在OpenSSL上設定多階層CA以產生IOS XE憑證

目錄

簡介

必要條件

- 需求
- 採用元件

設定

- 概觀
- 準備OpenSSL組態檔
- 為證書頒發機構建立初始檔案
- 建立根CA證書
- 建立中繼CA憑證
- 建立裝置證書
  - 建立Cisco IOS XE裝置證書
  - 可選-建立端點證書

將證書導入到Cisco IOS XE裝置

驗證

- 驗證OpenSSL上的憑證資訊

疑難排解

- 撤銷檢查已就緒

相關資訊

簡介

本文描述建立多級CA以建立與Cisco IOS® XE裝置相容的一般用途證書的方法。

必要條件

需求

思科建議您瞭解以下主題：

- 如何使用 OpenSSL 應用程式.
- 公開金鑰基礎架構(PKI)和數位憑證。

採用元件

本文中的資訊係根據以下軟體和硬體版本：

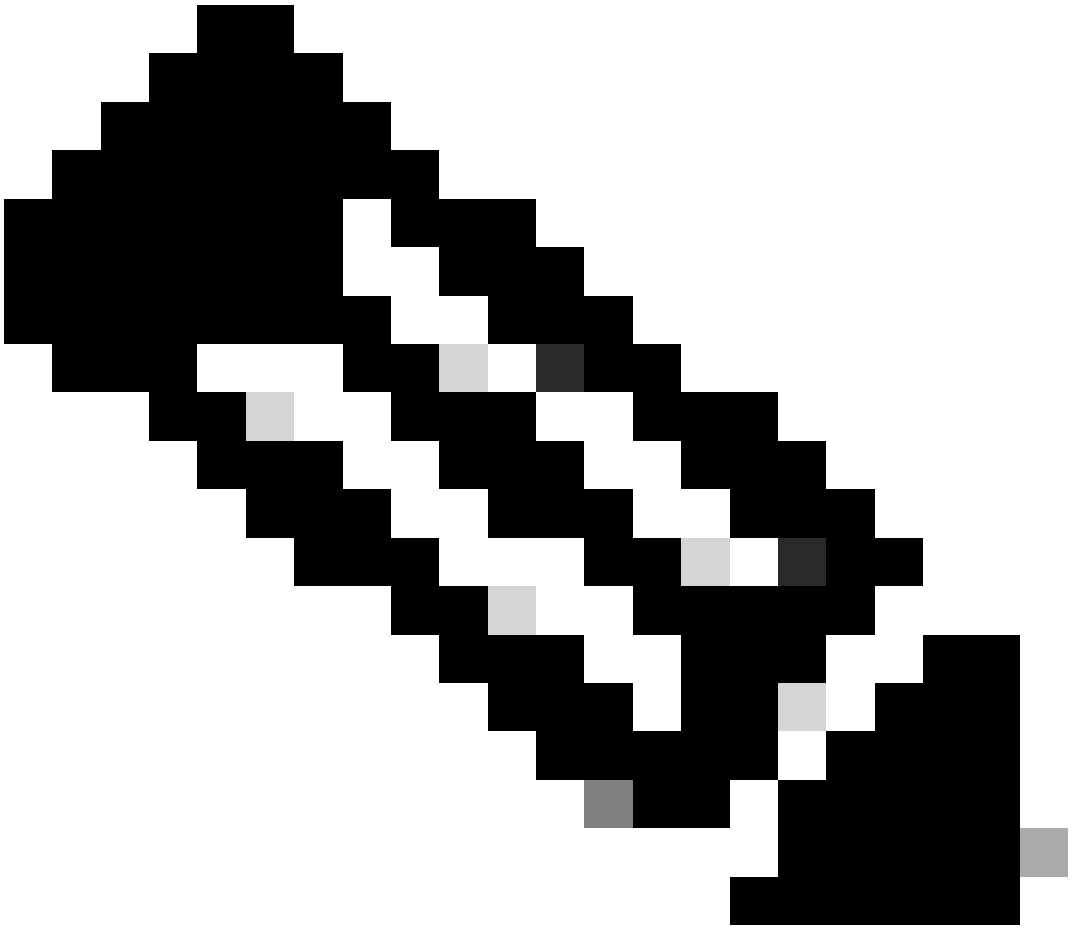
- OpenSSL應用程式（版本3.0.2）。
- 9800 WLC（Cisco IOS XE版本17.12.3）。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

