تكوين CA متعدد المستويات على OpenSSL لإنشاء شهادات IOS XE

المحتويات

المقدمة

المتطلبات الأساسية

المتطلبات

المكونات المستخدمة

التكوين

قم بإعداد التكوين

تحضير ملف تكوين OpenSSL

إنشاء ملفات أولية للهيئات المصدقة

إنشاء شهادة مرجع مصدق فرعي

إنشاء شهادة CA وسيطة

إنشاء حزم الشهادات

إنشاء حزمة شهادات Cisco IOS XE

إنشاء حزمة شهادات CA - اختياري

استيراد الشهادة إلى حزمة Cisco IOS XE

التحقق من الصحة

التحقق من معلومات الشهادة على OpenSSL

استكشاف الأخطاء وإصلاحها

التحقق من الباطل الموجود في المكان

معلومات ذات صلة

المقدمة

يصف هذا المستند طريقة إنشاء CA متعدد المستويات لإنشاء شهادات الأغراض العامة المتوافقة مع أجهزة Cisco IOS® XE.

المتطلبات الأساسية

المتطلبات

توصي Cisco بأن تكون لديك معرفة بالمواضيع التالية:

- كيفية إستخدام تطبيق OpenSSL.
- البنية الأساسية للمفتاح العام (PKI) والشهادات الرقمية.

المكونات المستخدمة

تستند المعلومات الواردة في هذا المستند إلى إصدارات البرامج والمكونات المادية التالية:

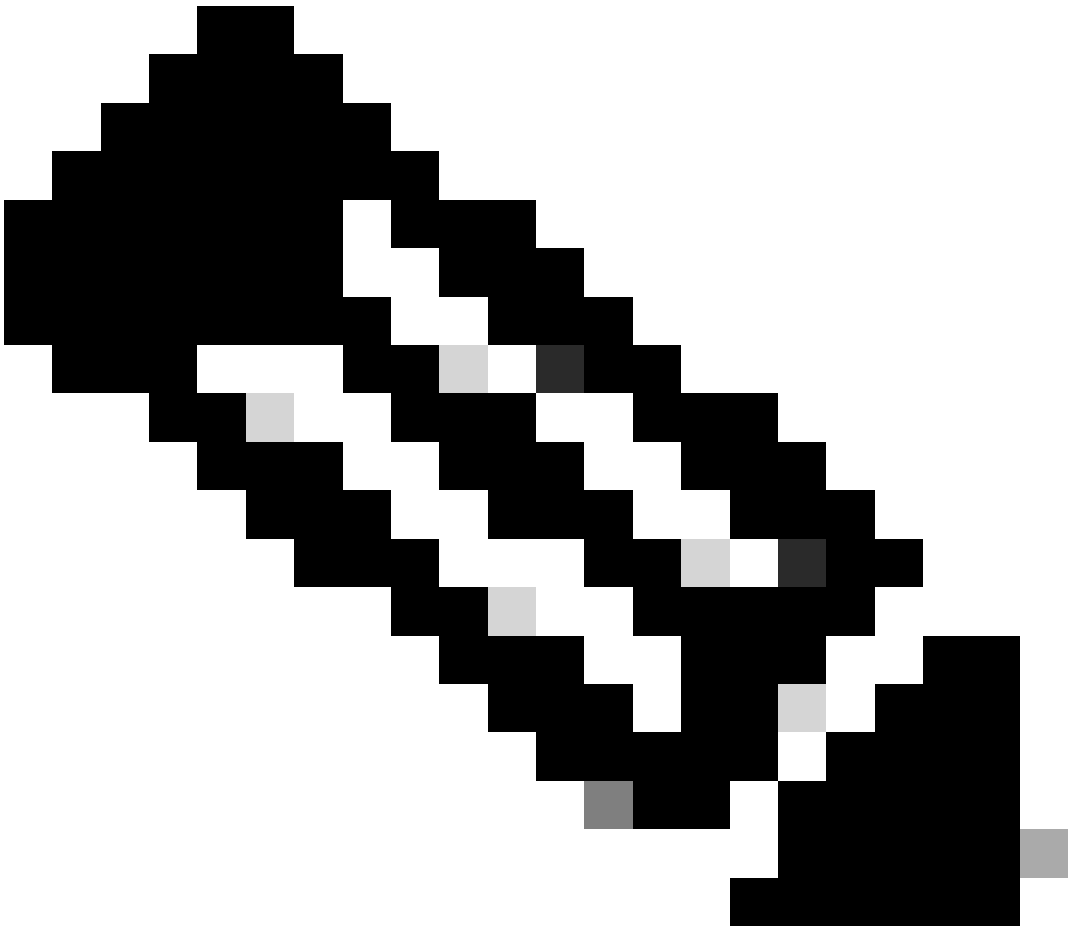
- تطبيق OpenSSL (الإصدار 3.0.2).
- 9800 WLC (Cisco IOS XE، الإصدار 17.12.3).

