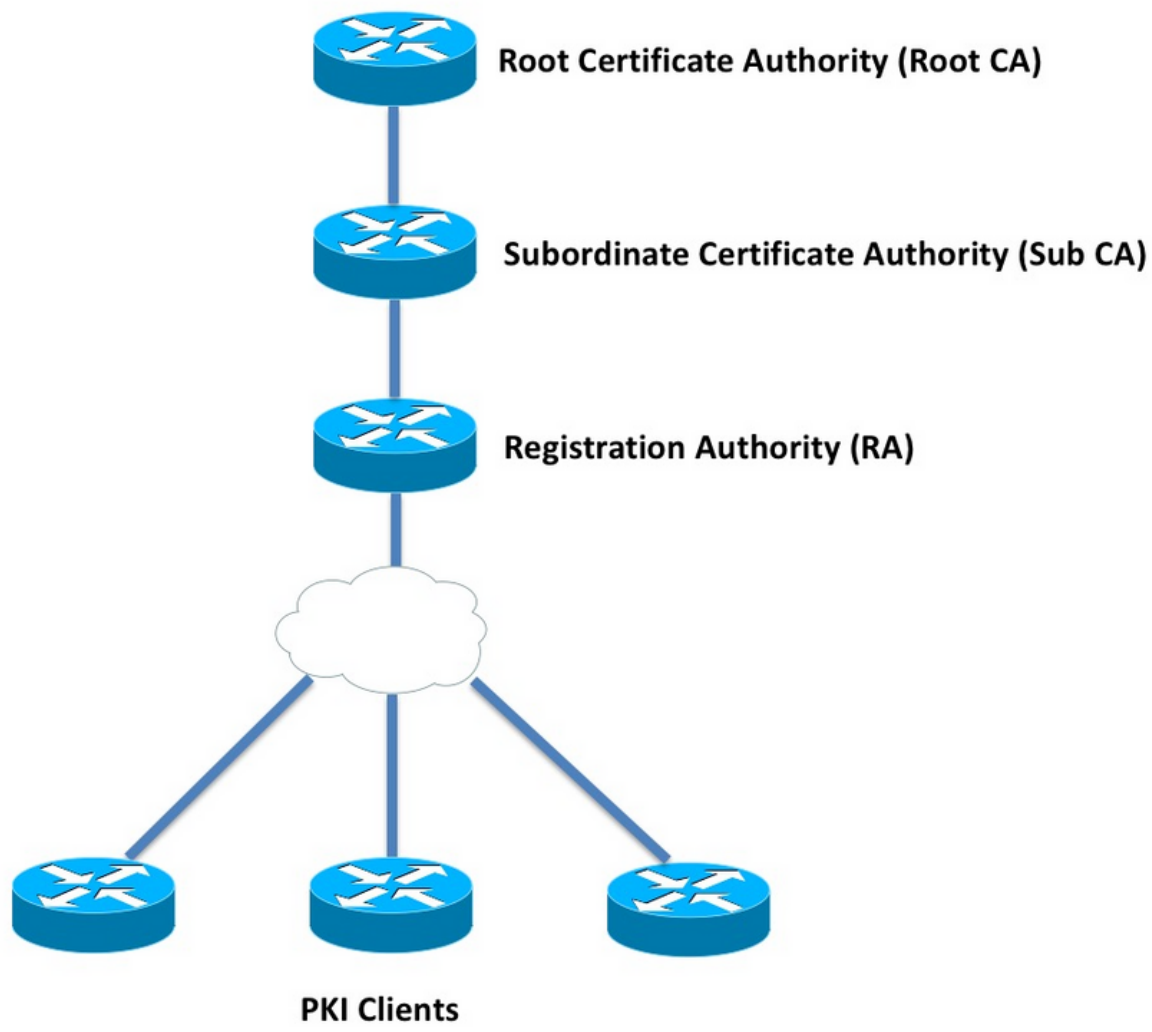
Guía de despliegue IOS PKI: Diseño inicial y despliegue

Contenido

Introducción

Este documento describe el servidor pki y las funcionalidades del cliente IOS detalladamente. Dirige el diseño inicial y las consideraciones sobre la instrumentación IOS PKI.

Infraestructura PKI



Certificate Authority

El Certificate Authority (CA), también designado el servidor pki en el documento, es una entidad

confiable esa los Certificados de los problemas. El PKI se basa en la confianza, y la confianza-jerarquía comienza en el Certificate Authority de la raíz (raíz CA). Porque raíz CA está en la cima de la jerarquía, tiene un certificado autofirmado.

Certificate Authority subordinado

En la Confianza-jerarquía PKI toda la raíz abajo de las autoridades de certificación se conoce como autoridades de certificación subordinadas (Sub-CA). Evidentemente, un certificado de Sub-CA es publicado por CA, que es un nivel arriba.

El PKI no impone ningún límite ante el número de Sub-CAS en una jerarquía dada. Sin embargo, en un despliegue en empresas con más de 3 niveles de autoridades de certificación puede llegar a ser difícil de manejar.

