Configuración de CA multinivel en OpenSSL para generar certificados IOS XE

Contenido

Introducción

Este documento describe un método para crear una CA multinivel para crear certificados de propósito general compatibles con los dispositivos Cisco IOS® XE.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Cómo utilizar la aplicación OpenSSL.
- Infraestructura de clave pública (PKI) y certificados digitales.

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y

hardware.

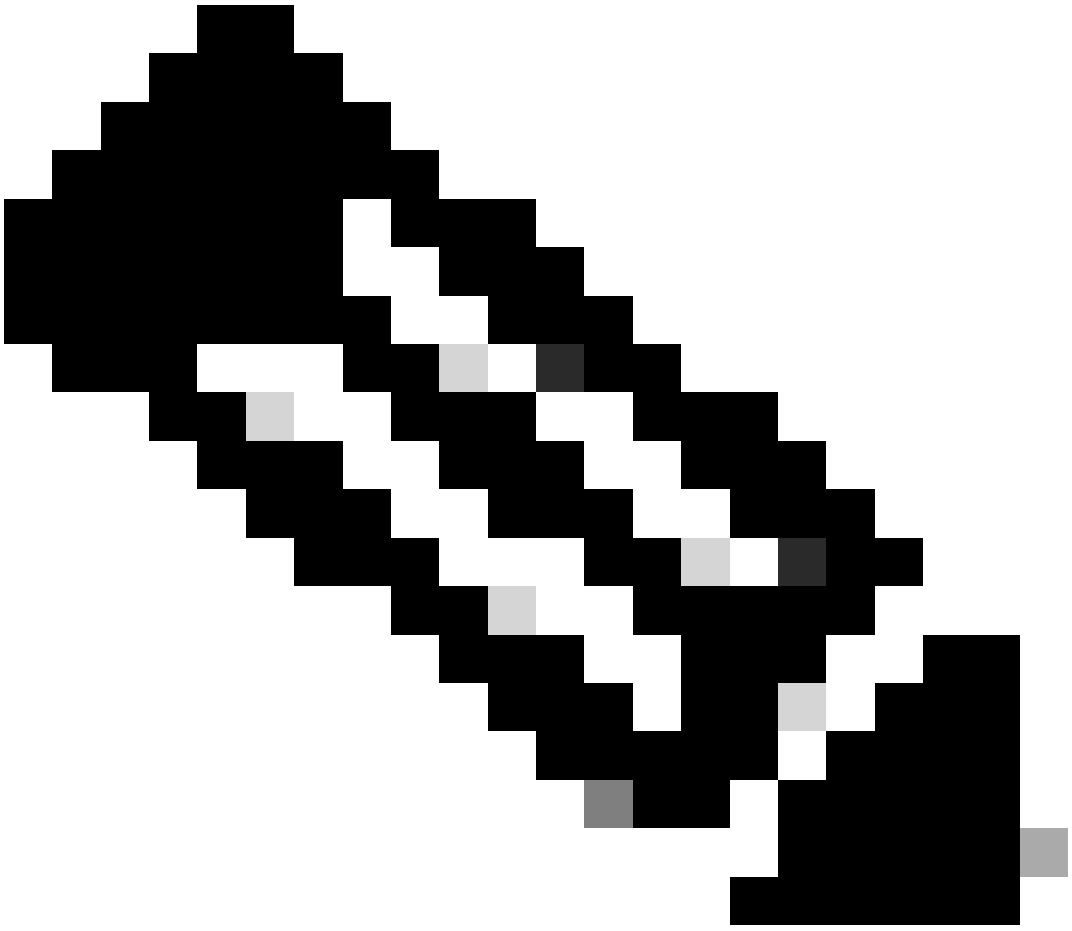
- Aplicación OpenSSL (versión 3.0.2).
- 9800 WLC (Cisco IOS XE versión 17.12.3).

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.

Configurar

Overview

El objetivo es crear una entidad emisora de certificados (CA) local de dos niveles con una CA raíz y una CA intermedia para firmar certificados de dispositivos. Una vez firmados los certificados, se importan al dispositivo Cisco IOS XE.



> Nota: Este documento utiliza comandos específicos de Linux para crear y organizar archivos. Los comandos se explican para que pueda realizar la misma acción en otros sistemas operativos donde OpenSSL esté disponible.

