Configurar CA multinível no OpenSSL para gerar certificados IOS XE

Contents

Introdução

Este documento descreve um método para criar uma CA de vários níveis para criar certificados de uso geral compatíveis com dispositivos Cisco IOS® XE.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Como usar a aplicação de OpenSSL.
- Public Key Infrastructure (PKI) e certificados digitais.

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

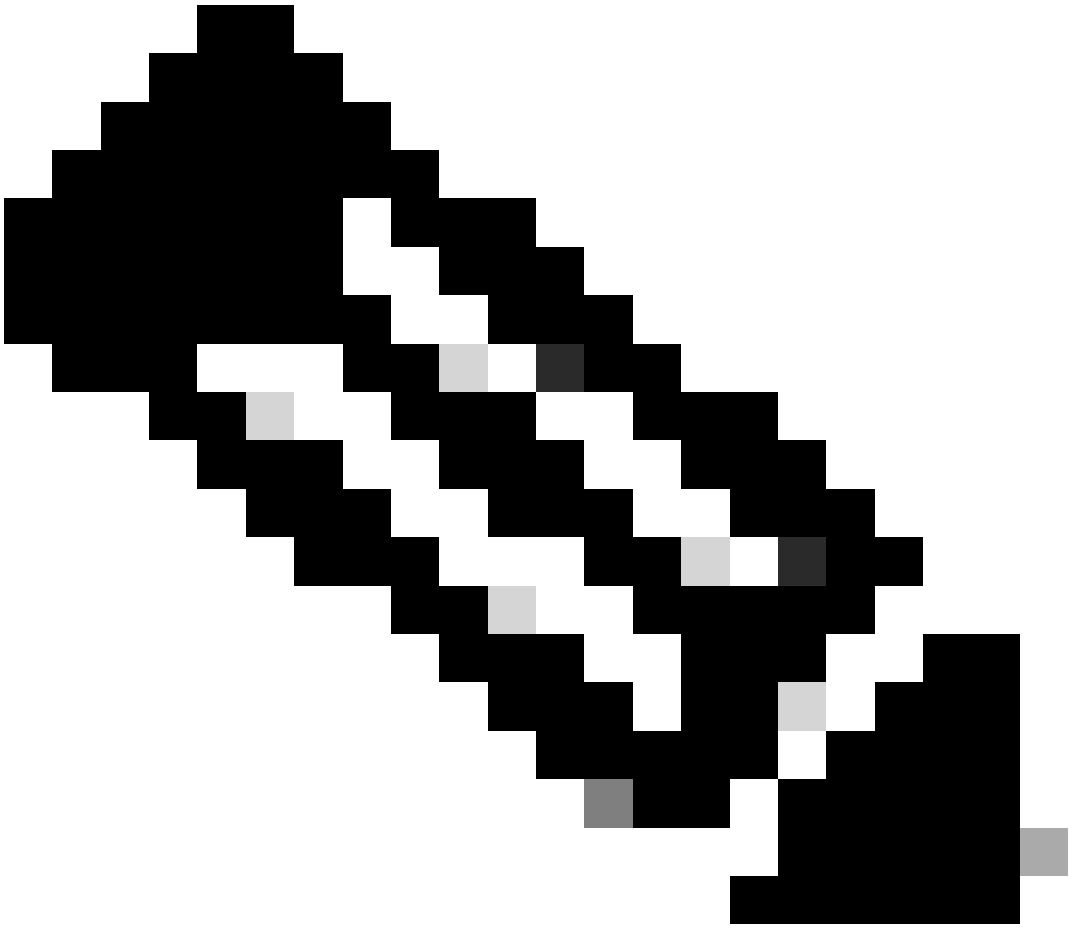
- Aplicativo OpenSSL (versão 3.0.2).
- 9800 WLC (Cisco IOS XE versão 17.12.3).

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Configurar

Overview

A finalidade é criar uma Autoridade de Certificação (CA) local de dois níveis com uma CA raiz e uma CA intermediária para assinar certificados de dispositivos. Uma vez assinados, os certificados são importados para o dispositivo Cisco IOS XE.



Observação: este documento usa comandos específicos do Linux para criar e organizar

---

arquivos. Os comandos são explicados para que você possa executar a mesma ação em outros sistemas operacionais em que o OpenSSL esteja disponível.

