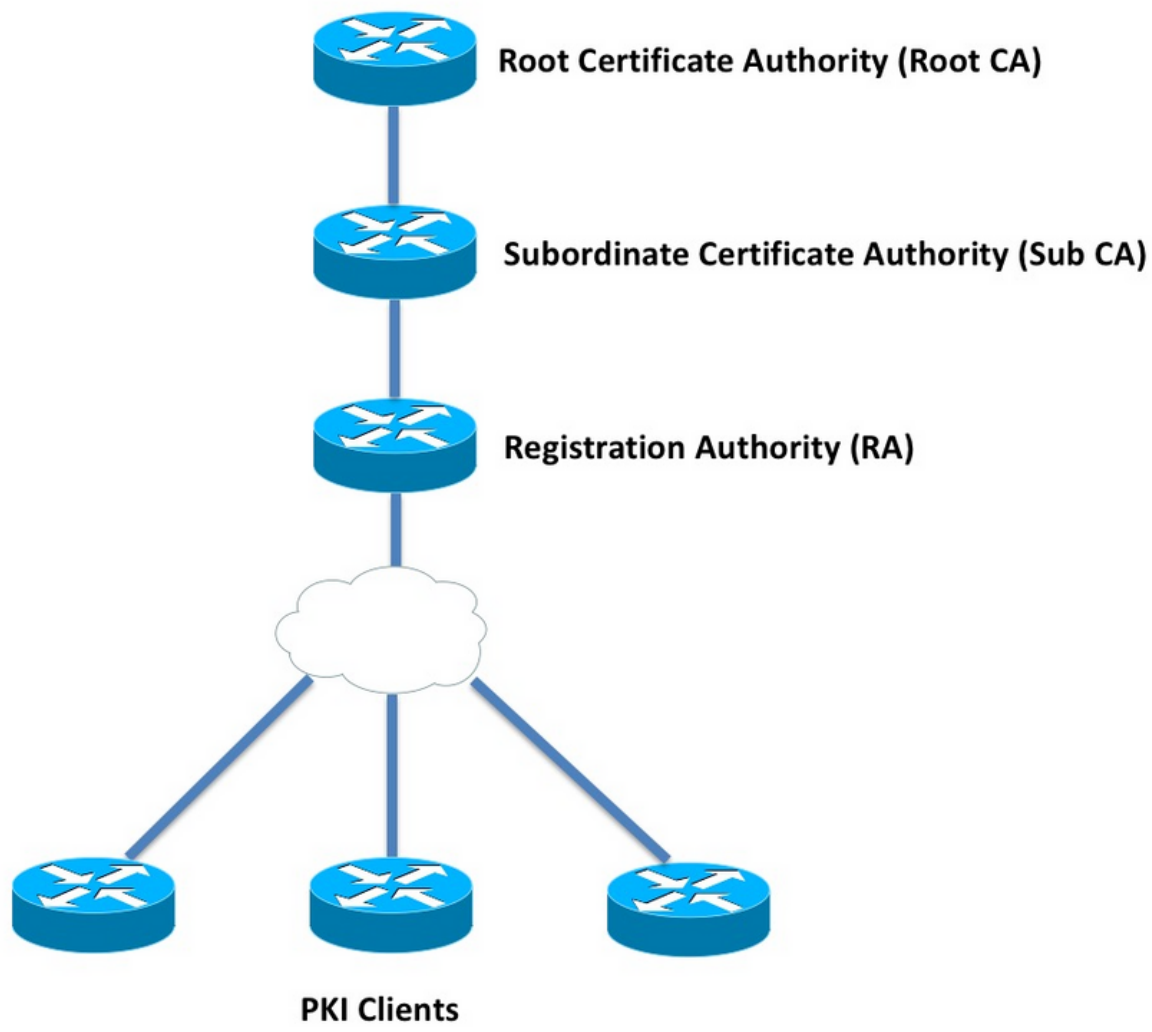
Guía de implementación de PKI de IOS: Diseño e implementación iniciales

Contenido

Introducción

Este documento describe detalladamente las funcionalidades del cliente y del servidor PKI de IOS. Aborda las consideraciones iniciales de diseño e implementación de la PKI de IOS.

Infraestructura PKI



Autoridad de certificados

La autoridad certificadora (CA), también denominada servidor PKI en todo el documento, es una

entidad de confianza que emite certificados. La PKI se basa en la confianza y la jerarquía de confianza comienza en la autoridad de certificados raíz (Root-CA). Debido a que la CA raíz está en la parte superior de la jerarquía, tiene un certificado autofirmado.

Autoridad de certificados subordinada

En la jerarquía de confianza de PKI, todas las autoridades de certificados siguientes a raíz se conocen como Autoridades de certificados subordinadas (Sub-CA). Evidentemente, la CA emite un certificado Sub-CA, que es un nivel anterior.