設定

概觀

其目的是建立具有根CA和中間CA的二級本地證書頒發機構(CA)來簽署裝置證書。一旦簽署憑證，便會將其匯入到Cisco IOS XE裝置。



> 注意：本文檔使用Linux特定命令來建立和排列檔案。這些指令會詳細說明，如此您便可在其他可使用OpenSSL的作業系統上執行相同的動作。

準備OpenSSL組態檔

在已安裝OpenSSL的機器上，從目前的工作目錄建立名為openssl.conf的文字檔。複製並貼上這些

行，以向OpenSSL提供憑證簽署所需的組態。您可以根據需要編輯此檔案。

```
[ 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
```

為證書頒發機構建立初始檔案

在當前目錄中建立名為RootCA的資料夾。在其中，再建立3個資料夾，分別名為RootCA.db.tmp、RootCA.db.certs和RootCA.db.crl。

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

在RootCA資料夾中建立名為RootCA.db.serial的檔案。此檔案需要包含證書序列號的初始值，01為此案例選擇的值。

在RootCA資料夾中建立名為RootCA.db.crlserial的檔案。此檔案需要包含證書撤銷清單編號的初始值，01是此案例中選定的值。

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

在RootCA資料夾中建立名為RootCA.db.index的檔案。

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

在RootCA資料夾中建立名為RootCA.db.rand的檔案，並用8192隨機位元組填充該檔案以作為內部隨機數生成器的種子。

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

在當前目錄中建立名為IntermCA的資料夾。在其中，再建立3個資料夾，分別名為IntermCA.db.tmp、IntermCA.db.certs和IntermCA.db.crl。