Preparar o arquivo de configuração do OpenSSL

Crie um arquivo de texto chamado openssl.conf a partir do diretório de trabalho atual na máquina em que o OpenSSL está instalado. Copie e cole essas linhas para fornecer ao OpenSSL as configurações necessárias para assinatura de certificado. Você pode editar esse arquivo para atender às suas necessidades.

```
[ 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
```

Criar Arquivos Iniciais para as Autoridades de Certificação

Crie uma pasta no diretório atual chamada RootCA. Dentro dele, crie mais 3 pastas chamadas RootCA.db.tmp, RootCA.db.certs e RootCA.db.crl.

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

Crie um arquivo chamado RootCA.db.serial dentro da pasta RootCA. Este arquivo precisa conter o valor inicial para o número de série dos certificados, 01 é o valor selecionado neste caso.

Crie um arquivo chamado RootCA.db.crlserial dentro da pasta RootCA. Este arquivo precisa conter o valor inicial para o número da lista de revogação de certificado, 01 é o valor selecionado neste caso.

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

Crie um arquivo chamado RootCA.db.index dentro da pasta RootCA.

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

Crie um arquivo chamado RootCA.db.rand dentro da pasta RootCA e preencha-o com 8192 bytes aleatórios para servir como semente do gerador de números aleatórios internos.

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

Crie uma pasta no diretório atual chamada IntermCA. Dentro dele, crie mais 3 pastas chamadas IntermCA.db.tmp, IntermCA.db.certs e IntermCA.db.crl.

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

Crie um arquivo chamado IntermCA.db.serial dentro da pasta IntermCA. Este arquivo precisa conter o valor inicial para o número de série dos certificados, 01 é o valor selecionado neste caso.

Crie um arquivo chamado IntermCA.db.crlserial dentro da pasta IntermCA. Este arquivo precisa conter o valor inicial para o número da lista de revogação de certificado, 01 é o valor selecionado neste caso.

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

Crie um arquivo chamado IntermCA.db.index dentro da pasta IntermCA.

Crie um arquivo chamado IntermCA.db.rand dentro da pasta IntermCA e preencha-o com 8192 bytes aleatórios para servir como semente do gerador de números aleatórios internos.

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

Crie um arquivo chamado IntermCA.db.rand dentro da pasta IntermCA e preencha-o com 8192 bytes aleatórios para servir como semente do gerador de números aleatórios internos.

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

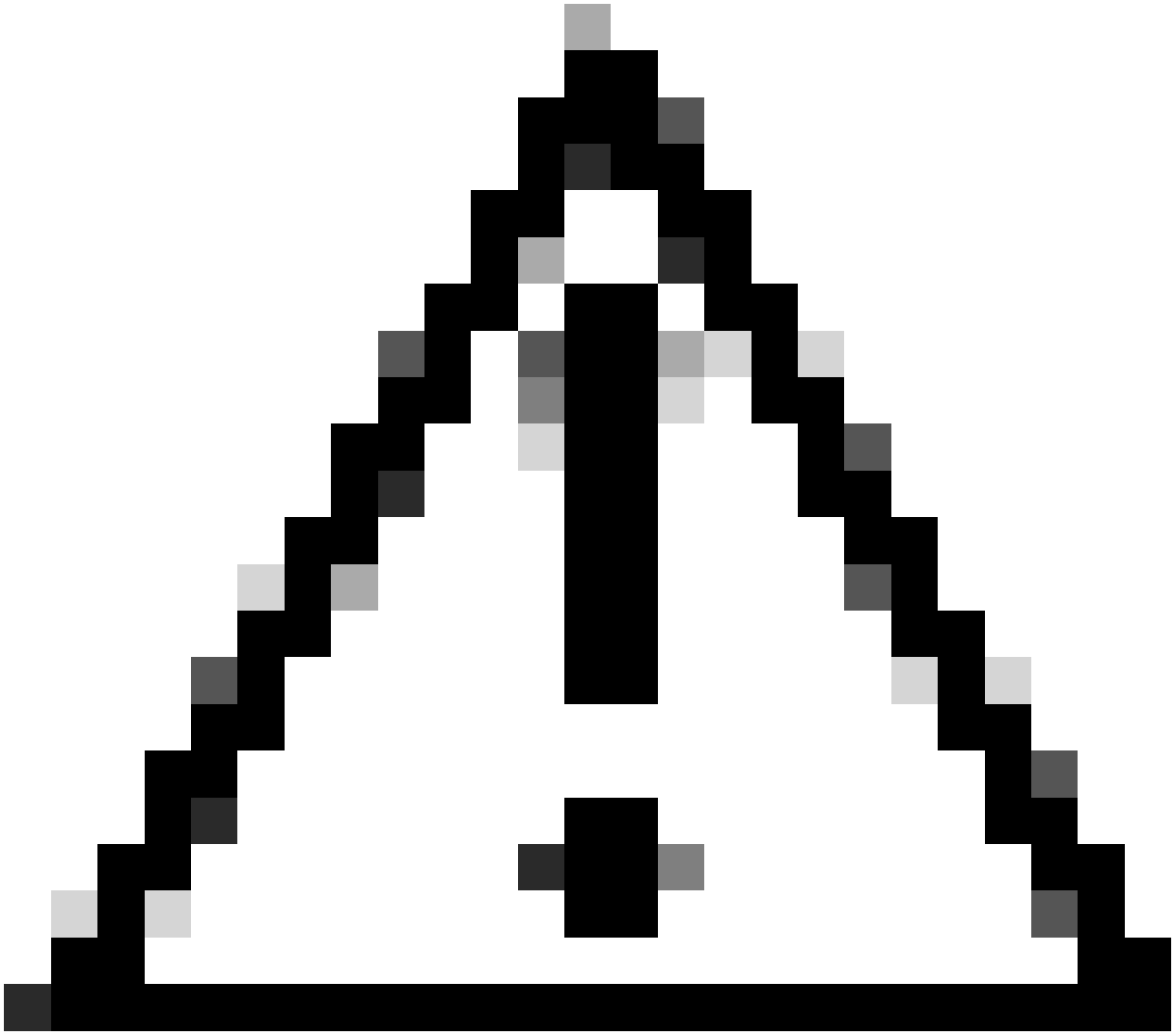
Esta é a estrutura do arquivo após a criação de todos os arquivos iniciais de CA raiz e intermediários.

```
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
```

