# Configurer une autorité de certification multiniveau sur OpenSSL pour générer des certificats IOS XE

## Table des matières



## Introduction

Ce document décrit une méthode de création d'une autorité de certification multiniveau pour créer des certificats à usage général compatibles avec les périphériques Cisco IOS® XE.

## Conditions préalables

### Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Comment utiliser l’application OpenSSL.
- Infrastructure à clé publique (PKI) et certificats numériques.

### Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Application OpenSSL (version 3.0.2).
- WLC 9800 (Cisco IOS XE version 17.12.3).

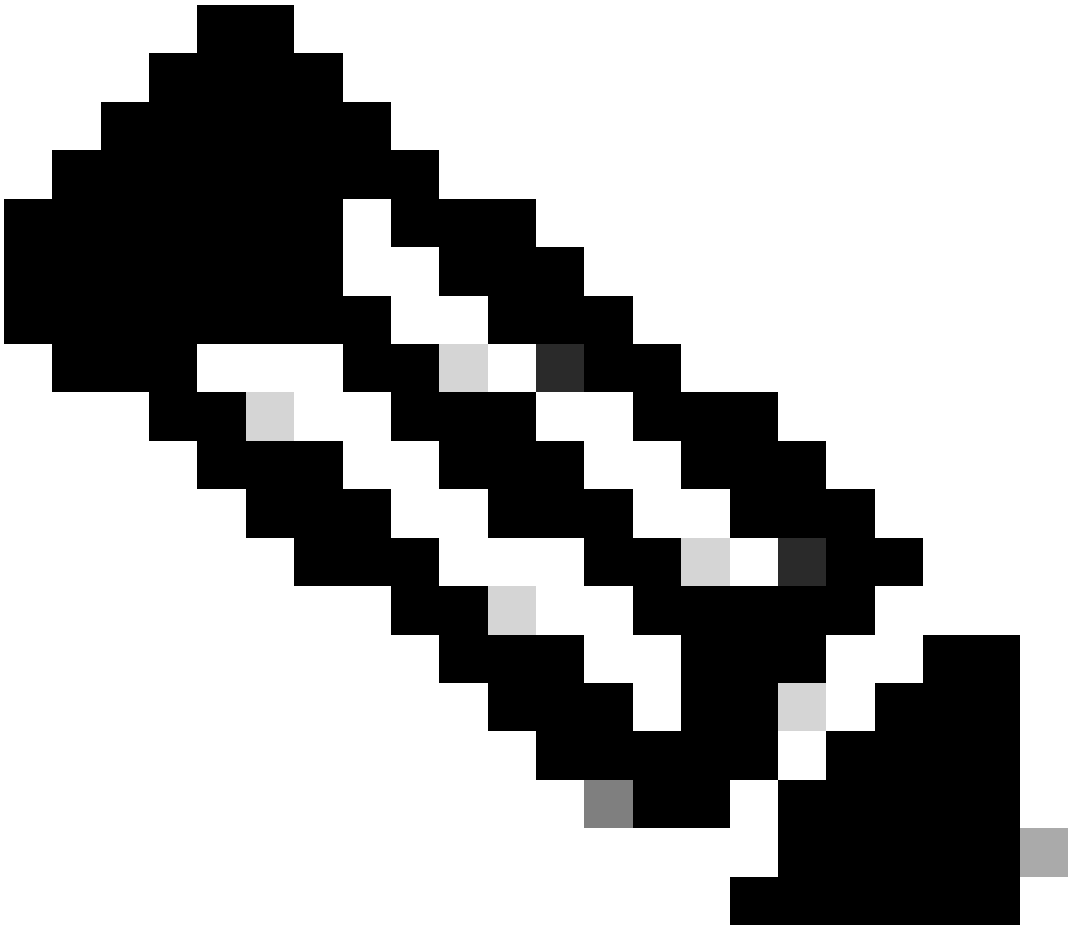
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

# Configurer

## Aperçu

L'objectif est de créer une autorité de certification locale à deux niveaux avec une autorité de certification racine et une autorité de certification intermédiaire pour signer les certificats des périphériques. Une fois les certificats signés, ils sont importés sur le périphérique Cisco IOS XE.

