Konfigurieren einer mehrstufigen Zertifizierungsstelle auf OpenSSL zum Generieren von IOS XE-Zertifikaten

Inhalt

Einleitung

In diesem Dokument wird eine Methode zur Erstellung einer mehrstufigen Zertifizierungsstelle beschrieben, mit der allgemeine Zertifikate erstellt werden können, die mit Cisco IOS® XE-Geräten kompatibel sind.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Verwendung der OpenSSL-Anwendung.
- Public Key Infrastructure (PKI) und digitale Zertifikate.

Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

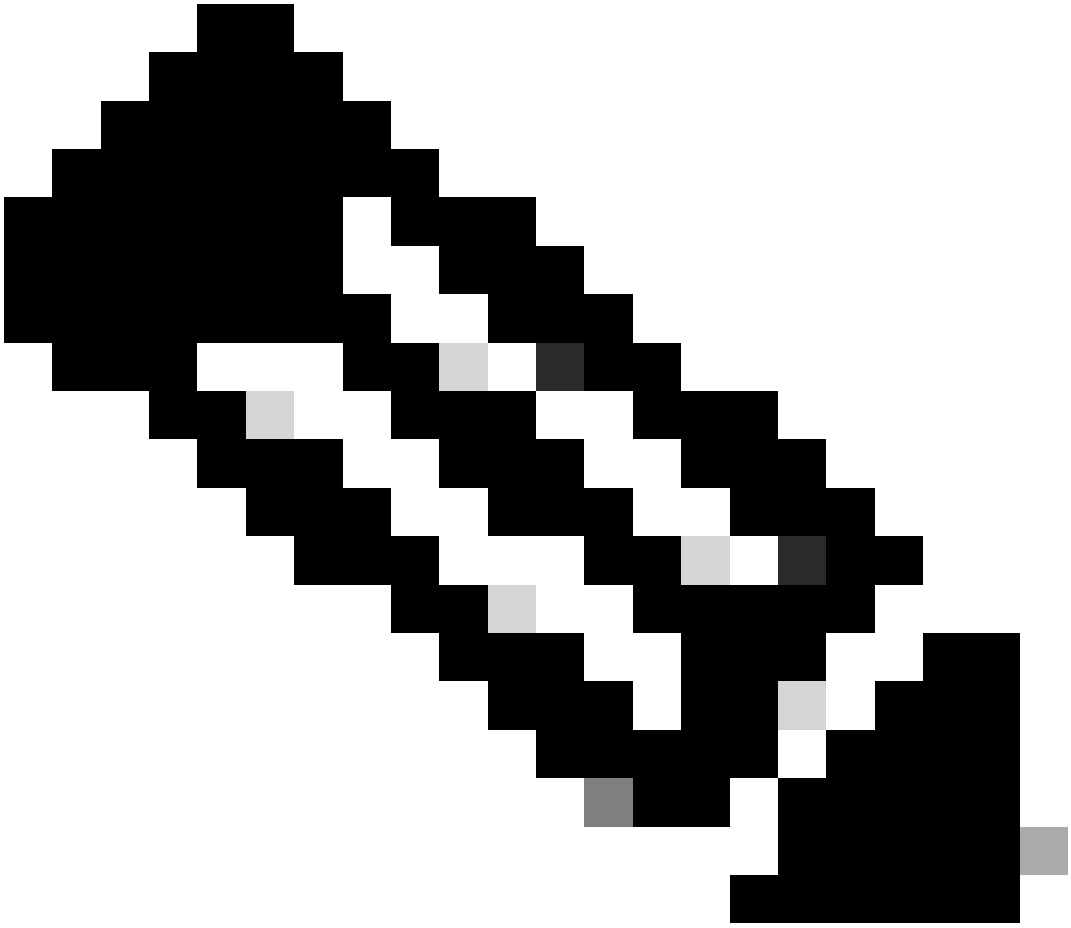
- OpenSSL-Anwendung (Version 3.0.2).
- 9800 WLC (Cisco IOS XE Version 17.12.3).

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Konfigurieren

Überblick

Der Zweck besteht darin, eine lokale Zertifizierungsstelle mit zwei Ebenen zu erstellen, mit einer Stammzertifizierungsstelle und einer Zwischenzertifizierungsstelle, um Gerätezertifikate zu signieren. Nachdem die Zertifikate signiert wurden, werden sie in das Cisco IOS XE-Gerät importiert.



