Configureer meerlaagse CA op OpenSSL om IOS XE-certificaten te genereren

Inhoud

Inleiding

Dit document beschrijft een methode om een CA op meerdere niveaus te maken om certificaten voor algemene doeleinden te maken die compatibel zijn met Cisco IOS® XE-apparaten.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Hoe de OpenSSL-toepassing te gebruiken.
- Public Key Infrastructure (PKI) en digitale certificaten.

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

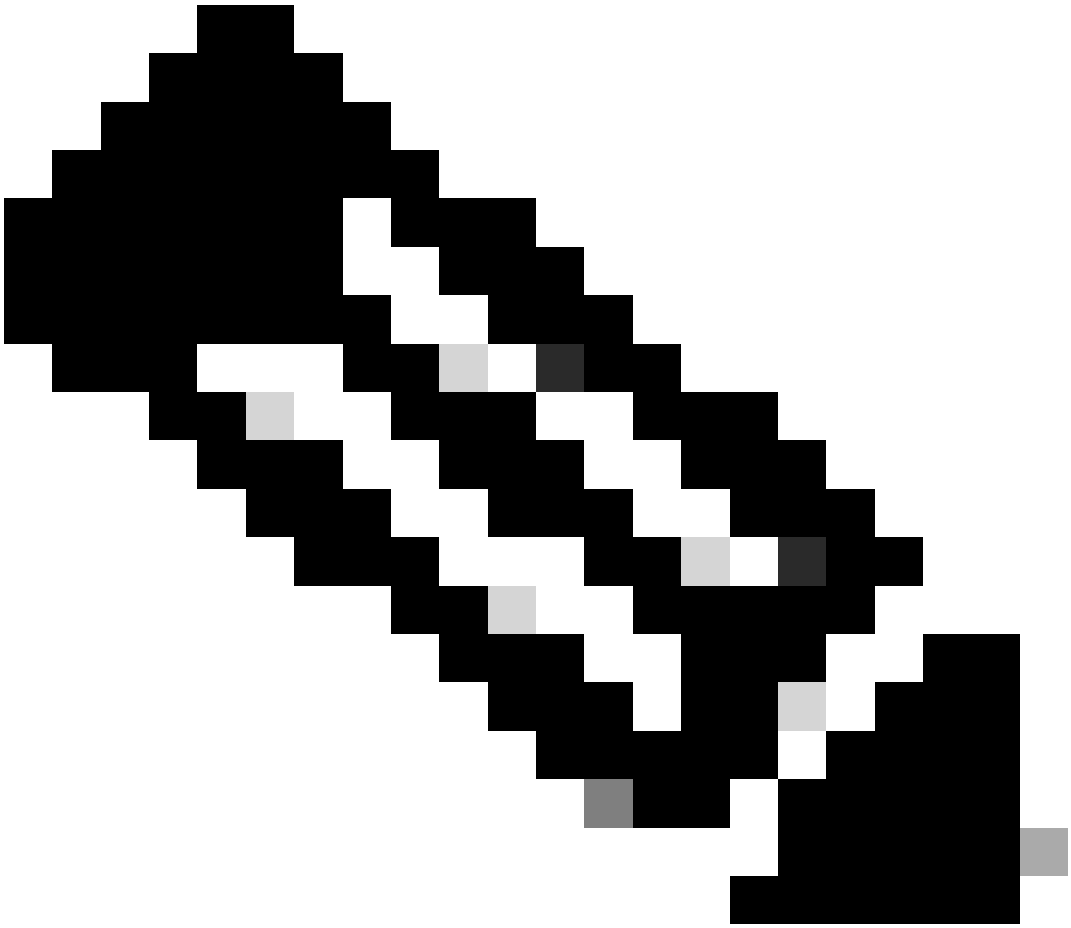
- OpenSSL-toepassing (versie 3.0.2).
- 980 WLC (Cisco IOS XE versie 17.12.3).

