqWK7iVoy3vtV6GoFOrxRoo05QVFehS0/m4NFQI
mXA0eTEgujSaQi4iWte/UxruO/3p/eHr67MtZXLRl0YDFgaQd7vD7fCsDx5pquKV
jNEUT6FNHdsnqrAKqodO
quit
% Enrollment request pending, reqId=1
```

Se la CA è in modalità di concessione automatica, il certificato concesso viene visualizzato in formato PEM. Quando la CA è in modalità di concessione manuale, la richiesta di certificato viene contrassegnata come **in sospeso**, viene assegnato un valore ID e accodata nella coda delle richieste di registrazione.

```
SUBCA#show crypto pki server SUBCA requests
Enrollment Request Database:

Router certificates requests:
ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------
1      pending    7710276982EA176324393D863C9E350E serialNumber=104+hostname=PKI-Client-
1.cisco.com,cn=PKI-Client,ou=MGMT,ou=TAC,o=Cisco
```

Passaggio 3. Concedere manualmente la richiesta utilizzando questo comando:

```
SUBCA# crypto pki server SUBCA grant 1
% Granted certificate:
-----BEGIN CERTIFICATE-----
MIIDcDCCAligAwIBAgIBAzANBgkqhkiG9w0BAQQFADAuMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ4wDAYDVQQDEwVTdWJDQTAeFw0xNTEwMTkyMDM1MDZa
Fw0xNjEwMTgyMDM1MDZaMHUxDjAMBgNVBAoTBUNpc2NvMQwwCgYDVQQLEwNUQUMx
DTALBgNVBAsTBE1HTVQxEzARBgNVBAMTClBLSS1DbGllbnQxMTAKBgNVBAUTAzEw
NDAjBgkqhkiG9w0BCQIWFlBLSS1DbGllbnQtMS5jaXNjby5jb20wggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQDcGu4PgycRue7DINNtMNRXb/fpiGekeJYr
27e76AG1vI6c0JTL+JVd6+rpUybpoQb1e8SPtWYo1z9BKqkGSzW6GL9+1EyYJvNw
xjueyPEU5/Q3w0KR4qr4bTbguUt0ggv4bwsRV53zV6gt3ZH7lZFVqCYi2WPO0eo+
OI1KtLUy+XfLepc9i9AXnpMyiZTrO94DjcdFYEMiP1ow4hMC9MReAzR1EWmMjsQV
iXO/wabAwn+9++pm+1189CwfvhPEr7zPy4uvskM2L9S2oK9G/AuLhPrQg8RuTp4j
Q4kvNiV7FXeN7ykRVvOVtrLKxJYJLlgL0tNe15cCvlA19tXbRXX1AgMBAAGjUjBQ
```

```
MA4GA1UdDwEB/wQEAwIFoDAfBgNVHSMEGDAWgBRTr/MbR0aIzHSeuaENjxQXg+aB
4jAdBgNVHQ4EFgQUK+9/lrlL+TyYxvsgxzPwwrhmS5UwDQYJKoZIhvcNAQEEBQAD
ggEBAIrLrzFLnm9z7ula1uRh03r6dSCFy9XkOk6ZaHfksbENoDmkcgIwKoAsSF9E
rQmA9W5qXVU7PEsqOmcu8zEv7uuiqM4D4nDP69HsyToPjxVcoG7PSyKJYnXRgkVa
IYyMaSaRKWlhb2uWj3XPLzS0/ZBOGAG9rMBVzaqLfLAZgnQUVJvwsNofe+ASojk9
mCRsEHD8WVuAzcnwYKXx3j3x/T7jbB3ibPfbYKQqlS12XFHhJoK+HfSA2fyZBFLF
syN/B2Ow0bvc71YlYOQuYwz3XOMIHD6vARTO4f0ZIQti2dy1kHc+5lIdhLsn/bA5
yUo7WxnAE8LOoYIf9iU9q0mqkMU=
-----END CERTIFICATE-----
```

Nota: La registrazione manuale di una CA secondaria a una CA radice non è consentita.

Nota: Una CA in stato disabilitato a causa di un server HTTP disabilitato può concedere manualmente le richieste di certificato.

Registrazione tramite SCEP

Configurazione client PKI:

```
crypto pki trustpoint MGMT
enrollment url http://172.16.1.2:80
 serial-number
 ip-address none
 password 7 110A1016141D5A5E57
 subject-name CN=PKI-Client,OU=MGMT,OU=TAC,O=Cisco
 revocation-check crl
 rsakeypair PKI-Key 2048
```

Configurazione server PKI:

```
SUBCA# show run all | section pki server
crypto pki server SUBCA
 database level complete
 database archive pkcs12 password 7 01100F175804575D72
 issuer-name CN=SubCA,OU=TAC,O=Cisco
 lifetime crl 12
 lifetime certificate 365
 lifetime ca-certificate 1095
 lifetime enrollment-request 168
 mode sub-cs
 auto-rollover 85
 database url ftp:://10.1.1.1/CA/SUB/
database url crl ftp:://10.1.1.1/CA/SUB/
database url crl publish ftp:://10.1.1.1/WWW/CRL/SUB/
```

La modalità predefinita di concessione della richiesta di certificato è manuale:

```
SUBCA# show crypto pki server
Certificate Server SUBCA:
   Status: enabled
   State: enabled
   Server's configuration is locked  (enter "shut" to unlock it)
   Issuer name: CN=SubCA,OU=TAC,O=Cisco
   CA cert fingerprint: DBE6AFAC 9E1C3697 01C5466B 78E0DFE3
```

```
Server configured in subordinate server mode
Upper CA cert fingerprint: CD0DE4C7 955EFD60 296B7204 41FB6EF6
 Granting mode is: manual
Last certificate issued serial number (hex): 4
CA certificate expiration timer: 21:42:27 CET Oct 17 2018
CRL NextUpdate timer: 09:42:37 CET Oct 20 2015
Current primary storage dir: unix:/SUB/
Current storage dir for .crl files: unix:/SUB/
Database Level: Complete - all issued certs written as <serialnum>.cer
Auto-Rollover configured, overlap period 85 days
Autorollover timer: 21:42:27 CET Jul 24 2018
```

Concessione manuale

Passaggio 1. Client PKI: Come primo passaggio, obbligatorio, autenticare il trust point sul client PKI:

```
PKI-Client-1(config)# crypto pki authenticate MGMT
Trustpoint 'MGMT' is a subordinate CA and holds a non self signed cert
Certificate has the following attributes:
      Fingerprint MD5: DBE6AFAC 9E1C3697 01C5466B 78E0DFE3
     Fingerprint SHA1: EAD41B32 BB37BC11 6E0FBC13 41701BFE 200DC46E

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

Passaggio 2. Client PKI: Dopo l'autenticazione del trust point, è possibile registrare il client PKI per un certificato.

> **Nota:** Se la registrazione automatica è configurata, il client eseguirà automaticamente l'iscrizione.

```
config terminal
crypto pki enroll MGMT
```

Dietro le quinte, si svolgono questi eventi:

- IOS cerca una coppia di chiavi RSA chiamata PKI-Key. Se esiste, viene selezionato per richiedere un certificato di identità. In caso contrario, IOS crea una coppia di chiavi a 2048 bit denominata PKI-Key e la utilizza per richiedere un certificato di identità.
- IOS crea una richiesta di firma del certificato nel formato PKCS10.
- IOS quindi cripta questo CSR utilizzando una chiave simmetrica casuale. La chiave simmetrica casuale viene crittografata utilizzando la chiave pubblica del destinatario, ovvero la SUBCA (la chiave pubblica della SUBCA è disponibile a causa dell'autenticazione del trust point). Il CSR crittografato, la chiave simmetrica casuale crittografata e le informazioni sul destinatario vengono riuniti nei dati con busta PKCS#7.
- Durante la registrazione iniziale, i dati con busta PKCS#7 vengono firmati con un certificato autofirmato temporaneo. I dati protetti da busta PKCS#7, il certificato di firma utilizzato dal client e la firma del client vengono riuniti in un pacchetto di dati firmato PKCS#7. Codifica base64, quindi codifica URL. Il BLOB di dati risultante viene inviato come argomento

"messaggio" nell'URI HTTP inviato alla CA:

```
GET /cgi-bin/pkiclient.exe?operation=PKIOperation&message=MII…. HTTP/1.0
```

Passaggio 3. Server PKI:

Quando il server PKI IOS riceve la richiesta, verifica quanto segue:

1. Controlla se il database delle richieste di registrazione contiene una richiesta di certificato con lo stesso ID transazione associato alla nuova richiesta.

> **Nota:** Un ID transazione è un hash MD5 della chiave pubblica, per il quale il client richiede un certificato di identità.

2. Controlla se il database di richiesta di registrazione contiene una richiesta di certificato con la stessa password di richiesta di verifica inviata dal client.

> **Nota:** Se (1) restituisce true o entrambi (1) e (2) insieme restituiscono true, un server CA è in grado di rifiutare la richiesta a causa di una richiesta di identità duplicata. In questo caso, tuttavia, il server PKI IOS sostituisce la richiesta precedente con quella più recente.

Passaggio 4. Server PKI:

Concedere manualmente le richieste nel server PKI:

Per visualizzare la richiesta:

```
show crypto pki server SUBCA requests
```

Per concedere una richiesta specifica o tutte le richieste:

```
crypto pki server SUBCA grant <id|all>
```

Passaggio 5. Client PKI:

Nel frattempo, un client PKI avvia un timer POLL. In questo caso, IOS esegue GetCertInitial a intervalli regolari finché SCEP CertRep = GRANT non riceve il certificato concesso dal client.

Una volta ricevuto il certificato concesso, IOS lo installa automaticamente.