Autoridad de registro

El PKI define un Certificate Authority especial conocido como registration authority (RA), que es responsable de autorizar a los clientes PKI de alistar a Sub-CA dado o raíz CA. El RA no publica los Certificados a los clientes PKI, en lugar decide a qué PKI-cliente puede o no puede ser publicado un certificado por Sub-CA o raíz CA.

El papel principal de un RA es descargar la validación básica de la petición del certificado del cliente de CA, y protege CA contra la exposición directa a los clientes. Esta manera, RA se coloca entre los clientes PKI y CA, así protegiendo CA contra cualquier clase de establecimientos de rechazo del servicio.

Cliente PKI

Cualquier dispositivo petición un certificado basado en un par clave público-privado residente para probar su identidad a los otros dispositivos se conoce como cliente PKI.

Un cliente PKI debe ser capaz de la generación o de salvar un par clave público-privado tal como RSA o DSA o ECDSA.

Un certificado es una prueba de la identidad y de la validez de una clave pública dada, con tal que la clave privada correspondiente exista en el dispositivo.

Servidor pki IOS

Evolución de la característica del servidor pki IOS del cuadro 1.

Función	IOS [ISR-G1, ISR-G2]	IOS-XE [ASR1K, ISR4K]
Servidor IOS CA/PKI	'12.3(4)T'	XE 3.14.0/15.5(1)S
Renovación del certificado de servidor pki IOS	12.4(1)T	XE 3.14.0/15.5(1)S
IOS PKI HA	el 15.0(1)M	[Implicit Inter-RP Redundancy is available] NA
IOS RA para las de	15.1(3)T	XE 3.14.0/15.5(1)S

otras compañías CA

Antes de conseguir en la configuración de servidor pki, el administrador debe entender estos conceptos de la base.

Fuente autoritaria de tiempo

Una de las fundaciones de la infraestructura PKI es tiempo. El reloj del sistema define si un certificado es válido o no. Por lo tanto, en el IOS, el reloj se debe hacer autoritario o digno de confianza. Sin una fuente autoritaria de tiempo, el servidor pki puede no funcionar como se esperaba, y se recomienda altamente para hacer el reloj en el IOS autoritario usando estos métodos:

NTP (protocolo Network Time Protocol)

La sincronización del reloj del sistema con un Servidor de tiempo es la única manera verdadera de hacer el reloj del sistema digno de confianza. Un router IOS puede ser configurado como cliente NTP a un servidor NTP bien conocido y estable en la red:

```
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
```

El IOS se puede también configurar como servidor NTP, que marcará el reloj de sistema local como autoritario. En el despliegue a escala reducida PKI, el servidor pki puede ser configurado como servidor NTP para sus clientes 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
```

Reloj de hardware de marcado según lo confiado en

En el IOS, el reloj de hardware se puede marcar como usar autoritario:

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

Esto se puede configurar junto con el NTP, y la razón dominante de hacer esto es mantener el reloj del sistema autoritario cuando las recargas de router, por ejemplo debido a una interrupción de la alimentación eléctrica, y a los servidores NTP no son accesibles. En esta etapa, los temporizadores PKI pararán el funcionar, que a su vez lleva para certificar los errores de la renovación/de la renovación. **el clock calendar-valid** actúa como salvaguardia en tales situaciones.

Mientras que configura esto, es dominante entender que el reloj del sistema saldrá de sincronice si muere la batería del sistema, y el PKI comenzará a confiar en un reloj del hacia fuera-de-sincronizar. Sin embargo, es relativamente más seguro configurar esto, que no teniendo una fuente autoritaria de tiempo en absoluto.

> Nota: agregaron al **comando clock calendar-valid** en la versión XE 3.10.0/15.3(3)S IOS-XE hacia adelante.

Nombre de host y Domain Name

Se recomienda para configurar un nombre de host y un Domain Name en el Cisco IOS como uno de los primeros pasos antes de configurar cualesquiera servicios relacionados PKI. El nombre del host del router y el Domain Name se utilizan en los escenarios siguientes:

- El nombre predeterminado del par clave RSA es derivado combinando el nombre de host y el Domain Name
- Al alistar para un certificado, el tema-nombre predeterminado consiste en el atributo del nombre de host y un no estructurado-nombre, que es nombre de host y Domain Name juntados.

En cuanto al servidor pki, el nombre de host y el Domain Name no se utilizan:

- El nombre predeterminado del par clave será lo mismo que el del nombre de servidor pki
- El Tema-nombre predeterminado consiste en el CN, que es lo mismo que el del nombre de servidor pki.

La recomendación general es configurar un nombre de host apropiado y un Domain Name.

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

Servidor HTTP

Habilitan al servidor pki IOS solamente si habilitan al servidor HTTP. Es importante observar que, si el servidor pki es inhabilitado debido al servidor HTTP que es inhabilitado, puede continuar

concediendo el [via terminal] offline de las peticiones. La capacidad del servidor HTTP se requiere para procesar las peticiones SCEP, y envía las respuestas SCEP.

Habilitan al servidor HTTP IOS usando:

```
ip http server
```

Y el puerto de servidor HTTP predeterminado se puede cambiar a partir del 80 a cualquier número del puerto válido usando:

```
ip http port 8080
```

MAX-conexión HTTP

Uno de los embotellamientos, mientras que el IOS que despliega como servidor pki que usa el SCEP es conexiones HTTP simultáneas máximas y conexiones HTTP medias por el minuto. Actualmente, las conexiones concurrentes máximas en un servidor HTTP IOS se limitan a 5 por abandono y se pueden aumentar a 16, que se recomienda altamente en un despliegue de la escala media:

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

Las instalaciones este IOS permiten las conexiones HTTP simultáneas máximas hasta 1000:

- IOS UniversalK9 con el licencia-conjunto uck9

El CLI se cambia automáticamente para validar un argumento numérico entre 1 a 1000

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

El servidor HTTP IOS permite 80 conexiones por el minuto [580 en el caso de las versiones del IOS donde las sesiones concurrentes máximas HTTP pueden ser aumentadas a 1000] y cuando este límite se alcanza dentro de un minuto, módulo de escucha IOS HTTP comienza a estrangular las conexiones HTTP entrantes apagando al módulo de escucha por 15 segundos. Esto lleva a las peticiones de conexión cliente que son caído debido al **límite de la** cola de conexiones TCP **alcanzado**. Más información sobre esto se puede encontrar aquí

Par clave RSA

El par clave RSA para las funciones del servidor pki en el IOS puede auto-ser generado o ser generado manualmente.
Mientras que configura a un servidor pki, el IOS crea automáticamente un trustpoint por el mismo nombre que el servidor pki para salvar el certificado de servidor pki.

Manualmente generación del par clave del servidor pki RSA:

Paso 1. Cree un par clave RSA con el mismo nombre que el del servidor pki:

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

Paso 2. Antes de habilitar al servidor pki, modifique el trustpoint del servidor pki:

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

> Nota: El valor del módulo del par clave RSA mencionado bajo el trustpoint del servidor pki no se toma en la consideración hasta el ver 15.4(3)M4 IOS, y esto es advertencias conocidas. El módulo de la clave predeterminada es 1024 bits.

Auto-generación del par clave del servidor pki RSA:

Al habilitar al servidor pki, el IOS genera automáticamente un par clave RSA con el mismo nombre que el del servidor pki, y el tamaño dominante del módulo es 1024 bits.

Comenzando el ver 15.4(3)M5 IOS, esta configuración crea un par clave RSA con **<LABEL>** pues el nombre y la invulnerabilidad de la clave estarán según el módulo definido del **<MOD>**.

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

Alerón

El servidor pki IOS CSCuu73408 debe tener en cuenta el tamaño de clave no valor por defecto para el CERT de la renovación.

El servidor pki IOS CSCuu73408 debe tener en cuenta el tamaño de clave no valor por defecto para el CERT de la renovación.

Los estándares actuales de la industria son utilizar un mínimo de par clave de 2048 bits RSA.

consideración del temporizador de la Auto-renovación

Actualmente, el servidor pki IOS no genera un certificado de la renovación por abandono, y tiene que ser habilitado explícitamente bajo el servidor pki que usa el comando del **<days-before-expiry> de la auto-renovación**. Más en la renovación del certificado se explica adentro

Este comando especifica cuántos días antes de que el vencimiento del certificado PKI Server/CA si el IOS crea un certificado de CA de la renovación. Observe que expira el certificado de CA de la renovación está activado una vez el certificado de CA activo actual. El valor predeterminado es actualmente 30 días. Este valor se debe fijar a un valor razonable dependiendo del curso de la vida del certificado de CA, y éste a su vez influencia la configuración del temporizador del auto-alistar en el cliente PKI.

> Nota: el temporizador de la Auto-renovación debe accionar siempre antes de auto-alista el temporizador en el cliente durante CA y el [known as] de la renovación del certificado del cliente

Consideraciones CRL

La infraestructura IOS PKI soporta dos maneras de distribuir el CRL:

Publique el CRL a un servidor HTTP

El servidor pki IOS puede ser configurado para publicar el archivo CRL a una ubicación específica en un servidor HTTP que usa este comando bajo el servidor pki:

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

Y el servidor pki puede ser configurado para integrar esta ubicación CRL en todos los certificados del cliente PKI usando este comando bajo el servidor pki:

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

Método SCEP GetCRL

El servidor pki IOS salva automáticamente el archivo CRL en la ubicación de la base de datos específica, que por abandono es nvram, y se recomienda altamente guardar una copia en un servidor SCP/FTP/TFTP usando este comando bajo el servidor pki:

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

Por abandono, el servidor pki IOS no integra la ubicación CDP en los certificados del cliente PKI. Si configuran a los clientes IOS PKI para realizar el control de la revocación, pero el certificado que es validado no tiene un CDP integrado en él, y el trustpoint de CA que valida se configura con la ubicación de CA (usando el <CA-Servidor-IP o FQDN> de http://), las caídas IOS de nuevo al método basado SCEP de GetCRL por abandono.