Preparación del archivo de configuración de OpenSSL

Cree un archivo de texto llamado openssl.conf desde el directorio de trabajo actual en el equipo en el que está instalado OpenSSL. Copie y pegue estas líneas para proporcionar a OpenSSL las configuraciones necesarias para la firma de certificados. Puede editar este archivo según sus necesidades.

```
[ ca ]
default_ca  = IntermCA

[ RootCA ]

dir     = ./RootCA
certs       = $dir/RootCA.db.certs
crl_dir = $dir/RootCA.db.crl
database    = $dir/RootCA.db.index
unique_subject = yes
new_certs_dir   = $dir/RootCA.db.certs
certificate = $dir/RootCA.crt
serial      = $dir/RootCA.db.serial
#crlnumber   = $dir/RootCA.db.crlserial
private_key = $dir/RootCA.key
RANDFILE    = $dir/RootCA.db.rand
name_opt    = ca_default
cert_opt    = ca_default
############################## Modify default days for certificates signed by Root CA (Intermediate certs
default_days    = 360
default_md  = sha256
preserve    = no
policy          = optional_policy

[ IntermCA ]

dir     = ./IntermCA
certs       = $dir/IntermCA.db.certs
crl_dir = $dir/IntermCA.db.crl
database    = $dir/IntermCA.db.index
unique_subject = yes
new_certs_dir   = $dir/IntermCA.db.certs
certificate = $dir/IntermCA.crt
serial      = $dir/IntermCA.db.serial
private_key = $dir/IntermCA.key
RANDFILE    = $dir/IntermCA.db.rand
name_opt    = ca_default
cert_opt    = ca_default
# Certificate field options
############################## Modify default days for certificates signed by Intermediate CA cert (devi
default_days    = 1000
#default_crl_days  = 1000
default_md  = sha256
# use public key default MD
```

```
preserve    = no
policy          = optional_policy


[ optional_policy ]
countryName     = optional
stateOrProvinceName = optional
localityName        = optional
organizationName    = optional
organizationalUnitName  = optional
commonName      = supplied

[ req ]
default_bits        = 2048
default_keyfile     = privkey.pem
distinguished_name  = req_distinguished_name
attributes      = req_attributes
x509_extensions = v3_ca # The extentions to add to the signed cert
string_mask = nombstr

[ req_distinguished_name ]
countryName         = Country Name
countryName_default     = MX
countryName_min         = 2
countryName_max         = 2

stateOrProvinceName     = State or province
stateOrProvinceName_default = CDMX

localityName            = Locality
localityName_default        = CDMX

organizationName      = Organization name
organizationName_default  = Cisco lab

organizationalUnitName      = Organizational unit
organizationalUnitName_default  = Cisco Wireless

commonName          = Common name
commonName_max          = 64



[ req_attributes ]
# challengePassword     = A challenge password
# challengePassword_min     = 4
# challengePassword_max     = 20

#This section contains the extensions used for the Intermediate CA certificate

[ v3_ca ]
# Extensions for a typical CA
basicConstraints = CA:true
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid:always,issuer:always
subjectAltName = @Intermediate_alt_names



[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
```

```
extendedKeyUsage = serverAuth, clientAuth

[ crl_ext ]
# CRL extensions.
#authorityKeyIdentifier=keyid:always,issuer:always


#DEFINE HERE SANS/IPs NEEDED for Intermediate CA device certificates
[Intermediate_alt_names]
DNS.1   = Intermediate.example.com
DNS.2   = Intermediate2.example.com


#Section for endpoint certificate CSR generation
[ endpoint_req_ext ]
subjectAltName = _alt_names

#Section for endpoint certificate sign by CA
[ Endpoint ]
basicConstraints=CA:FALSE
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always
#Change the key usage according to the certificate usage needs
extendedKeyUsage = clientAuth
subjectAltName = _alt_names

#Define here SANS/IPs needed for  Endpoint certificates
[endpoint_alt_names]
DNS.1   = Endpoint.example.com
DNS.2   = Endpoint2.example.com

#Section for IOS-XE device certificate CSR generation
[ device_req_ext ]
subjectAltName = @IOS_alt_names

#Section for IOS-XE certificate sign by CA
[ IOS_cert ]
basicConstraints=CA:FALSE
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always
#Change the key usage according to the certificate usage needs
extendedKeyUsage = clientAuth , serverAuth
subjectAltName = @IOS_alt_names

#Define here SANS/IPs needed for IOS-XE certificates
[IOS_alt_names]
DNS.1   = IOSXE.example.com
DNS.2   = IOSXE2.example.com
```

Crear archivos iniciales para las autoridades de certificados

Cree una carpeta en el directorio actual llamada RootCA. Dentro de él, cree 3 carpetas más llamadas RootCA.db.tmp, RootCA.db.certs y RootCA.db.crl.

```
mkdir RootCA
mkdir RootCA/RootCA.db.tmp
mkdir RootCA/RootCA.db.certs
```

```
mkdir RootCA/RootCA.db.crl
```

Cree un archivo llamado RootCA.db.serial dentro de la carpeta RootCA. Este archivo debe contener el valor inicial para el número de serie del certificado, 01 es el valor seleccionado en este caso.

Cree un archivo llamado RootCA.db.crlserial dentro de la carpeta RootCA. Este archivo debe contener el valor inicial para el número de lista de revocación de certificados; 01 es el valor seleccionado en este caso.

```
echo 01 > RootCA/RootCA.db.serial
echo 01 > RootCA/RootCA.db.crlserial
```

Cree un archivo denominado RootCA.db.index dentro de la carpeta RootCA.

```
touch RootCA/RootCA.db.index
```

Cree un archivo denominado RootCA.db.rand dentro de la carpeta RootCA y rellénelo con 8192 bytes aleatorios para que sirva como semilla del generador interno de números aleatorios.

```
openssl rand -out RootCA/RootCA.db.rand 8192
```

Cree una carpeta en el directorio actual denominada IntermCA. Dentro de él, cree 3 carpetas más llamadas IntermCA.db.tmp, IntermCA.db.certs y IntermCA.db.crl.

```
mkdir IntermCA
mkdir IntermCA/IntermCA.db.tmp
mkdir IntermCA/IntermCA.db.certs
mkdir IntermCA/IntermCA.db.crl
```

Cree un archivo denominado IntermCA.db.serial dentro de la carpeta IntermCA. Este archivo debe contener el valor inicial para el número de serie del certificado, 01 es el valor seleccionado en este caso.

