Straal DTLS op ISE en 9800 WLC configureren

Inhoud

Inleiding

Dit document beschrijft een methode om de benodigde certificaten te maken om RADIUS DTLS tussen ISE en de 9800 WLC te configureren.

Achtergrond

RADIUS DTLS is een beveiligde vorm van het RADIUS-protocol waarin de RADIUS-berichten worden verzonden via een DTLS-tunnel (Data Transport Layer Security). Om deze tunnel te maken tussen de verificatieserver en de verificator, is een set certificaten nodig. Deze set van certificaten vereist dat bepaalde Extended Key Usage (EKU) certificaatuitbreidingen worden ingesteld, met name clientverificatie op het WLC-certificaat en zowel serververificatie als clientverificatie voor het ISE-certificaat.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Hoe de 9800 WLC, het access point (AP) te configureren voor basisbediening
- De OpenSSL-toepassing gebruiken
- Public Key Infrastructure (PKI) en digitale certificaten

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

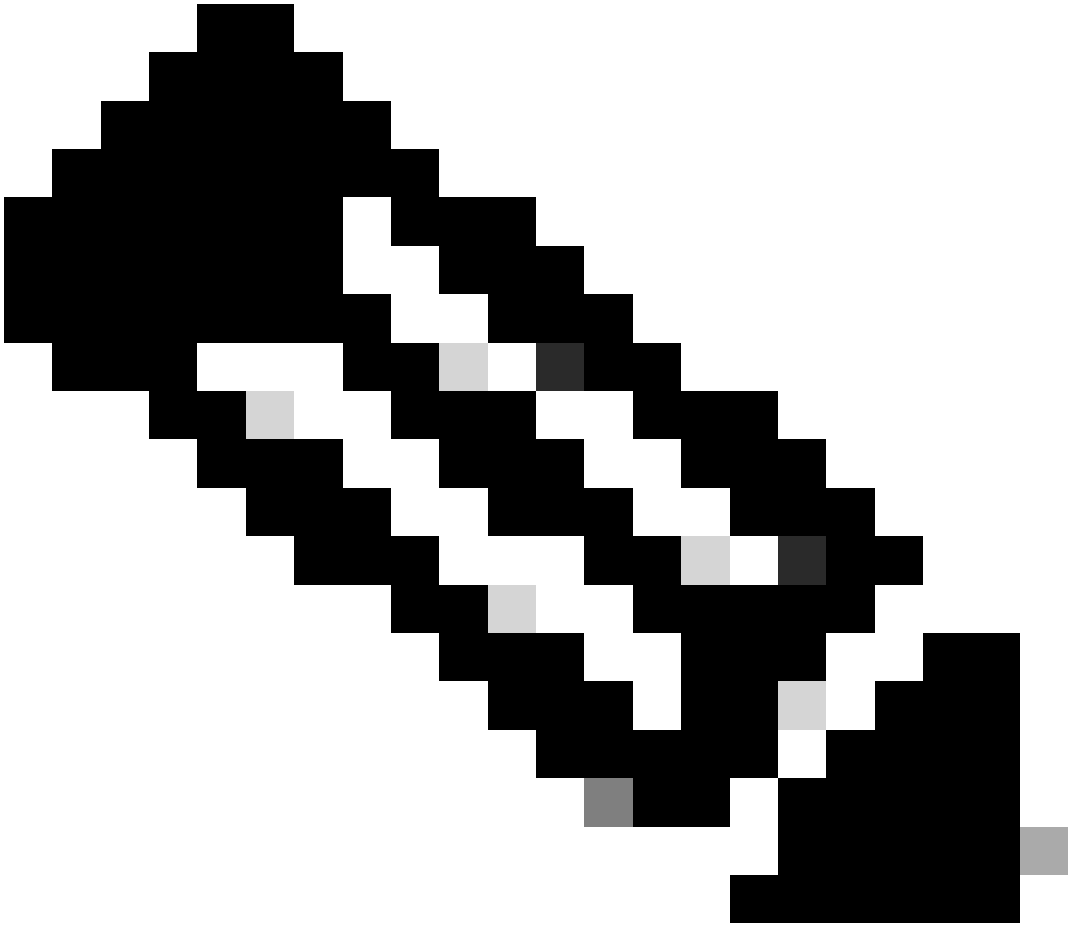
- OpenSSL-toepassing (versie 3.0.2).
- ISE (versie 3.1.0.518)
- 9800 WLC (versie 17.12.3)