La PKI no impone ningún límite al número de sub-CA en una jerarquía determinada. Sin embargo, en una implementación empresarial con más de 3 niveles de autoridades certificadoras puede resultar difícil de gestionar.

Autoridad de registro

PKI define una autoridad de certificación especial conocida como Autoridad de Registro (RA), que es responsable de autorizar a los clientes PKI a inscribirse en una sub-CA o raíz-CA determinada. RA no emite certificados a clientes PKI, sino que decide qué cliente PKI puede o no puede ser emitido por la Sub-CA o la CA raíz.

La función principal de una RA es descargar la validación básica de la solicitud de certificado de cliente de la CA y proteger la CA de la exposición directa a los clientes. De esta manera, RA se interpone entre los clientes PKI y la CA, protegiendo así a la CA de cualquier tipo de ataques de denegación de servicio.

Cliente PKI

Cualquier dispositivo que solicite un certificado basado en un par de claves público-privada residente para probar su identidad a otros dispositivos se conoce como cliente PKI.

Un cliente PKI debe ser capaz de generar o almacenar un par de claves público-privadas como RSA o DSA o ECDSA.

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

Servidor PKI de IOS

Tabla 1. Evolución de la Función del Servidor PKI de IOS

Función	IOS [ISR-G1, ISR-G2]	IOS-XE [ASR1K, ISR4K]
Servidor CA/PKI de IOS	‘12.3(4)T’	XE 3.14.0/15.5(1)S
Renovación de certificado de servidor PKI de IOS	12.4(1)T	XE 3.14.0/15.5(1)S
IOS PKI HA	15.0(1)M	NA [La redundancia implícita entre RP está disponible]

IOS RA para CA de 3 terceros	15.1(3)T	XE 3.14.0/15.5(1)S	

Antes de entrar en la configuración del servidor PKI, el administrador debe entender estos conceptos centrales.

Fuente de tiempo autorizada

Uno de los cimientos de la infraestructura de PKI es Time. El reloj del sistema define si un certificado es válido o no. Por lo tanto, en IOS, el reloj debe convertirse en autoritativo o confiable. Sin una fuente de tiempo autorizada, es posible que el servidor PKI no funcione como se espera, y se recomienda encarecidamente hacer que el reloj en IOS sea autoritario usando estos métodos:

NTP (protocolo de tiempo de red)

Sincronizar el reloj del sistema con un servidor de hora es la única forma real de hacer que el reloj del sistema sea fiable. Un router IOS se puede configurar como cliente NTP a un servidor NTP 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
```

IOS también se puede configurar como un servidor NTP, que marcará el reloj del sistema local como autoritativo. En la implementación de PKI a pequeña escala, el servidor PKI se puede configurar 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
```

Marcación del reloj de hardware como fiable

En IOS, el reloj de hardware se puede marcar como autoritativo usando:

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

Esto se puede configurar junto con NTP, y la razón clave para hacerlo es mantener el reloj del sistema autorizado cuando se recarga un router, por ejemplo debido a una interrupción de la alimentación, y los servidores NTP no son accesibles. En esta etapa, los temporizadores PKI dejarán de funcionar, lo que a su vez conduce a fallas de Renovación/Renovación de Certificados. **clock calendar-valid** actúa como salvaguardia en tales situaciones.

Al configurar esto, es clave comprender que el reloj del sistema se quedará sin sincronización si la batería del sistema muere, y PKI comenzará a confiar en un reloj fuera de sincronización. Sin embargo, es relativamente más seguro configurarlo que no tener una fuente de tiempo autorizada.

> **Nota: clock calendar-valid** se agregó en IOS-XE versión XE 3.10.0 / 15.3(3)S posterior.

Nombre de host y nombre de dominio

Se recomienda configurar un nombre de host y un nombre de dominio en Cisco IOS como uno de los primeros pasos antes de configurar cualquier servicio relacionado con PKI. El nombre de host y el nombre de dominio del router se utilizan en los siguientes escenarios:

- El nombre predeterminado del par de llaves RSA se deriva combinando el nombre de host y el nombre de dominio
- Al inscribirse en un certificado, el nombre de asunto predeterminado consiste en el atributo hostname y un nombre no estructurado, que es nombre de host y nombre de dominio juntos.

En cuanto al servidor PKI, el nombre de host y el nombre de dominio no se utilizan:

- El nombre predeterminado del par de llaves será el mismo que el del nombre del servidor PKI
- El nombre de asunto predeterminado consiste en CN, que es el mismo que el nombre del servidor PKI.

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

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

Servidor HTTP

El servidor PKI de IOS está habilitado sólo si el servidor HTTP está habilitado. Es importante tener en cuenta que, si el servidor PKI está inhabilitado debido a que el servidor HTTP está inhabilitado, puede continuar concediendo solicitudes sin conexión [a través de terminal]. Se requiere la capacidad de servidor HTTP para procesar solicitudes SCEP y enviar respuestas SCEP.