Cree un archivo denominado IntermCA.db.crlserial dentro de la carpeta IntermCA. Este archivo debe contener el valor inicial para el número de lista de revocación de certificados; 01 es el valor seleccionado en este caso.

```
echo 01 > IntermCA/IntermCA.db.serial
echo 01 > IntermCA/IntermCA.db.crlserial
```

Cree un archivo denominado IntermCA.db.index dentro de la carpeta IntermCA.

Cree un archivo denominado IntermCA.db.rand dentro de la carpeta IntermCA y rellénelo con 8192 bytes aleatorios para que sirva como semilla del generador interno de números aleatorios.

```
touch IntermCA/IntermCA.db.index
```

Cree un archivo denominado IntermCA.db.rand dentro de la carpeta IntermCA y rellénelo con 8192 bytes aleatorios para que sirva como semilla del generador interno de números aleatorios.

```
openssl rand -out IntermCA/IntermCA.db.rand 8192
```

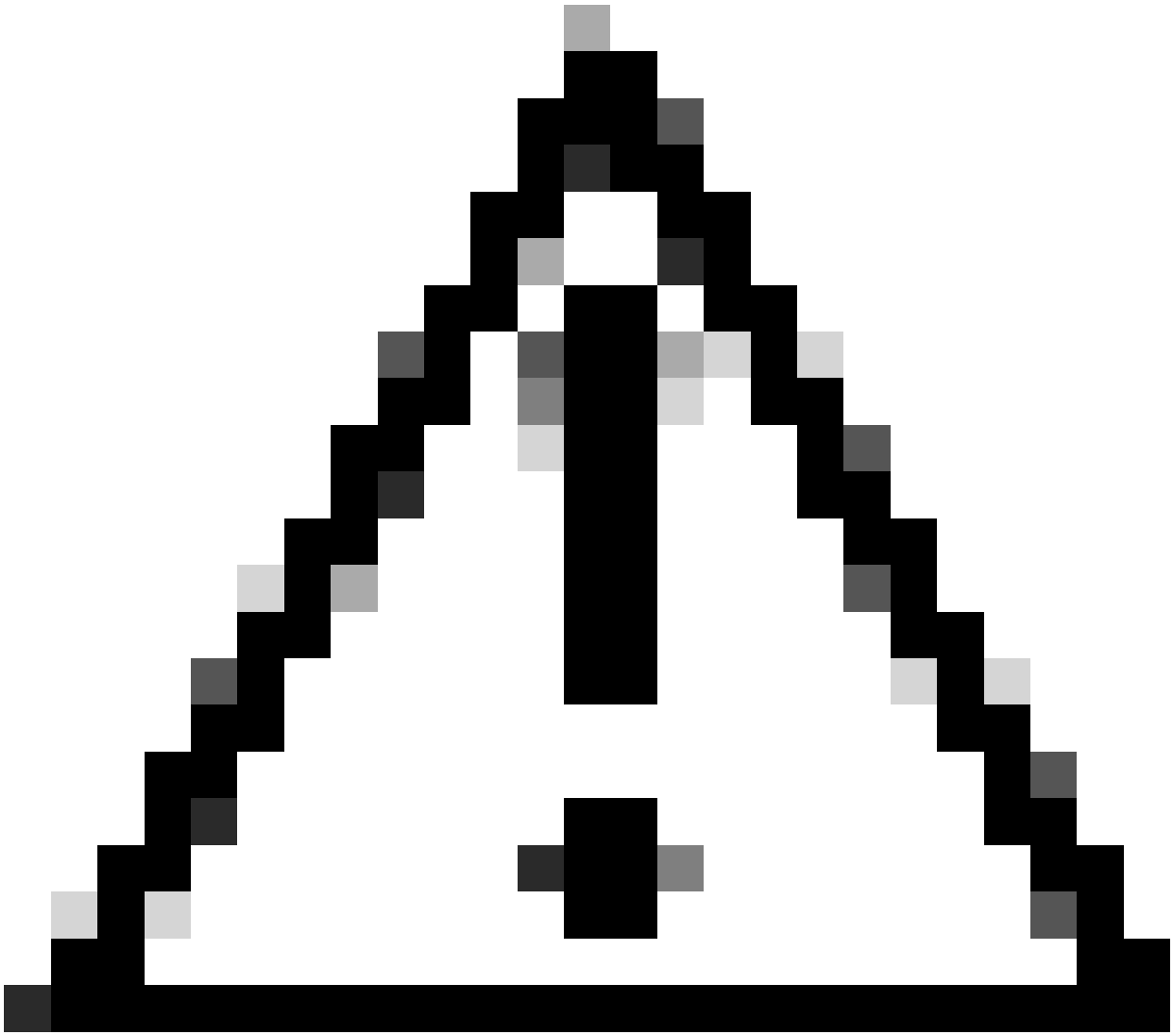
Esta es la estructura de archivos después de la creación de todos los archivos raíz y CA intermedia iniciales.

```
mariomed@CSCO-W-PF320YP6:/mnt/c/Users/mariomed/radsecfiles1$ tree
.
├── IntermCA
│   ├── IntermCA.db.certs
│   ├── IntermCA.db.crl
│   ├── IntermCA.db.crlserial
│   ├── IntermCA.db.index
│   ├── IntermCA.db.rand
│   ├── IntermCA.db.serial
│   └── IntermCA.db.tmp
├── RootCA
│   ├── RootCA.db.certs
│   ├── RootCA.db.crl
│   ├── RootCA.db.crlserial
│   ├── RootCA.db.index
│   ├── RootCA.db.rand
│   ├── RootCA.db.serial
│   └── RootCA.db.tmp
└── openssl.cnf
```