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

在IntermCA資料夾中建立名為IntermCA.db.serial的檔案。此檔案需要包含證書序列號的初始值，01為此案例選擇的值。

在IntermCA資料夾中建立名為IntermCA.db.crlserial的檔案。此檔案需要包含證書撤銷清單編號的初始值，01是此案例中選定的值。

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

在IntermCA資料夾中建立名為IntermCA.db.index的檔案。

在IntermCA資料夾中建立名為IntermCA.db.rand的檔案，並用8192隨機位元組填充該檔案，以作為內部隨機數生成器的種子。

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

在IntermCA資料夾中建立名為IntermCA.db.rand的檔案，並用8192隨機位元組填充該檔案，以作為內部隨機數生成器的種子。

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

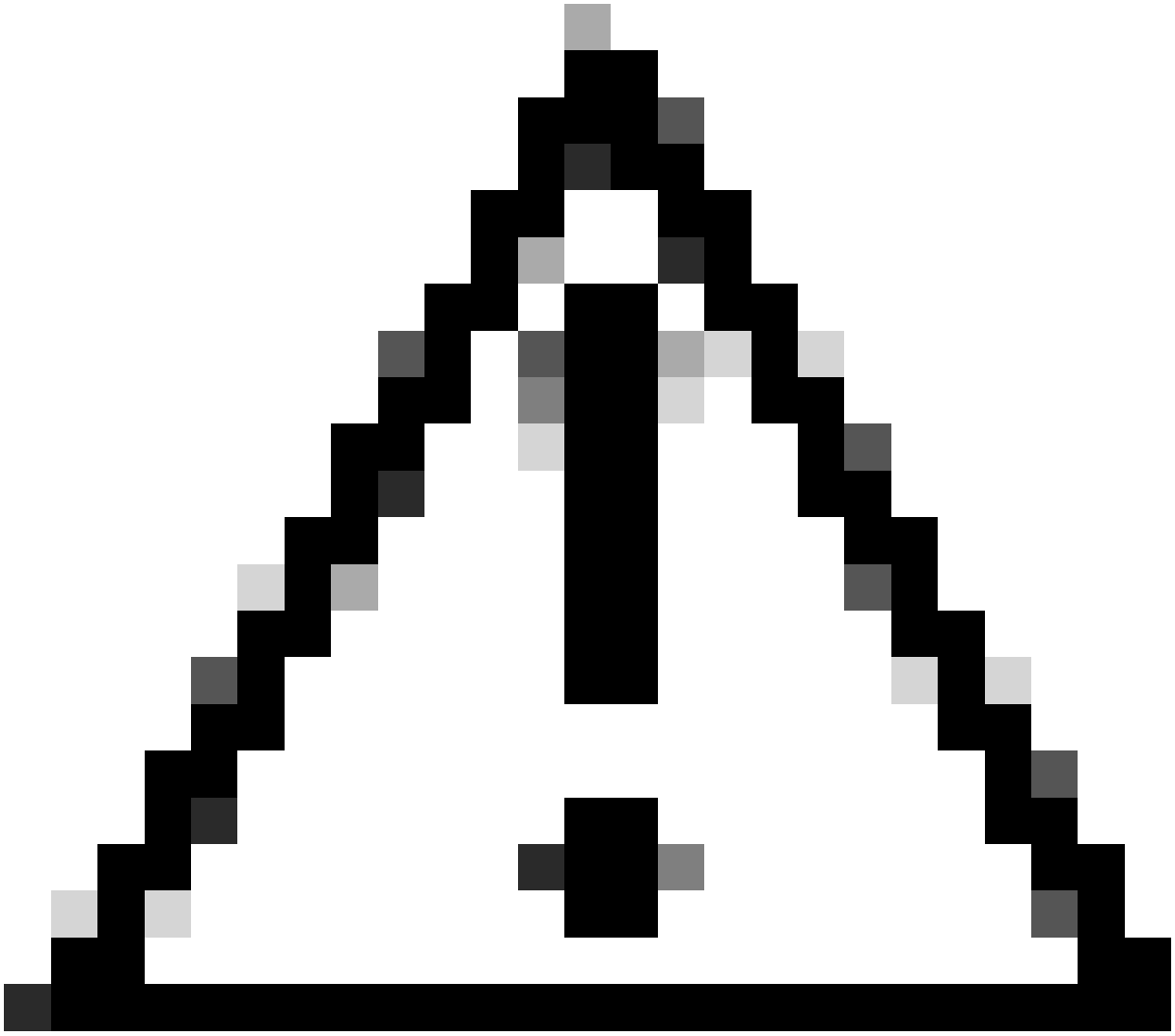
這是建立所有初始根和中間CA檔案之後的檔案結構。

```
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
```

建立根CA證書

運行此命令可為根CA建立私鑰。

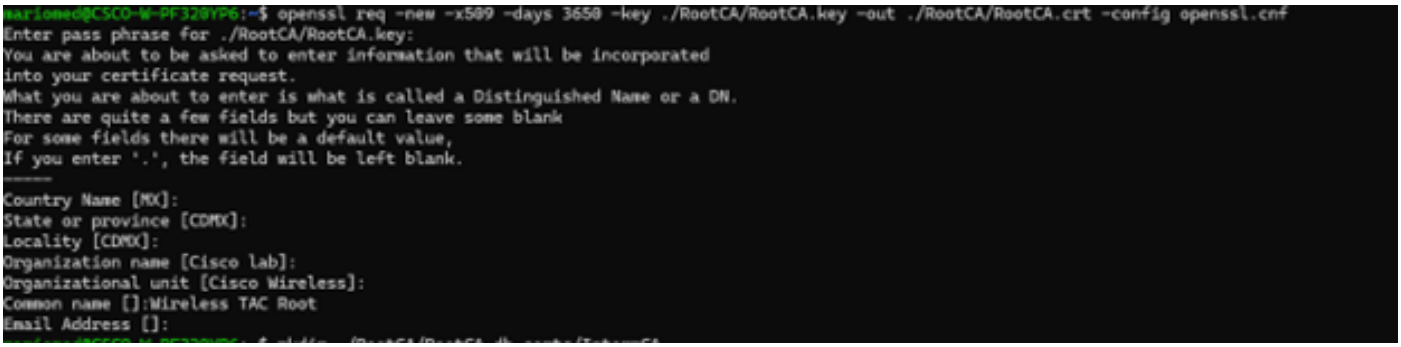
```
openssl genrsa -des3 -out ./RootCA/RootCA.key 4096
```