Hinweis: In diesem Dokument werden Linux-spezifische Befehle zum Erstellen und Anordnen von Dateien verwendet. Die Befehle werden erläutert, damit Sie die gleiche Aktion auf anderen Betriebssystemen ausführen können, auf denen OpenSSL verfügbar ist.

OpenSSL-Konfigurationsdatei vorbereiten

Erstellen Sie eine Textdatei namens openssl.conf aus Ihrem aktuellen Arbeitsverzeichnis auf dem Computer, auf dem OpenSSL installiert ist. Kopieren Sie diese Zeilen, und fügen Sie sie ein, um OpenSSL die erforderlichen Konfigurationen für die Zertifikatssignatur bereitzustellen. Sie können diese Datei an Ihre Anforderungen anpassen.

```
[ 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 cert
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
```

Erste Dateien für die Zertifizierungsstellen erstellen

Erstellen Sie im aktuellen Verzeichnis einen Ordner mit dem Namen RootCA. Erstellen Sie in diesem Ordner drei weitere Ordner mit den Namen RootCA.db.tmp, RootCA.db.certs und RootCA.db.crl.

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

Erstellen Sie eine Datei mit dem Namen RootCA.db.serial im Ordner RootCA. Diese Datei muss den Anfangswert für die Seriennummer des Zertifikats enthalten. 01 ist der in diesem Fall ausgewählte Wert.

Erstellen Sie eine Datei mit dem Namen RootCA.db.crlserial im Ordner RootCA. Diese Datei muss den Anfangswert für die Nummer der Zertifikatssperrliste enthalten. 01 ist der in diesem Fall ausgewählte Wert.

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

Erstellen Sie eine Datei mit dem Namen RootCA.db.index im Ordner RootCA.

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

Erstellen Sie eine Datei mit dem Namen RootCA.db.rand im RootCA-Ordner, und füllen Sie sie mit 8192 zufälligen Bytes aus, um als Seed des internen Zufallszahlengenerators zu dienen.

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

Erstellen Sie einen Ordner im aktuellen Verzeichnis mit dem Namen IntermCA. Erstellen Sie in diesem Ordner drei weitere Ordner mit den Namen IntermCA.db.tmp, IntermCA.db.certs und IntermCA.db.crl.

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

Erstellen Sie im IntermCA-Ordner eine Datei mit dem Namen IntermCA.db.serial. Diese Datei muss den Anfangswert für die Seriennummer des Zertifikats enthalten. 01 ist der in diesem Fall ausgewählte Wert.

Erstellen Sie eine Datei mit der Bezeichnung IntermCA.db.crlserial im Ordner IntermCA. Diese Datei muss den Anfangswert für die Nummer der Zertifikatssperrliste enthalten. 01 ist der in diesem Fall ausgewählte Wert.

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

Erstellen Sie eine Datei mit dem Namen IntermCA.db.index im Ordner IntermCA.

Erstellen Sie eine Datei mit dem Namen IntermCA.db.rand im Ordner IntermCA, und füllen Sie sie mit 8192 zufälligen Bytes aus, um als Seed des internen Zufallszahlengenerators zu dienen.

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

Erstellen Sie eine Datei mit dem Namen IntermCA.db.rand im Ordner IntermCA, und füllen Sie sie mit 8192 zufälligen Bytes aus, um als Seed des internen Zufallszahlengenerators zu dienen.

