Configuración de certificados firmados de CA con IOS XE PKI

Contenido

Introducción

Este documento sirve como guía general para configurar certificados IOS XE firmados por una autoridad de certificación (CA) de terceros.

Este documento detallará cómo importar una cadena firmada de CA multinivel como para que el dispositivo sirva como certificado de identidad (ID), así como cómo importar otros certificados de terceros para validar certificados.

Prerequisites

Requirements

El NTP y el tiempo de reloj **DEBEN** configurarse cuando se utilizan las funciones de IOS PKI.

Si un administrador no configura NTP, es posible que tenga problemas con un certificado generado con una fecha/hora futura/pasada. Este sesgo en la fecha o la hora puede causar problemas de importación y otros problemas en el futuro.

Ejemplo de configuración de NTP:

```
ntp server 192.168.1.1
clock timezone EST -5
clock summer-time EDT recurring
```

Componentes Utilizados

- Router de Cisco que ejecuta Cisco IOS® XE17.11.1a

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Antecedentes

Tenga en cuenta que algunas características detalladas en este documento pueden no estar disponibles en versiones anteriores de IOS XE. Siempre que ha sido posible, se ha tenido cuidado de documentar cuándo se ha introducido o modificado un comando o función.

Consulte siempre la documentación oficial de las funciones IOS XE PKI de una versión determinada para comprender cualquier limitación o cambio que pueda ser relevante para su versión específica:

Examples:

- IOS 15 M/T: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_pki/configuration/15-mt/sec-pki-15-mt-book/sec-pki-overview.html
- IOS XE 16.12.x: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_pki/configuration/xe-16-12/sec-pki-xe-16-12-book/sec-est-client-supp-pki.html
- IOS XE 17.x: https://www.cisco.com/c/en/us/td/docs/routers/ios/config/17-x/sec-vpn/b-security-vpn/m_sec-pki-overview-0.html

Configuración IOS XE PKI

En un nivel superior, un administrador debe realizar las siguientes acciones al trabajar con certificados PKI IOS XE:

1. Crear una clave para utilizarla con una función o servicio (**generación de clave criptográfica**)
2. Configure un punto de confianza con varios parámetros y vincule la clave. (**crypto pki trustpoint**)
3. Generar una solicitud de firma de certificado (CSR) (**crypto pki enroll**)
4. Proporcione el CSR a una CA para su firma (*no se trata en este documento*)
5. Autenticar los certificados de CA raíz o intermedia (**crypto pki authenticate**)
6. Importar los certificados de dispositivo (**crypto pki import**)
7. Opcional: autentique certificados de CA de peer (**crypto pki authenticate**)

Estos pasos se detallan en las próximas secciones agrupadas por los comandos necesarios para la acción determinada.

crypto key generate

Muchos administradores han ingresado este comando para habilitar Secure Socket Shell (SSH) en un router o como parte de alguna guía de configuración para una función. Sin embargo, pocos no han diseccionado lo que el comando realmente hace.

Tomemos por ejemplo los siguientes comandos:

```
crypto key generate rsa general-keys modulus 2048 label rsaKey exportable
crypto key generate ec keysize 521 exportable label ecKey
```

Si se analizan estos comandos en las partes específicas, se detallará el uso:

- La primera parte del comando en negro (crypto key generate) le indica al router que crearemos una nueva clave. Existen otras opciones, como la exportación de claves criptográficas, la importación de claves criptográficas o el tamaño cero de claves criptográficas, que se detallarán más adelante.
- La siguiente parte del comando en **verde** (rsa general-keys, ec) le indica al router exactamente qué tipo de clave estamos creando. En la mayoría de los casos, se utilizará un par de claves Rivest-Shamir-Adleman (RSA) que consta de una clave pública/privada, pero un administrador también puede configurar una curva elíptica (EC) para utilizarla con funciones como las que requieren certificados ECDSA o para utilizarla con los saludos ECDHE.
- El comando en **naranja** define el tamaño de nuestra clave.
    - Para RSA, el módulo es la terminología y los valores entre 512-4096 son opciones disponibles. El tamaño predeterminado del módulo varía según la versión, pero se recomienda seguir las prácticas recomendadas de Cisco para la criptografía de última generación y utilizar claves superiores a 2048.
    - Para EC, se requiere el comando key-size para especificar el número de bits en la clave. Las opciones son 256, 384 o 512.
- El comando en **morado** define la etiqueta para esta clave. Esto es importante porque un administrador puede necesitar definir múltiples claves para diversos propósitos en el mismo dispositivo IOS XE. La etiqueta se utiliza para especificar la clave exacta que se utilizará con una función determinada. Siempre que sea posible, utilice una etiqueta para distinguir las claves en uso y facilitar la asignación de claves a las funciones. Por ejemplo: etiqueta SSH, etiqueta CUBE, etiqueta HTTPS creará dos claves para usarlas con diferentes servicios o funciones.
    - La etiqueta predeterminada para una clave es device hostname.domain. Algunos dispositivos pueden generar claves RSA en el primer arranque. Al no introducir un postfijo de etiqueta, un administrador puede correr el riesgo de sobrescribir o regenerar inadvertidamente la clave incorrecta
- El último comando en **azul** es el postfijo exportable. Este comando detalla que la clave se puede utilizar con el comando **crypto pki export** para exportar y utilizar con otros sistemas. Un ejemplo puede ser importar a un dispositivo de alta disponibilidad par para que los miembros de un par HA utilicen una sola clave o para su uso en herramientas de solución de problemas como Wireshark para descifrar sesiones TLS basadas en RSA. Cualquiera que sea la razón por la que se debe declarar que las claves RSA solo se pueden crear como exportables desde el principio. Si un administrador crea una clave RSA no exportable, esta clave no se puede establecer como exportable sin regenerar la clave, lo que puede afectar a otras características, como la invalidación de todos los certificados creados con esa clave. Dicho esto, una clave exportable se puede degradar a no exportable sin regenerar la clave mediante el comando **crypto key move rsaKeyLabel non-exportable**

Ejemplos de Configuración:

<#root>

```
Router(config)#
```

```
crypto key generate rsa general-keys modulus 2048 label rsaKey exportable
```

```
The name for the keys will be: rsaKey

% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be exportable...
[OK] (elapsed time was 1 seconds)
```

```
Router(config)#
```

```
crypto key generate ec keysize 521 exportable label ecKey
```

```
The name for the keys will be: ecKey
```

Ejemplos de verificación:

<#root>

```
Router#
```

```
show crypto key mypubkey rsa rsaKey
```

```
% Key pair was generated at: 10:21:42 EDT Apr 14 2023
Key name: rsaKey
Key type: RSA KEYS      2048 bits
 Storage Device: not specified
 Usage: General Purpose Key
 Key is exportable. Redundancy enabled.
 Key Data:
  30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
  [..truncated..]
  9F020301 0001
```

```
Router#
```

```
show crypto key mypubkey ec ecKey
```

```
% Key pair was generated at: 10:03:05 EDT Apr 14 2023
Key name: ecKey
Key type: EC KEYS      p521 curve
 Storage Device: private-config
 Usage: Signature Key
 Key is exportable. Redundancy enabled.
 Key Data:
  30819B30 1006072A 8648CE3D 02010605 2B810400 23038186 000401A2 A77FCD34
  [..truncated..]
  93FAC967 96ADA79E 4A245881 B2AD2F4A 279A362D F390A20F C06D5845 06DA
```

crypto pki trustpoint

Los Trustpoints son un concepto "tipo carpeta" para almacenar y administrar certificados PKI dentro de IOS XE. (Sintaxis de comandos)

A un nivel superior:

1. Cada punto de confianza IOS XE puede contener un único certificado de CA raíz o intermedio

cargado mediante el comando **crypto pki authenticate**. Considere los puntos de confianza autenticados como la adición de certificados en los que ahora confía el dispositivo.
2. Cada Trustpoint IOS XE también puede importar un único certificado de identidad (ID) cargado mediante el comando **crypto pki import**. El certificado de ID es este certificado de dispositivos que generalmente está vinculado a algún servicio o función.
3. Un administrador puede utilizar el comando **authenticate** e **import** en el mismo punto de confianza (lo cual es necesario para importar un certificado de ID que se discuta más adelante). Al utilizar el flujo de trabajo de autenticación/importación, el punto de confianza contendrá dos certificados (raíz/intermedio + certificado de identidad).
4. Cuando los puntos de confianza se utilizan para almacenar certificados de CA intermedia/raíz de peer de confianza sólo el **crypto pki authenticate** es obligatorio. En esta situación, un punto de confianza sólo contendrá el certificado único autenticado por el administrador.

