Configurazione di una CA multilivello su OpenSSL per generare certificati IOS XE

Sommario

Introduzione

Questo documento descrive un metodo per creare una CA multilivello per creare certificati per scopi generali compatibili con i dispositivi Cisco IOS® XE.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Come utilizzare l'applicazione OpenSSL.
- Infrastruttura a chiave pubblica (PKI) e certificati digitali.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e

hardware:

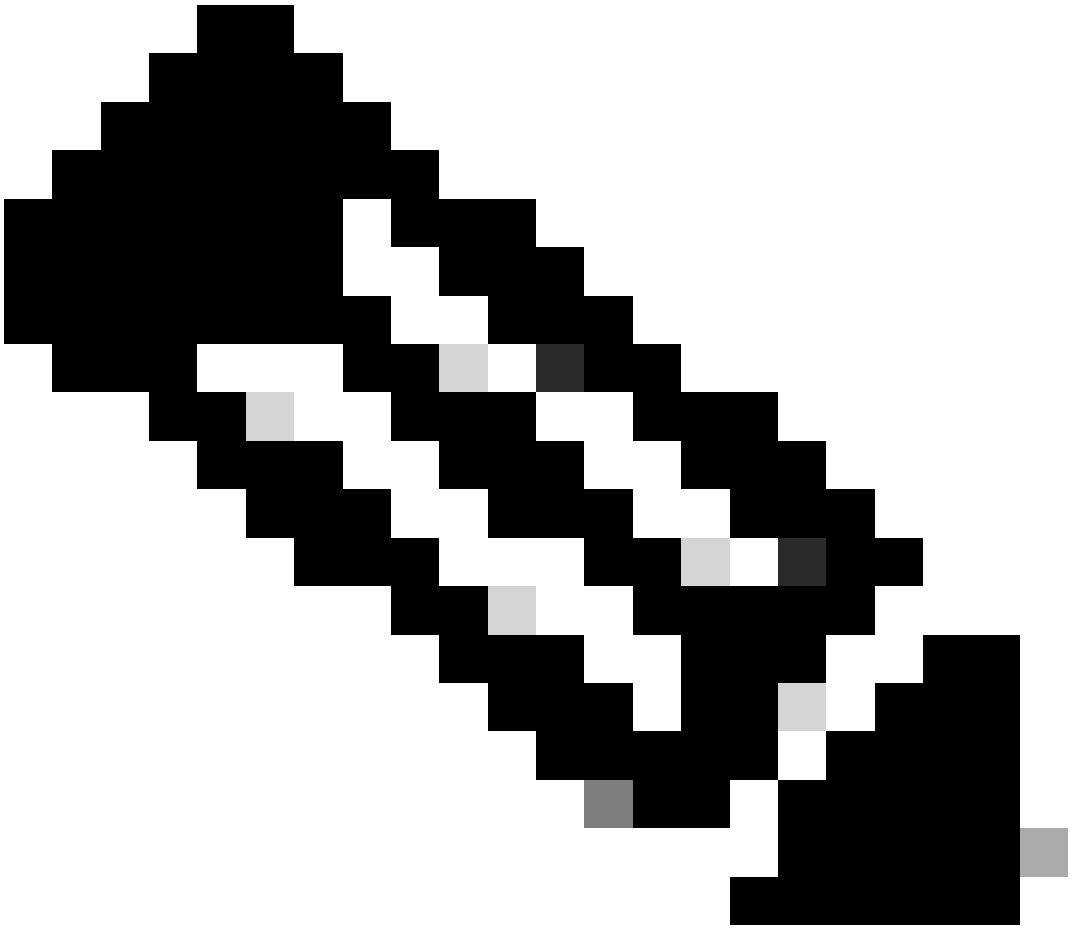
- Applicazione OpenSSL (versione 3.0.2).
- 9800 WLC (Cisco IOS XE versione 17.12.3).

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Configurazione

Panoramica

Lo scopo è creare un'Autorità di certificazione (CA) locale a due livelli con una CA radice e una CA intermedia per firmare i certificati dei dispositivi. Dopo aver firmato i certificati, questi vengono importati nel dispositivo Cisco IOS XE.



Nota: questo documento utilizza comandi specifici di Linux per creare e disporre i file. I comandi vengono spiegati in modo da poter eseguire la stessa azione su altri sistemi operativi in cui è disponibile OpenSSL.