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Configureren

Overzicht

Het doel is om een certificeringsinstantie op twee niveaus te creëren met een Root CA en een Intermediate CA om eindpuntcertificaten te ondertekenen. Zodra de certificaten zijn ondertekend, worden ze geïmporteerd in de WLC en ISE. Tot slot worden de apparaten geconfigureerd om RADIUS DTLS-verificatie met die certificaten uit te voeren.



> Opmerking: dit document gebruikt Linux-specifieke opdrachten om bestanden te maken en te rangschikken. De opdrachten worden uitgelegd zodat u dezelfde actie kunt uitvoeren op andere besturingssystemen waar OpenSSL beschikbaar is.

Optioneel - WLC en ISE RADIUS DTLS-apparaatcertificaat maken

Het RADIUS DTLS-protocol moet certificaten uitwisselen tussen ISE en WLC om de DTLS-tunnel te maken. Als u nog geen geldige certificaten hebt, kunt u een lokale CA maken om de certificaten te genereren, raadpleegt u Een certificeringsinstantie op meerdere niveaus configureren op OpenSSL om Cisco IOS® XE Compatible Certificates te genereren en voert u de stappen uit die vanaf het begin tot het einde van de stap op het document zijn beschreven Maak een tussentijds CA-certificaat aan.

Configuratiesecties toevoegen aan openssl.cnf-bestand

Open uw configuratiebestand openssl.cnf en kopieer en plak onderaan de WLC- en ISE-secties

die worden gebruikt om een geldige certificaataanvraag te genereren.

Zowel ISE_device_req_ext als WLC_device_req_ext secties wijzen elk op een lijst van SAN's die in de MVO moeten worden opgenomen:

```
#Section used for CSR generation, it points to the list of subject alternative names to add them to CSR
[ ISE_device_req_ext ]
subjectAltName = @ISE_alt_names

[ WLC_device_req_ext ]
subjectAltName = @WLC_alt_names

#DEFINE HERE SANS/IPs NEEDED for **ISE** device certificates
[ISE_alt_names]
DNS.1   = ISE.example.com
DNS.2   = ISE2.example.com

#DEFINE HERE SANS/IPs NEEDED for **WLC** device certificates
[WLC_alt_names]
DNS.1   = WLC.example.com
DNS.2   = WLC2.example.com
```

Als veiligheidsmaatregel, treedt CA alle SAN's op een CSR af om het te ondertekenen zodat onbevoegde apparaten geen geldig certificaat kunnen ontvangen voor een naam die ze niet mogen gebruiken. Als u de SAN's weer aan het ondertekende certificaat wilt toevoegen, gebruikt u de parameter subjectAltName om naar dezelfde lijst SAN's te wijzen als de SAN's die worden gebruikt voor de productie van MVO.

ISE vereist zowel serverAuth als clientAuth EKUs aanwezig op het certificaat terwijl de WLC alleen clientAuth nodig heeft. Ze worden toegevoegd aan het ondertekende certificaat met de extendedKeyUsage parameter.

Kopieer en plak de secties die gebruikt worden voor het certificaatteken onderaan het bestand openssl.cnf:

```
#This section contains the extensions used for the device certificate sign
[ ISE_cert ]
basicConstraints=CA:FALSE
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always
#EKU client and server is needed for RADIUS DTLS on ISE
extendedKeyUsage = serverAuth, clientAuth
subjectAltName = @ISE_alt_names

[ WLC_cert ]
basicConstraints=CA:FALSE
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always
#EKU client is needed for RADIUS DTLS on WLC
extendedKeyUsage = clientAuth
```

```
subjectAltName = @WLC_alt_names
```