Crear certificado de CA raíz

Ejecute este comando para crear la clave privada para la CA raíz.

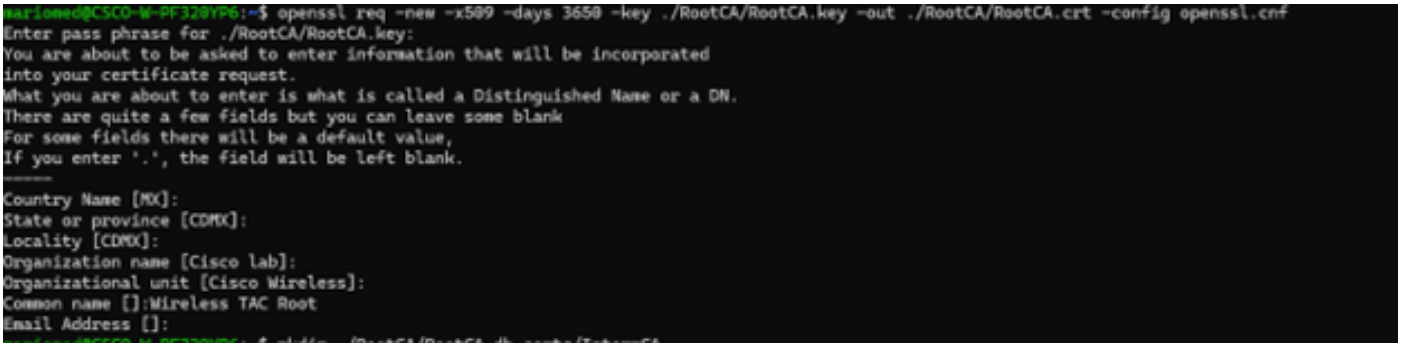
```
openssl genrsa -des3 -out ./RootCA/RootCA.key 4096
```



> Precaución: OpenSSL requiere que se proporcione una frase de contraseña cuando se genera una clave. Mantenga la frase de contraseña secreta y la clave privada generada en una ubicación segura. Cualquier persona con acceso a ella puede emitir certificados como CA raíz.

Cree el certificado autofirmado de la CA raíz mediante el`req`comando en openSSL. El `-x509` indicador crea internamente una solicitud de firma de certificado (CSR) y la firma automática automáticamente. Edite el`-days`parámetro y el nombre alternativo del asunto. El terminal le solicita que proporcione un nombre común. Asegúrese de que el nombre común que introduzca coincide con el nombre alternativo del sujeto (SAN).

```
openssl req -new -key ./RootCA/RootCA.key -out ./RootCA/RootCA.crt -config openssl.cnf -x509 -days 3650
```



Mensaje interactivo de nombre distintivo de OpenSSL

El archivo generado se denomina RootCA.crt y se encuentra dentro de la carpeta RootCA. Este archivo es el certificado de CA raíz.