El SCEP GetCRL realiza la extracción CRL ejecutando HTTP GET en este URL:

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

> Nota: ¿En IOS CLI, antes de ingresar? , secuencia clave del Ctrl +V de la prensa.

El servidor pki IOS puede también integrar este URL como la ubicación CDP. La ventaja de hacer esto es doble:

- Se asegura de que todos los clientes basados SCEP del no IOS PKI puedan realizar la extracción CRL.
- Sin un CDP integrado, los mensajes request IOS SCEP GetCRL se firman (usando un certificado autofirmado temporal) según lo definido en el proyecto SCEP. Sin embargo, los pedidos de recuperación CRL no necesitan ser firmados, e integrando el CDP URL para el método de GetCRL, la firma de las peticiones CRL puede ser evitada.

Curso de la vida del CRL

La vida útil de CRL del servidor pki IOS puede ser controlada usando este comando bajo el servidor pki:

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

El valor es en las horas. Por abandono el curso de la vida del CRL se fija a 6 horas. Dependiendo de cómo los Certificados se revocan con frecuencia, ajustar la vida útil de CRL a un valor óptimo aumenta el funcionamiento de la extracción CRL en la red.

Consideraciones de la base de datos

El servidor pki IOS utiliza el nvram como la ubicación de la base de datos predeterminada, y se recomienda altamente para utilizar un servidor FTP o TFTP o de SCP como la ubicación de la base de datos. Por abandono, el servidor pki IOS crea dos archivos:

- <Server-Name>.ser – Esto contiene el número de serie más reciente publicado por CA en el maleficio. El archivo está en el formato de texto sin formato, y contiene esta información:
  db_version = 1
  last_serial = 0x4

- <Server-Name>.crl – Éste es el archivo codificado DER CRL publicado por CA

El servidor pki IOS salva la información en la base de datos en 3 niveles configurables:

- Mínimo – Éste es el nivel predeterminado, y a este nivel no se crea ningún archivo en la base de datos, y por lo tanto no hay información disponible en el servidor de CA con respecto a los certificados del cliente concedida en el pasado.
- Nombres – A este nivel el servidor pki IOS crea un archivo nombrado <Serial-Number>.cnm para cada certificado del cliente publicado, donde el <Serial-Number> del nombre refiere al número de serie del certificado del cliente publicado y este archivo del cnm contiene el tema-nombre y la fecha de vencimiento del certificado del cliente.
- Completo – A este nivel, el servidor pki IOS crea dos archivos para cada certificado del cliente publicado:
- <Serial-Number>.cnm
- <Serial-Number>.crt

aquí, el archivo CRT es el archivo de certificado del cliente, que es DER codificado.

Estas puntas son importantes:

- Antes de publicar un certificado del cliente, el servidor pki IOS refiere a <Server-Name>.ser para determinar y para derivar el número de serie del certificado.
- Con el conjunto del nivel de la base de datos a los nombres o necesidad complete, <Serial-Number>.cnm y <Serial-Number>.crt de ser escrito a la base de datos antes de enviar el certificado concedido/publicado al cliente
- Con el conjunto URL de la base de datos a los nombres o complete, la base de datos URL

debe tener bastante espacio para salvar los archivos. Por lo tanto la recomendación es configurar un [FTP or TFTP or SCP] del servidor de archivo externo como la base de datos URL.
- Con la base de datos externa URL configurada, es absolutamente necesario aseegurarse que el servidor de archivos es accesible durante el proceso de la concesión del certificado, que marcaría de otra manera el servidor de CA como minusválidos. Y la intervención manual se requiere para traer la parte posterior del servidor de CA en línea.

Archivo de la base de datos

Mientras que despliega a un servidor pki, es importante considerar los escenarios de falla y ser preparado, debe haber un error del hadrware. Existes dos maneras para lograr esto:

1. Redundancia
   En este caso, dos dispositivos o unidades de procesamiento actúan como Activo-espera para proporcionar la Redundancia.
   El servidor pki IOS de gran disponibilidad puede ser alcanzado usando dos Routers habilitado HSRP ISR [ISR G1 y ISR G2] como se explica en

   El IOS XE basó los sistemas [ISR4K y ASR1k] no tiene opción de la redundancia del dispositivo disponible. Sin embargo, en el inter RP ASR1k la Redundancia está disponible por abandono.
2. Archivar el par clave y los archivos del servidor de CA
   El IOS proporciona un recurso para archivar el par clave del servidor pki y el certificado. El archivar se puede hacer usando dos tipos de archivos:
   PEM - El IOS crea los archivos formatados PEM para salvar la clave pública RSA, clave privada cifrada RSA, certificado de servidor de CA. El par clave y los Certificados de la renovación están archivados automáticamentePKCS12 - El IOS crea un solo archivo del PKCS12 que contiene el certificado de servidor de CA y la clave privada correspondiente RSA cifrados usando una contraseña.

El archivar de la base de datos se puede habilitar usando este comando bajo el servidor pki:

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

Es también posible salvar los ficheros de archivo a un servidor separado, posiblemente usando un protocolo seguro (SCP) usando el siguiente comando bajo el servidor pki:

```
crypto pki server <PKI-SERVER-Name>
 database url {p12 | pem} <URL>
```

De todos los archivos en la base de datos a excepción de los ficheros de archivo y. El archivo de Ser, el resto de los archivos está en el texto claro y no plantea ninguna amenaza real si está perdido, y por lo tanto se puede salvar en un servidor separado sin incurrir en muchos gastos indirectos mientras que escribe los archivos, por ejemplo un servidor TFTP.

IOS como Sub-CA

El servidor pki IOS por abandono toma el papel de a raíz CA. Para configurar a un servidor pki subordinado (Sub-CA), primero habilite este comando conforme a la sección de configuración del servidor pki (antes de habilitar al servidor pki):

```
crypto pki server <Sub-PKI-SERVER-Name>
 mode sub-cs
```
Usando esta configuración el URL de Raíz-CA bajo el trustpoint del servidor pki:

```
crypto pki trustpoint <Sub-PKI-SERVER-Name>
 enrollment url <Root-CA URL>
```
Habilitar a este servidor pki ahora acciona estos eventos:

- El trustpoint del servidor pki se autentica para instalar certificado raíz CA.
- Después de que raíz CA se autentique, el IOS genera un CSR para el obstáculo básico de Subordinado-CA [x509 que contiene CA: El indicador VERDADERO] y lo envía al raíz CA

Con independencia del modo de la concesión configurado en raíz CA, el IOS pone los pedidos de certificado de CA (o RA) en la cola pendiente. Un administrador tiene que conceder manualmente los Certificados de CA.
Para ver el pedido de certificado pendiente y la petición-identificación:

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

Para conceder la petición:

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

- Usando esto, la operación subsiguiente de la ENCUESTA SCEP (GetCertInitial) descarga el certificado de Sub-CA y lo instala en el router, que habilita al servidor pki subordinado

IOS como RA

El servidor pki IOS puede ser configurado como autoridad de registro a un subordinado dado o raíz CA. Para configurar al servidor pki como autoridad de registro, primero habilite este comando conforme a la sección de configuración del servidor pki (antes de habilitar al servidor pki):