WLC-apparaatcertificaat maken

Maak nieuwe map om WLC certs op te slaan op de machine die OpenSSL heeft geïnstalleerd in de tussenliggende CA cert map genaamd IntermCA.db.certs. De nieuwe map wordt WLC genoemd:

```
mkdir ./IntermCA/IntermCA.db.certs/WLC
```

Wijzig de DNS-parameters in het [WLC_alt_names] gedeelte van het bestand openssl.cnf. Verander de voorbeeldnamen die voor de gewenste waarden zijn opgegeven. Deze waarden vullen het SAN-veld van het WLC-certificaat in:

```
[WLC_alt_names]
DNS.1   = WLC.example.com    <-----Change the values after the equals sign
DNS.2   = WLC2.example.com   <-----Change the values after the equals sign
```

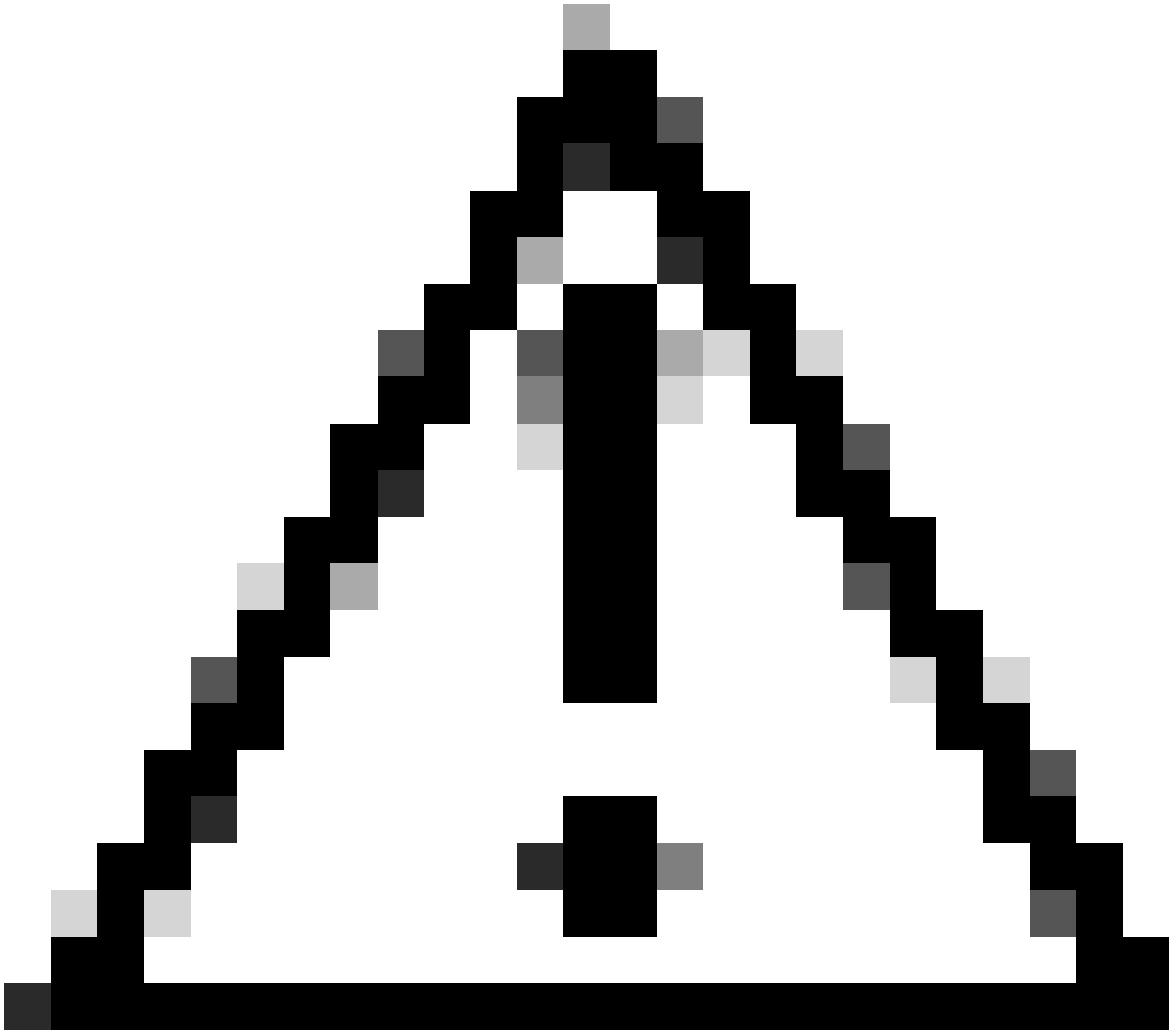
Maak de WLC private key en WLC CSR met informatie uit sectie WLC_device_req_ext voor SAN's:

```
openssl req -newkey rsa:4096 -keyout ./IntermCA/IntermCA.db.certs/WLC/WLC.key -nodes -config openssl.cn
```