Nota: Las próximas secciones de **crypto pki authenticate** y **crypto pki import** y las secciones posteriores que detallan ejemplos de autenticación/importación para certificados multinivel proporcionarán contexto adicional a estas cuatro viñetas.

Los Trustpoints pueden tener varios comandos configurados. Estos comandos pueden utilizarse para influir en los valores de una Solicitud de firma de certificado (CSR) creada por el dispositivo mediante el comando **crypto pki enroll** en un punto de confianza.

Hay muchos comandos diferentes disponibles para un punto de confianza (demasiados para detallar en este documento), pero algunos ejemplos más comunes se detallan en el punto de confianza de ejemplo y en la tabla siguiente:

```
crypto pki trustpoint labTrustpoint
 enrollment terminal pem
 serial-number none
 fqdn none
 ip-address none
 subject-name cn=router.example.cisco.com
 subject-alt-name myrouter.example.cisco.com
 revocation-check none
 rsakeypair rsaKey
 hash sha256
```

Comando	**Descripción**
crypto pki trustpoint labTrustpoint	Etiqueta de configuración legible por personas para este punto de confianza. Se utiliza para vincular funciones o servicios en comandos posteriores.
enrollment terminal pem	Determina qué acción llevará a cabo el comando **crypto pki enroll**. En este ejemplo, **enrollment terminal pem** indica que la solicitud de firma de certificado (CSR) se enviará al terminal en un texto con formato PEM Base64.

	Otras opciones como **enrollment selfsigned** se pueden utilizar para crear un certificado autofirmado o **enrollment url** se puede configurar para definir una URL HTTP y aprovechar el protocolo Simple Certificate Enrollment Protocol (SCEP). Ambos métodos están fuera del alcance de este documento.
serial-number none	Determina si los dispositivos IOS XE en serie se agregarán al CSR. Esto también inhabilita la indicación durante el comando crypto pki enroll.
fqdn none	Determina si el nombre de dominio completo (FQDN) se agregará a CSR. Esto también inhabilita la indicación durante el comando crypto pki enroll.
ip-address none	Determina si la dirección IP de los dispositivos IOS XE se agregará al CSR. Esto también inhabilita la indicación durante el comando crypto pki enroll.
subject-name cn=router.example.cisco.com	Indica el X500 formateado que se agregará al CSR.
subject-alt-name myrouter.example.cisco.com	A partir de IOS XE 17.9.1, se puede agregar una lista separada por comas de valores de nombre alternativo de sujeto (SAN) al CSR.
revocation-check none	Indica cómo debe comprobar el dispositivo IOS XE la validez del certificado. Se pueden utilizar opciones como la lista de revocación de certificados (CRL) o el protocolo de estado de certificados en línea (OCSP) si son compatibles con la autoridad de certificación que se elija. Esto se utiliza principalmente cuando el punto de confianza es utilizado por alguna otra función o servicio IOS XE configurado. El estado de revocación también se comprueba cuando un certificado se autentica con un punto de confianza.
rsakeypair rsaKey	Indica al comando que utilice el par de claves RSA con esta etiqueta específica. Para los certificados ECDSA, utilice el comando "eckeypair ecKey" que hace referencia a la etiqueta de la clave EC
hash sha256	Este comando influye en el tipo de algoritmo de hash que se va a utilizar. Las opciones son SHA1, SHA256, SHA384, SHA512

crypto pki enroll

El comando **crypto pki enroll** se utiliza para activar el comando enrollment en un punto de confianza determinado. (Sintaxis del comando)
Para el punto de confianza de ejemplo mostrado anteriormente, el comando **crypto pki enroll labTrustpoint** mostrará la solicitud de firma de certificado (CSR) al terminal en formato de texto PEM Base64, como se muestra en el ejemplo siguiente.