注意：生成金鑰時，OpenSSL需要您提供口令。將密碼和生成的私鑰儲存在安全位置。任何有權存取此軟體的人都可以將憑證作為您的根CA簽發。

在openSSL上使用req命令建立根CA自簽名證書。-x509標誌在內部建立證書簽名請求(CSR)並自動對其進行自簽名。編輯-days引數和主題備用名稱。會提示您提供一般名稱。確保您輸入的公用名與主題備用名(SAN)匹配。

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



OpenSSL辨別名稱互動提示

生成的檔名為RootCA.crt，位於RootCA資料夾中。此檔案是根CA證書。

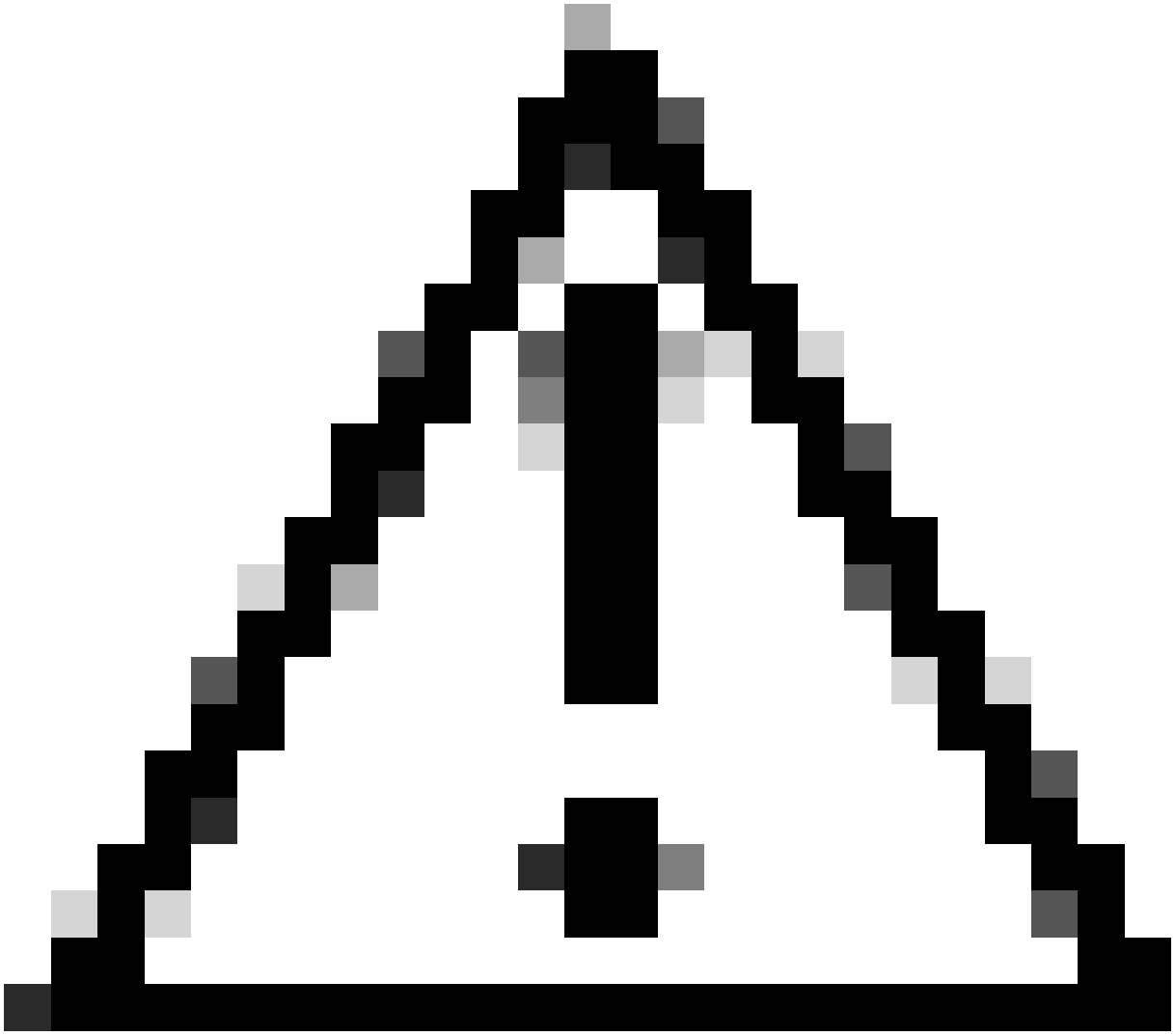
建立中繼CA憑證

建立資料夾，將簽署的中間CA憑證儲存在根資料夾中。

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

為中間證書建立私鑰。

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



注意：生成金鑰時，OpenSSL需要您提供口令。將密碼和生成的私鑰儲存在安全位置。任何具有存取權的人均可發出憑證作為您的中繼CA。

建立中間CA證書簽名請求。終端會提示您輸入憑證資訊。

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

使用openssl.cnf檔案的RootCA部分簽署中間CSR。

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

產生的檔案稱為IntermCA.crt，位於RootCA資料夾內。此檔案是根CA證書。

將中間證書和金鑰移動到您建立作為中間CA初始檔案一部分的其自己的資料夾中。

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

這是為初始根和中間CA建立私鑰和證書之後的檔案結構。

```
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
```

建立裝置證書

建立Cisco IOS XE裝置證書

建立一個新資料夾以儲存Cisco IOS XE裝置證書。

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

建立裝置私鑰IOSdevice.key和裝置CSR IOSdevice.csr。使用device_req_ext部分可將上述部分下的SAN增加到CSR中。