Preparare il file di configurazione OpenSSL

Creare un file di testo denominato openssl.conf dalla directory di lavoro corrente sul computer in cui è installato OpenSSL. Copiare e incollare queste righe per fornire a OpenSSL le configurazioni necessarie per la firma dei certificati. È possibile modificare questo file in base alle proprie esigenze.

```
[ 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
```

Crea file iniziali per le Autorità di certificazione

Creare una cartella denominata RootCA nella directory corrente. All'interno, creare altre 3 cartelle denominate RootCA.db.tmp, RootCA.db.certs e RootCA.db.crl.

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

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

Creare un file denominato RootCA.db.serial nella cartella RootCA. Questo file deve contenere il valore iniziale per il numero di serie dei certificati, 01 è il valore selezionato in questo caso.

Creare un file denominato RootCA.db.crlserial nella cartella RootCA. Il file deve contenere il valore iniziale per il numero dell'elenco di revoche di certificati, 01 è il valore selezionato in questo caso.

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

Creare un file denominato RootCA.db.index nella cartella RootCA.

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

Creare un file denominato RootCA.db.rand all'interno della cartella RootCA e popolarlo con 8192 byte casuali da utilizzare come valore di inizializzazione del generatore di numeri casuali interno.

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

Creare una cartella denominata IntermCA nella directory corrente. All'interno, creare altre 3 cartelle denominate IntermCA.db.tmp, IntermCA.db.certs e IntermCA.db.crl.

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

Creare un file denominato IntermCA.db.serial nella cartella IntermCA. Questo file deve contenere il valore iniziale per il numero di serie dei certificati, 01 è il valore selezionato in questo caso.

Creare un file denominato IntermCA.db.crlserial nella cartella IntermCA. Il file deve contenere il valore iniziale per il numero dell'elenco di revoche di certificati, 01 è il valore selezionato in questo caso.

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

Creare un file denominato IntermCA.db.index nella cartella IntermCA.

Creare un file denominato IntermCA.db.rand all'interno della cartella IntermCA e popolarlo con 8192 byte casuali da utilizzare come base del generatore di numeri casuali interno.

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

Creare un file denominato IntermCA.db.rand all'interno della cartella IntermCA e popolarlo con 8192 byte casuali da utilizzare come base del generatore di numeri casuali interno.

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

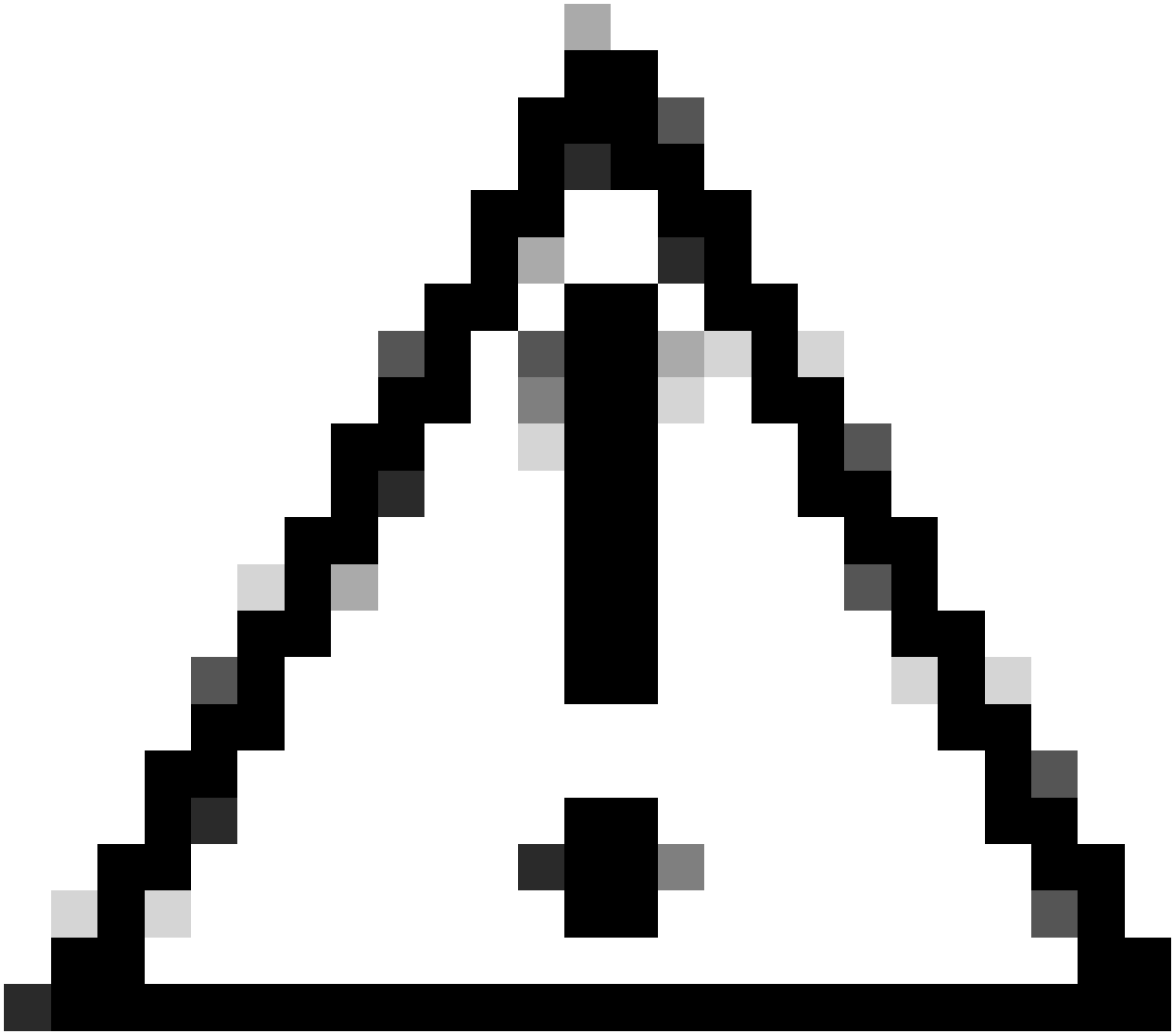
Questa è la struttura di file dopo la creazione di tutti i file CA radice e intermedi iniziali.

```
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
```