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

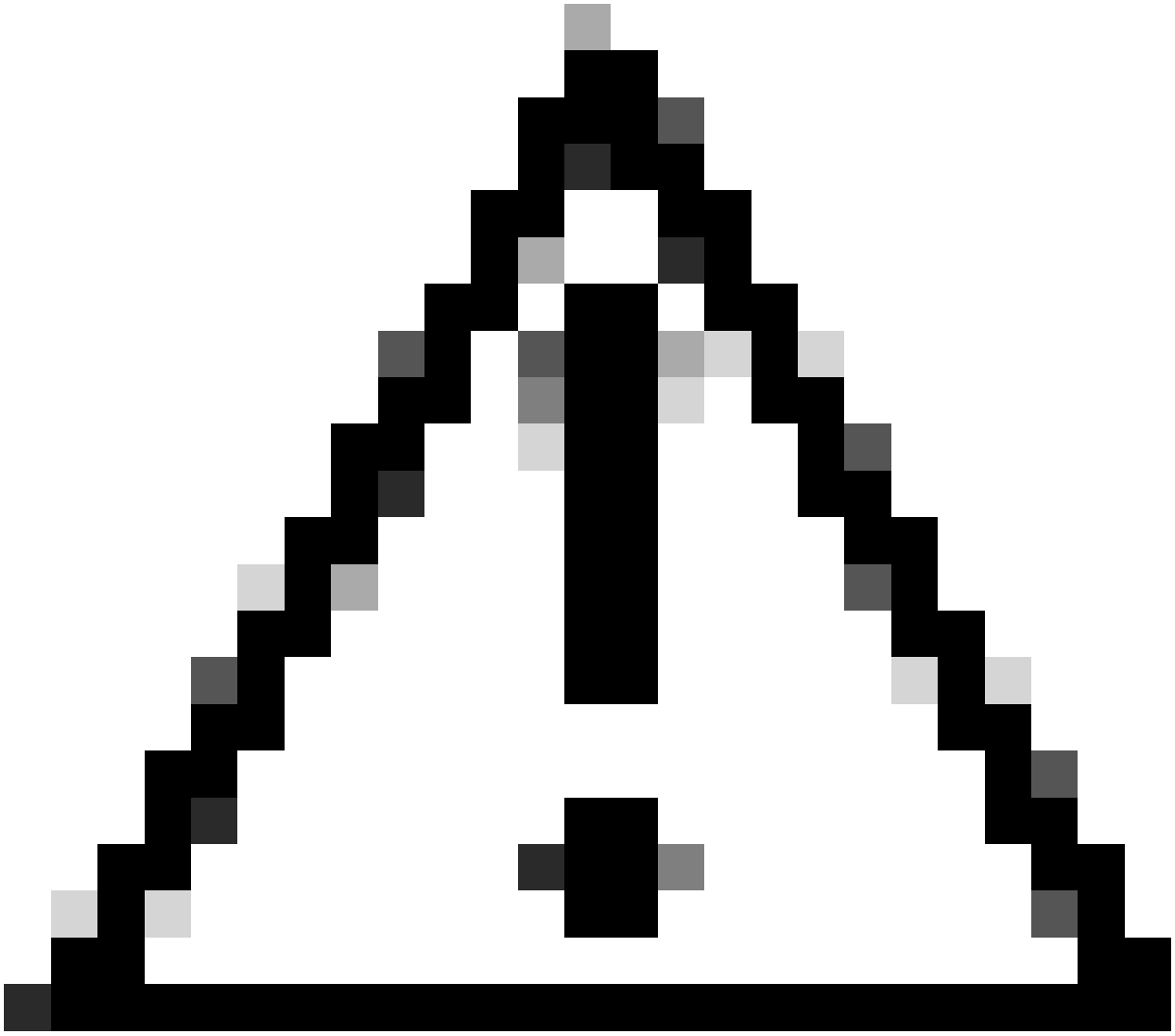
Dies ist die Dateistruktur nach der Erstellung aller anfänglichen Root- und Intermediate-CA-Dateien.

```
mariomed@CSCO-W-PF32OYP6:/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
```

Zertifikat der Stammzertifizierungsstelle erstellen

Führen Sie diesen Befehl aus, um den privaten Schlüssel für die Stammzertifizierungsstelle zu erstellen.

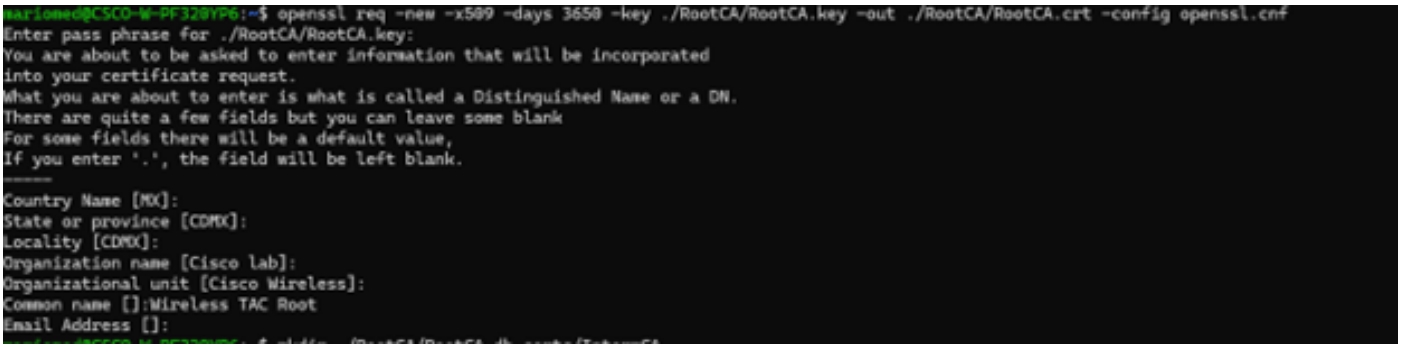
```
openssl genrsa -des3 -out ./RootCA/RootCA.key 4096
```