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Configureren

Overzicht

Het doel is om een lokale certificeringsinstantie (CA) op twee niveaus te maken met een root-CA en een intermediaire CA om apparaatcertificaten te ondertekenen. Zodra de certificaten zijn ondertekend, worden ze geïmporteerd naar het Cisco IOS XE-apparaat.



Opmerking: dit document gebruikt Linux-specifieke opdrachten om bestanden te maken

---

en te rangschikken. De opdrachten worden uitgelegd zodat u dezelfde actie kunt uitvoeren op andere besturingssystemen waar OpenSSL beschikbaar is.

Het OpenSSL-configuratiebestand voorbereiden

Maak een tekstbestand met de naam openssl.conf vanuit uw huidige werkmap op de machine OpenSSL is geïnstalleerd. Kopieer en plak deze lijnen om OpenSSL te voorzien van de nodige configuraties voor het ondertekenen van certificaten. U kunt dit bestand naar uw wensen bewerken.

```
[ 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
```

Eerste bestanden voor de certificeringsinstanties aanmaken

Maak een map aan in de huidige map RootCA genoemd. Maak er nog 3 mappen in: RootCA.db.tmp, RootCA.db.certs, en RootCA.db.crl.

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

Maak een bestand met de naam RootCA.db.serial binnen de RootCA-map. Dit bestand moet de initiële waarde voor het serienummer van het certificaat bevatten. 01 is de waarde die in deze case is geselecteerd.

Maak een bestand met de naam RootCA.db.crlseriële binnen de RootCA-map. Dit bestand moet de initiële waarde voor het nummer van de certificaatintrekkingslijst bevatten. 01 is de waarde die in deze case is geselecteerd.

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

Maak een bestand met de naam RootCA.db.index binnen de RootCA-map.

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

Maak een bestand met de naam RootCA.db.raen binnen de RootCA-map en vul het in met 8192 willekeurige bytes om te dienen als het zaad van de interne random number generator.

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

Maak een map aan in de huidige map IntermCA. Maak er nog 3 mappen in: IntermCA.db.tmp, IntermCA.db.certs, en IntermCA.db.crl.

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

Maak een bestand met de naam IntermCA.db.serial binnen de IntermCA-map. Dit bestand moet de initiële waarde voor het serienummer van het certificaat bevatten. 01 is de waarde die in deze case is geselecteerd.

Maak een bestand met de naam IntermCA.db.crlseriële in de IntermCA-map. Dit bestand moet de initiële waarde voor het nummer van de certificaatintrekkingslijst bevatten. 01 is de waarde die in deze case is geselecteerd.

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

Maak een bestand met de naam IntermCA.db.index binnen de IntermCA map.

Maak een bestand met de naam IntermCA.db.raen binnen de IntermCA-map en vul het in met 8192 willekeurige bytes om te dienen als het zaad van de interne random number generator.

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

Maak een bestand met de naam IntermCA.db.raen binnen de IntermCA-map en vul het in met 8192 willekeurige bytes om te dienen als het zaad van de interne random number generator.