El servidor HTTP de IOS está habilitado mediante:

```
ip http server
```

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

```
ip http port 8080
```

HTTP Max-connection

Uno de los cuellos de botella, mientras se implementa IOS como servidor PKI usando SCEP es el número máximo de conexiones HTTP simultáneas y el promedio de conexiones HTTP por minuto. Actualmente, el número máximo de conexiones simultáneas en un servidor HTTP de IOS está limitado a 5 de forma predeterminada y puede aumentarse a 16, lo que se recomienda en gran medida en una implementación a media escala:

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

Estas instalaciones del IOS permiten un máximo de conexiones HTTP simultáneas de hasta 1000:

- Universal K9 IOS con uck9 license-set

La CLI se cambia automáticamente para aceptar un argumento numérico entre 1 y 1000

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

El servidor HTTP de IOS permite 80 conexiones por minuto [580 en el caso de las versiones de IOS en las que las sesiones simultáneas máx. HTTP se pueden aumentar a 1000] y cuando este límite se alcanza en un minuto, el receptor HTTP de IOS comienza a limitar las conexiones HTTP entrantes cerrando el receptor durante 15 segundos. Esto lleva a que las solicitudes de conexión del cliente se descarten debido al **límite de cola de conexión** TCP. Puede encontrar más información sobre esto aquí

par de claves RSA

El par de llaves RSA para la funcionalidad del servidor PKI en IOS se puede generar automáticamente o manualmente.
Al configurar un servidor PKI, IOS crea automáticamente un punto de confianza con el mismo

nombre que el servidor PKI para almacenar el certificado del servidor PKI.

Par de clave RSA del servidor PKI que genera manualmente:

Paso 1. Cree un par de claves 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 el servidor PKI, modifique el punto de confianza del servidor PKI:

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

> **Nota:** El valor de módulo de par de llaves RSA mencionado en el punto de confianza del servidor PKI no se toma en consideración hasta IOS ver 15.4(3)M4, y esto es una advertencia conocida. El módulo de clave predeterminado es 1024 bits.

Par clave RSA de servidor PKI de generación automática:

Al habilitar el servidor PKI, IOS genera automáticamente un par de llaves RSA con el mismo nombre que el del servidor PKI, y el tamaño del módulo de claves es de 1024 bits.

A partir de IOS ver 15.4(3)M5, esta configuración crea un par de llaves RSA con **<LABEL>**, ya que el nombre y la fuerza de la clave serán según el módulo definido **<MOD>**.

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

Spoiler

CSCuu73408 El servidor PKI del IOS debe permitir un tamaño de clave no predeterminado para el certificado de renovación.

El servidor PKI del IOS CSCuu73408 debe permitir un tamaño de clave no predeterminado para la llave de sustitución de certificados.

El estándar actual del sector es utilizar un par de claves RSA mínimo de 2048 bits.

Consideración del temporizador de renovación automática

Actualmente, el Servidor PKI de IOS no genera un certificado de sustitución incremental de forma predeterminada, y debe habilitarse explícitamente bajo el servidor PKI usando el comando **auto-rollover <días antes de la expiración>**. Más información sobre la renovación de certificados se explica en

Este comando especifica cuántos días antes de que caduque el certificado de servidor PKI/CA si el IOS crea un certificado de CA de renovación. Tenga en cuenta que el certificado de CA de

renovación se activa una vez que caduque el certificado de CA activo actual. El valor predeterminado actualmente es 30 días. Este valor se debe establecer en un valor razonable dependiendo de la duración del certificado de CA, y esto a su vez influye en la configuración del temporizador de inscripción automática en el cliente PKI.

> **Nota:** El temporizador de renovación automática siempre debe activarse antes de la inscripción automática del temporizador en el cliente durante la renovación de certificados de CA y cliente [conocido como]

Consideraciones de CRL

La infraestructura PKI de IOS admite dos formas de distribuir CRL:

Publicar CRL en un servidor HTTP

El servidor PKI de IOS se puede configurar para publicar el archivo CRL en una ubicación específica de un servidor HTTP mediante este comando en el servidor PKI:

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

Y el servidor PKI se puede configurar para incrustar esta ubicación CRL en todos los certificados de cliente PKI usando este comando en el Servidor PKI:

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

Método GetCRL de SCEP

El servidor PKI de IOS almacena automáticamente el archivo CRL en la ubicación específica de la base de datos, que de forma predeterminada es nvram, y se recomienda encarecidamente conservar una copia en un servidor SCP/FTP/TFTP mediante este comando en el servidor PKI:

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

De forma predeterminada, el Servidor PKI de IOS no incrusta la ubicación CDP en los certificados de cliente PKI. Si los clientes PKI de IOS están configurados para realizar la verificación de revocación, pero el certificado que se está validando no tiene un CDP incrustado en él y el punto de confianza de CA de validación se configura con la ubicación de CA (mediante http://<CA-Server-IP o FQDN>), el IOS vuelve al método GetCRL basado en SCEP de forma predeterminada.
SCEP GetCRL realiza la recuperación de CRL ejecutando HTTP GET en esta URL:

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

Nota: En IOS CLI, antes de ingresar **?**, presione **Ctrl + V** key-sequence.