```
crypto pki server <RA-SERVER-Name>
 mode ra
```
Después de esto, configure el URL de CA bajo el trustpoint del servidor pki. Esto indica qué CA es protegido por el RA:

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```
Una autoridad de registro no publica los Certificados, por lo tanto la configuración del **nombre del emisor** bajo el RA no se requiere, y no es eficaz incluso si se configura. El tema-nombre de un RA se configura bajo el trustpoint RA usando el **comando subject-name**. Es importante configurar **OU = los ioscs RA** como parte del tema-nombre para que el IOS CA para identificar el IOS RA es decir para identificar los pedidos de certificado autorizados por el IOS RA.

El IOS puede actuar como autoridad de registro a las de otras compañías CA tales como Microsoft CA, y para permanecer compatible el IOS RA tiene que ser habilitado usando este comando conforme a la sección de configuración del servidor pki (antes de habilitar al servidor pki):

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```
En el modo predeterminado RA, el IOS firma los pedidos de cliente [PKCS#10] usando el

certificado RA. Esta operación indica al servidor pki IOS que el pedido de certificado ha sido autorizado por un RA.

Con el modo transparente RA, el IOS adelante los pedidos de cliente en su formato original sin la introducción del certificado RA, y éste es compatibles con Microsoft CA como ejemplo bien conocido.

Cliente IOS PKI

Uno de la entidad de configuración más importante del cliente IOS PKI es un trustpoint. Los parámetros de la configuración del trustpoint se explican detalladamente en esta sección.

Fuente autoritaria de tiempo

Como fuente anterior, autoritaria señalada de tiempo está un requisito en el cliente PKI también. El cliente IOS PKI puede ser configurado como usar del cliente NTP este configuración:

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

Nombre de host y Domain Name

Una recomendación general es configurar un nombre de host y un Domain Name en el router:

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

Par clave RSA

En el cliente IOS PKI, el par clave RSA para una inscripción dada del trustpoint puede ser generado automáticamente o ser generado manualmente.

El proceso de generación de claves automático RSA implica el siguiente:

- El IOS por abandono crea el par clave de 512 bits RSA
- El nombre automáticamente generado del par clave es hostname.domain-name, que es el nombre del host del dispositivo combinado con el Domain Name del dispositivo
- El par clave auto-generado no se marca como exportable.

El proceso de generación de claves automático RSA implica el siguiente:

- Opcionalmente, un par clave de los fines generales RSA de una fuerza conveniente se puede generar manualmente usando:
- `crypto pki trustpoint <RA-SERVER-Name>`
  `enrollment url <CA URL>`
  `subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco`Aquí, ESCRITURA DE LA ETIQUETA - el nombre del par clave RSA
  Mod - El módulo o la fuerza de la clave RSA en los bits entre 360 labra 4096, que son tradicionalmente 512, 1024, 2048 o 4096.
  La ventaja manualmente de generar el par clave RSA es la capacidad de marcar el par clave como exportable, que a su vez permite para que el certificado de identidad sea exportado

totalmente, que se puede entonces restablecer en otro dispositivo. Sin embargo, uno debe entender las consecuencias en la seguridad de esta acción.

- Un par clave RSA se conecta a un trustpoint antes de la inscripción usando este comando

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

Aquí, si existe un par clave RSA nombrado <LABEL> ya, después se coge durante la inscripción del trustpoint.

Si no existe un par clave RSA nombrado <LABEL>, después uno de la acción siguiente se ejecuta durante la inscripción:

- Si no se pasa ningún argumento del <MOD>, después se generan 512 bits <LABEL> nombrado par clave.

- si se pasa un argumento del <MOD>, después un par clave de fines generales de los bits del <MOD> nombrado <LABEL> se genera

- si se pasan dos argumentos del <MOD>, después se generan un par clave de la firma de los bits del <MOD> y un par clave del cifrado de los bits del <MOD>, ambos <LABEL> Nombrados

Trustpoint

Un trustpoint es un envase abstracto para sostener un certificado en el IOS. Un solo trustpoint es capaz de salvar dos Certificados activos en cualquier momento:

- Un certificado de CA - Cargando un certificado de CA en un trustpoint dado se conoce como proceso de autenticación del trustpoint.
- Un certificado ID publicado por CA - el cargamento o la importación de un certificado ID en un trustpoint dado se conoce como proceso de la inscripción del trustpoint.

Una configuración del trustpoint se conoce como directiva de confianza, y ésta define eso:

- ¿Qué certificado de CA se carga en el trustpoint?
- ¿Qué CA el trustpoint alista?
- ¿Cómo el IOS alista el trustpoint?
- ¿Cómo un certificado publicado por el [loaded in the trustpoint] dado de CA se valida?

Explican a los componentes principales de un trustpoint aquí.

Modo de la inscripción

Un modo de la inscripción del trustpoint, que también define al modo de autenticación del trustpoint, se puede realizar vía 3 métodos principales:

1. Inscripción terminal - método manual de realizar la autenticación y la inscripción del certificado del trustpoint usando la copia-goma en la terminal CLI.
2. Inscripción SCEP - Autenticación e inscripción del trustpoint usando el SCEP sobre el HTTP.
3. Perfil de la inscripción - Aquí, los métodos de la autenticación y de la inscripción se definen por separado. Junto con la terminal y los métodos de la inscripción SCEP, los perfiles de la inscripción proporcionan una opción para especificar los comandos HTTP/TFTP de realizar la extracción de archivo del servidor, que se define usando un URL de la autenticación o de la inscripción bajo perfil.

Interfaz de origen y VRF

La autenticación y la inscripción del trustpoint sobre HTTP (SCEP) o TFTP (perfil de la inscripción) utiliza el archivo de sistema de IOS para realizar las operaciones entrada-salida del archivo. Estos intercambios de paquetes pueden ser originados de una interfaz de origen específica y de un VRF.

En caso de la configuración clásica del trustpoint, estas funciones se habilitan usando los submandatos de la **interfaz de origen** y del **vrf** bajo el trustpoint.

En caso de los perfiles, de la **interfaz de origen** y de la **inscripción de la** inscripción **| el URL <http/tftp://Server-location de la autenticación > los** comandos del **<vrf-name> del vrf** proporcionan las mismas funciones.

Ejemplo de configuración:

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

o

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

Inscripción del certificado y renovación automáticas

El cliente IOS PKI puede ser configurado para realizar el enlistamiento automático y la renovación usando este comando bajo sección del trustpoint PKI:

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

Aquí, auto-**aliste a los** commandes sates del **[regenerate] del <percentage>** que el IOS debe realizar la renovación del certificado en el exactamente 80% del curso de la vida del certificado actual.

Los estados del **regenerado de la** palabra clave que el IOS debe regenerar el par clave RSA conocido como par clave de la sombra durante cada operación de la renovación del certificado.

Éste es el comportamiento del enlistamiento automático:

- Se configura el momento auto-**alista**, si se autentica el trustpoint, IOS realizará un enlistamiento automático al servidor situado en el URL mencionado como parte del **comando enrollment url** bajo sección del trustpoint PKI o bajo perfil de la inscripción.
- El momento que un trustpoint se alista con un servidor pki o CA, una RENOVACIÓN o un temporizador de la SOMBRA se inicializa en el cliente PKI basaron en el porcentaje del auto-**alistar del** certificado de identidad actual instalaron bajo el trustpoint. Este temporizador es visible bajo comando **crypto del temporizador del pki de la demostración**. Más en las funciones del temporizador *se refieren*
- El soporte de la capacidad de la renovación viene del servidor pki. Más en esto adentro
  El cliente IOS PKI realiza dos tipos de renovación:
  Renovación implícita: Si el servidor pki no envía la “renovación” como capacidad soportada,

el IOS realiza una inscripción inicial en definida auto-alista el porcentaje. es decir el IOS utiliza un certificado autofirmado para firmar el pedido de renovación.Renovación explícita: Cuando el servidor pki soporta la característica de la renovación del certificado del cliente PKI, hace publicidad de la “renovación” como capacidad soportada. El IOS toma esta capacidad en la consideración durante el IOS del renewal del certificado es decir utiliza el certificado de identidad activo actual para firmar el pedido de certificado del renewal.

El cuidado debe ser tomado mientras que configura auto-aliste el porcentaje. En cualquier cliente dado PKI en el despliegue, si se presenta una condición donde el certificado de identidad expira al mismo tiempo que el certificado de CA de publicación, después el valor del auto-alistar debe accionar siempre la operación de la renovación del [shadow] después de que CA haya creado el certificado de la renovación. *Refiera a la* sección de las **dependencias del temporizador PKI** adentro

Certifique el Revocación-control

Un trustpoint autenticado PKI es decir un trustpoint PKI que contiene un certificado de CA es capaz de realizar la validación de certificado durante un IKE o una negociación SSL, donde el certificado de peer se sujeta a una validación de certificado completa. Uno de los métodos de la validación es marcar el estado de anulación del certificado de peer que usa uno de los dos métodos siguientes:

- Listas de revocación de certificados (CRL) - Esto es un archivo que contiene los números de serie de los Certificados revocados por CA dado. Este archivo se firma usando el certificado de CA de publicación. El método CRL implica el descargar del archivo CRL usando el HTTP o el LDAP.

- Protocolo status en línea del certificado (OCSP) - El IOS establece el canal de comunicación con una entidad llamada como respondedor OCSP, que es un servidor señalado por CA de publicación. Un cliente tal como IOS envía una petición que contiene el número de serie del certificado se está validando. El respondedor OCSP responde con el estado de anulación del número de serie dado. El canal de comunicación se podría establecer usando cualquier aplicación compatible/el Transport Protocol, que sea generalmente HTTP.

El control de la revocación se puede definir usando éstos ordena bajo sección del trustpoint PKI:

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

Por abandono, un trustpoint se configura para realizar el control de la revocación usando el crl.

Los métodos pueden ser reordenados, y el control del estado de anulación se realiza en la orden definida. El método “ningunos” desvía el revocación-control.

Caché CRL

Con el revocación-control basado CRL, cada validación de certificado puede accionar una descarga fresca del archivo CRL. Y como el archivo CRL consigue más grande o si el CRL Distribution Point (CDP) está más lejos ausente, descargar el archivo durante cada proceso de validación obstaculiza el funcionamiento del según el protocolo en la validación de certificado. Por lo tanto, el almacenamiento en memoria inmediata CRL se realiza para mejorar el funcionamiento,

y el almacenamiento en memoria inmediata del CRL toma la validez CRL en la consideración.

La validez CRL se define usando dos parámetros del tiempo: **LastUpdate**, que es la última vez el CRL fue publicado por CA de publicación, y **NextUpdate**, que es el tiempo es el futuro en que una nueva versión del archivo CRL es publicada por CA de publicación.

El IOS oculta cada CRL descargado para mientras el CRL sea válido. Sin embargo, en determinadas circunstancias por ejemplo el CDP que no es accesible temporalmente, puede ser necesario conservar el CRL en el caché durante un largo período de tiempo. En el IOS un CRL ocultado puede conservado para mientras 24 horas después de que expira la validez CRL, y esto se pueda configurar usando este comando bajo sección del trustpoint PKI:

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

En determinadas circunstancias por ejemplo CA de publicación que revoca los Certificados dentro del período de validez CRL, el IOS puede confiugured para borrar el caché más con frecuencia. Borrando el CRL prematuramente, el IOS se fuerza para descargar el CRL más con frecuencia para mantener el caché CRL actualizado. Esta opción de configuración está disponible bajo sección del trustpoint PKI:

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

Y finalmente, el IOS se puede configurar para no ocultar el archivo CRL usando este comando bajo sección del trustpoint PKI:

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

Configuración recomendada

Un despliegue típico de CA con raíz CA y una configuración de Sub-CA está como abajo. El ejemplo también incluye una configuración de Sub-CA protegida por un RA.

Con el par clave de 2048 bits RSA en todos los ámbitos, este ejemplo recomienda una configuración donde:
Raíz CA tiene un curso de la vida de 8 años
Sub-CA tiene un curso de la vida de 3 años
Los certificados del cliente se publican por un año, que se configuran para pedir para un renewal del certificado, automáticamente.

RAÍZ CA - Configuración

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

SUBCA sin el RA - Configuración

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

SUBCA con el RA - Configuración

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

RA para SUBCA - Configuración

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

Inscripción del certificado

Registro manual

El Registro manual implica la generación offline CSR en el cliente PKI, que se copia manualmente encima al administrador CA firma manualmente la petición, que entonces se importa en el cliente.

Cliente PKI

Configuración del cliente PKI:

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

Paso 1. Primero autentique el trustpoint (esto se puede también realizar después del paso 2).

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

Paso 2. Genere el pedido de firma de certificado y lleve el CSR CA y consiga el certificat concedido:

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

Paso 3. Ahora importe el certificado concedido vía la terminal:

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

Servidor pki

Paso 1. Primero exporte el certificado de CA de publicación de CA, que en este caso es certificado SUBCA. Esto se importa durante el paso 1 antedicho en el cliente PKI, es decir autenticación del 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-----
```