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

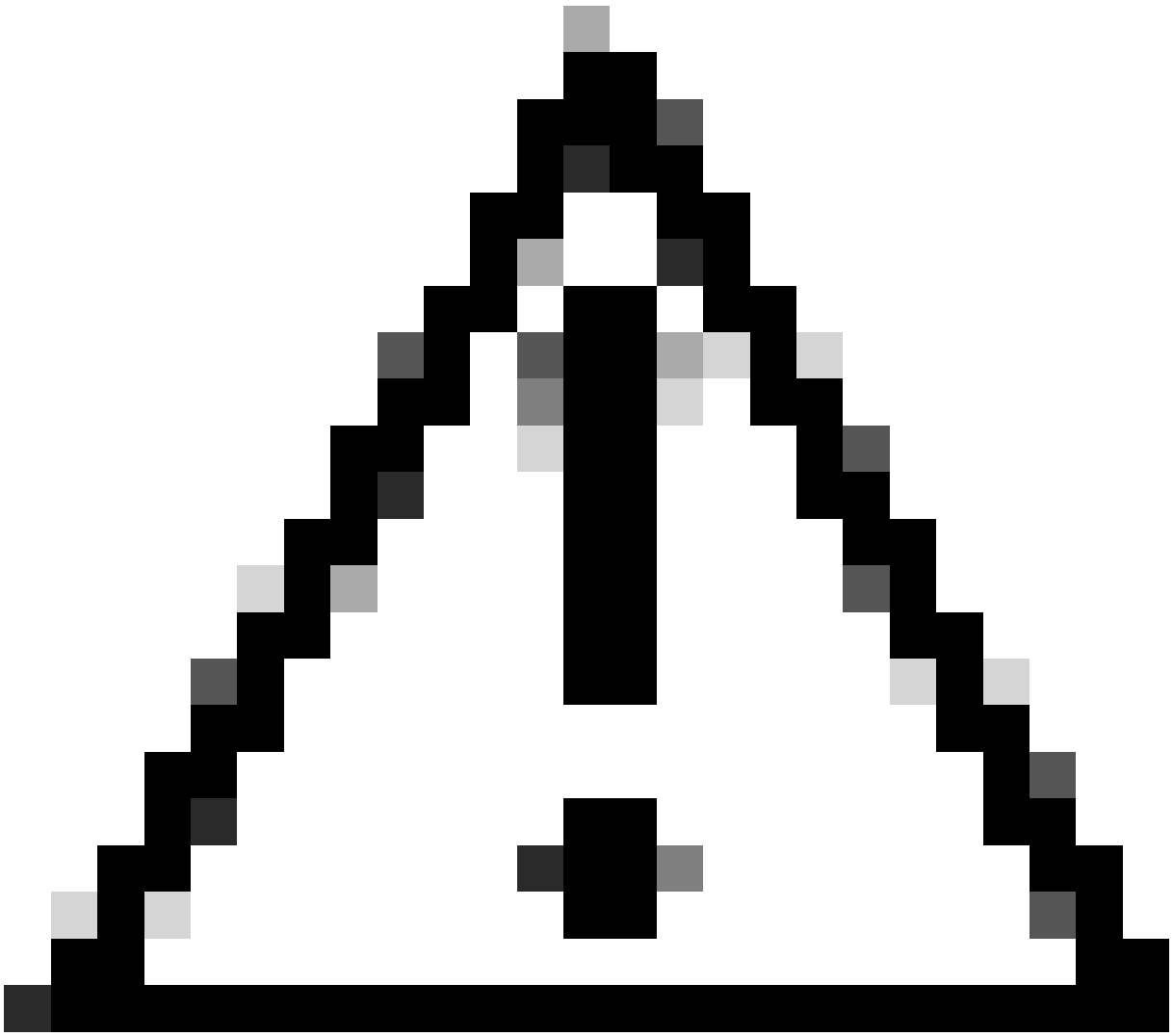
Dit is de bestandsstructuur na de aanmaak van alle eerste Root- en Tussenfase-CA-bestanden.

```
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
```

CA-certificaat hoofdmap maken

Voer deze opdracht uit om de privé-sleutel voor de Root CA te maken.