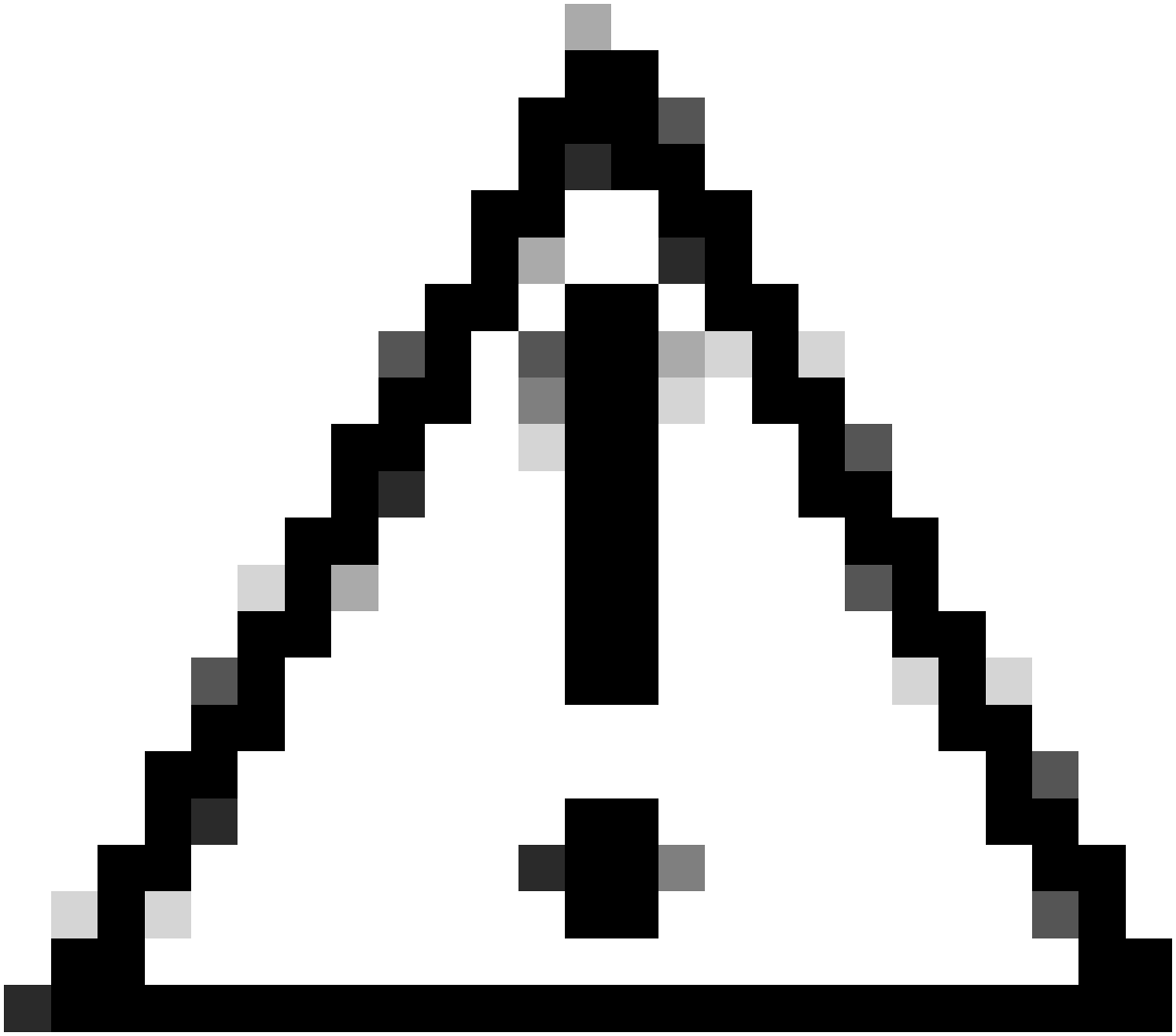
Crear certificado de CA intermedio

Cree una carpeta para almacenar el certificado de CA intermedia firmado dentro de la carpeta raíz.

```
mkdir ./RootCA/RootCA.db.certs/IntermCA
```

Cree una clave privada para el certificado intermedio.

```
openssl genrsa -des3 -out ./RootCA/RootCA.db.certs/IntermCA/IntermCA.key 4096
```



Precaución: OpenSSL requiere que se proporcione una frase de contraseña cuando se genera una clave. Mantenga la frase de contraseña secreta y la clave privada generada en una ubicación segura. Cualquier persona con acceso a ella puede emitir certificados como CA intermedia.

Cree una solicitud de firma de certificado de CA intermedia. El terminal le pide que introduzca la información del certificado.

```
openssl req -new -key ./RootCA/RootCA.db.certs/IntermCA/IntermCA.key -out ./RootCA/RootCA.db.certs/Inte
```

Firme el CSR intermedio con la sección RootCA del archivo openssl.cnf.

```
openssl ca -config openssl.cnf -name RootCA -extensions v3_ca -out ./RootCA/RootCA.db.certs/IntermCA/In
```

El archivo generado se denomina IntermCA.crt y se encuentra dentro de la carpeta RootCA. Este archivo es el certificado de CA raíz.

Mueva el certificado y la clave intermedios a su propia carpeta que creó como parte de los archivos iniciales de la CA intermedia.

```
cp ./RootCA/RootCA.db.certs/IntermCA/IntermCA.crt ./RootCA/RootCA.db.certs/IntermCA/IntermCA.key ./Inte
```

Ésta es la estructura de archivos después de la creación de la clave privada y los certificados para las CA raíz e intermedias iniciales.

```
mariomed@CSCO-W-PF320YP6:/mnt/c/Users/mariomed/radsecfiles$ tree
.
├── IntermCA
│   ├── IntermCA.crt  <------Intermediate CA certficate
│   ├── IntermCA.db.certs
│   ├── IntermCA.db.crl
│   ├── IntermCA.db.crlserial
│   ├── IntermCA.db.index
│   ├── IntermCA.db.rand
│   ├── IntermCA.db.serial
│   ├── IntermCA.db.tmp
│   └── IntermCA.key  <------Intermediate CA private key
├── RootCA
│   ├── RootCA.crt    <------Root CA certficate
│   ├── RootCA.db.certs
│   │   ├── 01.pem
│   │   └── IntermCA
│   │       ├── IntermCA.crt
│   │       ├── IntermCA.csr
│   │       └── IntermCA.key
│   ├── RootCA.db.crl
│   ├── RootCA.db.crlserial
│   ├── RootCA.db.index
│   ├── RootCA.db.index.attr
│   ├── RootCA.db.index.old
│   ├── RootCA.db.rand
│   ├── RootCA.db.serial
│   ├── RootCA.db.serial.old
│   ├── RootCA.db.tmp
│   └── RootCA.key    <------Root CA private key
└── openssl.cnf
```