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

修改opnessl.cnf檔案[IOS_alt_names]部分，以便在CSR上提供的公用名與SAN匹配。

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

使用中間CA IntermCA部分簽署IOS XE裝置CSR。使用-config指向openSSL配置檔案，使用-extensions指向IOS_cert部分。這樣會將SAN保留在簽署的憑證上。

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

完成此步驟後，您為名為IOSdevice.crt的IOS XE裝置建立了具有匹配私鑰IOSdevice.key的有效證書。

可選-建立端點證書

此時，您已部署本地CA，並為您的IOS XE裝置頒發了一個證書。您也可以使用此CA生成終端身份證書。這些憑證也有效，例如，在9800無線LAN控制器上執行本機EAP驗證，或甚至是使用RADIUS伺服器進行dot1x驗證。此部分可幫助您生成終端證書。

建立用於儲存終端證書的資料夾。

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

修改openSSL.cnf檔案[endpoint_alt_names]部分，以便在CSR上提供的公用名與SAN匹配。

```
#Define here SANS/IPs needed for  Endpoint certificates
[endpoint_alt_names]
```

```
DNS.1   = Endpoint.example.com
DNS.2   = Endpoint2.example.com
```

使用用於SAN的endpoint_req_ext部分建立終端私鑰和WLC CSR。

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

簽署終端裝置證書。

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

將證書導入到Cisco IOS XE裝置

根據導入到Cisco IOS XE裝置所需的資訊，在同一檔案中建立包含根CA和中間CA的檔案，並將其儲存到名為certfile.crt的./IntermCA/IntermCA.db.certs/WLC/資料夾中。