Esta solicitud de firma de certificado ahora se puede guardar en un archivo de texto o copiar y pegar desde la línea de comandos con el fin de proporcionar a cualquier CA de terceros para validar y firmar.

```
<#root>

Router(config)#

crypto pki enroll labTrustpoint

% Start certificate enrollment ..

% The subject name in the certificate will include: cn=router.example.cisco.com
% The fully-qualified domain name will not be included in the certificate
Display Certificate Request to terminal? [yes/no]:

yes

Certificate Request follows:

-----BEGIN CERTIFICATE REQUEST-----
MIICrTCCAZUCAQAwIzEhMB8GA1UEAxMYcm91dGVyLmV4YW1wbGUuY2lzY28uY29t
[..truncated..]
mGvBGUpn+cDIdFcNVzn8LQk=
-----END CERTIFICATE REQUEST-----

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

crypto pki authenticate

El comando **crypto pki authenticate** se utiliza para agregar un certificado de CA confiable a un punto de confianza dado. Cada punto de confianza se puede autenticar una sola vez. Es decir, un punto de confianza sólo puede contener una única raíz de CA o un certificado intermedio. Ejecutar el comando por segunda vez y agregar un nuevo certificado sobrescribirá el primer certificado.

Con el comando **enrollment terminal pem** configurado, el comando **crypto pki authenticate** solicitará al router que se cargue un certificado con formato PEM Base64 a través de la CLI. (Sintaxis del comando)

Un administrador puede autenticar un punto de confianza para agregar los certificados raíz y los certificados intermedios opcionales en una cadena de certificados con el fin de importar el certificado de identificación de un dispositivo más adelante.

Los administradores también pueden autenticar un punto de confianza para agregar otras CA raíz de confianza al dispositivo IOS XE con el fin de habilitar relaciones de confianza con dispositivos de peer durante los protocolos de enlace con ese dispositivo de peer.

Para ilustrarlo con más detalle, un dispositivo par puede incluir una cadena de certificados firmada por "CA raíz 1". Para que la validación de certificados durante el protocolo de enlace entre el dispositivo IOS XE y el dispositivo par sea exitosa; un administrador puede utilizar el comando **crypto pki authenticate** para agregar el certificado de CA a un punto de confianza en el dispositivo IOS XE.

El elemento principal que se debe recordar es que la autenticación de puntos de confianza mediante crypto pki authenticate es siempre para agregar certificados raíz o intermedios de CA a un punto de confianza, no para agregar certificados de identidad. Tenga en cuenta que este concepto también se aplica a la autenticación de certificados autofirmados desde otro dispositivo par.

El siguiente ejemplo muestra cómo autenticar un punto de confianza desde antes usando el comando **crypto pki authenticate**:

<#root>

```
Router(config)#
```

```
crypto pki authenticate labTrustpoint
```

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

-----BEGIN CERTIFICATE-----
[..truncated..]
-----END CERTIFICATE-----

Certificate has the following attributes:
       Fingerprint MD5: C955FC74 7AABC184 D8A75DE7 3C9E7218
      Fingerprint SHA1: 3A99FF61 1E9E6C7B D0E567A9 96D882F5 2279C534

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

```
yes
```

```
Trustpoint CA certificate accepted.
% Certificate successfully imported
```

crypto pki import

Este comando se utiliza para importar el certificado de identidad (ID) en un punto de confianza. Un único punto de confianza sólo puede contener un único certificado de ID y, si se ejecuta el comando por segunda vez, se solicitará que se sobrescriba el certificado importado anteriormente. (Sintaxis del comando)

El siguiente ejemplo muestra cómo importar un certificado de Identidad en el punto de confianza de ejemplo desde antes usando el comando **crypto pki import**.

<#root>

```
Router(config)#
```

```
crypto pki import labTrustpoint certificate
```

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

-----BEGIN CERTIFICATE-----
[..truncated..]
-----END CERTIFICATE-----

% Router Certificate successfully imported
```

Un administrador obtendrá un error si intenta importar un certificado antes de que el punto de confianza haya autenticado el certificado de CA utilizado para firmar directamente este certificado.

```
<#root>

Router(config)#

crypto pki import labTrustpoint certificate

% You must authenticate the Certificate Authority before
 you can import the router's certificate.
```

Autenticación de certificados de CA de peer

Los certificados de CA del mismo nivel se agregan a IOS XE con el mismo método de agregar cualquier certificado de CA. Es decir, se autentican contra un punto de confianza mediante el comando **crypto pki authenticate**.

El siguiente comando muestra cómo crear un punto de confianza y autenticar un certificado de CA de terceros del mismo nivel.