تم إنشاء المعلومات الواردة في هذا المستند من الأجهزة الموجودة في بيئة معملية خاصة. بدأت جميع الأجهزة المُستخدمة في هذا المستند بتكوين ممسوح (افتراضي). إذا كانت شبكتك قيد التشغيل، فتأكد من فهمك للتأثير المحتمل لأي أمر.

التكوين

نظرة عامة

الغرض هو إنشاء مرجع تصديق محلي مستويني (CA) مع مرجع مصدق رئيسي و CA متوسط لتوقيع شهادات الجهاز. بمجرد توقيع الشهادات، يتم إستيرادها إلى جهاز Cisco IOS XE.



ملاحظة: يستخدم هذا المستند أوامر Linux المحددة لإنشاء الملفات وترتيبها. يتم شرح الأوامر بحيث يمكن تنفيذ الإجراء نفسه على أنظمة التشغيل الأخرى حيث يتوفر OpenSSL.

تحضير ملف تكوين OpenSSL

إنشاء ملف نصي يسمى openssl.conf من دليل العمل الحالي على الجهاز المثبت OpenSSL.
قم بنسخ هذه الأسطر ولصقها لتوفير OpenSSL بالتكوينات اللازمة لتوقيع الشهادة.
يمكنك تحرير هذا الملف ليناسب إحتياجاتك.

```
[ 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
```

إنشاء ملفات أولية للهيئات المصدقة

قم بإنشاء مجلد على الدليل الحالي يسمى RootCA. بداخله، قم بإنشاء 3 مجلدات أخرى تسمى RootCA.db.tmp، RootCA.db.certs، وRootCA.db.crl.

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

قم بإنشاء ملف يسمى RootCA.db.seبالريال داخل مجلد RootCA. يجب أن يحتوي هذا الملف على القيمة الأولية للرقم التسلسلي للشهادات، 01 هي القيمة المحددة في هذه الحالة.

قم بإنشاء ملف يسمى RootCA.db.crSerial داخل مجلد RootCA. يجب أن يحتوي هذا الملف على القيمة الأولية لرقم قائمة إبطال الشهادة، 01 هي القيمة المحددة في هذه الحالة.

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

قم بإنشاء ملف يسمى RootCA.db.index داخل مجلد RootCA.

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

قم بإنشاء ملف باسم RootCA.db.rand داخل مجلد RootCA وقم بملئه بـ 8192 بايت عشوائي ليعمل كقاعدة لمولد الأرقام العشوائية الداخلي.

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

قم بإنشاء مجلد على الدليل الحالي يسمى IntermCA. بداخله، قم بإنشاء 3 مجلدات أخرى تسمى IntermCA.db.tmp، IntermCA.db.certs، وIntermCA.db.crl.

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

قم بإنشاء ملف يسمى IntermCA.db.serial داخل مجلد IntermCA. يجب أن يحتوي هذا الملف على القيمة الأولية للرقم التسلسلي للشهادات، 01 هي القيمة المحددة في هذه الحالة.

قم بإنشاء ملف يسمى IntermCA.db.crlserial داخل مجلد IntermCA. يجب أن يحتوي هذا الملف على القيمة الأولية لرقم قائمة إبطال الشهادة، 01 هي القيمة المحددة في هذه الحالة.

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

قم بإنشاء ملف باسم IntermCA.db.index داخل مجلد IntermCA.

قم بإنشاء ملف باسم IntermCA.db.rand داخل مجلد IntermCA وقم بملئه بـ 8192 بايت عشوائي ليعمل كقاعدة لمولد الأرقام العشوائية الداخلي.