Achtung: OpenSSL erfordert die Eingabe einer Passphrase, wenn ein Schlüssel generiert wird. Behalten Sie die Passphrase und den generierten privaten Schlüssel an einem sicheren Ort. Jeder, der darauf Zugriff hat, kann Zertifikate als Stammzertifizierungsstelle ausstellen.

Erstellen Sie das selbstsignierte Zertifikat der Stammzertifizierungsstelle mit dem`req`Befehl auf openSSL. Das `-x509` Flag erstellt intern eine Zertifikatsanforderung (CSR) und signiert sie automatisch selbst. Bearbeiten Sie den Alternativnamen für den`-days`Parameter und den Betreff. Der Terminal fordert Sie auf, einen allgemeinen Namen anzugeben. Stellen Sie sicher, dass der eingegebene allgemeine Name mit dem alternativen Antragstellernamen (SAN) übereinstimmt.

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



Interaktive Aufforderung OpenSSL Distinguished Name

Die generierte Datei heißt RootCA.crt und befindet sich im Ordner RootCA. Diese Datei ist das Zertifikat der Stammzertifizierungsstelle.

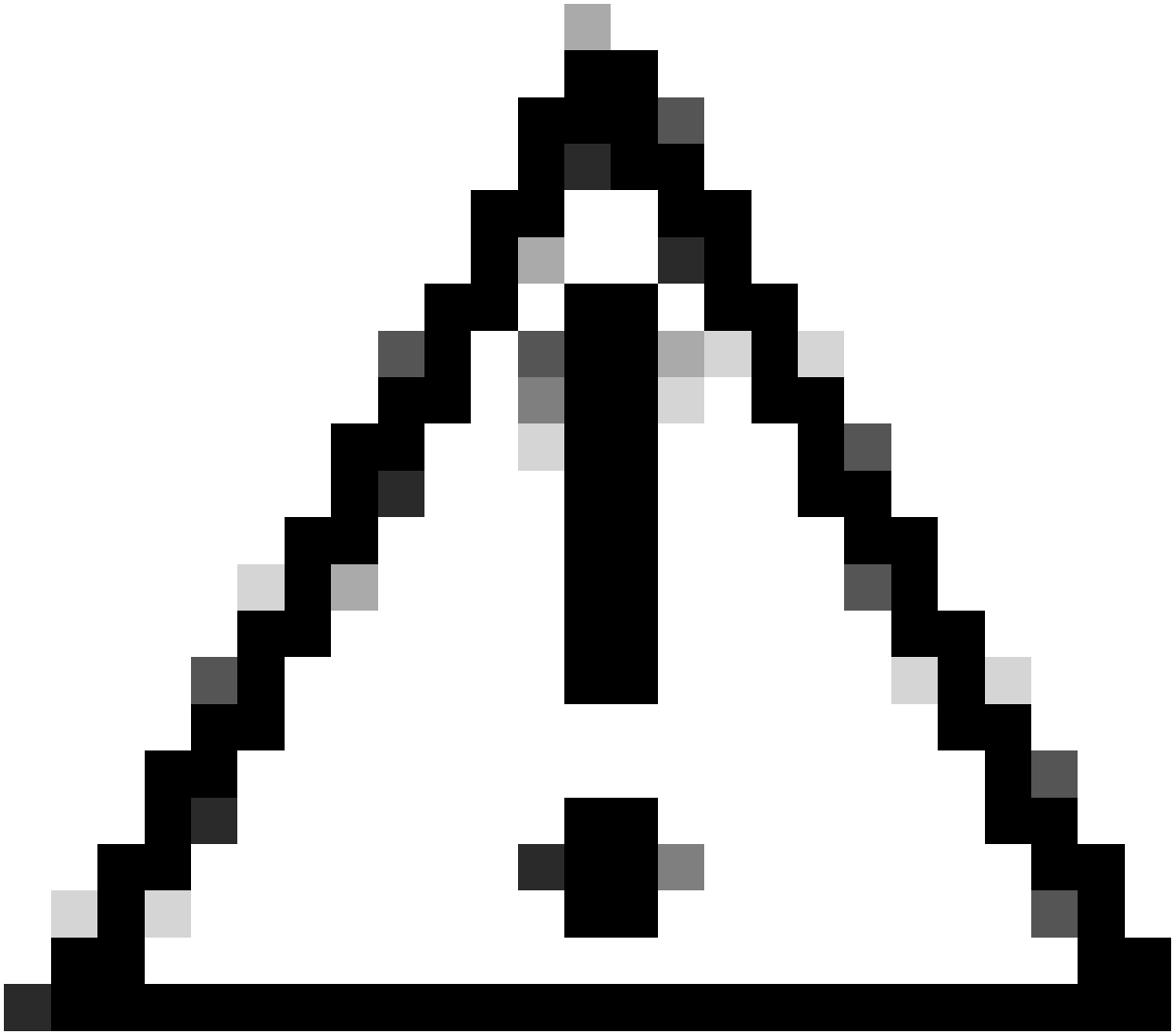
Zwischenzertifikat erstellen

Ordner zum Speichern des signierten Zwischenzertifikats der Zertifizierungsstelle im Stammordner erstellen.

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

Erstellen Sie einen privaten Schlüssel für ein Zwischenzertifikat.

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



Achtung: OpenSSL erfordert die Eingabe einer Passphrase, wenn ein Schlüssel generiert wird. Behalten Sie die Passphrase und den generierten privaten Schlüssel an einem sicheren Ort. Jeder, der darauf Zugriff hat, kann Zertifikate als Zwischenzertifikat ausstellen.

Erstellen Sie eine Signierungsanforderung für ein zwischengeschaltetes Zertifizierungsstellenzertifikat. Das Terminal fordert Sie auf, die Zertifikatinformationen einzugeben.

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