1. Primero, cree un punto de confianza con un nombre descriptivo que contenga el certificado de CA del par
2. configure **enrollment terminal pem** para que el comando crypto pki authenticate solicite el certificado a través de la línea de comandos.
3. Configure **revocation-check none** para omitir la comprobación de CRL/OCSP durante el proceso de importación
4. Autenticar el punto de confianza y proporcionar el certificado
5. Repita los pasos del 1 al 4 para los certificados de CA de peer (recuerde sólo un certificado de CA por punto de confianza).

```
<#root>

Router(config)#

crypto pki trustpoint PEER-ROOT

Router(ca-trustpoint)#

enrollment terminal pem

Router(ca-trustpoint)#

revocation-check none



Router(ca-trustpoint)#

crypto pki authenticate PEER-ROOT


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

-----BEGIN CERTIFICATE-----
[..truncated..]
```

```
-----END CERTIFICATE-----

Certificate has the following attributes:
       Fingerprint MD5: 62D1381E 3E03D06A 912BAC4D 247EEF17
      Fingerprint SHA1: 3C97CBB4 491FC8D6 3D12B489 0C285481 64198EDB

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

yes

```
Trustpoint CA certificate accepted.
% Certificate successfully imported
```

Autenticación de uno o más certificados intermedios

Los ejemplos anteriores detallan cómo generar un CSR usando **crypto pki enroll**, autenticar el certificado de CA raíz usando **crypto pki authenticate** y luego importar el certificado de identidad usando **crypto pki import**.
Sin embargo, al introducir certificados intermedios, el proceso difiere ligeramente. No tema, los mismos conceptos y comandos todavía se aplican! La diferencia radica en la forma en que se establecen los puntos de confianza que contienen los certificados.

Recuerde que cada punto de confianza sólo puede contener un único certificado de CA raíz o intermedio. Por lo tanto, en un ejemplo donde tenemos una cadena de CA como la que se muestra a continuación, es imposible utilizar el comando crypto pki authenticate para agregar más de un certificado de CA:

<#root>

```
- Root CA


 - Intermediate CA 1


   - Identity Certificate
```

Solución:

1. Cree un punto de confianza que contenga la CA raíz autenticada.
2. A continuación, autentique el certificado intermedio con el punto de confianza utilizado para crear la CSR
3. Por último, importe el certificado de identidad en el punto de confianza final.

Usando la tabla a continuación se puede ilustrar el certificado para ordenar la asignación de punto de confianza con colores que corresponden a la cadena anterior para ayudar con la visualización.

Nombre del certificado	Punto de confianza a utilizar	Comando a utilizar
CA raíz	crypto pki trustpoint **ROOT-CA**	crypto pki authenticate **ROOT-CA**
CA 1 intermedia	crypto pki trustpoint **labTrustpoint**	crypto pki authenticate **labTrustpoint**

Certificado de identidad	crypto pki trustpoint **labTrustpoint**	crypto pki import **labTrustpoint** certificate

La misma lógica se puede aplicar a una cadena de certificados con dos certificados de CA intermedios. Nuevamente, se proporcionan colores para ayudar con la visualización de dónde se aplica la nueva CA intermedia a la configuración de IOS XE.

<#root>

- Root CA

- Intermediate CA 1

- Intermediate CA 2

- Identity Certificate

Nombre del certificado	Punto de confianza a utilizar	Comando a utilizar
CA raíz	crypto pki trustpoint **ROOT-CA**	crypto pki authenticate **ROOT-CA**
CA 1 intermedia	crypto pki trustpoint **INTER-CA**	crypto pki authenticate **INTER-CA**
CA 2 intermedia	crypto pki trustpoint **labTrustpoint**	crypto pki authenticate **labTrustpoint**
Certificado de identidad	crypto pki trustpoint **labTrustpoint**	crypto pki import **labTrustpoint** certificate

Mirando de cerca uno puede notar dos patrones:

1. Todos los certificados raíz o intermedios se cargan en los puntos de confianza mediante **crypto pki authenticate** (independientemente de cuántos haya).
2. También se puede notar que el certificado final antes del certificado de identidad del dispositivo (lea el que firmó directamente el certificado de identidad) siempre se autentica en el mismo punto de confianza donde se va a importar el certificado de identidad.
    - De forma similar al error que se muestra anteriormente, IOS XE no permitirá que un administrador importe un certificado sin autenticar primero el certificado de CA utilizado para firmar directamente este certificado.

Estos dos patrones anteriores se pueden utilizar para cualquier número de certificados intermedios más allá de dos, aunque en la mayoría de las implementaciones es probable que un administrador vea más de dos CA intermedias en una cadena de certificados.

También se proporciona la siguiente tabla de certificados de identidad/raíz para completar la información:

<#root>

- Root CA

- Identity Certificate

Nombre del certificado	Punto de confianza a utilizar	Comando a utilizar
CA raíz	crypto pki trustpoint **labTrustpoint**	crypto pki authenticate **labTrustpoint**
Certificado de identidad	crypto pki trustpoint **labTrustpoint**	crypto pki import **labTrustpoint** certificate

Verificación

- Durante el proceso de autenticación o importación, IOS XE realiza varias comprobaciones de integridad para asegurarse de que el certificado es válido y está bien formado. Estos errores se imprimirán en la pantalla o los registros (show logging) buscarán líneas que comiencen con "CRYPTO_PKI"

A continuación se detallan algunos ejemplos comunes:

Las comprobaciones válidas antes y después se realizan en función del tiempo configurado frente al encontrado en el certificado

<#root>

004458:

Aug 9

21:05:34.403: CRYPTO_PKI: trustpoint labTrustpoint authentication status = 0

%CRYPTO_PKI: Cert not yet valid or is expired -

start date: 05:54:04 EDT

Aug 29

2019
end date: 05:54:04 EDT Aug 28 2022

si la comprobación de revocación no está deshabilitada, IOS XE realizará una comprobación de revocación a través del método configurado antes de importar el certificado

<#root>

003375: Aug 9 20:24:14:

%PKI-3-CRL_FETCH_FAIL: CRL fetch for trustpoint ROOT failed

003376: Aug 9 20:24:14.121:

CRYPTO_PKI: enrollment url not configured

Para ver los detalles sobre la configuración, la autenticación o la importación del punto de confianza, utilice los siguientes comandos:

```
show crypto pki trustpoints trustpoint_name
show crypto pki certificates trustpoint_name
show crypto pki certificates verbose trustpoint_name
```

Resolución de problemas

Al depurar problemas de importación u otros problemas de PKI, utilice los siguientes debugs.

```
debug crypto pki messages
debug crypto pki transactions
debug crypto pki validation
debug crypto pki api
debug crypto pki callback
!
debug ssl openssl error
debug ssl openssl msg
debug ssl openssl states
debug ssl openssl ext
```

Conceptos avanzados de IOS PKI

Importar un certificado con formato PKCS12

Algunos proveedores de CA pueden proporcionar archivos de vuelta en formato PKCS#12 (.pfx, .p12).

PKCS#12 es un tipo especial de formato de certificado en el que toda la cadena de certificados, desde el certificado raíz hasta el certificado de identidad, se agrupa junto con el par de claves rsa.

Este formato es muy útil para importar con IOS XE y se puede importar fácilmente usando el siguiente comando:

<#root>

```
Router(config)#
```

crypto pki import PKCS12-TP pkcs12 terminal password Cisco123

or

```
Router(config)#
```

crypto pki import PKCS12-TP pkcs12 ftp://cisco:cisco@192.168.1.1/certificate.pfx password Cisco123

```
% Importing pkcs12...
Address or name of remote host [192.168.1.1]?
Source filename [certificate.pfx]?
Reading file from ftp://cisco@192.168.1.1/certificate.pfx!
```

```
[OK - 2389/4096 bytes]
% You already have RSA keys named PKCS12.
% If you replace them, all router certs issued using these keys
% will be removed.
% Do you really want to replace them? [yes/no]:
```

yes

```
CRYPTO_PKI: Imported PKCS12 file successfully.
```

Exportación de certificados PKCS12 o PEM

Un administrador puede exportar certificados al terminal como PEM de texto sin formato Base64, texto sin formato cifrado Base64 o formato PKCS12 para importarlos a otros dispositivos pares.

Esto es útil cuando se traen nuevos dispositivos de peer y un administrador necesita compartir un certificado de CA raíz que firmó el certificado de identidad de los dispositivos.

A continuación se muestra algún ejemplo de sintaxis:

<#root>

```
Router(config)#
```

crypto pki export labTrustpoint pem terminal

```
Router(config)#
```

crypto pki export labTrustpoint pem terminal 3des password Cisco!123