OpenSSL opent een interactieve prompt voor u om Distinguished Name (DN) details in te voeren:

```
.........+..+.......+...........+...+......+.+............+..++++++++++++++
++++++
-----
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 []:WLC.example.com
```

Prompt voor WLC-certificaat onder verschillende namen



Waarschuwing: de algemene naam (CN) die u opgeeft in de interactieve prompt moet

---

identiek zijn aan een van de namen in de sectie [WLC_alt_names] van het bestand openssl.cnf.

Gebruik de CA met de naam IntermCA om de WLC CSR met de naam WLC.csr te ondertekenen met de extensies die zijn gedefinieerd onder [WLC_cert] en sla het ondertekende certificaat op in ./IntermCA/IntermCA.db.certs/WLC. Het WLC-apparaatcertificaat wordt WLC.crt genoemd:

```
openssl ca -config openssl.cnf -extensions WLC_cert -name IntermCA -out ./IntermCA/IntermCA.db.certs/WL
```

9800 WLC heeft een certificaat nodig in pfx-formaat om het te kunnen importeren. Maak een nieuw bestand dat de keten van CA's bevat die het WLC-certificaat hebben ondertekend. Dit wordt een certfile genoemd:

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

Om uw .pfx bestand te maken voert u een van deze opdrachten uit volgens de WLC-versie.

Voor uitvoeringen ouder dan 17.12.1:

```
openssl pkcs12 -export -macalg sha1 -legacy -descert -out ./IntermCA/IntermCA.db.certs/WLC/WLC.pfx -ink
```

Voor versie 17.12.1 of hoger:

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

ISE-apparaatcertificaat maken

Maak een nieuwe map om ISE-certs op te slaan op de machine waarop OpenSSL is geïnstalleerd in de tijdelijke CA cert-map IntermCA.db.certs. De nieuwe map wordt ISE genoemd:

```
mkdir ./IntermCA/IntermCA.db.certs/ISE
```

Wijzig de DNS-parameters in het gedeelte [ISE_alt_names] van het bestand openssl.cnf. Verander de voorbeeldnamen die voor uw gewenste waarden worden verstrekt, bevolken deze waarden het gebied van SANs van het WLC- certificaat:

```
[ISE_alt_names]
DNS.1   = ISE.example.com   <-----Change the values after the equals sign
DNS.2   = ISE2.example.com  <-----Change the values after the equals sign
```