Paso 2. Después de Step-2 en el PKI-cliente, tome el CSR del cliente y proporcionelo para firmar en el SUBCA usando este comando:

```
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-----
```

Este comando sugiere que el SUBCA valide un pedido de firma de certificado de la terminal, y concedido una vez, los datos del certificado se imprimen en el formato PEM.

```
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-----
```

Si CA está en el modo de la auto-concesión, el certificado concedido se visualiza en el formato PEM arriba. Cuando CA está en el modo de concesión manual, el pedido de certificado se marca como **pendiente**, se asigna un valor identificación y se hace cola en la cola de petición de la inscripción.

```
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-----
```

Paso 3. Conceda manualmente esta petición usando este comando:

```
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-----
```

> Nota: El Registro manual de Sub-CA a raíz CA no es posible.

> Nota: CA en un estado inhabilitado debido inhabilitó al servidor HTTP puede conceder manualmente los pedidos de certificado.

Inscripción usando el SCEP

La configuración del cliente PKI es:

```
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
```

La configuración de servidor pki es:

```
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
```

El modo predeterminado de concesión del pedido de certificado es manual:

```
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
```

Concesión manual

Paso 1. Cliente PKI: En primer lugar, que es obligatorio, autentique el trustpoint en el cliente PKI:

```
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
```

Paso 2. PKI-cliente: Después de la autenticación del trustpoint, el cliente PKI puede ser alistado

para un certificado.

Nota: Si auto-aliste se configura, el cliente realiza automáticamente la inscripción.

```
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
```

Detrás de las escenas, estos eventos ocurren:

- El IOS busca un par clave RSA nombrado PKI-Clave. Si existe, se coge para pedir un certificado de identidad. Si no, el IOS crea 2048 un bit PKI-Clave nombrada par clave, y después lo utiliza para pedir un certificado de identidad.

- El IOS crea un pedido de firma de certificado en el formato PKCS10.

- El IOS entonces cifra este CSR usando una clave simétrica al azar. La clave simétrica al azar se cifra usando la clave pública del beneficiario, que es el SUBCA (la clave pública SUBCA es disponible debido a la autenticación del trustpoint). El CSR cifrado, la clave simétrica al azar cifrada y la información del receptor se pone junta en los datos PKCS-7 envueltos.

- Estos datos PKCS-7 envueltos se firman usando un certificado autofirmado temporal durante la inscripción inicial. Los datos PKCS-7 envueltos, el certificado de firma usado por el cliente y la firma del cliente se juntan en un paquete de datos PKCS-7 firmado. Éste es base64 codificado, y entonces URL codificado. La gota resultante de los datos se envía como argumento del “mensaje” en HTTP URI enviado a CA:

  ```
  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
```

Paso 3. Servidor pki:

Cuando el servidor pki IOS recibe la petición, marca éstos:

1. Marca si la base de datos de la petición de la inscripción contiene un pedido de certificado con el mismo ID de transacción asociado a la nueva petición.

> Nota: Un ID de transacción es un hash MD5 de la clave pública, para la cual un certificado de identidad está siendo pedido por el cliente.

2. Marca si la base de datos de la petición de la inscripción contiene un pedido de certificado con la misma contraseña de impugnación que la que está enviada por el cliente.

> Nota: Si (1) las devoluciones verdad o (1) y (2) juntas verdad de vuelta, después un servidor de CA es capaz de rechazar la petición sobre la base de la petición duplicado de la identidad. Sin embargo, en tal caso el servidor pki IOS substituye la más vieja petición por la más nueva petición.

Paso 4. Servidor pki:

Conceda manualmente las peticiones en el servidor pki:

Para ver la petición:

```
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
```

Para conceder una petición específica o todas las peticiones:

```
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
```

Paso 5. PKI-cliente:

Mientras tanto, un cliente PKI comienza un temporizador de la ENCUESTA. Aquí, el IOS realiza GetCertInitial a intervalos regulares hasta que el SCEP CertRep = CONCEDIDO junto con el certificado concedido es recibido por el cliente.

Una vez que se recibe el certificado concedido, el IOS lo instala automáticamente.