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

قم بإنشاء ملف باسم IntermCA.db.rand داخل مجلد IntermCA وقم بملئه بـ 8192 بايت عشوائي ليعمل كقاعدة لمولد الأرقام العشوائية الداخلي.

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

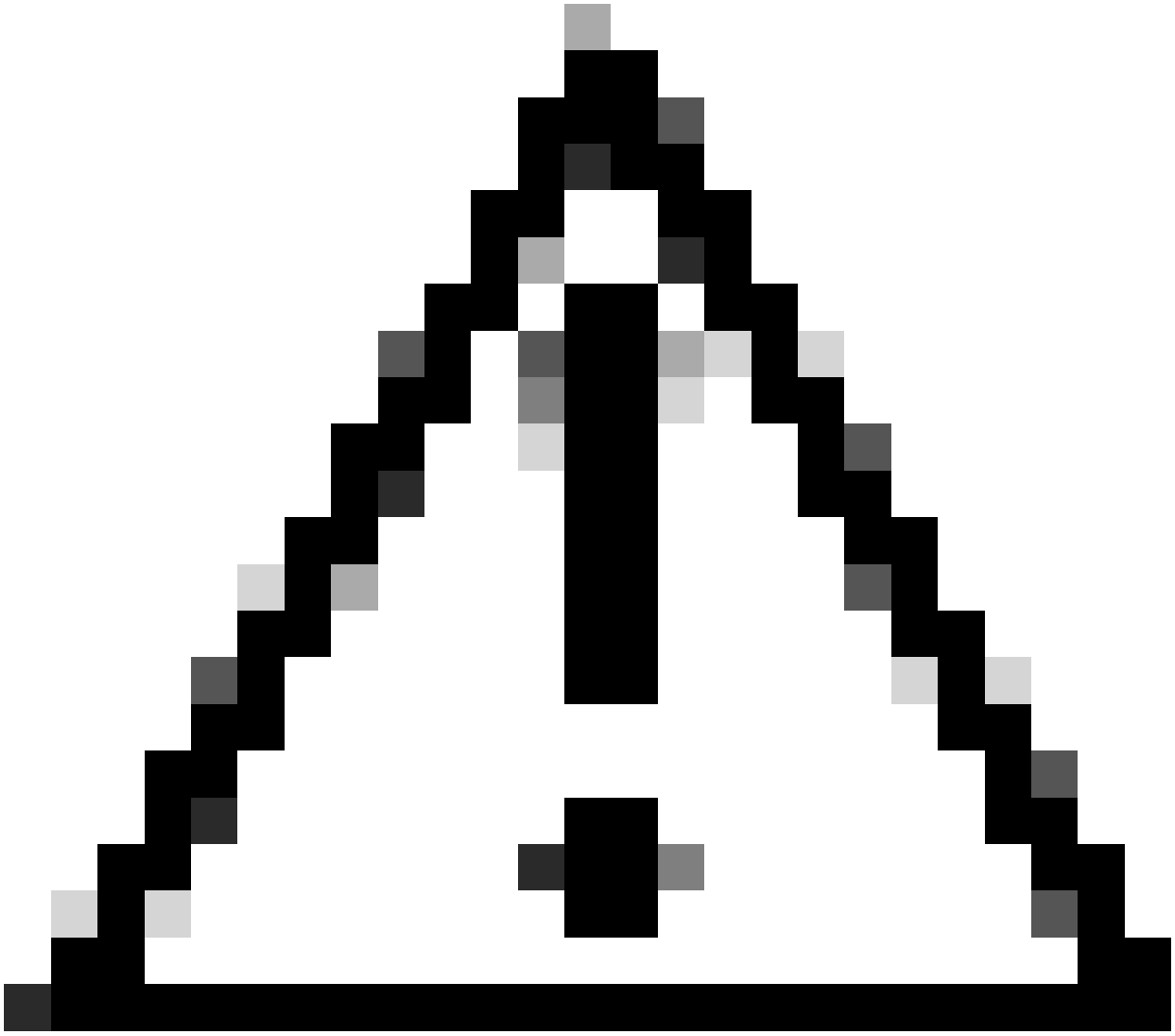
هذا هو هيكل الملف بعد إنشاء كل ملفات CA الجذر الأولي والمتوسط.

```
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
```

إنشاء شهادة مرجع مصدق جذري

قم بتشغيل هذا الأمر لإنشاء المفتاح الخاص لـ CA الجذر.