Signieren Sie Zwischen-CSR mit dem RootCA-Abschnitt der Datei openssl.cnf.

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

Die generierte Datei heißt IntermCA.crt und befindet sich im Ordner RootCA. Diese Datei ist das Zertifikat der Stammzertifizierungsstelle.

Verschieben Sie das Zwischenzertifikat und den Schlüssel in den eigenen Ordner, den Sie als Teil der ursprünglichen Dateien für die Zwischenzertifikatdatei erstellt haben.

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

Dies ist die Dateistruktur nach der Erstellung des privaten Schlüssels und der Zertifikate für die anfängliche Root- und die intermediäre Zertifizierungsstelle.

```
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
```

Erstellen von Gerätezertifikaten

Cisco IOS XE-Gerätezertifikat erstellen

Erstellen Sie einen neuen Ordner zum Speichern der Cisco IOS XE-Gerätezertifikate.

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

Erstellen Sie den privaten Geräteschlüssel IOSdevice.key und die CSR-Datei IOSdevice.csr. Verwenden Sie den Abschnitt device_req_ext, um die SANs unter diesem Abschnitt dem CSR hinzuzufügen.

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

Ändern Sie die Datei opnessl.cnf [IOS_alt_names] so, dass der vom CSR angegebene allgemeine Name mit dem SAN übereinstimmt.

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

Signieren des IOS XE-Geräte-CSR mit einem CA IntermCA-Abschnitt für die Zwischenzeit. Zeigen Sie -config auf die openSSL-Konfigurationsdatei und -extensions auf den Abschnitt IOS_cert. Dadurch bleibt das SAN auf dem signierten Zertifikat erhalten.

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

Nach diesem Schritt haben Sie ein gültiges Zertifikat für das IOS XE-Gerät mit dem Namen IOSdevice.crt mit dem passenden privaten Schlüssel IOSdevice.key erstellt.