Criar Certificado CA Raiz

Execute este comando para criar a chave privada para a CA raiz.

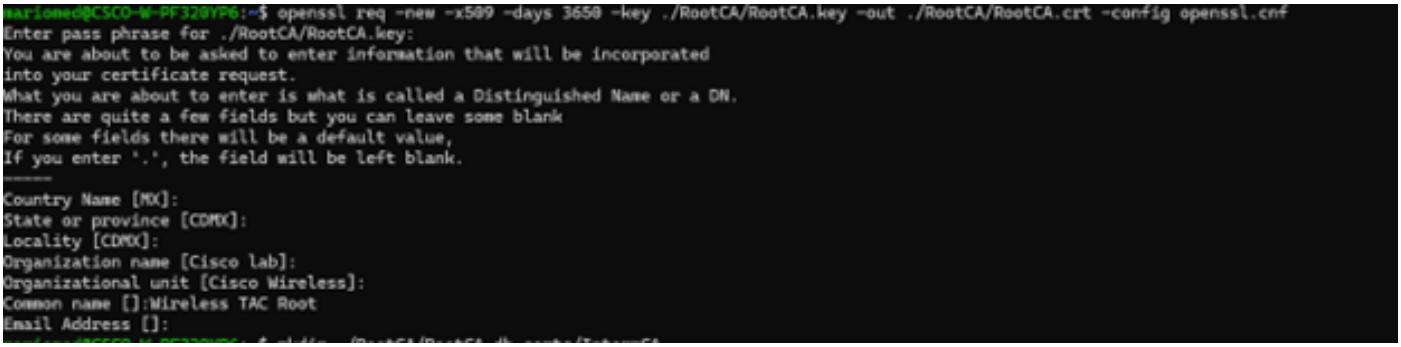
```
openssl genrsa -des3 -out ./RootCA/RootCA.key 4096
```



Cuidado: o OpenSSL requer que você forneça uma senha quando uma chave for gerada. Mantenha a senha secreta e a chave privada gerada em um local seguro. Qualquer pessoa com acesso a ele pode emitir certificados como sua CA raiz.

Crie o certificado autoassinado da autoridade de certificação raiz usando o comando`req`em openSSL. `-x509` O sinalizador cria internamente uma solicitação de assinatura de certificado (CSR) e a autosassina automaticamente. Edite o parâmetro`-days`e o nome alternativo do assunto. O terminal solicita que você forneça um nome comum. Certifique-se de que o nome comum digitado corresponda ao Nome alternativo do assunto (SAN).

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



Prompt Interativo de Nome Distinto OpenSSL

O arquivo gerado é chamado RootCA.crt e está localizado dentro da pasta RootCA. Este arquivo é o certificado CA raiz.