---



---

Remarque : ce document utilise des commandes Linux spécifiques pour créer et organiser des fichiers. Les commandes sont expliquées afin que vous puissiez effectuer la même action sur d'autres systèmes d'exploitation où OpenSSL est disponible.

## Préparation du fichier de configuration OpenSSL

Créez un fichier texte appelé openssl.conf à partir de votre répertoire de travail actuel sur l'ordinateur sur lequel OpenSSL est installé. Copiez et collez ces lignes pour fournir à OpenSSL les configurations nécessaires à la signature des certificats. Vous pouvez modifier ce fichier en fonction de vos besoins.

```
[ 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
```

## Créer des fichiers initiaux pour les autorités de certification

Créez un dossier sur le répertoire courant appelé RootCA. À l'intérieur, créez 3 autres dossiers appelés RootCA.db.tmp, RootCA.db.certs et RootCA.db.crl.

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

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

Créez un fichier appelé RootCA.db.serial dans le dossier RootCA. Ce fichier doit contenir la valeur initiale pour le numéro de série des certificats, 01 est la valeur sélectionnée sur ce cas.

Créez un fichier appelé RootCA.db.crlserial dans le dossier RootCA. Ce fichier doit contenir la valeur initiale du numéro de liste de révocation de certificats, 01 est la valeur sélectionnée dans ce cas.

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

Créez un fichier appelé RootCA.db.index dans le dossier RootCA.

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

Créez un fichier nommé RootCA.db.rand dans le dossier RootCA et remplissez-le avec 8192 octets aléatoires pour servir de valeur de départ du générateur de nombres aléatoires interne.

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

Créez un dossier sur le répertoire actuel appelé IntermCA. À l'intérieur, créez 3 autres dossiers appelés IntermCA.db.tmp, IntermCA.db.certs et IntermCA.db.crl.

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

Créez un fichier appelé IntermCA.db.serial dans le dossier IntermCA. Ce fichier doit contenir la valeur initiale pour le numéro de série des certificats, 01 est la valeur sélectionnée sur ce cas.

Créez un fichier appelé IntermCA.db.crlserial dans le dossier IntermCA. Ce fichier doit contenir la valeur initiale du numéro de liste de révocation de certificats, 01 est la valeur sélectionnée dans ce cas.

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

Créez un fichier nommé IntermCA.db.index dans le dossier IntermCA.

Créez un fichier nommé IntermCA.db.rand dans le dossier IntermCA et remplissez-le avec 8192 octets aléatoires pour servir de valeur de départ du générateur de nombres aléatoires interne.

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

Créez un fichier nommé IntermCA.db.rand dans le dossier IntermCA et remplissez-le avec 8192 octets aléatoires pour servir de valeur de départ du générateur de nombres aléatoires interne.

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

Il s'agit de la structure de fichiers après la création de tous les fichiers CA racine et intermédiaire initiaux.