Optional - Erstellen eines Endgerätezertifikats

An diesem Punkt haben Sie eine lokale Zertifizierungsstelle bereitgestellt und ein Zertifikat für Ihr IOS XE-Gerät ausgestellt. Sie können diese Zertifizierungsstelle auch zum Generieren von Endpunkt-Identitätszertifikaten verwenden. Diese Zertifikate sind beispielsweise auch für die lokale EAP-Authentifizierung auf 9800 Wireless LAN-Controllern oder die 802.1x-Authentifizierung

mit RADIUS-Servern gültig. In diesem Abschnitt können Sie ein Endpunktzertifikat erstellen.

Erstellen Sie einen Ordner zum Speichern der Endpunktzertifikate.

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

Ändern Sie den Abschnitt openSSL.cnf [endpoint_alt_names] so, dass der auf dem CSR angegebene allgemeine Name mit dem SAN übereinstimmt.

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

Erstellen Sie den privaten Endpunktschlüssel und den WLC-CSR unter Verwendung des Abschnitts endpoint_req_ext für SANs.

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

Signieren des Endgerätezertifikats

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

Zertifikat in Cisco IOS XE-Gerät importieren

Erstellen Sie eine Datei, die die Stammzertifizierungsstelle und die Zwischen-Zertifizierungsstelle in derselben Datei enthält, und speichern Sie sie in ./IntermCA/IntermCA.db.certs/WLC/Ordner mit dem Namen certfile.crt, wie dies für den Import auf das Cisco IOS XE-Gerät erforderlich ist.

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

Der WLC der Serie 9800 verwendet verschiedene Befehle, um die PFX-Datei für den Zertifikatimport zu erstellen. Führen Sie zum Erstellen der PFX-Datei einen dieser Befehle gemäß

der Cisco IOS XE-Version aus.

Detaillierte Informationen zum Zertifikatsimportprozess finden Sie unter Generate and Download CSR Certificates on Catalyst 9800 WLCs (CSR-Zertifikate für Catalyst 9800-WLCs generieren und herunterladen).

Bei älteren Versionen als 17.12.1:

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

Für Version 17.12.1 oder höher:

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

Importieren Sie das Zertifikat IOSdevice.pfx in das Cisco IOS XE-Gerät:

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

```
      pkcs12 [tftp://
```

```
       /
```

```
        | ftp://
```