Crear certificados de dispositivo

Crear certificado de dispositivo Cisco IOS XE

Cree una nueva carpeta para almacenar los certificados de dispositivo de Cisco IOS XE.

```
mkdir ./IntermCA/IntermCA.db.certs/IOSdevice
```

Cree la clave privada del dispositivo IOSdevice.key y la CSR del dispositivo IOSdevice.csr. Utilice la sección device_req_ext para agregar las SAN de dicha sección a la CSR.

```
openssl req -newkey rsa:4096 -sha256 -keyout ./IntermCA/IntermCA.db.certs/IOSdevice/IOSdevice.key -node
```

Modifique la sección del archivo opnessl.cnf [IOS_alt_names] para que el nombre común que proporcione en el CSR coincida con la SAN.

```
#Define here SANS/IPs needed for IOS-XE certificates
[IOS_alt_names]
DNS.1   = IOSXE.example.com
DNS.2   = IOSXE2.example.com
```

Firme la CSR del dispositivo IOS XE con la sección CA IntermCA intermedia. Utilícelo -config para señalar al archivo de configuración openSSL y -extensions para señalar a la sección IOS_cert. Esto mantiene la SAN en el certificado firmado.

```
openssl ca -config openssl.cnf -extensions IOS_cert -name IntermCA -out ./IntermCA/IntermCA.db.certs/IO
```

Después de este paso, ha creado un certificado válido para el dispositivo IOS XE llamado IOSdevice.crt con la clave privada IOSdevice.key coincidente.

Opcional - Crear certificado de terminal

En este momento, ha implementado una CA local y ha emitido un certificado para su dispositivo IOS XE. También puede utilizar esta CA para generar certificados de identidad de extremos. Estos certificados son válidos para, por ejemplo, realizar la autenticación EAP local en controladores LAN inalámbricos 9800 o incluso la autenticación dot1x con servidores RADIUS. Esta sección le ayuda a generar un certificado de terminal.

Cree una carpeta para almacenar los certificados de extremo.

```
mkdir ./IntermCA/IntermCA.db.certs/Endpoint
```

Modifique la sección del archivo openSSL.cnf [endpoint_alt_names] para que el nombre común que proporcione en el CSR coincida con la SAN.

```
#Define here SANS/IPs needed for  Endpoint certificates
[endpoint_alt_names]
DNS.1   = Endpoint.example.com
DNS.2   = Endpoint2.example.com
```

Cree la clave privada del terminal y la CSR del WLC con el uso de la sección endpoint_req_ext para las SAN.

```
openssl req -newkey rsa:2048 -keyout ./IntermCA/IntermCA.db.certs/Endpoint/Endpoint.key -nodes -config
```

Firme el certificado del dispositivo de terminal.

```
openssl ca -config openssl.cnf -extensions Endpoint -name IntermCA -out ./IntermCA/IntermCA.db.certs/En
```

Importar certificado al dispositivo Cisco IOS XE

Cree un archivo que contenga la CA raíz y la CA intermedia en el mismo archivo y guárdelo en la carpeta ./IntermCA/IntermCA.db.certs/WLC/ con el nombre certfile.crt que se requiere para la importación al dispositivo Cisco IOS XE.

```
cat ./RootCA/RootCA.crt ./IntermCA/IntermCA.crt > ./IntermCA/IntermCA.db.certs/IOSdevice/certfile.crt
```

El WLC de la serie 9800 utiliza diversos comandos para crear el archivo pfx para la importación del certificado. Para crear su archivo pfx, ejecute uno de estos comandos según la versión de Cisco IOS XE.

Consulte Generación y Descarga de Certificados CSR en WLC Catalyst 9800 para obtener información detallada sobre el proceso de importación de certificados

Para versiones anteriores a 17.12.1:

```
openssl pkcs12 -export -macalg sha1 -legacy -descert -out ./IntermCA/IntermCA.db.certs/IOSdevice/IOSdev
```

Para la versión 17.12.1 o posterior:

```
openssl pkcs12 -export -out ./IntermCA/IntermCA.db.certs/IOSdevice/IOSdevice.pfx -inkey ./IntermCA/Inte
```