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



تحذير: يتطلب OpenSSL منك توفير عبارة مرور عند إنشاء مفتاح. أبق عبارة المرور سرية والمفتاح الخاص الذي تم إنشاؤه على موقع آمن. يمكن لأي شخص يمكنه الوصول إليها إصدار شهادات CA الجذر الخاص بك.

قم بإنشاء شهادة CA موقعة ذاتيا الجذر باستخدامreqالأمر على OpenSSL. تنشئ -x509 العلامة داخليا طلب توقيع شهادة (CSR) وتوقعه ذاتيا تلقائيا. قم بتحرير-daysالعلامة واسم الموضوع البديل. يطالبك المفتاح بتقديم اسم مشترك. تأكد من تطابق الاسم الشائع الذي أدخلته مع الاسم البديل للموضوع (SAN).

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

```
mariomed@CSCO-W-PF328YP6:~$ openssl req -new -x509 -days 3650 -key ./RootCA/RootCA.key -out ./RootCA/RootCA.crt -config openssl.cnf
Enter pass phrase for ./RootCA/RootCA.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name [MX]:
State or province [CDMX]:
Locality [CDMX]:
Organization name [Cisco lab]:
Organizational unit [Cisco Wireless]:
Common name []:Wireless TAC Root
Email Address []:
```

المطالبة التفاعلية لاسم OpenSSL المميز

يسمى الملف الذي تم إنشاؤه RootCA.crt ويوجد داخل مجلد RootCA. هذا الملف هو شهادة المرجع المصدق الجذر.