El servidor PKI de IOS también puede incrustar esta URL como la ubicación CDP. La ventaja de hacerlo es doble:

- Se asegura de que todos los clientes PKI que no están basados en SCEP de IOS puedan realizar la recuperación de CRL.
- Sin un CDP incrustado, los mensajes de solicitud GetCRL de IOS SCEP se firman (mediante un certificado autofirmado temporal) como se define en el borrador SCEP. Sin embargo, las solicitudes de recuperación de CRL no necesitan firmarse y al incrustar la URL CDP para el método GetCRL, se puede evitar la firma de las solicitudes de CRL.

Duración de CRL

La vida útil de CRL del servidor PKI de IOS se puede controlar mediante este comando en el Servidor PKI:

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

El valor es en horas. De forma predeterminada, la duración de la CRL se establece en 6 horas. En función de la frecuencia con la que se revoquen los certificados, la adaptación de la duración de CRL a un valor óptimo aumenta el rendimiento de recuperación de CRL en la red.

Consideraciones sobre la base de datos

El servidor PKI de IOS utiliza nvram como la ubicación de base de datos predeterminada y se recomienda utilizar un servidor FTP o TFTP o SCP como la ubicación de la base de datos. De forma predeterminada, el Servidor PKI de IOS crea dos archivos:

- <Server-Name>.ser: contiene el último número de serie emitido por la CA en hexadecimal. El archivo está en formato de texto sin formato y contiene esta información:
  db_version = 1
  last_serial = 0x4
- <Server-Name>.crl - Se trata del archivo CRL codificado DER publicado por la CA

El servidor PKI de IOS almacena información en la base de datos en 3 niveles configurables:

- Mínimo: este es el nivel predeterminado y, en este nivel, no se crea ningún archivo en la base de datos y, por lo tanto, no hay información disponible en el servidor de la CA sobre los certificados de cliente concedidos en el pasado.

- Nombres - En este nivel, el servidor PKI de IOS crea un archivo denominado <Serial-Number>.cnm para cada certificado de cliente emitido, donde el nombre <Serial-Number> se refiere al número de serie del certificado de cliente emitido Y este archivo cnm contiene el nombre del asunto y la fecha de vencimiento del certificado de cliente.
- Complete - En este nivel, el Servidor PKI de IOS crea dos archivos para cada certificado de cliente emitido:
- <Serial-Number>.cnm
- <Serial-Number>.crt

aquí, el archivo crt es el archivo de certificado del cliente, que está codificado en DER.

Estos puntos son importantes:

- Antes de emitir un certificado de cliente, el Servidor PKI de IOS hace referencia a <Server-Name>.ser para determinar y derivar el número de serie del certificado.
- Con el nivel de base de datos establecido en Nombres o Completos, <Serial-Number>.cnm y <Serial-Number>.crt deben escribirse en la base de datos antes de enviar el certificado concedido/emitido al cliente
- Con la url de la base de datos establecida en Nombres o Complete, la URL de la base de datos debe tener suficiente espacio para guardar los archivos. Por lo tanto, se recomienda configurar un servidor de archivos externo [FTP o TFTP o SCP] como la URL de la base de datos.
- Con la URL de base de datos externa configurada, es absolutamente necesario asegurarse de que el servidor de archivos esté accesible durante el proceso de concesión de certificados, lo que, de lo contrario, marcaría el servidor de la CA como inhabilitado. Además, se requiere una intervención manual para volver a conectar el servidor de la CA.

Archivo de base de datos

Al implementar un servidor PKI, es importante considerar los escenarios de falla y prepararse, en caso de que se produzca una falla del hardware. Existes dos maneras para lograr esto:

1. Redundancia
   En este caso, dos dispositivos o unidades de procesamiento actúan como Active-Standby para proporcionar redundancia.
   La alta disponibilidad del servidor PKI de IOS se puede lograr mediante dos routers ISR habilitados para HSRP [ISR G1 e ISR G2], como se explica en

   Los sistemas basados en IOS XE [ISR4K y ASR1k] no tienen disponible la opción de redundancia de dispositivos. Sin embargo, en ASR1k la redundancia Inter-RP está disponible de forma predeterminada.
2. Archivando el par de claves y los archivos del servidor de la CA
   IOS proporciona una función para archivar el par de claves del servidor PKI y el certificado. El archiving se puede realizar utilizando dos tipos de archivos:
   PEM: IOS crea archivos con formato PEM para almacenar la clave pública RSA, la clave privada RSA cifrada y el certificado del servidor de la CA. El par de claves de sustitución incremental y los certificados se archivan automáticamentePKCS12 - El IOS crea un único archivo PKCS12 que contiene el certificado del servidor de la CA y la clave privada RSA correspondiente cifrada mediante una contraseña.