Crea certificato CA radice

Eseguire questo comando per creare la chiave privata per la CA radice.

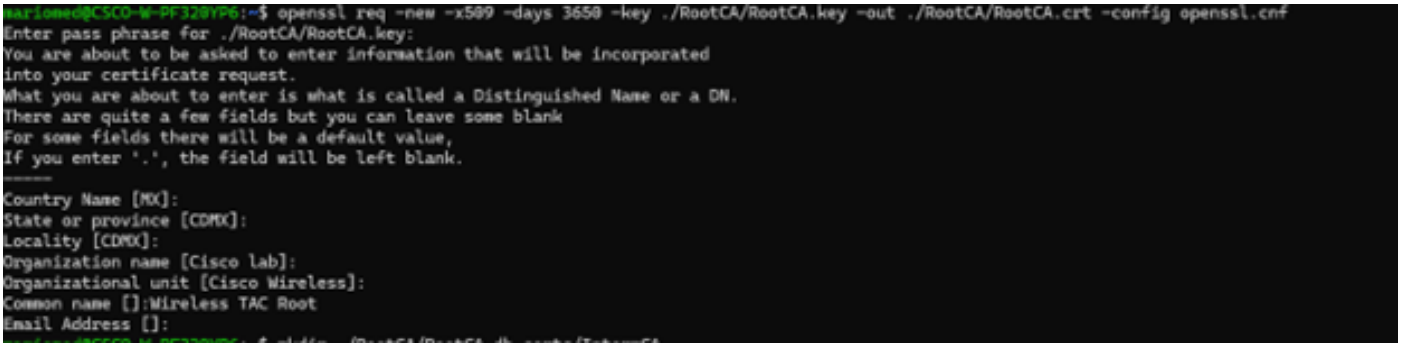
```
openssl genrsa -des3 -out ./RootCA/RootCA.key 4096
```



Attenzione: OpenSSL richiede di fornire una passphrase quando viene generata una chiave. Mantenere la passphrase segreta e la chiave privata generata in un percorso sicuro. Chiunque disponga dell'accesso può emettere certificati come CA radice.

Creare il certificato autofirmato della CA radice utilizzando il comando`req`su openSSL. Il `-x509` flag crea internamente una richiesta di firma del certificato (CSR) e la firma automaticamente. Modificare il nome alternativo del`-days`parametro e del soggetto. Viene richiesto di specificare un nome comune. Assicurarsi che il nome comune immesso corrisponda al nome alternativo del soggetto (SAN).