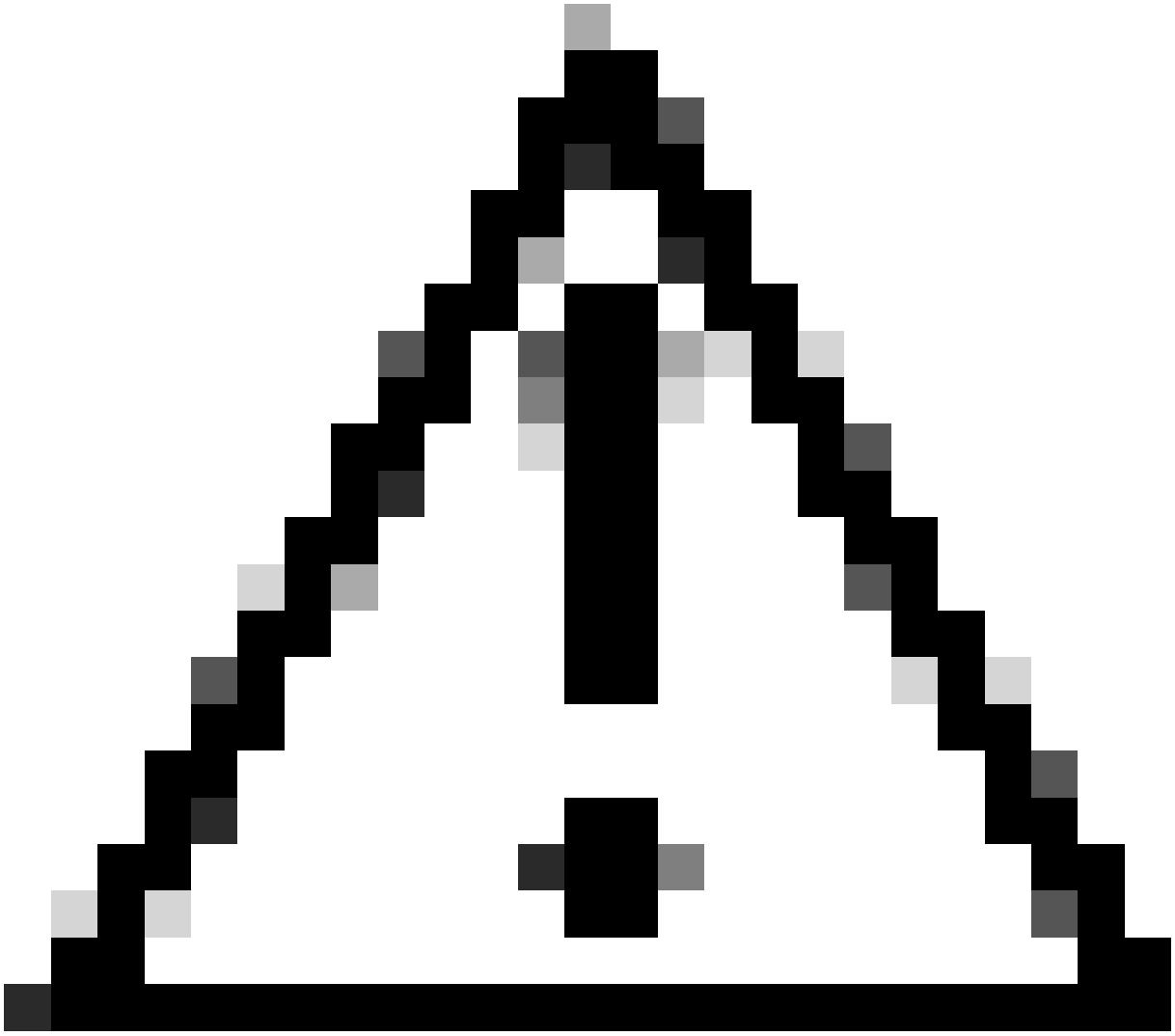
إنشاء شهادة CA وسيطة

قم بإنشاء مجلد لتخزين شهادة CA الوسيطة الموقعة داخل المجلد الجذر.

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

إنشاء مفتاح خاص للشهادة الوسيطة.

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



تحذير: يتطلب OpenSSL منك توفير عبارة مرور عند إنشاء مفتاح. أبق عبارة المرور سرية والمفتاح الخاص الذي تم إنشاؤه على موقع آمن. يمكن لأي شخص لديه حق الوصول إليه إصدار شهادات CA الوسيط الخاص بك.

إنشاء طلب توقيع شهادة مرجع مصدق وسيط. تطلب منك المحطة الطرفية إدخال معلومات الشهادة.

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

قم بتوقيع CSR متوسط باستخدام قسم RootCA من ملف openssl.cnf.

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

يسمى الملف الذي تم إنشاؤه IntermCA.crt ويوجد داخل مجلد RootCA. هذا الملف هو شهادة المرجع المصدق الجذر.

انقل الشهادة الوسيطة والمفاتيح إلى مجلده الذي أنشأته كجزء من الملفات الأولية للمرجع المصدق الوسيط.

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

هذا هو هيكل الملف بعد إنشاء المفاتيح الخاص والتاريخي لكل من الجذر الأولي والتاريخي الوسيطة.

```
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
```

إنشاء شهادات جهاز

إنشاء شهادة جهاز Cisco IOS XE

إنشاء مجلد جديد لتخزين شهادات جهاز Cisco IOS XE.

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

قم بإنشاء الجهاز مفاتيح خاص IOSdevice.key والجهاز CSR IOSdevice.csr. أستخدم القسم device_req_ext لإضافة شبكات SAN ضمن القسم المذكور إلى CSR.

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

قم بتعديل قسم ملف [IOS_ALT_NAMES] openSSL.cnf بحيث يتطابق الاسم الشائع الذي توفره على CSR مع شبكة منطقة التخزين (SAN).

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

توقيع IOS XE جهاز CSR مع قسم CA IntermCA الوسيط. أستخدم -config للإشارة إلى ملف قسم IOS_CERT. وللإشارة إلى قسم -extensions OpenSSL تكوين هذا إلى إبقاء شبكة منطقة الذي يؤدي إلى التخزين على الشهادة الموقعة.

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

بعد هذه الخطوة، قمت بإنشاء شهادة صالحة لجهاز IOS XE تسمى IOSDevice.crt مع الملفات الخاص المطابق IOSdevice.key.

إختياري - إنشاء شهادة نقطة نهاية

عند هذه النقطة، قمت بنشر مرجع مصدق محلي وإصدار شهادة واحدة لجهاز IOS XE الخاص بك. يمكنك أيضا إستخدام المرجع المصدق هذا لإنشاء شهادات هوية نقطة النهاية. وهذه الشهادات صالحة أيضا، على سبيل المثال، لإجراء مصادقة EAP محلية على وحدات تحكم 9800 وأحدث مع dot1x مصادقة RADIUS. هذا يساعدك على إنشاء شهادة نقطة نهاية القسم.