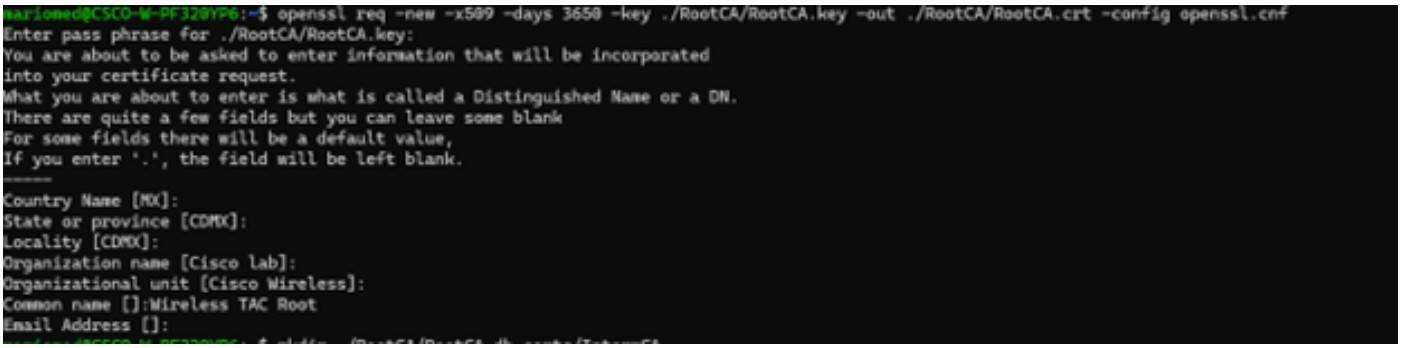
```
openssl genrsa -des3 -out ./RootCA/RootCA.key 4096
```



Waarschuwing: OpenSSL vereist dat u een wachtwoord opgeeft wanneer een sleutel wordt gegenereerd. Houd het wachtwoord geheim en de gegenereerde privé sleutel op een veilige locatie. Iedereen met toegang tot het kan certificaten uitgeven als uw Root CA.

`req`Maak het root CA zelf ondertekende certificaat met behulp van de opdracht op openSSL. De `-x509` vlag maakt intern een certificaat ondertekeningsaanvraag (CSR) en ondertekent deze automatisch zelf. Bewerk de`-days`parameter en de alternatieve onderwerpnaam. Het randnummer vraagt u een algemene naam op te geven. Zorg ervoor dat de veelvoorkomende naam die u invoert, overeenkomt met de alternatieve onderwerpnaam (SAN).

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



Prompt voor OpenSSL Distinguished Name Interactive

Het gegenereerde bestand heet RootCA.crt en bevindt zich in de RootCA-map. Dit bestand is het Root CA-certificaat.