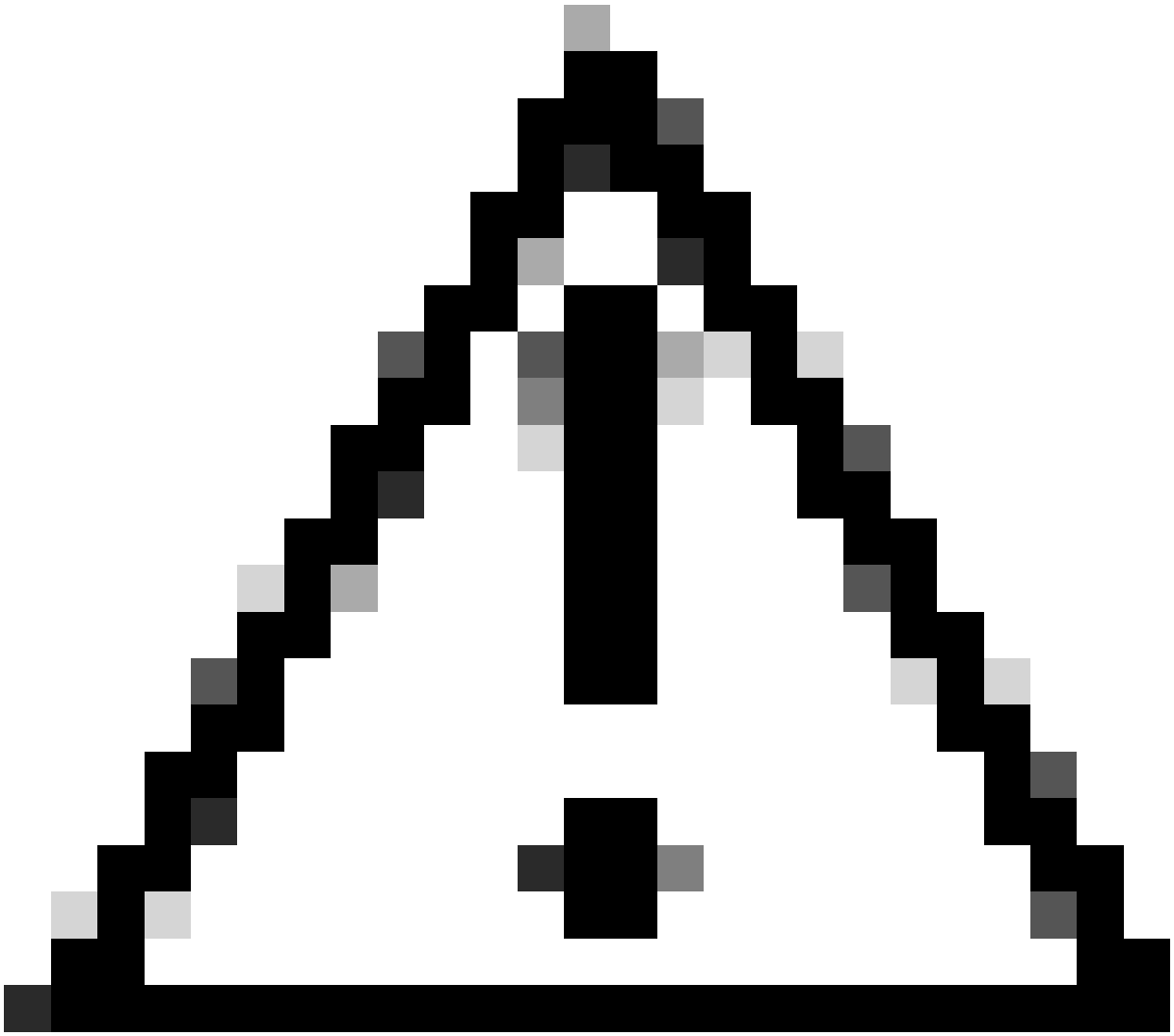
```
mariomed@CSCO-W-PF320YP6:/mnt/c/Users/mariomed/radsecfiles1$ tree
.
├── IntermCA
│   ├── IntermCA.db.certs
│   ├── IntermCA.db.crl
│   ├── IntermCA.db.crlserial
│   ├── IntermCA.db.index
│   ├── IntermCA.db.rand
│   ├── IntermCA.db.serial
│   └── IntermCA.db.tmp
├── RootCA
│   ├── RootCA.db.certs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

## Créer un certificat CA racine

Exécutez cette commande pour créer la clé privée pour l'autorité de certification racine.

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

---

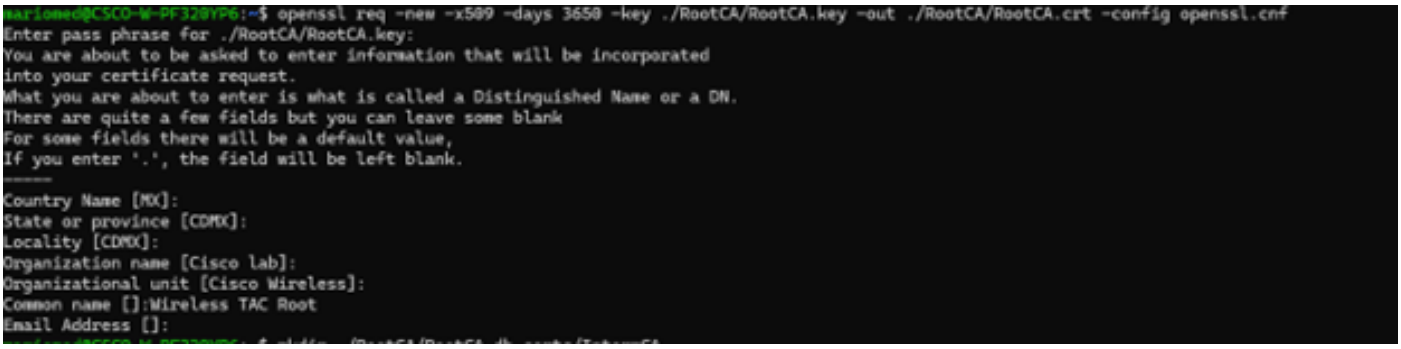


Attention : OpenSSL nécessite la saisie d'une phrase de passe lors de la génération d'une clé. Conservez la phrase de passe secrète et la clé privée générée dans un emplacement sécurisé. Toute personne y ayant accès peut émettre des certificats en tant qu'autorité de certification racine.

---