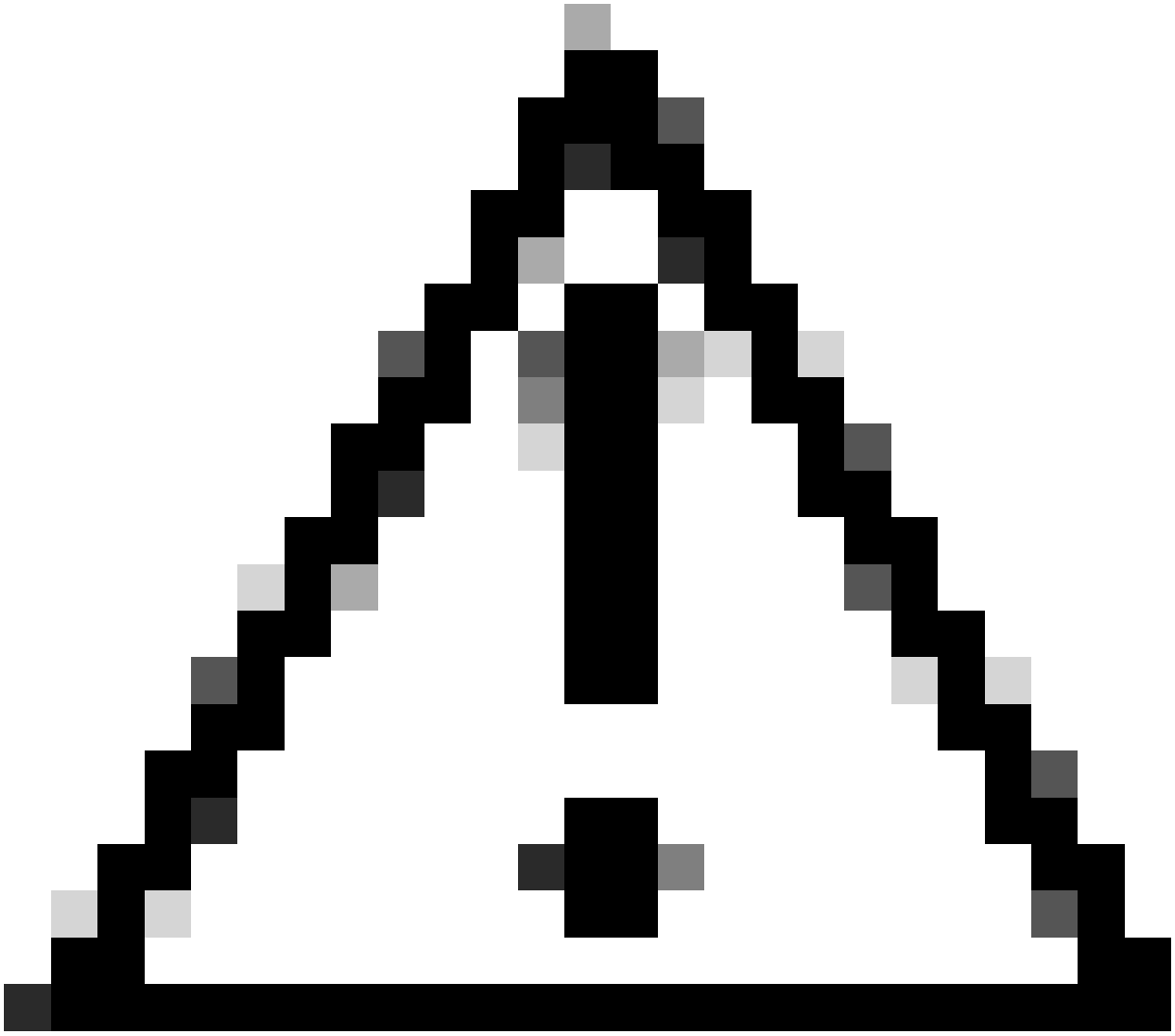
Criar Certificado CA Intermediário

Crie uma pasta para armazenar o certificado CA intermediário assinado dentro da pasta raiz.

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

Criar chave privada para certificado intermediário.

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



Cuidado: o OpenSSL requer que você forneça uma senha quando uma chave for gerada. Mantenha a senha secreta e a chave privada gerada em um local seguro. Qualquer pessoa com acesso a ele pode emitir certificados como sua CA intermediária.