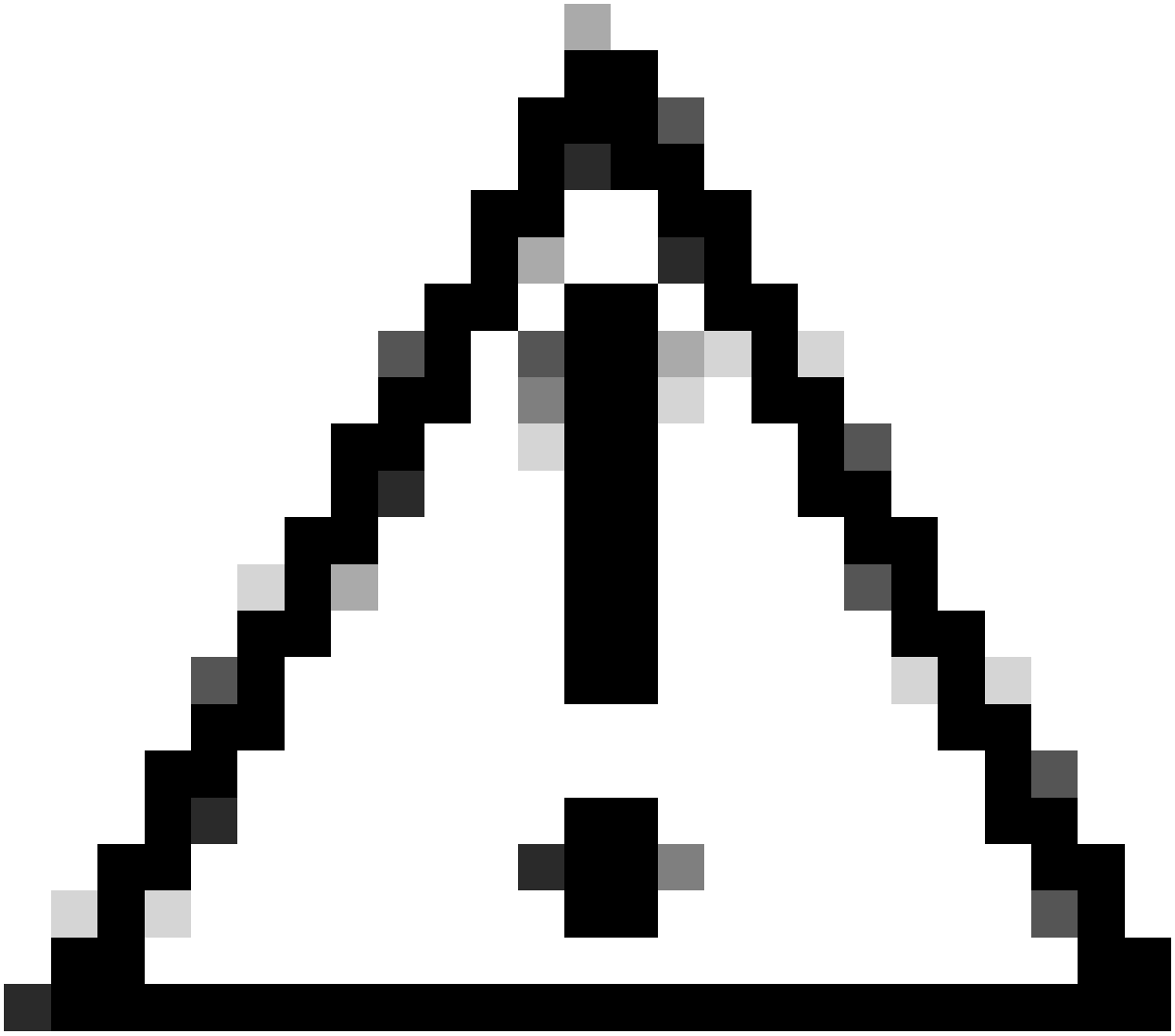
Maak de ISE private key en ISE CSR met informatie uit sectie ISE_device_req_ext voor SAN's:

```
openssl req -newkey rsa:2048 -sha256 -keyout ./IntermCA/IntermCA.db.certs/ISE/ISE.key -nodes -config ope
```

OpenSSL opent een interactieve prompt voor u om Distinguished Name (DN) details in te voeren:

```
-----
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 []:ISE.example.com
```

Prompt voor ISE-certificaat onder verschillende namen



Waarschuwing: de CN die u opgeeft op de interactieve prompt moet precies hetzelfde zijn als een van de Namen op de [ISE_alt_names] sectie van het bestand openssl.cnf.