Créez le certificat auto-signé de l'autorité de certification racine à l'aide de la commande`req`on openSSL. L'`-x509`indicateur crée en interne une demande de signature de certificat (CSR) et la signe automatiquement. Modifiez le paramètre et le`-days`nom alternatif de l'objet. Le terminal vous invite à fournir un nom commun. Assurez-vous que le nom commun que vous entrez correspond au nom alternatif du sujet (SAN).

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




Invite interactive OpenSSL Distinguished Name

Le fichier généré est appelé RootCA.crt et se trouve dans le dossier RootCA. Ce fichier est le certificat de l'autorité de certification racine.

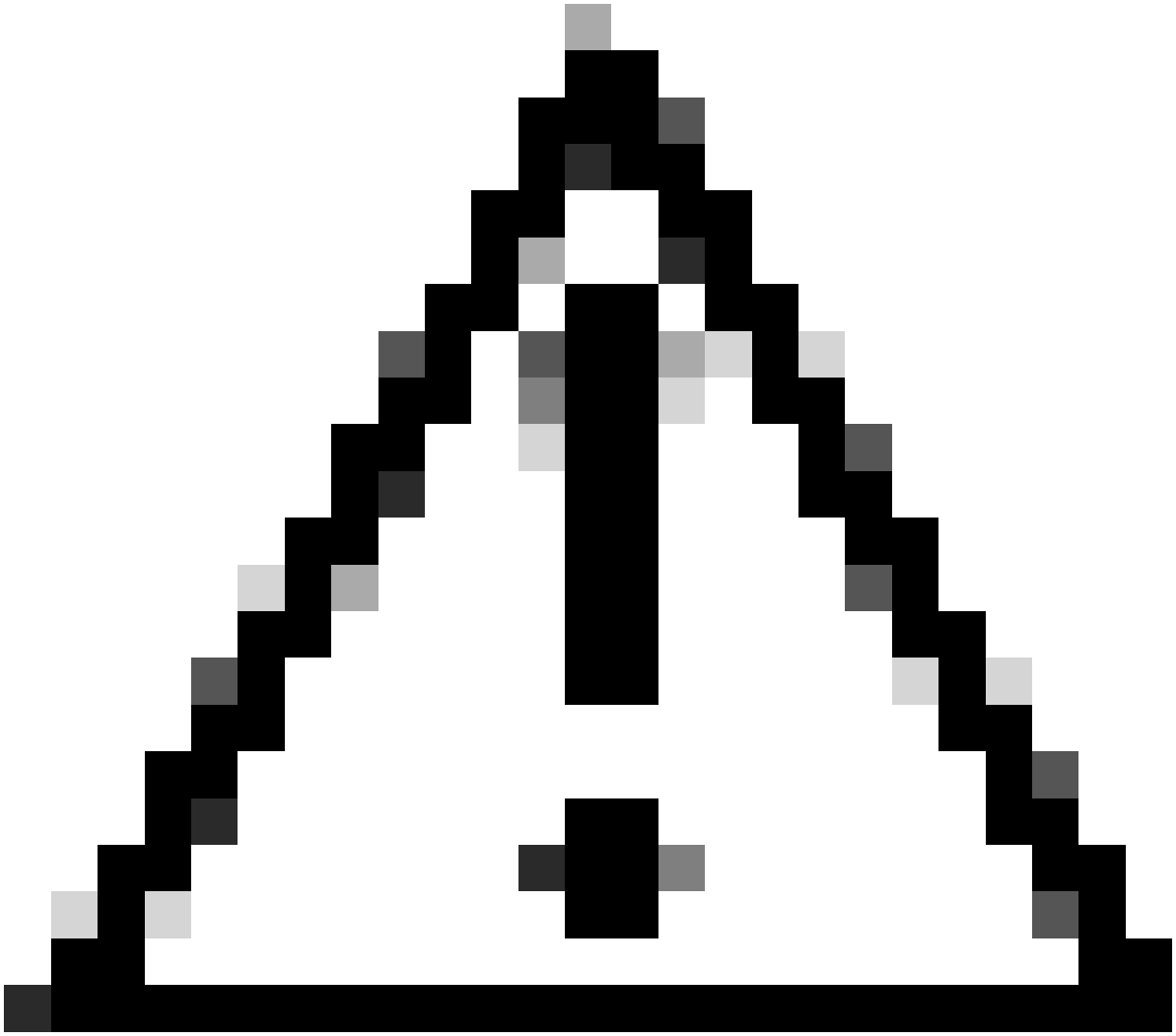
## Créer un certificat CA intermédiaire

Créez un dossier pour stocker le certificat AC intermédiaire signé dans le dossier racine.

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

Créez une clé privée pour le certificat intermédiaire.

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



Attention : OpenSSL nécessite la saisie d'une phrase de passe lors de la génération d'une clé. Conservez la phrase de passe secrète et la clé privée générée dans un emplacement sécurisé. Toute personne y ayant accès peut émettre des certificats en tant qu'autorité de certification intermédiaire.

Créer une demande de signature de certificat CA intermédiaire. Le terminal vous invite à saisir les informations du certificat.

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

Signez le CSR intermédiaire avec la section RootCA du fichier openssl.cnf.

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

Le fichier généré s'appelle IntermCA.crt et se trouve dans le dossier RootCA. Ce fichier est le certificat de l'autorité de certification racine.

Déplacez le certificat intermédiaire et la clé vers son propre dossier que vous avez créé dans le cadre des fichiers initiaux de l'autorité de certification intermédiaire.

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

Il s'agit de la structure de fichiers après la création de la clé privée et des certificats pour les autorités de certification racines initiales et intermédiaires.