El archivo de base de datos se puede habilitar mediante este comando en el Servidor PKI:

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

También es posible almacenar los archivos archivados en un servidor independiente, posiblemente mediante un protocolo seguro (SCP) utilizando el siguiente comando en el servidor PKI:

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

De todos los archivos de la base de datos, excepto los archivos archivados y el archivo .Ser, todos los demás archivos están en texto sin formato y no plantean ninguna amenaza real si se pierden, por lo que se pueden almacenar en un servidor independiente sin incurrir en demasiada sobrecarga al escribir los archivos, por ejemplo, un servidor TFTP.

IOS como Sub-CA

El servidor PKI de IOS de forma predeterminada asume la función de una CA raíz. Para configurar un servidor PKI subordinado (Sub-CA), active primero este comando en la sección de configuración del servidor PKI (antes de habilitar el servidor PKI):

```
crypto pki server <Sub-PKI-SERVER-Name>
 mode sub-cs
```

Con esto, configure la URL de Root-CA bajo el punto de confianza del servidor PKI:

```
crypto pki trustpoint <Sub-PKI-SERVER-Name>
 enrollment url <Root-CA URL>
```

La activación de este servidor PKI desencadena ahora estos eventos:

- El punto de confianza del servidor PKI se autentica para instalar el certificado de la CA raíz.
- Después de autenticar la raíz-CA, IOS genera un CSR para la subordinate-CA [restricción básica x509 que contiene el indicador CA:TRUE] y lo envía a la raíz-CA

Independientemente del modo de concesión configurado en la CA raíz, el IOS coloca las solicitudes de certificado de CA (o RA) en cola pendiente. Un administrador debe conceder manualmente los certificados de CA.
Para ver la solicitud de certificado pendiente y la ID de solicitud:

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

Para conceder la solicitud:

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

- Con esto, la operación SCEP POLL (GetCertInicial) subsiguiente descarga el certificado de la sub-CA e lo instala en el router, lo que habilita el servidor PKI subordinado

IOS como RA

El servidor PKI de E/S se puede configurar como autoridad de registro para una CA raíz o subordinada determinada. Para configurar el servidor PKI como autoridad de registro, active primero este comando en la sección de configuración del servidor PKI (antes de habilitar el servidor PKI):

```
crypto pki server <RA-SERVER-Name>
 mode ra
```

A continuación, configure la URL de la CA bajo el punto de confianza del servidor PKI. Esto indica qué CA está protegida por la 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 emite certificados, por lo tanto la configuración **nombre del emisor** en la RA no es necesaria y no es efectiva incluso si está configurada. El subject-name de una RA se configura bajo el trustpoint de RA usando el comando **subject-name**. Es importante configurar **OU = ioscs RA** como parte del nombre del asunto para que la CA del IOS identifique la RA del IOS, es decir, para identificar las solicitudes de certificado autorizadas por la RA del IOS.

IOS puede actuar como autoridad de registro para CA de terceros como Microsoft CA, y para mantener la compatibilidad, IOS RA debe habilitarse usando este comando en la sección de configuración del servidor PKI (antes de habilitar el servidor PKI):

```
mode ra transparent
```

En el modo RA predeterminado, el IOS firma las solicitudes del cliente [PKCS#10] mediante el certificado RA. Esta operación indica al Servidor PKI del IOS que una RA ha autorizado la solicitud de certificado.

Con el modo de RA transparente, IOS reenvía las solicitudes de cliente en su formato original sin introducir el certificado de RA, y esto es compatible con Microsoft CA como un ejemplo bien conocido.

Cliente PKI de IOS

Una de las entidades de configuración más importantes en el cliente PKI de IOS es un punto de confianza. Los parámetros de configuración del punto de confianza se explican en detalle en esta sección.

Fuente de tiempo autorizada

Como se ha señalado anteriormente, la fuente de tiempo autorizada también es un requisito para el cliente de PKI. El cliente PKI de IOS se puede configurar como cliente NTP usando esta configuración:

```
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
```

Nombre de host y nombre de dominio

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

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

Par de Llaves RSA

En el cliente PKI de IOS, el par de llaves RSA para una inscripción de punto de confianza determinada puede generarse automáticamente o generarse manualmente.

El proceso de generación automática de claves RSA implica lo siguiente:

- El IOS de forma predeterminada crea un par de claves RSA de 512 bits
- El nombre del par de claves generado automáticamente es hostname.domain-name, que es el nombre de host del dispositivo combinado con el nombre de dominio del dispositivo
- El par de claves generado automáticamente no se marca como exportable.