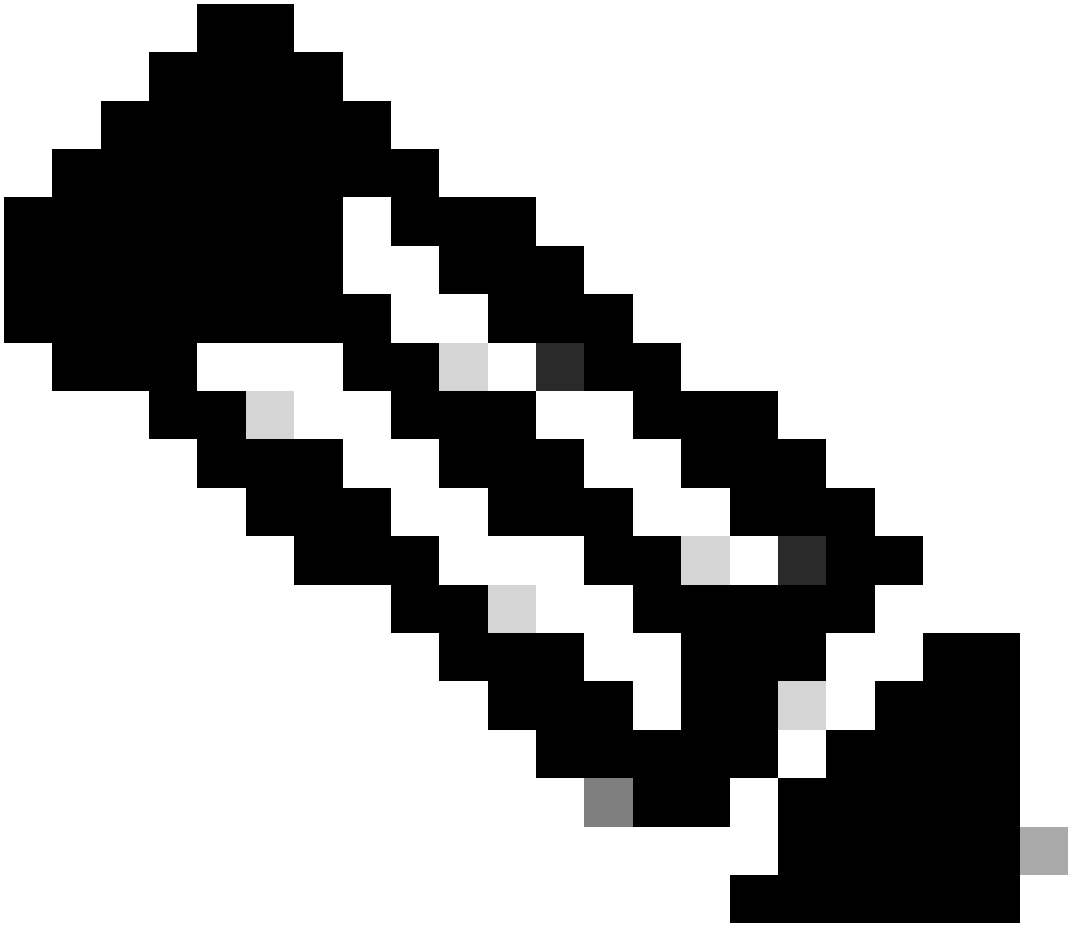
```
         /
```

| http://

/

| bootflash:

] password



Hinweis: Stellen Sie sicher, dass die für dieses Handbuch erstellten Zertifizierungsstellenzertifikate den Geräten vertrauen, die das Gerätezertifikat überprüfen müssen. Wenn das Gerätezertifikat beispielsweise für Webadministrationszwecke auf dem Cisco IOS XE-Gerät verwendet wird, müssen die Zertifizierungsstellenzertifikate auf jedem Computer oder Browser, der auf das Admin-Portal zugreift, im Vertrauensspeicher gespeichert sein.

Deaktivieren Sie die Sperrprüfung für die Zertifikate, da es keine Online-Zertifikatsperrliste gibt, die das Cisco IOS XE-Gerät von der bereitgestellten Zertifizierungsstelle überprüfen kann.
Sie müssen sie für alle Vertrauenspunkte deaktivieren, die Teil des Überprüfungspfads sind. Der Stamm-CA-Vertrauenspunkt hat denselben Namen wie der Vertrauenspunkt zwischen Geräten/Geräten, wobei die Zeichenfolge -rrr1 am Ende angehängt wird.

```
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
```

Überprüfung

Überprüfen der Zertifikatinformationen auf OpenSSL

Um die Zertifikatinformationen für die erstellten Zertifikate zu überprüfen, führen Sie auf dem Linux-Terminal den folgenden Befehl aus:

```
openssl x509 -in

      -text -noout
```

Es zeigt die vollständigen Zertifikatinformationen an.

```
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:
```

Informationen zu Cisco IOS XE-Gerätezertifikaten wie in OpenSSL dargestellt

Überprüfen der Zertifikatinformationen auf dem Cisco IOS XE-Gerät

Der Befehl`show crypto pki certificates verbose`druckt die Zertifikatinformationen aller auf dem Gerät verfügbaren Zertifikate.

```
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
```

Fehlerbehebung

Sperrprüfung wurde eingerichtet

Wenn die Zertifikate in Cisco IOS XE importiert werden, ist für die neu erstellten Vertrauenspunkte die Sperrprüfung aktiviert. Wenn dem Gerät, das die importierten Zertifikatvertrauensstellen für die Validierung verwenden muss, ein Zertifikat vorgelegt wird, sucht das Gerät nach einer nicht vorhandenen Zertifikatsperrliste und schlägt fehl. Die Nachricht wird auf das Terminal gedruckt.

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

Stellen Sie sicher, dass jeder Vertrauenspunkt im Prüfpfad für die Zertifikate den Befehl`revocation-check none`enthält.

Zugehörige Informationen

- Erstellen und Herunterladen von CSR-Zertifikaten auf Catalyst 9800 WLCs
- Konfigurieren von Zertifikaten der Zertifizierungsstelle mithilfe der IOS XE PKI
- Sicherheits- und VPN-Konfigurationsleitfaden, Cisco IOS XE 17.x
- Informationen zum Zertifikat für die Erstellung einer Kette für den 9800 WLC

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.