```
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
```

## Créer des certificats de périphérique

Créer un certificat de périphérique Cisco IOS XE

Créez un nouveau dossier pour stocker les certificats de périphérique Cisco IOS XE.

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

Créez la clé privée du périphérique IOSdevice.key et le périphérique CSR IOSdevice.csr. Utilisez la section device_req_ext pour ajouter les SAN sous cette section sur le CSR.

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

Modifiez la section du fichier opnessl.cnf [IOS_alt_names] afin que le nom commun que vous fournissez sur le CSR corresponde au SAN.

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

Signez le CSR du périphérique IOS XE avec la section intermédiaire CA IntermCA. Utilisez -config pour pointer vers le fichier de configuration openSSL et -extensions pour pointer vers la section IOS_cert. Le SAN reste ainsi sur le certificat signé.

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

Après cette étape, vous avez créé un certificat valide pour le périphérique IOS XE appelé IOSdevice.crt avec la clé privée correspondante IOSdevice.key.

Facultatif - Créer un certificat de terminal

À ce stade, vous avez déployé une autorité de certification locale et émis un certificat pour votre périphérique IOS XE. Vous pouvez également utiliser cette autorité de certification pour générer des certificats d'identité de point de terminaison. Ces certificats sont également valides, par exemple, pour effectuer l'authentification EAP locale sur les contrôleurs LAN sans fil 9800 ou même l'authentification dot1x avec les serveurs RADIUS. Cette section vous aide à générer un certificat de point de terminaison.

Créez un dossier pour stocker les certificats de point de terminaison.

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

Modifiez la section openSSL.cnf file [ endpoint_alt_names ] afin que le nom commun que vous fournissez sur le CSR corresponde au SAN.

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

Créez la clé privée du point de terminaison et le CSR du WLC à l'aide de la section endpoint_req_ext pour les SAN.

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

Signez le certificat de périphérique du terminal.

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

## Importer un certificat sur le périphérique Cisco IOS XE

Créez un fichier qui contient l'autorité de certification racine et l'autorité de certification intermédiaire sur le même fichier et enregistrez-le dans le dossier ./IntermCA/IntermCA.db.certs/WLC/ avec le nom certfile.crt comme requis pour l'importation vers le périphérique Cisco IOS XE.

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

Le WLC de la gamme 9800 utilise différentes commandes pour créer le fichier pfx pour l'importation de certificat. Pour créer votre fichier pfx, exécutez l'une de ces commandes selon la version de Cisco IOS XE.

Référez-vous à Générer et télécharger des certificats CSR sur des WLC Catalyst 9800 pour des

informations détaillées sur le processus d'importation de certificat

Pour les versions antérieures à 17.12.1 :

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

Pour la version 17.12.1 ou ultérieure :

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

Importez le certificat IOSdevice.pfx vers le périphérique Cisco IOS XE :

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

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