Criar Solicitação de Assinatura de Certificado de Autoridade de Certificação intermediária. O terminal solicita que você insira as informações do certificado.

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

Assine o CSR intermediário com a seção RootCA do arquivo openssl.cnf.

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

O arquivo gerado é chamado de IntermCA.crt e está localizado dentro da pasta RootCA. Este arquivo é o certificado CA raiz.

Mova o certificado intermediário e a chave para sua própria pasta que você criou como parte dos arquivos iniciais da autoridade de certificação intermediária.

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

Esta é a estrutura de arquivos após a criação da chave privada e dos certificados para as CAs raiz e intermediárias iniciais.

```
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
```

Criar Certificados de Dispositivo

Criar certificado do dispositivo Cisco IOS XE

Crie uma nova pasta para armazenar os certificados do dispositivo Cisco IOS XE.

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

Crie a chave privada do dispositivo IOSdevice.key e o dispositivo CSR IOSdevice.csr. Use a seção device_req_ext para adicionar as SANs sob a seção ao CSR.

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

Modifique a seção do arquivo opnessl.cnf [IOS_alt_names] para que o nome comum fornecido no CSR corresponda à SAN.

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

Assine o dispositivo IOS XE CSR com a seção intermediária CA IntermCA. Use -config para apontar para o arquivo de configuração openSSL e -extensions para apontar para a seção IOS_cert. Isso mantém a SAN no certificado assinado.

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

Após esta etapa, você criou um certificado válido para o dispositivo IOS XE chamado IOSdevice.crt com a chave privada correspondente IOSdevice.key.

Opcional - Criar Certificado de Ponto de Extremidade

Neste ponto, você implantou uma CA local e emitiu um certificado para seu dispositivo IOS XE. Você também pode usar esta CA para gerar certificados de identidade de ponto de extremidade. Esses certificados são válidos também, por exemplo, para executar a autenticação EAP Local em controladores LAN Wireless 9800 ou até a autenticação dot1x com servidores RADIUS. Esta seção o ajuda a gerar um certificado de ponto de extremidade.

Crie uma pasta para armazenar os certificados de ponto de extremidade.

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

Modifique o arquivo openSSL.cnf [endpoint_alt_names] para que o nome comum fornecido no CSR corresponda à SAN.

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

Crie a chave privada do endpoint e o CSR da WLC com o uso da seção endpoint_req_ext para SANs.

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

Assinar o certificado do dispositivo de Ponto de Extremidade.

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

Importar certificado para o dispositivo Cisco IOS XE

Crie um arquivo que contenha a CA raiz e a CA intermediária no mesmo arquivo e salve-o na pasta ./IntermCA/IntermCA.db.certs/WLC/ com o nome certfile.crt conforme necessário para importar para o dispositivo Cisco IOS XE.

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

A WLC 9800 Series usa comandos diferentes para criar o arquivo pfx para importação de certificado. Para criar o arquivo pfx, execute um desses comandos de acordo com a versão do Cisco IOS XE.

Consulte Gerar e Fazer Download de Certificados CSR em Catalyst 9800 WLCs para obter informações detalhadas sobre o processo de importação de certificados

Para versões anteriores a 17.12.1:

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

Para a versão 17.12.1 ou posterior:

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

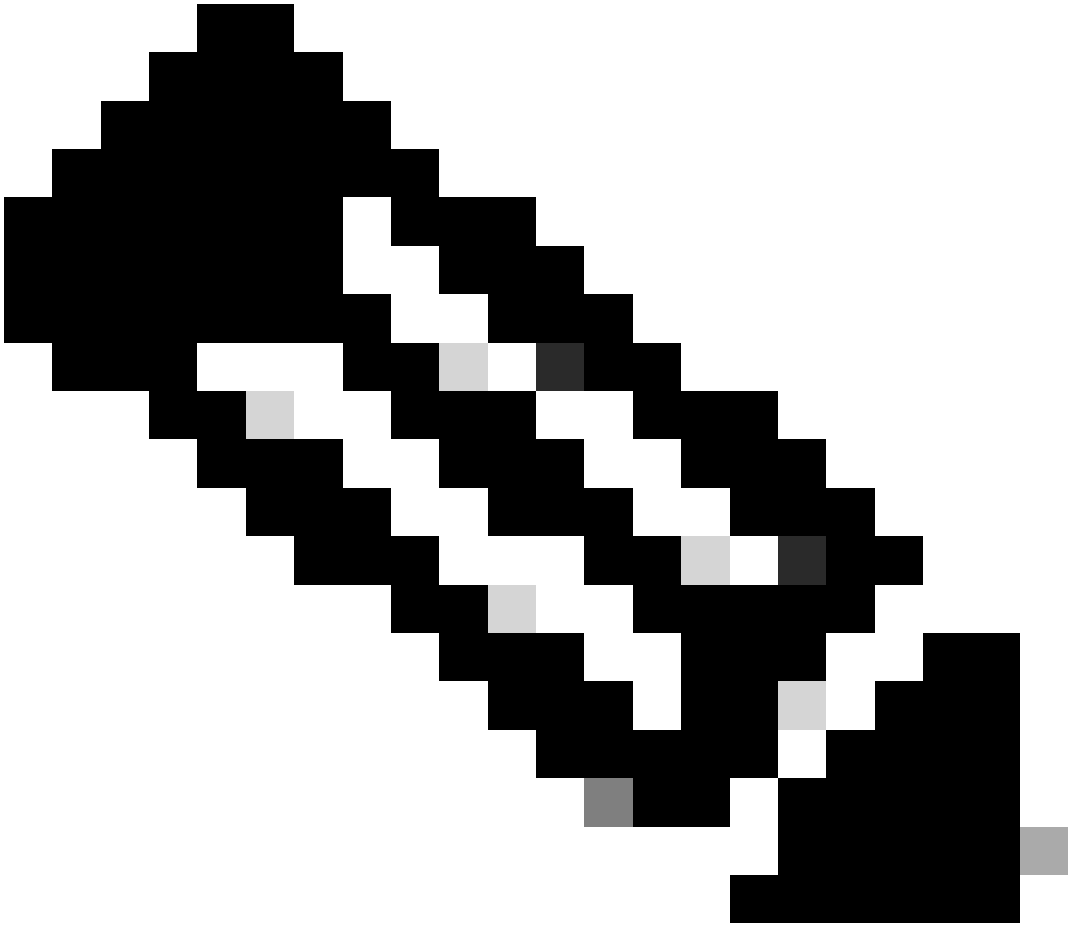
Importe o certificado IOSdevice.pfx para o dispositivo Cisco IOS XE:

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


      pkcs12 [tftp://


       /


        | ftp://


         /


          | http://
```

/

| bootflash:

] password



Observação: certifique-se de que os certificados de CA criados para este guia sejam confiáveis para os dispositivos que precisam verificar o certificado do dispositivo. Por exemplo, se o certificado do dispositivo for usado para fins de administração da Web no dispositivo Cisco IOS XE, qualquer computador ou navegador que acesse o portal do administrador precisará ter os certificados CA em seu armazenamento confiável.

Desabilite a verificação de revogação para os certificados, pois não há lista de revogação de certificados online que o dispositivo Cisco IOS XE possa verificar na CA que você implantou. Você deve desabilitá-lo em todos os pontos confiáveis que fazem parte do caminho de verificação. O ponto de confiança de CA raiz tem o mesmo nome que o ponto de confiança Intermediário/Dispositivo com a cadeia de caracteres -rrr1 anexada ao 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
```

Verificar

Verificar informações de certificado no OpenSSL

Para verificar as informações de certificado para os certificados criados, no terminal Linux, execute o comando:

```
openssl x509 -in

      -text -noout
```

Ela mostra as informações completas do 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:
```

Informações de certificado do dispositivo Cisco IOS XE conforme mostrado pelo OpenSSL

Verifique as informações de certificado no dispositivo Cisco IOS XE.

O comando`show crypto pki certificates verbose`imprime as informações de certificado de todos os certificados disponíveis no 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
```

Troubleshooting

Verificação de Revogação em Vigor

Quando os certificados são importados para o Cisco IOS XE, os pontos confiáveis recém-criados têm a verificação de revogação habilitada. Se um certificado for apresentado ao dispositivo que precisa usar os pontos de confiança do certificado importado para validação, o dispositivo procurará uma Lista de Revogação de Certificado inexistente e falhará. A mensagem é impressa no terminal.

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

Certifique-se de que cada ponto confiável no caminho de verificação para os certificados contenha o comando`revocation-check none`.

Informações Relacionadas

- Gerar e fazer download de certificados CSR em WLCs Catalyst 9800
- Configurar certificados CA assinados com IOS XE PKI
- Guia de configuração de segurança e VPN, Cisco IOS XE 17.x
- Entender as informações do certificado para criar uma cadeia de WLC 9800

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.