```
Router(config)#
```

crypto pki export labTrustpoint pkcs12 terminal password cisco!123

Exportar claves RSA

Es posible que sea necesario exportar claves RSA para importarlas a algún otro dispositivo o para utilizarlas en los esfuerzos de resolución de problemas. Suponiendo que el par de claves se creó como exportable, las claves se pueden exportar mediante el comando crypto key export junto con un método de cifrado (DES, 3DES, AES) y una contraseña.

Ejemplo de Uso:

<#root>

```
Router(config)#
```

crypto key export rsa rsaKey pem terminal aes Cisco!123

```
% Key name: IOS-VG
   Usage: General Purpose Key
   Key data:
-----BEGIN PUBLIC KEY-----
[..truncated..]
-----END PUBLIC KEY-----
```

```
base64 len 1664-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-256-CBC,40E087AFF0886DA7C468D2084A0DECFB

[..truncated..]
-----END RSA PRIVATE KEY-----
```

Si la clave no se puede exportar, se mostrará un error.

<#root>

```
Router(config)#
```

crypto key export rsa kydavis.cisco.com pem terminal 3des mySecretPassword

```
% RSA keypair kydavis.cisco.com' is not exportable.
```

Importar llaves RSA generadas fuera de la caja

Algunos administradores pueden realizar la creación de RSA y certificados fuera de la caja, es posible importar las claves RSA usando el comando **crypto key import** como se muestra a continuación usando la contraseña.

<#root>

```
Router(config)#
```

crypto key import rsa rsaKey general-purpose exportable terminal mySecretPassword

```
% Enter PEM-formatted public General Purpose key or certificate.
% End with a blank line or "quit" on a line by itself.
-----BEGIN PUBLIC KEY-----
[..truncated..]
-----END PUBLIC KEY-----

% Enter PEM-formatted encrypted private General Purpose key.
% End with "quit" on a line by itself.
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,9E31AAD9B7463502
[..truncated..]
-----END RSA PRIVATE KEY-----
quit
% Key pair import succeeded.
```

Eliminar claves RSA

Utilice el comando **crypto key zeroize rsa rsaKey** para eliminar un par de llaves RSA denominado rsaKey.

Importar paquete Cisco Trusted CA a través de Trustpool

Los grupos de confianza varían ligeramente desde un punto de confianza, pero el uso principal es el mismo. Donde los puntos de confianza suelen contener un solo certificado de CA, un grupo de confianza contendrá varias CA de confianza.

Cisco publica paquetes de CA en https://www.cisco.com/security/pki/

Un uso común es descargar el archivo ios_core.p7b usando el siguiente comando:

<#root>

```
Router(config)#
```

```
crypto pki trustpool import clean url http://www.cisco.com/security/pki/trs/ios_core.p7b
```

```
Reading file from http://www.cisco.com/security/pki/trs/ios_core.p7b
Loading http://www.cisco.com/security/pki/trs/ios_core.p7b
% PEM files import succeeded.
Router(config)#
```

Preguntas Frecuentes

¿La eliminación de un punto de confianza invalida la CSR o una cadena de certificados otorgada desde una CSR determinada?

No, una vez que se ha generado y guardado la CSR, el punto de confianza se puede eliminar y volver a agregar sin invalidar la CSR.

El soporte técnico de Cisco suele utilizarlo para empezar de cero cuando la autenticación/importación de certificados ha fallado.

Siempre y cuando el administrador o el ingeniero de soporte técnico no vuelvan a generar las claves RSA; la cadena de certificado firmado o CSR se puede importar y se puede autenticar/importar.

¡Importante! La eliminación del punto de confianza **ELIMINARÁ** cualquier certificado autenticado/importado que podría ser más problemático si se asume que esos certificados están siendo utilizados actualmente por algún servicio o función.

¿La generación de una CSR en un punto de confianza invalidará el certificado existente?

No, esto es común cuando los certificados están a punto de caducar. Un administrador puede ejecutar un comando **crypto pki enroll** para crear una nueva CSR e iniciar el proceso de firma de certificados con una CA mientras los certificados existentes que se han autenticado/importado permanecen en uso. El momento en que un administrador reemplaza los certificados por **crypto pki authenticate**/**crypto pki import** es el momento en que se reemplazan los certificados antiguos.

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).