```
       /
```

```
        | ftp://
```

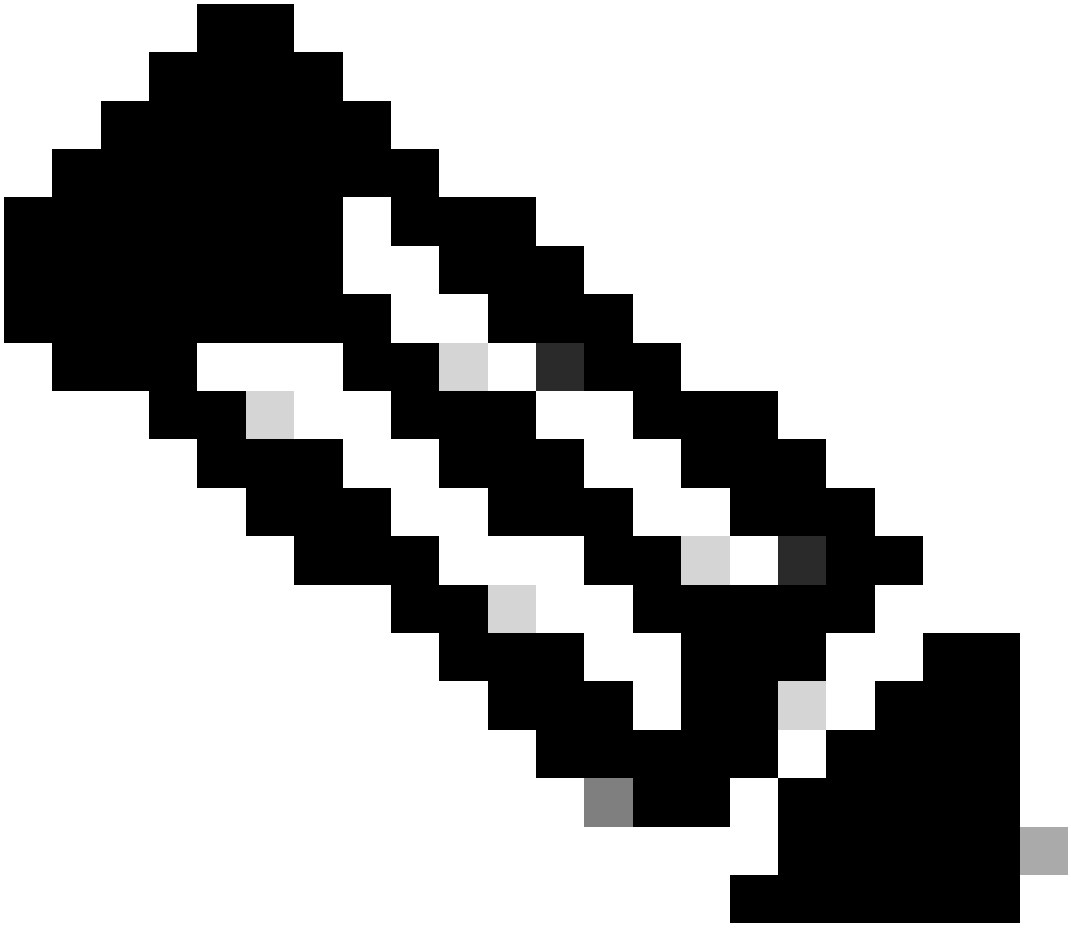
```
         /
```