إنشاء مجلد لتخزين نتائج نقطة النهاية.

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

قم بتعديل قسم ملف [endpoint_alt_names>OpenSSL.cnf بحيث يتطابق الاسم الشائع الذي توفره على CSR مع شبكة منطقة التخزين (SAN).

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

قم بإنشاء مفتاح نقطة النهاية الخاص و WLC CSR باستخدام نقطة نهاية القسم_req_ext لشبكات منطقة التخزين (SANs).

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

توقيع شهادة جهاز نقطة النهاية.

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

إستيراد الشهادة إلى جهاز Cisco IOS XE

قم بإنشاء ملف يحتوي على CA الجذر والنسخة الوسيطة على الملف نفسه واحفظه إلى المجلد الذي يحمل الاسم certfile.crt كما هو مطلوب ./IntermCA/IntermCA.db.certs/WLC/ للاستيراد إلى جهاز Cisco IOS XE.

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

تستخدم وحدة التحكم في الشبكة المحلية اللاسلكية (WLC) من السلسلة 9800 أوامر مختلفة لإنشاء ملف PFX الشهادة. لإنشاء ملف PFX الخاص بك، قم بتشغيل أحد هذه الأوامر وفقا لإصدار Cisco IOS XE.

راجع إنشاء وتنزيل شهادات CSR على Catalyst 9800 WLCs للحصول على معلومات تفصيلية حول عملية إستيراد الشهادة

للإصدارات الأقدم من 17.12.1:

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

للإصدار 17.12.1 وأ إصدار أحدث:

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

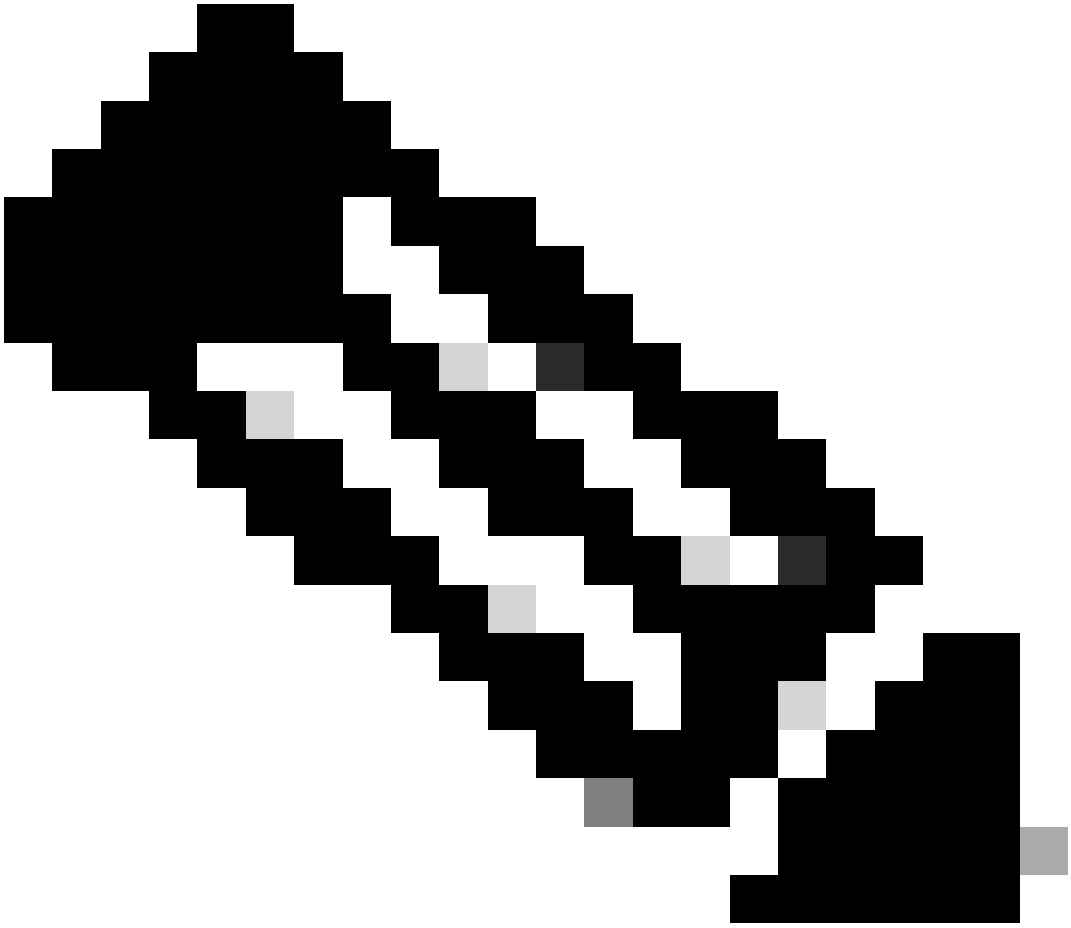
إستيراد شهادة IOSDevice.pfx إلى جهاز Cisco IOS XE:

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


      pkcs12 [tftp://


       /


        | ftp://


         /


          | http://
```

/

| bootflash:

] password



ملاحظة: تأكد من أن شهادات CA التي تم إنشاؤها لهذا الدليل موثوق بها من قبل الأجهزة التي تحتاج إلى التحقق من شهادة الجهاز. على سبيل المثال، إذا تم استخدام شهادة الجهاز لأغراض مسؤول الويب على جهاز Cisco IOS XE، فإن أي كمبيوتر أو مستعرض يصل إلى مدخل المسؤول يحتاج إلى وجود شهادات CA على مخزن الثقة الخاص به.

قم بتعطيل التحقق من الإبطال للشهادات نظرا لعدم وجود قائمة إبطال للشهادات على الإنترنت يمكن لجهاز Cisco IOS XE فحصها من المرجع المصدق الذي قمت بنشره.
يجب تعطيلها على كافة نقاط الثقة التي تعد جزءا من مسار التحقق. يحتوي Root CA -rrr1 المرفقة في النهاية مع السلسلة Intermediate/Device TrustPoint على نفس اسم TrustPoint.

```
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
```

التحقق من الصحة

التحقق من معلومات الشهادة على OpenSSL

للتحقق من معلومات الشهادة للشهادات التي تم إنشاؤها، على محطة ليونكس الطرفية قم بتشغيل الأمر:

```
openssl x509 -in

      -text -noout
```

إنه يعرض معلومات الشهادة كاملة.

```
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:
```

معلومات شهادة جهاز Cisco IOS XE كما هو موضح باستخدام OpenSSL

تحقق من معلومات الشهادة على جهاز Cisco IOS XE.

يطبع الأمرshow crypto pki certificates verboseمعلومات الترخيص من كل التراخيص المتاحة على الجهاز.

```
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
```

استكشاف الأخطاء وإصلاحها

التحقق من الإبطال موجود في المكان

عند إستيراد الشهادات إلى Cisco IOS XE، تم تمكين فحص إبطال نقاط الثقة التي تم إنشاؤها حديثا. إذا تم تقديم شهادة إلى الجهاز الذي يحتاج إلى إستخدام نقاط الثقة للشهادات المستوردة للتحقق منها، يبحث الجهاز عن قائمة إبطال شهادات غير موجودة ويفشل. تتم طباعة الرسالة على الوحدة الطرفية.

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

تأكد من أن كل نقطة ثقة في مسار التحقق من الشهادات تحتوي على الأمرrevocation-check none.

معلومات ذات صلة

- [إنشاء وتنزيل شهادات CSR على Catalyst 9800 WLCs]
- [تكوين الشهادات الموقعة من CA باستخدام IOS XE PKI]

- دليل تكوين الأمان والشبكة الخاصة الظاهرية (VPN)، Cisco IOS XE 17.x
- فهم معلومات الترخيص لإنشاء سلسلة لـ 9800 WLC

حول هذه الترجمة
ترجمت Cisco هذا المستند باستخدام مجموعة من التقنيات الآلية والبشرية لتقديم محتوى دعم للمستخدمين في جميع أنحاء العالم بلغتهم الخاصة. يُرجى ملاحظة أن أفضل ترجمة آلية لن تكون دقيقة كما هو الحال مع الترجمة الاحترافية التي يقدمها مترجم محترف. تخلي Cisco Systems مسئوليتها عن دقة هذه الترجمات وتُوصي بالرجوع دائمًا إلى المستند الإنجليزي الأصلي (الرابط متوفر).