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



Prompt interattivo nome distinto OpenSSL

Il file generato è denominato RootCA.crt e si trova all'interno della cartella RootCA. Questo file è il certificato CA radice.

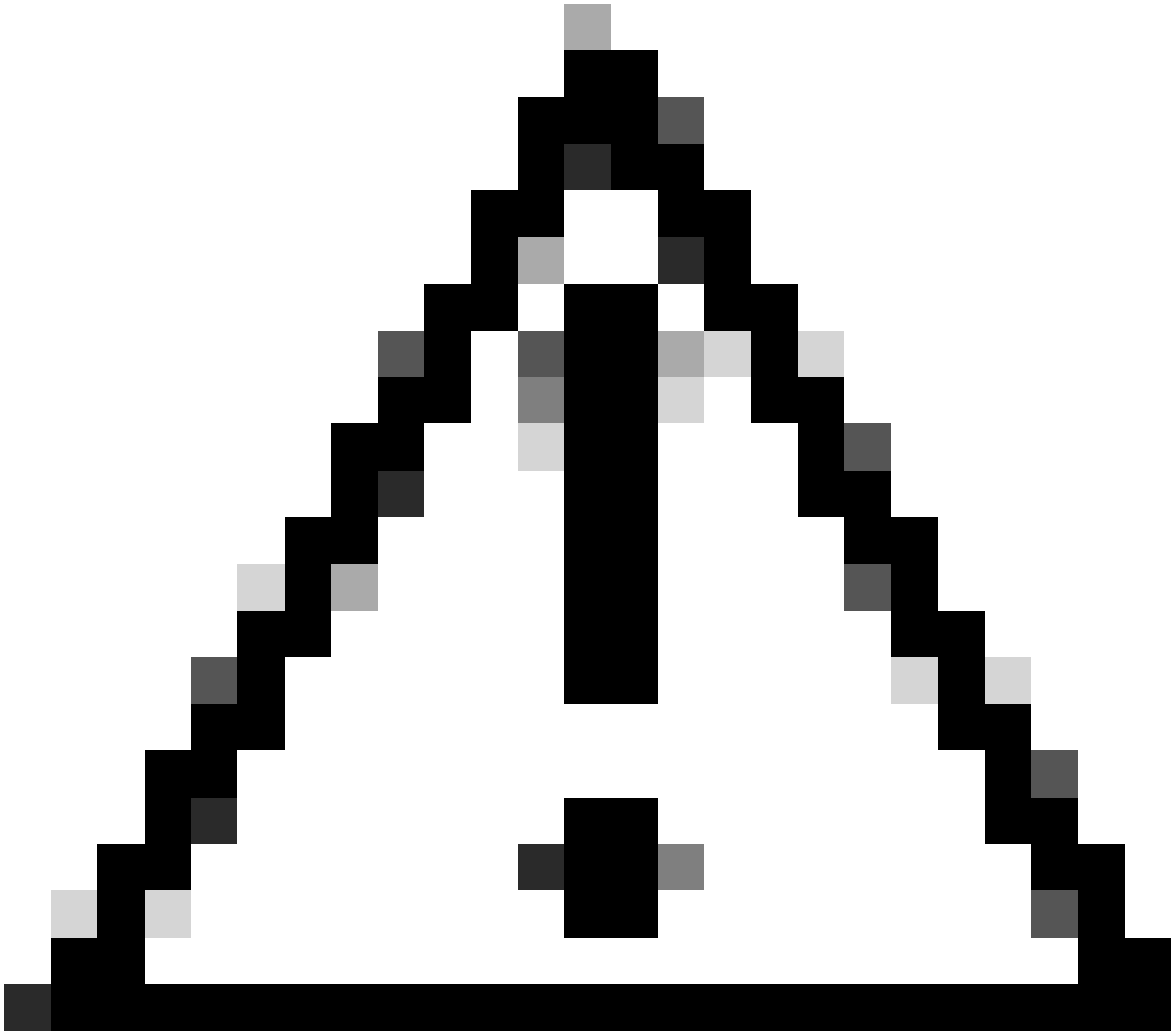
Crea certificato CA intermedio

Creare una cartella in cui archiviare il certificato CA intermedio firmato all'interno della cartella radice.

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

Crea la chiave privata per il certificato intermedio.