```
          | http://
```

/

| bootflash:

] password



> Remarque : assurez-vous que les certificats d'autorité de certification créés pour ce guide sont approuvés par les périphériques qui doivent vérifier le certificat de périphérique. Par exemple, si le certificat du périphérique est utilisé à des fins d'administration Web sur le périphérique Cisco IOS XE, tout ordinateur ou navigateur accédant au portail d'administration doit disposer des certificats CA dans son magasin de confiance.

Désactivez la vérification de révocation des certificats car il n'existe pas de liste de révocation des certificats en ligne que le périphérique Cisco IOS XE peut vérifier à partir de l'autorité de certification que vous avez déployée.
Vous devez le désactiver sur tous les points de confiance qui font partie du chemin de vérification. Le point de confiance de l'autorité de certification racine a le même nom que le point de confiance Intermédiaire/Périphérique avec la chaîne -rrr1 ajoutée à la fin.

```
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
```

# Vérifier

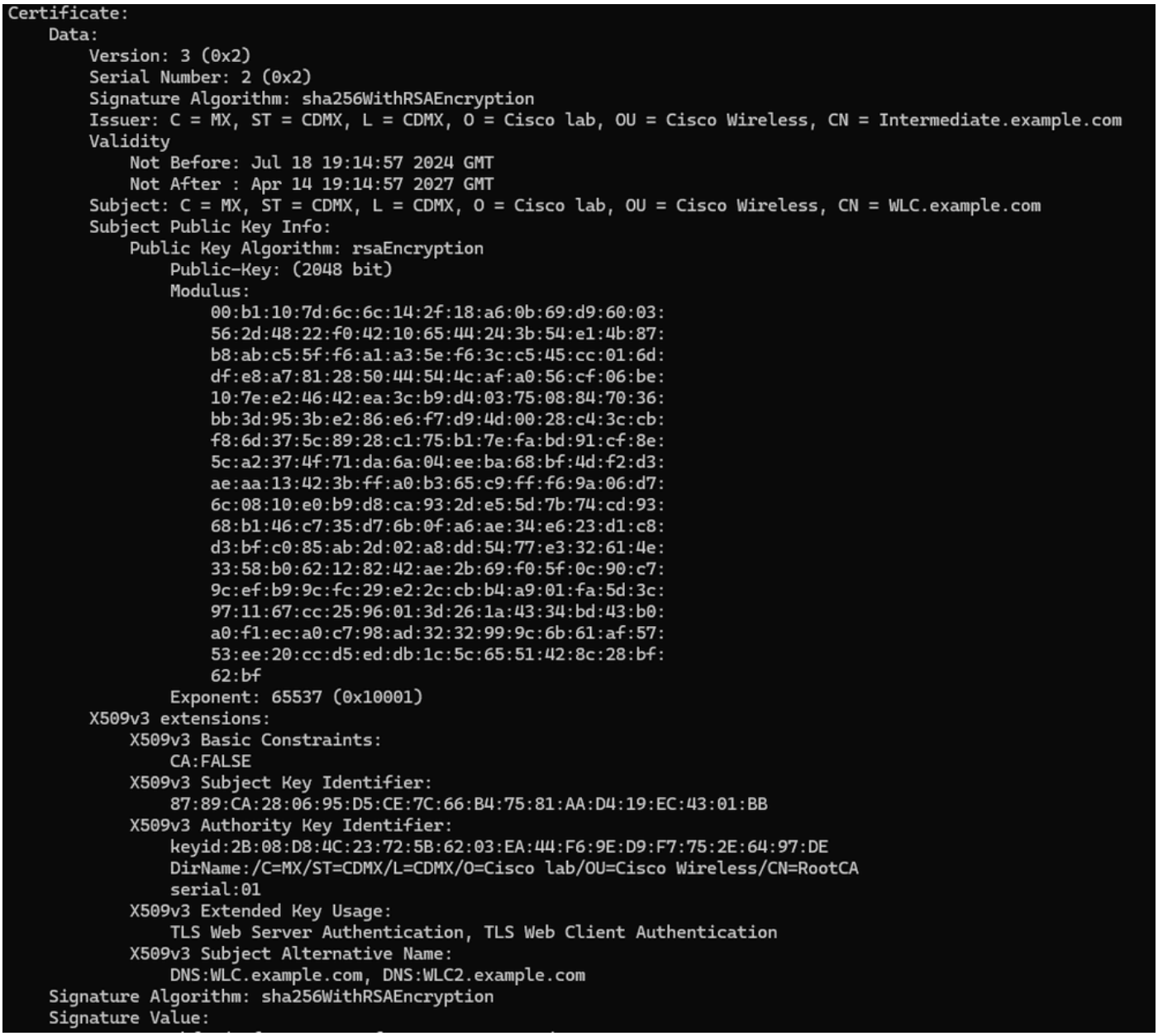
## Vérifier les informations de certificat sur OpenSSL

Pour vérifier les informations de certificat pour les certificats créés, exécutez la commande suivante sur le terminal Linux :

```
openssl x509 -in

      -text -noout
```

Elle affiche les informations complètes du certificat.




Informations de certificat du périphérique Cisco IOS XE, comme indiqué par OpenSSL

Vérifiez les informations de certificat sur le périphérique Cisco IOS XE.

Cette commande`show crypto pki certificates verbose`imprime les informations de certificat de tous les certificats disponibles sur le périphérique.

```
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
```

# Dépannage

## La vérification de révocation est en place

Lorsque les certificats sont importés dans Cisco IOS XE, le contrôle de révocation est activé sur les points de confiance nouvellement créés. Si un certificat est présenté au périphérique qui doit utiliser les points de confiance de certificat importés pour validation, le périphérique recherche une liste de révocation de certificats inexistante et échoue. Le message est imprimé sur le terminal.

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

Assurez-vous que chaque point de confiance du chemin de vérification des certificats contient la commande`revocation-check none`.

# Informations connexes

- Générer et télécharger des certificats CSR sur les WLC Catalyst 9800
- Configuration des certificats signés CA avec IOS XE PKI
- Guide de configuration de la sécurité et du VPN, Cisco IOS XE 17.x
- Comprendre les informations de certificat pour créer une chaîne pour le WLC 9800

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.