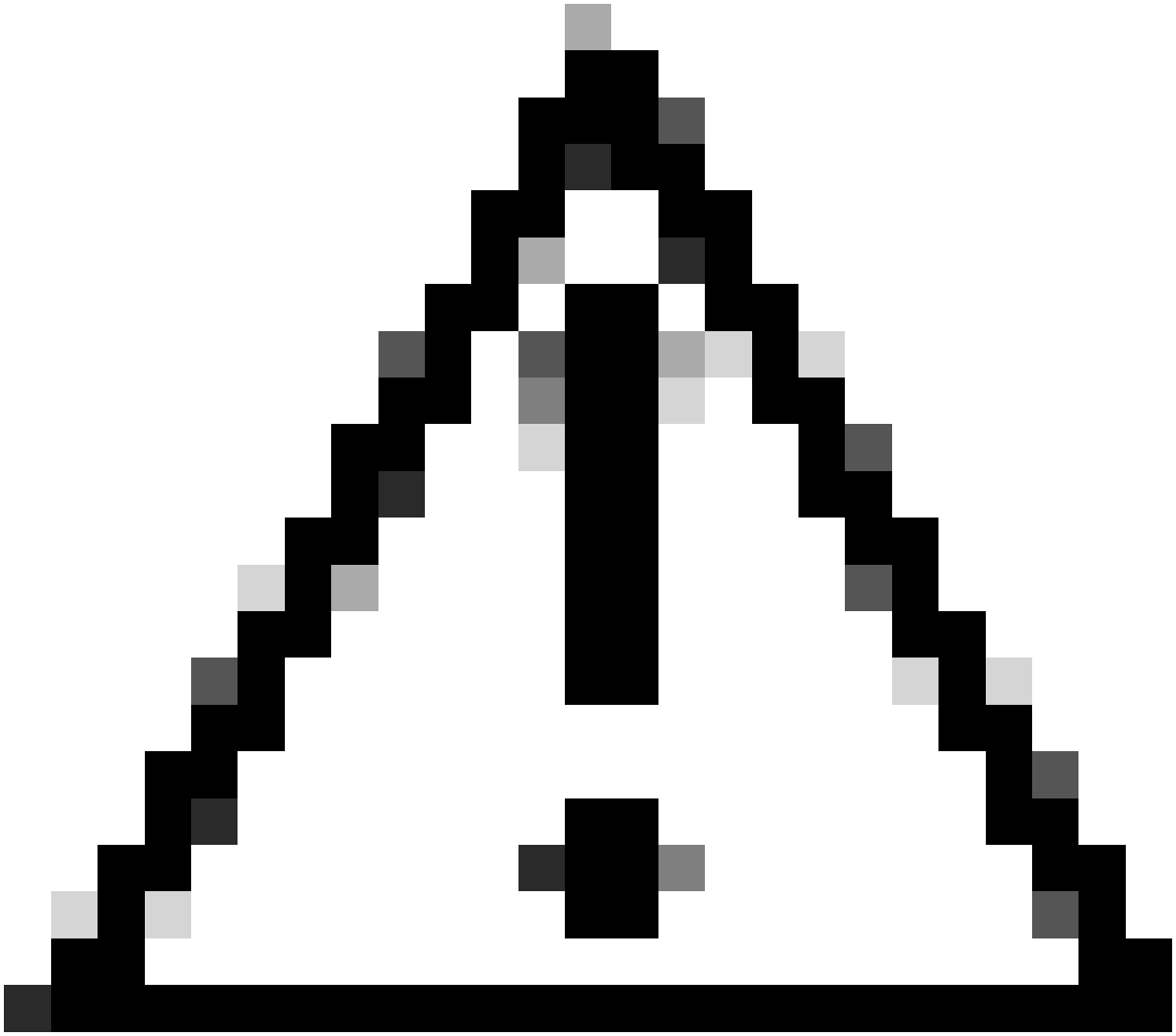
Tussentijds CA-certificaat maken

Map maken om het ondertekende Tussentijdse CA-certificaat op te slaan in de hoofdmap.

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

Maak private sleutel voor tussenliggend certificaat.

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



Waarschuwing: OpenSSL vereist dat u een wachtwoord opgeeft wanneer een sleutel wordt gegenereerd. Houd het wachtwoord geheim en de gegenereerde privé sleutel op een veilige locatie. Iedereen met toegang tot het kan certificaten uitgeven als uw Intermediate CA.

Aanvraag voor tussentijds CA-certificaat aanmaken. De terminal vraagt u om de certificaatinformatie in te voeren.

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

Onderteken Tussenfase CSR met de RootCA sectie van het openssl.cnf bestand.

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

Het gegenereerde bestand heet IntermCA.crt en bevindt zich in de RootCA-map. Dit bestand is het Root CA-certificaat.

Verplaats het tussenliggende certificaat en de sleutel naar de eigen map die u hebt gemaakt als deel van de eerste bestanden voor de tussenliggende CA.

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

Dit is de bestandsstructuur na het aanmaken van de private sleutel en certificaten voor zowel de eerste Root en de tussenliggende CA's.

```
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
```

Apparaatcertificaten maken

Cisco IOS XE-apparaatcertificaat maken

Maak een nieuwe map om de Cisco IOS XE-apparaatcertificaten op te slaan.

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

Maak het apparaat private key IOS device.key en apparaat CSR IOSdevice.csr. Gebruik de sectie device_req_ext om de SAN's onder die sectie toe te voegen aan de CSR.

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

Wijzig het bestand opnessl.cnf [IOS_alt_names] sectie zodat de veelvoorkomende naam die u op de CSR geeft overeenkomt met het SAN.

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

Teken IOS XE-apparaat CSR met tussenliggende CA IntermCA-sectie. `-config` Gebruik dit om naar het openSSL-configuratiebestand te wijzen en naar de IOS_cert-sectie te `-extensions` wijzen. Dit houdt de SAN op het ondertekende certificaat.