El proceso de generación automática de claves RSA implica lo siguiente:

- Opcionalmente, se puede generar manualmente un par de claves RSA de uso general de una fortaleza adecuada mediante:
- `crypto key generate rsa general-keys label <LABEL> modulus < MOD> [exportable]`
  Aquí, ETIQUETA - el nombre del par de llaves RSA
  MOD: modulador o fuerza de clave RSA en bits entre 360 y 4096, que tradicionalmente es 512, 1024, 2048 o 4096.
  La ventaja de generar manualmente el par de llaves RSA es la capacidad de marcar el par de llaves como exportable, lo que a su vez permite que el certificado de identidad se exporte completamente, lo que luego se puede restaurar en otro dispositivo. Sin embargo, hay que comprender las consecuencias de esta medida para la seguridad.
- Un par de llaves RSA se enlaza a un punto de confianza antes de la inscripción usando este comando

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

Aquí, si ya existe un par de claves RSA con el nombre **<LABEL>**, se recuperará durante la inscripción al punto de confianza.

Si no existe un par de claves RSA denominado **<LABEL>**, se ejecuta una de las siguientes acciones durante la inscripción:

- Si no se pasa ningún argumento <MOD>, se genera un par de llaves de 512 bits denominado <LABEL>.

- si se pasa un argumento <MOD>, se genera un par de llaves de uso general de <MOD> bits denominado <LABEL>

- si se pasan dos argumentos <MOD>, se genera un par de claves de firma de bits <MOD> y un par de claves de cifrado de bits <MOD>, ambos denominados <LABEL>

Punto de confianza

Un punto de confianza es un contenedor abstracto para contener un certificado en IOS. Un único punto de confianza puede almacenar dos certificados activos en un momento dado:

- Un certificado de CA - La carga de un certificado de CA en un punto de confianza determinado se conoce como proceso de autenticación de punto de confianza.
- Un certificado de ID emitido por la CA - Cargando o Importando un certificado de ID en un punto de confianza determinado se conoce como proceso de inscripción de punto de confianza.

Una configuración de punto de confianza se conoce como política de confianza y esto define que:

- ¿Qué certificado de CA se carga en el punto de confianza?
- ¿A qué CA se inscribe el punto de confianza?
- ¿Cómo se registra el IOS en el punto de confianza?
- ¿Cómo se valida un certificado emitido por la CA dada [cargado en el punto de confianza]?

Los componentes principales de un punto de confianza se explican aquí.

Modo de inscripción

Un modo de inscripción de punto de confianza, que también define el modo de autenticación de punto de confianza, se puede realizar a través de 3 métodos principales:

1. Inscripción de terminal: método manual para realizar la autenticación de punto de confianza y la inscripción de certificados mediante el comando copy-pasta en el terminal CLI.
2. Inscripción en SCEP: autenticación e inscripción en Trustpoint mediante SCEP sobre HTTP.
3. Perfil de inscripción: Aquí, los métodos de autenticación y de inscripción se definen por separado. Junto con los métodos de inscripción de terminal y SCEP, los perfiles de inscripción proporcionan una opción para especificar comandos HTTP/TFTP para realizar la recuperación de archivos desde el servidor, que se define mediante una url de autenticación o inscripción bajo el perfil.

Interfaz de Origen y VRF

La autenticación de Trustpoint y la inscripción en HTTP (SCEP) o TFTP (perfil de inscripción) utilizan el sistema de archivos IOS para realizar operaciones de E/S de archivos. Estos intercambios de paquetes se pueden originar desde una interfaz de origen específica y un VRF.

En el caso de la configuración clásica de trustpoint, esta funcionalidad se habilita usando los subcomandos **source interface** y **vrf** en el trustpoint.

En el caso de los perfiles de inscripción, la **interfaz de origen** y la **inscripción** Los comandos **| authentication url <http/tftp://Server-location> vrf <vrf-name>** proporcionan la misma funcionalidad.

Ejemplo de configuración:

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

crypto pki trustpoint MGMT
 source interface Ethernet0/0
 vrf MGMT
```

or

```
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
```

Inscripción y renovación automáticas de certificados

El cliente PKI de IOS se puede configurar para realizar la inscripción y renovación automáticas usando este comando en la sección del punto de confianza de PKI:

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

Aquí, el comando **auto-enroll <porcentaje> [regenerate]** establece que el IOS debe realizar la renovación del certificado exactamente al 80% de la vida útil del certificado actual.

La palabra clave **regenerate** establece que el IOS debe regenerar el par de llaves RSA conocido como par de llaves centrales sombra durante cada operación de renovación de certificados.

Este es el comportamiento de inscripción automática:

- El momento en el que se configura **auto-enroll**, si se autentica el punto de confianza, IOS realizará una inscripción automática en el servidor ubicado en la URL mencionada como parte del comando **enrollment url** en la sección del punto de confianza PKI o en el perfil de inscripción.
- En el momento en que se inscribe un punto de confianza con un servidor PKI o una CA, se inicializa un temporizador RENEW o SHADOW en el cliente PKI basándose en el porcentaje **de inscripción automática** del certificado de identidad actual instalado en el punto de confianza. Este temporizador está visible bajo el comando **show crypto pki timer**. Más información sobre las funciones del temporizador *consulte*
- La compatibilidad con la capacidad de renovación proviene del servidor PKI. Más información sobre esto en
  El cliente PKI de IOS realiza dos tipos de renovación:
  Renovación implícita: Si el servidor PKI no envía "Renovación" como una capacidad soportada, IOS realiza una inscripción inicial en el porcentaje de inscripción automática definido. es decir, IOS utiliza un certificado autofirmado para firmar la solicitud de

renovación.Renovación explícita: Cuando el servidor PKI admite la función de renovación de certificados de cliente PKI, anuncia "Renovación" como una capacidad admitida. IOS toma esta capacidad en consideración durante la renovación del certificado, es decir, IOS utiliza el certificado de identidad activo actual para firmar la solicitud de certificado de renovación.

Se debe tener cuidado al configurar el porcentaje de inscripción automática. En cualquier cliente PKI dado en la implementación, si surge una condición en la que el certificado de identidad caduca al mismo tiempo que el certificado CA emisor, el valor auto-enroll siempre debe activar la operación de renovación [Shadow] después de que la CA haya creado el certificado de renovación. *Consulte la* sección **Dependencias del Temporizador PKI** en

Verificación de revocación de certificados

Un punto de confianza PKI autenticado, es decir, un punto de confianza PKI que contenga un certificado CA, puede realizar la validación de certificados durante una negociación IKE o SSL, donde el certificado Peer se somete a una validación de certificado exhaustiva. Uno de los métodos de validación es verificar el estado de revocación del certificado de peer utilizando uno de los dos métodos siguientes:

- Lista de revocación de certificados (CRL): archivo que contiene los números de serie de los certificados revocados por una CA determinada. Este archivo se firma mediante el certificado CA de emisión. El método CRL implica descargar el archivo CRL mediante HTTP o LDAP.

- Protocolo de estado de certificado en línea (OCSP): el IOS establece un canal de comunicación con una entidad denominada Respondedor de OCSP, que es un servidor designado por la CA emisora. Un Cliente como IOS envía una solicitud que contiene el número de serie del certificado que se está validando. El respondedor OCSP responde con el estado de revocación del número de serie dado. El canal de comunicación se puede establecer utilizando cualquier aplicación/protocolo de transporte soportado, que normalmente es HTTP.

La verificación de revocación se puede definir mediante estos comandos en la sección Punto de confianza PKI:

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

De forma predeterminada, se configura un punto de confianza para realizar la verificación de revocación mediante crl.

Los métodos se pueden reordenar y la comprobación del estado de revocación se realiza en el orden definido. El método "none" omite la revocación-comprobación.

caché CRL

Con la comprobación de revocación basada en CRL, cada validación de certificado puede activar una nueva descarga de archivo CRL. A medida que el archivo CRL aumenta o que el punto de distribución de CRL (CDP) se encuentra más lejos, la descarga del archivo durante cada proceso de validación dificulta el rendimiento del protocolo en función de la validación del certificado. Por lo tanto, el almacenamiento en caché de CRL se realiza para mejorar el rendimiento y el almacenamiento en caché de CRL toma en consideración la validez de CRL.

La validez de CRL se define utilizando dos parámetros de tiempo: **LastUpdate**, que es la última vez que la CA emisora publicó la CRL, y **NextUpdate**, que es el momento en que la CA emisora publica una nueva versión del archivo CRL.

IOS almacena en memoria caché cada CRL descargada mientras la CRL sea válida. Sin embargo, en ciertas circunstancias, como que el CDP no se pueda alcanzar temporalmente, puede ser necesario retener el CRL en la memoria caché durante un período de tiempo prolongado. En IOS, una CRL almacenada en caché puede conservarse hasta 24 horas después de que caduque la validez de la CRL, y esto puede configurarse usando este comando en la sección Punto de confianza de PKI:

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

Bajo ciertas circunstancias, como la emisión de certificados de revocación de CA dentro del período de validez de CRL, el IOS puede configurarse para eliminar la memoria caché con más frecuencia. Al eliminar la CRL prematuramente, IOS se ve obligado a descargar la CRL con más frecuencia para mantener la memoria caché de CRL actualizada. Esta opción de configuración está disponible en la sección Punto de confianza PKI:

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

Y, por último, se puede configurar IOS para que no almacene en caché el archivo CRL mediante este comando en la sección Punto de confianza PKI:

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

Configuración recomendada

A continuación se muestra una implementación típica de CA con la configuración de la CA raíz y la subCA. El ejemplo también incluye una configuración Sub-CA protegida por un RA.

Con un par de claves RSA de 2048 bits en toda la placa, este ejemplo recomienda una configuración donde:
Root-CA tiene una vida útil de 8 años
La sub-CA tiene una vida útil de 3 años
Los certificados de cliente se emiten durante un año y se configuran para solicitar una renovación de certificado automáticamente.

CA RAÍZ - Configuración

```
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 sin RA - Configuración

```
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 - Configuración

```
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 para SUBCA - Configuración

```
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
```

Inscripción de certificados

Inscripción manual

La inscripción manual implica la generación de CSR sin conexión en el cliente PKI, que se copia manualmente a la CA. El administrador firma manualmente la solicitud, que luego se importa al cliente.