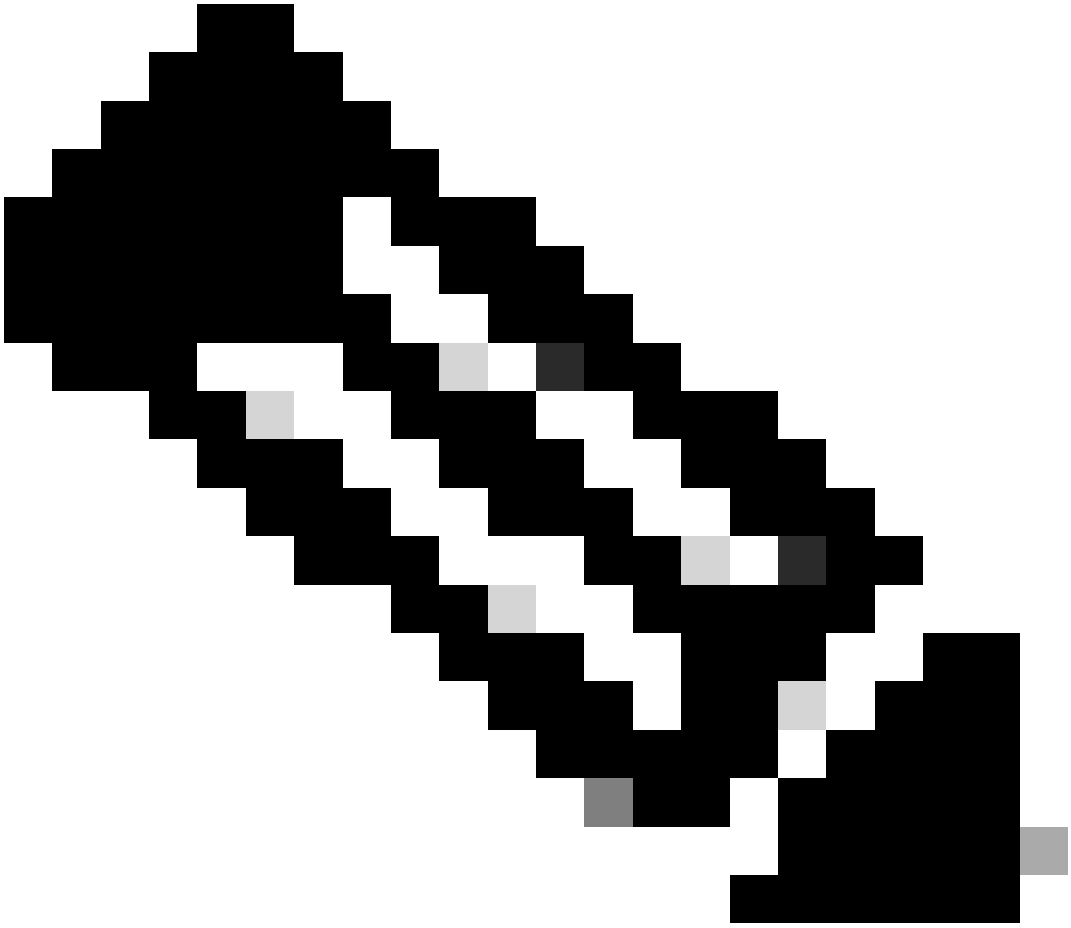
Importe el certificado IOSdevice.pfx al dispositivo Cisco IOS XE:

```
WLC# configure terminal
WLC(config)#crypto pki import


      pkcs12 [tftp://


       /


        | ftp://


         /


          | http://
```

/

| bootflash:

] password



Nota: asegúrese de que los dispositivos que necesitan comprobar el certificado de dispositivo confían en los certificados de CA creados para esta guía. Por ejemplo, si el certificado del dispositivo se utiliza para fines de administración web en el dispositivo Cisco IOS XE, cualquier equipo o explorador que acceda al portal de administración debe tener los certificados de CA en su almacén de confianza.

Deshabilite la comprobación de revocación para los certificados, ya que no hay ninguna lista de revocación de certificados en línea que el dispositivo Cisco IOS XE pueda comprobar desde la CA que ha implementado.
Debe desactivarla en todos los puntos de confianza que forman parte de la ruta de verificación. El punto de confianza de la CA raíz tiene el mismo nombre que el punto de confianza Intermedio/Dispositivo con la cadena -rrr1 anexada al final.

```
9800#configure terminal

9800(config)#crypto pki trustpoint IOSdevice.pfx
9800(config)#revocation-check none
```

```
9800(config)#exit

9800(config)#crypto pki trustpoint IOSdevice.pfx-rrr1
9800(config)#revocation-check none
9800(config)#exit
```

Verificación

Verificar la información del certificado en OpenSSL

Para verificar la información del certificado para los certificados creados, en el terminal Linux ejecute el comando:

```
openssl x509 -in

      -text -noout
```