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

Na deze stap, hebt u een geldig certificaat voor het IOS XE apparaat genoemd IOSdevice.crt met passende privé sleutel IOSdevice.key gemaakt.

Optioneel - Endpoint certificaat maken

Op dit punt, hebt u een lokale CA opgesteld en één certificaat voor uw IOS XE apparaat verstrekt. U kunt deze CA ook gebruiken om endpointidentiteitscertificaten te genereren. Deze certificaten zijn ook geldig, bijvoorbeeld, voor het uitvoeren van lokale EAP-verificatie op 9800 draadloze LAN-controllers of zelfs dot1x-verificatie met RADIUS-servers. Deze sectie helpt u een endpointcertificaat te genereren.

Maakt een map voor het opslaan van de endpointcertificaten.

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

Wijzig het openSSL.cnf bestand [endpoint_alt_names] sectie zodat de veelvoorkomende naam die u op de CSR geeft overeenkomt met het SAN.

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

Maak de endpoint private key en WLC CSR met het gebruik van sectie endpoint_req_ext voor SAN's.

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

Onderteken het apparaatcertificaat Endpoint.

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

Certificaat importeren naar het Cisco IOS XE-apparaat

Maak een bestand dat de root-CA en tussenliggende CA bevat in hetzelfde bestand en sla dit op naar ./IntermCA/IntermCA.db.certs/WLC/folder met de naam certfile.crt zoals vereist voor het importeren naar het Cisco IOS XE-apparaat.

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

De 9800 Series WLC gebruikt verschillende opdrachten om het pfx-bestand te maken voor het importeren van certificaten. Om uw pfx-bestand te maken voert u een van deze opdrachten uit volgens de Cisco IOS XE-versie.

Raadpleeg CSR-certificaten genereren en downloaden op Catalyst 9800 WLC's voor meer informatie over het proces voor het importeren van certificaten

Voor uitvoeringen ouder dan 17.12.1:

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

Voor versie 17.12.1 of hoger:

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

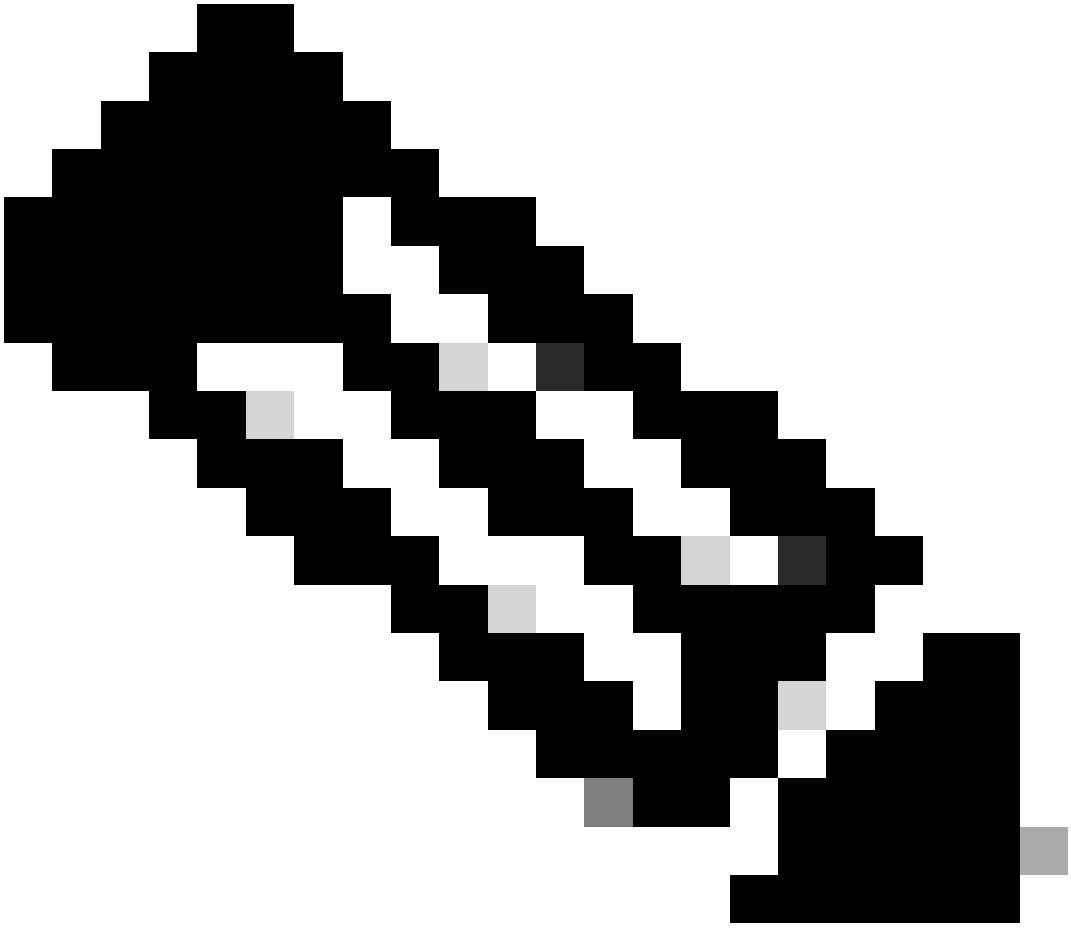
Voer het IOS device.pfx-certificaat in naar het Cisco IOS XE-apparaat:

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


      pkcs12 [tftp://


       /


        | ftp://


         /


          | http://
```

/

| bootflash:

] password



> Opmerking: Zorg ervoor dat de CA-certificaten die voor deze handleiding zijn gemaakt, worden vertrouwd door de apparaten die het apparaatcertificaat moeten verifiëren. Als het apparaatcertificaat bijvoorbeeld wordt gebruikt voor webbeheerdoeleinden op het Cisco IOS XE-apparaat, moeten alle computers of browser die toegang krijgen tot het beheerportal, beschikken over de CA-certificaten in de vertrouwensopslag.

Schakel de herroepingscontrole voor de certificaten uit omdat er geen online lijst is met de herroeping van certificaten die het Cisco IOS XE-apparaat kan controleren vanaf de CA die u hebt geïmplementeerd.
U moet het op alle trustpoints die deel uitmaken van het verificatiepad uitschakelen. De wortel CA trustpoint heeft dezelfde naam als het Tussenpersoon/Apparaat trustpoint met de string -rr1 toegevoegd aan het eind.

```
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
```

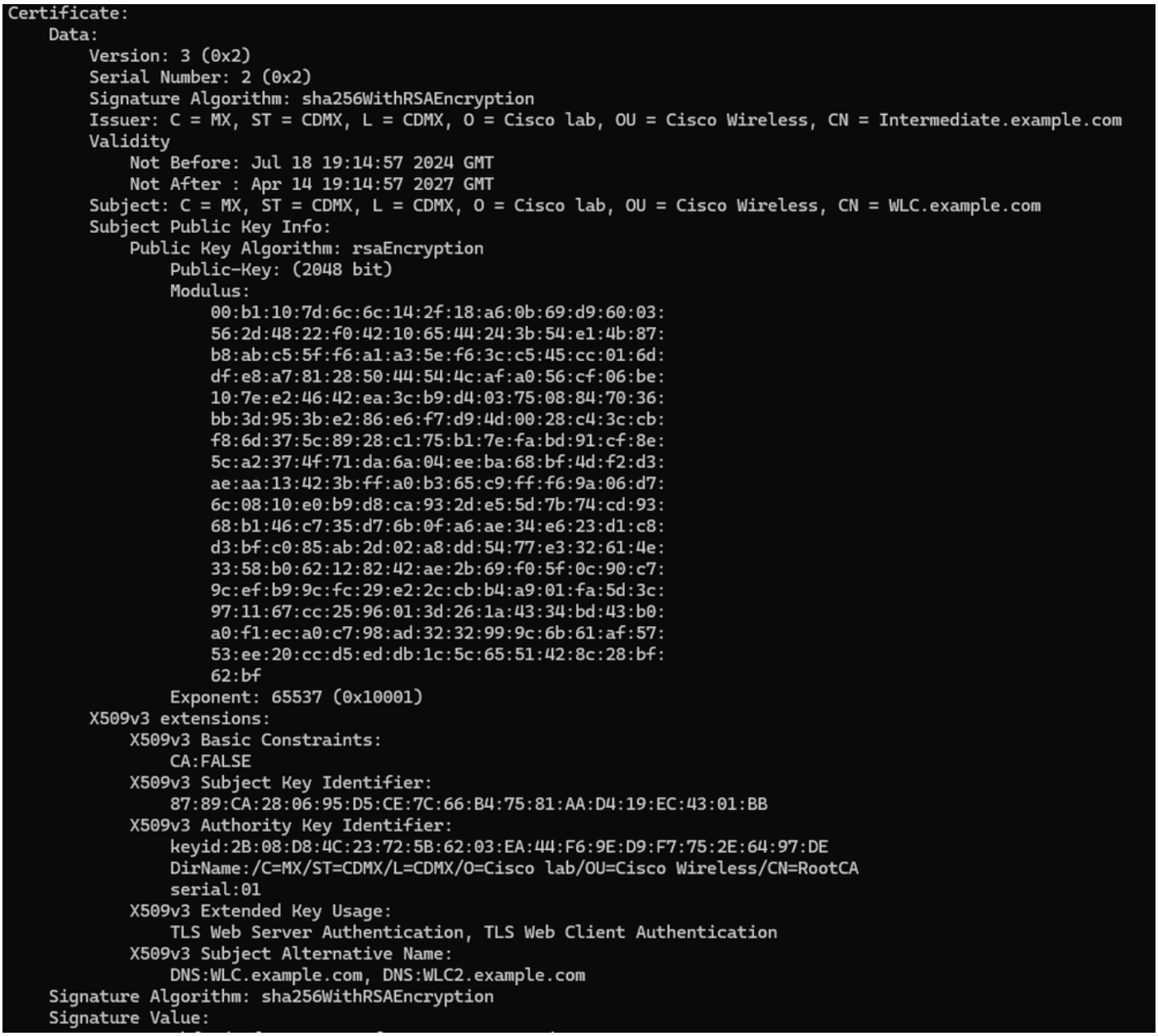
Verifiëren

Controleer de certificaatinformatie op OpenSSL

Om de certificaatinformatie voor de gemaakte certificaten te verifiëren, voert u op de Linux-terminal de opdracht uit:

```
openssl x509 -in

      -text -noout
```