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



Attenzione: OpenSSL richiede di fornire una passphrase quando viene generata una chiave. Mantenere la passphrase segreta e la chiave privata generata in un percorso sicuro. Chiunque disponga dell'accesso può emettere certificati come CA intermedia.

Creare una richiesta intermedia di firma del certificato CA. Il terminale chiede di immettere le informazioni sul certificato.

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

Firmare CSR intermedio con la sezione RootCA del file openssl.cnf.

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

Il file generato è denominato IntermCA.crt e si trova all'interno della cartella RootCA. Questo file è il certificato CA radice.

Spostare il certificato e la chiave intermedi nella relativa cartella creata come parte dei file iniziali per la CA intermedia.

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

Questa è la struttura di file dopo la creazione della chiave privata e dei certificati per le CA radice e intermedie iniziali.

```
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
```

Crea certificati dispositivo

Crea certificato dispositivo Cisco IOS XE

Creare una nuova cartella in cui archiviare i certificati dei dispositivi Cisco IOS XE.

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

Creare la chiave privata del dispositivo IOSdevice.key e il dispositivo CSR IOSdevice.csr. Utilizzare la sezione device_req_ext per aggiungere le SAN in tale sezione al CSR.

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

Modificare la sezione del file opnessl.cnf [IOS_alt_names] in modo che il nome comune specificato nel CSR corrisponda alla SAN.

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

Firmare il CSR del dispositivo IOS XE con la sezione intermedia CA IntermCA. Utilizzare -config per puntare al file di configurazione openSSL e -extensions per puntare alla sezione IOS_cert. In questo modo la SAN rimane sul certificato firmato.

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

Dopo questo passaggio, è stato creato un certificato valido per il dispositivo IOS XE denominato IOSdevice.crt con la chiave privata IOSdevice.key corrispondente.

Facoltativo - Crea certificato endpoint

A questo punto, è stata distribuita una CA locale e rilasciato un certificato per il dispositivo IOS XE. È inoltre possibile utilizzare questa CA per generare certificati di identità per gli endpoint. Questi certificati sono validi, ad esempio, per eseguire l'autenticazione EAP locale sui controller LAN wireless 9800 o anche l'autenticazione dot1x con i server RADIUS. Questa sezione consente di generare un certificato per l'endpoint.

Creare una cartella in cui archiviare i certificati dell'endpoint.

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

Modificare la sezione openSSL.cnf file [endpoint_alt_names] in modo che il nome comune specificato nel CSR corrisponda alla SAN.

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

Creare la chiave privata dell'endpoint e il CSR WLC con la sezione endpoint_req_ext per le SAN.

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

Firmare il certificato del dispositivo endpoint.

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

Importa certificato nel dispositivo Cisco IOS XE

Creare un file contenente la CA radice e la CA intermedia nello stesso file e salvarlo nella cartella ./IntermCA/IntermCA.db.certs/WLC/certfile.crt come richiesto per l'importazione nel dispositivo Cisco IOS XE.

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

Il WLC serie 9800 utilizza diversi comandi per creare il file pfx per l'importazione del certificato. Per creare il file pfx, eseguire uno di questi comandi in base alla versione di Cisco IOS XE.

Per informazioni dettagliate sul processo di importazione dei certificati, consultare il documento sulla generazione e il download dei certificati CSR sui WLC di Catalyst 9800

Per le versioni precedenti alla 17.12.1:

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

Per la versione 17.12.1 o successive:

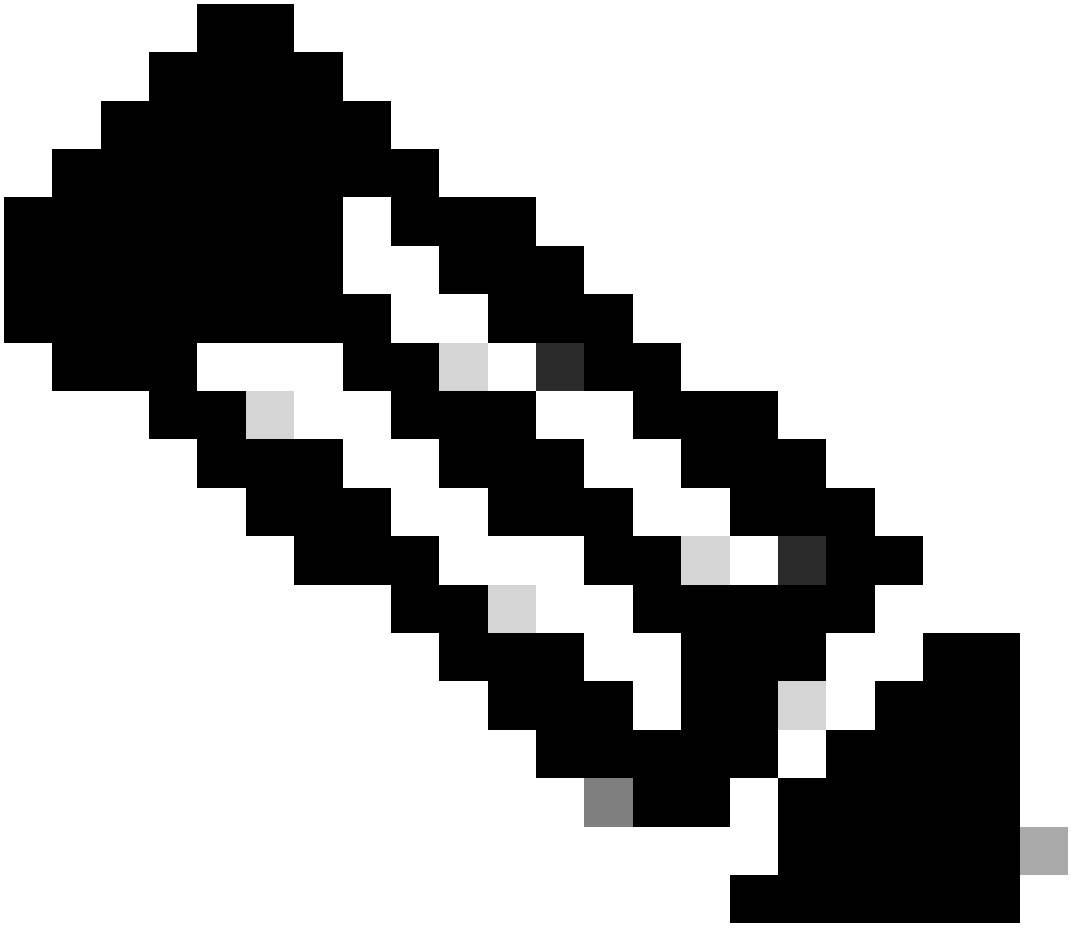
```
openssl pkcs12 -export -out ./IntermCA/IntermCA.db.certs/IOSdevice/IOSdevice.pfx -inkey ./IntermCA/Inte
```

Importare il certificato IOSdevice.pfx nel dispositivo Cisco IOS XE:

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


      pkcs12 [tftp://


       /


        | ftp://


         /


          | http://


           /
```

| bootflash:

] password



> Nota: verificare che i certificati CA creati per questa guida siano attendibili per i dispositivi che devono verificare il certificato del dispositivo. Ad esempio, se il certificato del dispositivo viene utilizzato per scopi di amministrazione Web sul dispositivo Cisco IOS XE, tutti i computer o i browser che accedono al portale di amministrazione devono disporre dei certificati CA nel relativo archivio attendibile.

Disabilitare il controllo di revoca per i certificati poiché non esiste un elenco di revoche di certificati online che il dispositivo Cisco IOS XE può verificare dalla CA distribuita.
È necessario disattivarla in tutti i trust point che fanno parte del percorso di verifica. Il trust point CA radice ha lo stesso nome del trust point intermedio/dispositivo con la stringa -rr1 aggiunta alla fine.