Cliente PKI

Configuración del cliente 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
```

Paso 1. Primero, autentique el punto de confianza (también se puede realizar después del paso 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
```

Paso 2. Genere la solicitud de firma de certificado y lleve la CSR a la CA y obtenga el certificado concedido:

```
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
```

Paso 3. Ahora importe el certificado concedido a través de la terminal:

```
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
```

Servidor PKI

Paso 1. Primero exporte el certificado de CA de emisión de la CA, que en este caso es el certificado SUBCA. Esto se importa durante el paso 1 anterior en el cliente PKI, es decir, autenticación de punto de confianza.

```
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 del Paso 2 en el Cliente PKI, tome el CSR del cliente y proporciónelo para iniciar sesión en SUBCA usando este comando:

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

Este comando sugiere que la SUBCA acepta una solicitud de firma de certificado del terminal y, una vez concedida, los datos del certificado se imprimen en 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
```

Si la CA está en modo de concesión automática, el certificado concedido se muestra en el formato PEM anterior. Cuando la CA se encuentra en modo de concesión manual, la solicitud de certificado se marca como **pendiente**, se le asigna un valor de ID y se coloca en cola en la cola de solicitud de inscripción.

```
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
```

Paso 3. Permita manualmente esta solicitud mediante este 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: No es posible la inscripción manual de una Sub-CA en una Root-CA.

Nota: Una CA en estado desactivado debido a un servidor HTTP deshabilitado puede conceder manualmente las solicitudes de certificado.

Inscripción mediante 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 del servidor PKI es:

```
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/
```

El modo predeterminado de concesión de solicitud 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
```

Permiso manual

Paso 1. Cliente PKI: Como primer paso, que es obligatorio, autentique el punto de confianza en el cliente 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.
```

Paso 2. Cliente PKI: Después de la autenticación del punto de confianza, el cliente PKI puede estar inscrito para un certificado.

> **Nota:** Si se configura la inscripción automática, el cliente realizará la inscripción automáticamente.

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

Entre bastidores, estos eventos tienen lugar:

- IOS busca un par de llaves RSA llamado PKI-Key. Si existe, se recopila para solicitar un certificado de identidad. Si no es así, IOS crea un par de claves de 2048 bits denominado PKI-Key y luego lo utiliza para solicitar un certificado de identidad.

- IOS crea una solicitud de firma de certificado en formato PKCS10.

- A continuación, IOS cifra esta CSR utilizando una clave simétrica aleatoria. La clave simétrica aleatoria se cifra mediante la clave pública del destinatario, que es la SUBCA (la clave pública de SUBCA está disponible debido a la autenticación de punto de confianza). La CSR cifrada, la clave simétrica aleatoria cifrada y la información del destinatario se agrupan en los datos envueltos PKCS#7.

- Estos datos PKCS#7 se firman mediante un certificado temporal autofirmado durante la inscripción inicial. Los datos encapsulados PKCS#7, el certificado de firma utilizado por el cliente y la firma del cliente se combinan en un paquete de datos firmado PKCS#7. Esto está codificado en base64 y, a continuación, está codificado en URL. El blob de datos resultante se envía como argumento "message" en el URI HTTP enviado a la CA:

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

Paso 3. Servidor PKI:

Cuando el servidor PKI de IOS recibe la solicitud, verifica lo siguiente:

1. Comprueba si la base de datos de solicitudes de inscripción contiene una solicitud de certificado con la misma ID de transacción asociada a la nueva solicitud.

> **Nota:** Un ID de transacción es un hash MD5 de la clave pública, para el cual el cliente solicita un certificado de identidad.

2. Comprueba si la base de datos de solicitudes de inscripción contiene una solicitud de certificado con la misma contraseña de desafío que la enviada por el cliente.

> **Nota:** Si (1) devuelve true o ambos (1) y (2) devuelven true, un servidor de CA puede rechazar la solicitud por motivos de solicitud de identidad duplicada. Sin embargo, en tal caso, el servidor PKI de IOS reemplaza la solicitud anterior por la más reciente.

Paso 4. Servidor PKI:

Permita manualmente las solicitudes en el servidor PKI:

Para ver la solicitud:

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

Para conceder una solicitud específica o todas las solicitudes:

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

Paso 5. Cliente PKI:

Mientras tanto, un cliente PKI inicia un temporizador POLL. Aquí, IOS realiza GetCertInicial a intervalos regulares hasta que SCEP CertRep = GRANTED junto con el certificado concedido sea recibido por el cliente.

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