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

9800系列WLC使用不同的命令來建立用於憑證匯入的pfx檔案。要建立pfx檔案，請根據Cisco IOS XE版本運行以下命令之一。

有關證書導入過程的詳細資訊，請參閱在Catalyst 9800 WLC上生成和下載CSR證書

對於早於17.12.1的版本：

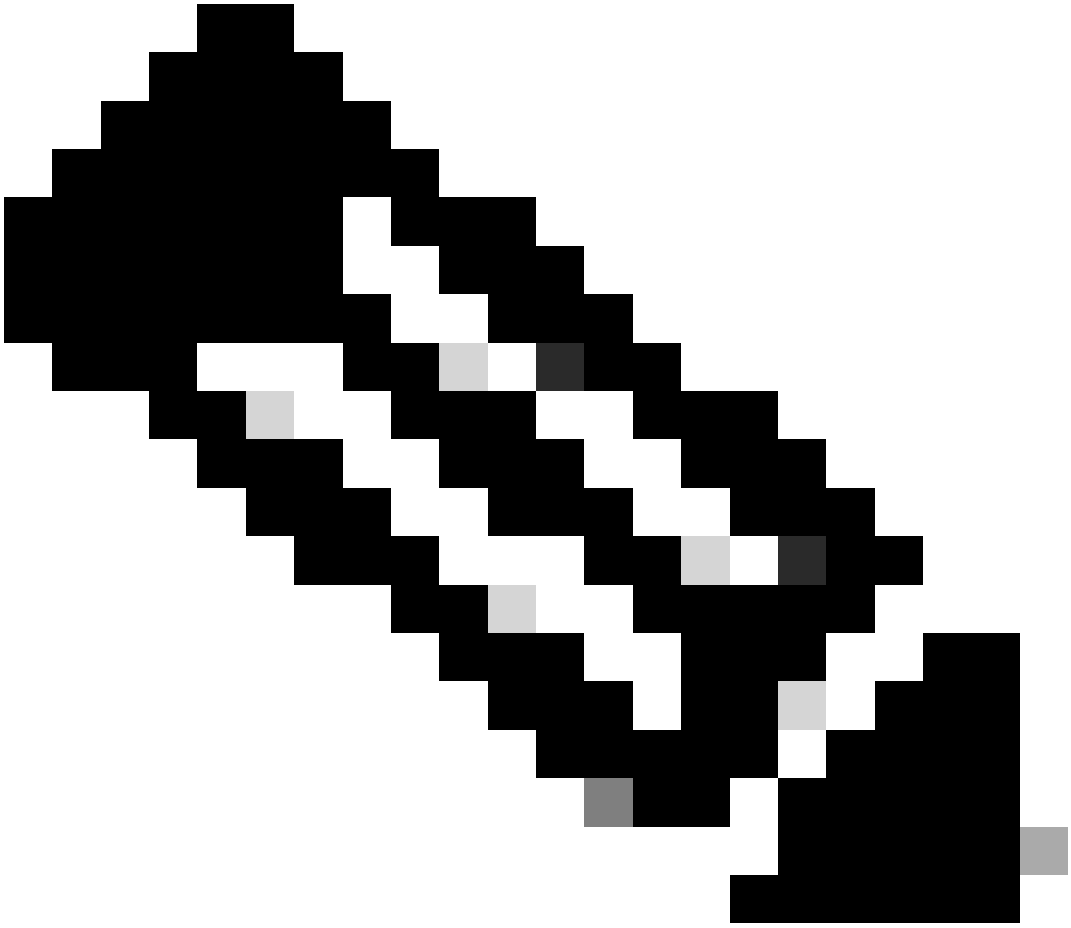
```
openssl pkcs12 -export -macalg sha1 -legacy -descert -out ./IntermCA/IntermCA.db.certs/IOSdevice/IOSdev
```

對於版本17.12.1或更高版本：

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

將IOSdevice.pfx證書導入到Cisco IOS XE裝置：

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




      pkcs12 [tftp://




       /




        | ftp://




         /




          | http://




           /




            | bootflash:




             ] password
```



注意：請確保需要驗證裝置證書的裝置信任為此指南建立的CA證書。例如，如果裝置證書用於Cisco IOS XE裝置上的Web管理目的，則訪問管理門戶的任何電腦或瀏覽器都需要在信任庫上擁有CA證書。

停用證書的吊銷檢查，因為Cisco IOS XE裝置可以從已部署的CA檢查沒有聯機證書吊銷清單。您必須在驗證路徑中的所有信任點上停用它。根CA信任點與中間/裝置信任點具有相同名稱，字串-rrr1附加在末尾。

```
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
```

驗證

驗證OpenSSL上的憑證資訊

要驗證已建立證書的證書資訊，請在Linux終端上運行命令：

```
openssl x509 -in

      -text -noout
```

它會顯示完整的憑證資訊。

```
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:
```

Cisco IOS XE裝置證書資訊，如OpenSSL所示

驗證Cisco IOS XE裝置上的證書資訊。

命令show crypto pki certificates verbose用於列印裝置上所有可用證書的證書資訊。

```
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
```

疑難排解

撤銷檢查已就緒

當證書導入到Cisco IOS XE時，新建立的信任點啟用了撤銷檢查。如果向需要使用導入的證書信任點進行驗證的裝置提供證書，則裝置會搜尋不存在的證書撤銷清單並失敗。消息被列印在終端上。

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

確保證書驗證路徑中的每個信任點都包含命令revocation-check none。

相關資訊

- 在Catalyst 9800 WLC上產生和下載CSR憑證
- 使用IOS XE PKI配置CA簽名證書
- 安全和VPN配置指南，Cisco IOS XE 17.x
- 瞭解憑證資訊，以為9800 WLC建立鏈結

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。