Muestra la información completa del certificado.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = MX, ST = CDMX, L = CDMX, O = Cisco lab, OU = Cisco Wireless, CN = Intermediate.example.com
        Validity
            Not Before: Jul 18 19:14:57 2024 GMT
            Not After : Apr 14 19:14:57 2027 GMT
        Subject: C = MX, ST = CDMX, L = CDMX, O = Cisco lab, OU = Cisco Wireless, CN = WLC.example.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:b1:10:7d:6c:6c:14:2f:18:a6:0b:69:d9:60:03:
                    56:2d:48:22:f0:42:10:65:44:24:3b:54:e1:4b:87:
                    b8:ab:c5:5f:f6:a1:a3:5e:f6:3c:c5:45:cc:01:6d:
                    df:e8:a7:81:28:50:44:54:4c:af:a0:56:cf:06:be:
                    10:7e:e2:46:42:ea:3c:b9:d4:03:75:08:84:70:36:
                    bb:3d:95:3b:e2:86:e6:f7:d9:4d:00:28:c4:3c:cb:
                    f8:6d:37:5c:89:28:c1:75:b1:7e:fa:bd:91:cf:8e:
                    5c:a2:37:4f:71:da:6a:04:ee:ba:68:bf:4d:f2:d3:
                    ae:aa:13:42:3b:ff:a0:b3:65:c9:ff:f6:9a:06:d7:
                    6c:08:10:e0:b9:d8:ca:93:2d:e5:5d:7b:74:cd:93:
                    68:b1:46:c7:35:d7:6b:0f:a6:ae:34:e6:23:d1:c8:
                    d3:bf:c0:85:ab:2d:02:a8:dd:54:77:e3:32:61:4e:
                    33:58:b0:62:12:82:42:ae:2b:69:f0:5f:0c:90:c7:
                    9c:ef:b9:9c:fc:29:e2:2c:cb:b4:a9:01:fa:5d:3c:
                    97:11:67:cc:25:96:01:3d:26:1a:43:34:bd:43:b0:
                    a0:f1:ec:a0:c7:98:ad:32:32:99:9c:6b:61:af:57:
                    53:ee:20:cc:d5:ed:db:1c:5c:65:51:42:8c:28:bf:
                    62:bf
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                87:89:CA:28:06:95:D5:CE:7C:66:B4:75:81:AA:D4:19:EC:43:01:BB
            X509v3 Authority Key Identifier:
                keyid:2B:08:D8:4C:23:72:5B:62:03:EA:44:F6:9E:D9:F7:75:2E:64:97:DE
                DirName:/C=MX/ST=CDMX/L=CDMX/O=Cisco lab/OU=Cisco Wireless/CN=RootCA
                serial:01
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 Subject Alternative Name:
                DNS:WLC.example.com, DNS:WLC2.example.com
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
```

Información del certificado del dispositivo Cisco IOS XE tal como la muestra OpenSSL

Verifique la información del certificado en el dispositivo Cisco IOS XE.

El comando`show crypto pki certificates verbose`imprime la información de certificado de todos los certificados disponibles en el dispositivo.

```
9800#show crypto pki certificates verbose
CA Certificate <------Type of certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 2A352E27C69021ECE1AA61751CA1F233E0636FB1
  Certificate Usage: General Purpose
  Issuer: <-------------DN for issuer
    cn=RootCA
    ou=Cisco Wireless
    o=Cisco lab
    l=CDMX
    st=CDMX
```

```
    c=MX
  Subject: <--------------DN for subject
    cn=RootCA
    ou=Cisco Wireless
    o=Cisco lab
    l=CDMX
    st=CDMX
    c=MX
  Validity Date:  <--------Validity date
    start date: 14:54:02 Central Jul 22 2024
    end   date: 14:54:02 Central Jul 20 2034
  Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)  <--------------Key size
  Signature Algorithm: SHA256 with RSA Encryption
  Fingerprint MD5: 432021B5 B4BE15F5 A537385C 4FAB9A94
  Fingerprint SHA1: 86D18427 BE619A2A 6C20C314 9EDAAEB2 6B4DFE87
  X509v3 extensions:
    X509v3 Subject Key ID: 57DEEBD8 3214CA05 176F0CD6 6C842EBC 9ABFF7D8
    X509v3 Basic Constraints:
        CA: TRUE
    X509v3 Subject Alternative Name:
        RootCA  <------------SANs
        IP Address :
        OtherNames :
    X509v3 Authority Key ID: 57DEEBD8 3214CA05 176F0CD6 6C842EBC 9ABFF7D8
    Authority Info Access:
  Cert install time: 16:42:09 Central Jul 22 2024
  Associated Trustpoints: WLC.pfx-rrr1  <----------Associated trustpoint
  Storage: nvram:RootCA#6FB1CA.cer
```

Troubleshoot

La comprobación de revocación está activa

Cuando los certificados se importan a Cisco IOS XE, los puntos de confianza recién creados tienen habilitada la comprobación de revocación. Si se presenta un certificado al dispositivo que necesita utilizar los puntos de confianza de certificados importados para la validación, el dispositivo busca una lista de revocación de certificados inexistente y falla. El mensaje se imprime en el terminal.

```
Jul 17 21:50:39.068: %PKI-3-CRL_FETCH_FAIL: CRL fetch for trustpoint WLC1.pfx failed
                      Reason : Enrollment URL not configured.
```

Asegúrese de que cada punto de confianza de la ruta de verificación de los certificados contenga el comando`revocation-check none`.

Información Relacionada

- Generar y descargar certificados CSR en WLC Catalyst 9800
- Configuración de certificados firmados de CA con IOS XE PKI
- Guía de Configuración de Seguridad y VPN, Cisco IOS XE 17.x
- Información sobre certificados para crear una cadena para el WLC 9800

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).