Het toont de volledige certificaatinformatie.



Cisco IOS XE-apparaatcertificaatinformatie zoals weergegeven door OpenSSL

Controleer de certificaatinformatie op het Cisco IOS XE-apparaat.

De opdracht`show crypto pki certificates verbose`drukt de certificaatinformatie van alle beschikbare certificaten op het apparaat af.

```
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
```

Problemen oplossen

Herroepingscontrole is uitgevoerd

Wanneer de certificaten in Cisco IOS XE worden geïmporteerd, is de herroepingscontrole ingeschakeld voor de nieuw gemaakte trustpoints. Als een certificaat wordt voorgelegd aan het apparaat dat de geïmporteerde certificatentrustpoints voor validatie moet gebruiken, zoekt het apparaat naar een niet-bestaande certificaatintrekkingslijst en mislukt het. Het bericht is afgedrukt op de terminal.

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

Zorg ervoor dat elk trustpoint in het verificatiepad voor de certificaten de opdracht bevat`revocation-check none`.

Gerelateerde informatie

- CSR-certificaten genereren en downloaden op Catalyst 9800 WLC's
- CA Signed Certificates configureren met IOS XE PKI
- Beveiligings- en VPN-configuratiehandleiding, Cisco IOS XE 17.x
- Begrijp certificaatinformatie om een ketting voor 9800 WLC te creëren

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.