```
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
```

Verifica

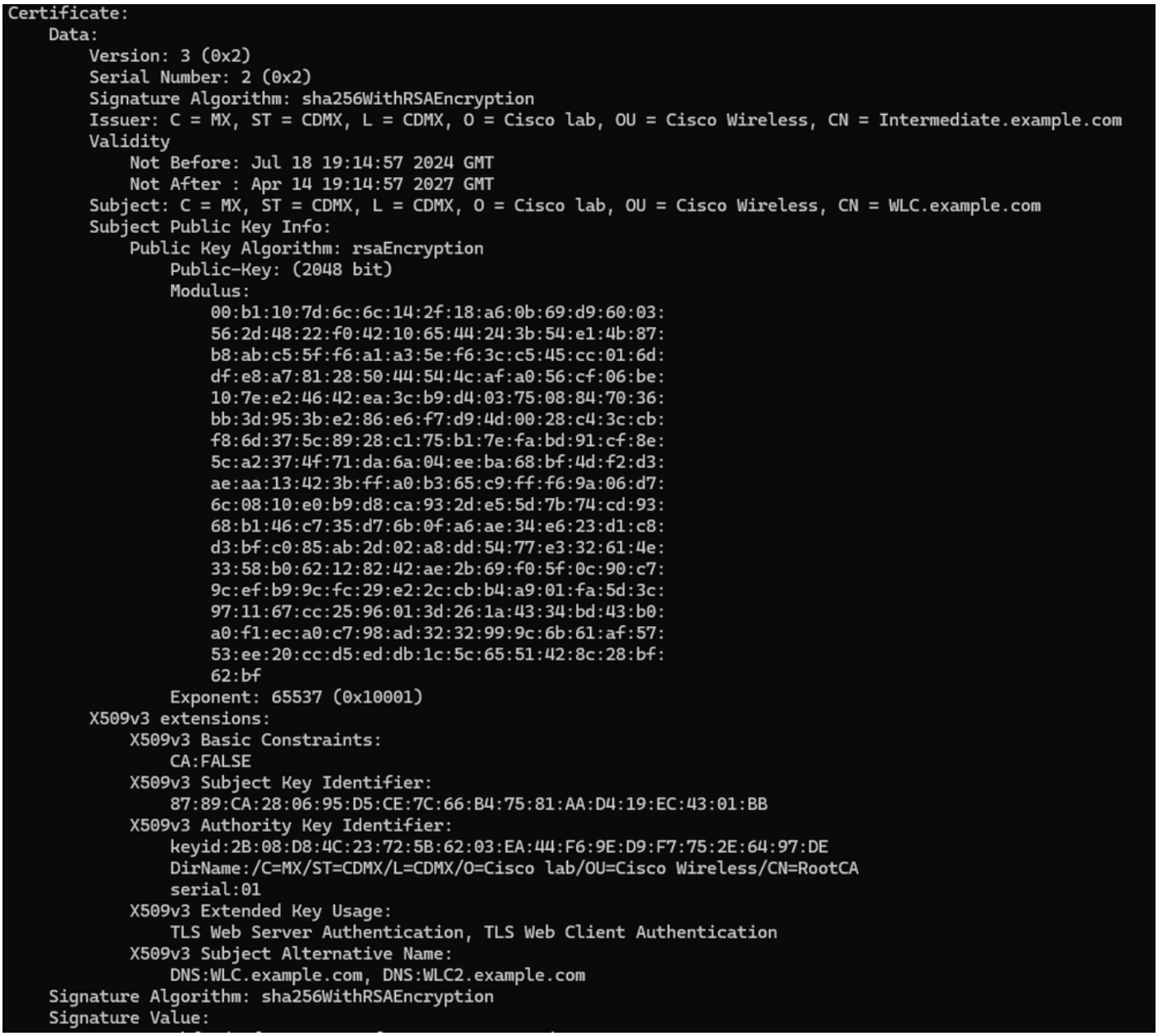
Verifica delle informazioni sul certificato in OpenSSL

Per verificare le informazioni relative ai certificati creati, sul terminale Linux eseguire il comando:

```
openssl x509 -in
```

```
      -text -noout
```

Visualizza le informazioni complete sul certificato.



Informazioni sul certificato del dispositivo Cisco IOS XE mostrate da OpenSSL

Verificare le informazioni sul certificato sul dispositivo Cisco IOS XE.

Il comando`show crypto pki certificates verbose`stampa le informazioni di tutti i certificati disponibili nel dispositivo.

```
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
```

Risoluzione dei problemi

Verifica revoca in corso

Quando i certificati vengono importati in Cisco IOS XE, per i trust appena creati è abilitata la verifica di revoca. Se un certificato viene presentato al dispositivo che deve utilizzare i punti di attendibilità dei certificati importati per la convalida, il dispositivo cerca un elenco di revoche di certificati inesistente e l'operazione non riesce. Il messaggio viene stampato sul terminale.

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

Verificare che ogni trust point nel percorso di verifica dei certificati contenga il comando`revocation-check none`.

Informazioni correlate

- Generazione e download dei certificati CSR sui WLC di Catalyst 9800
- Configurare i certificati firmati dalla CA con PKI IOS XE
- Guida alla sicurezza e alla configurazione VPN, Cisco IOS XE 17.x
- Informazioni sui certificati per creare una catena per 9800 WLC

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).