Gebruik de CA met de naam IntermCA om de ISE CSR met de naam ISE.csr te ondertekenen met de extensies die zijn gedefinieerd onder [ISE_cert] en sla het ondertekende certificaat op in ./IntermCA/IntermCA.db.certs/WLC. Het ISE-apparaatcertificaat wordt ISE.crt genoemd:

```
openssl ca -config openssl.cnf -extensions ISE_cert -name IntermCA -out ./IntermCA/IntermCA.db.certs/IS
```

Certificaten importeren in apparaten

Certificaten importeren naar ISE

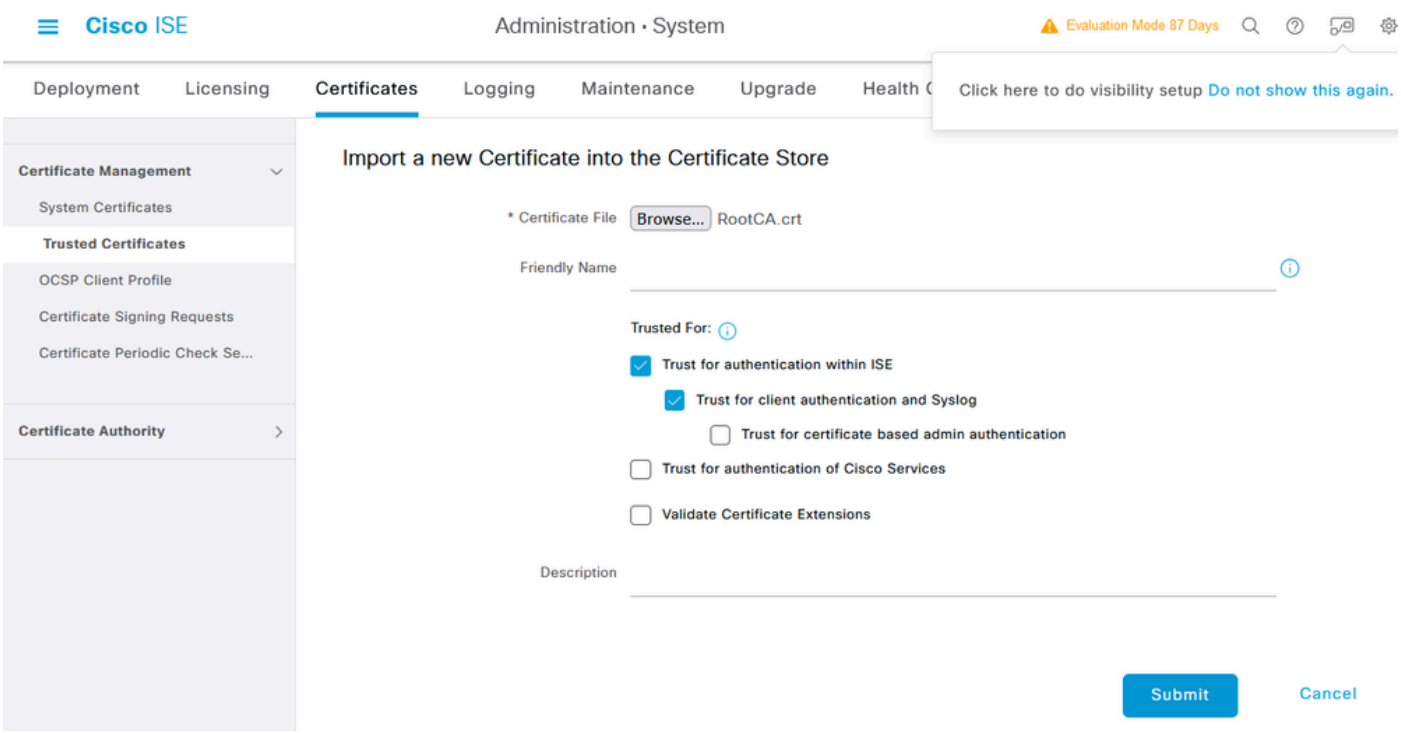
1. Importeer het Root CA-certificaat van de ISE-certificaatketen naar het vertrouwde

certificaatarchief.

2. Navigeer naar Beheer>Systeem>Certificaten>Betrouwbare certificaten.

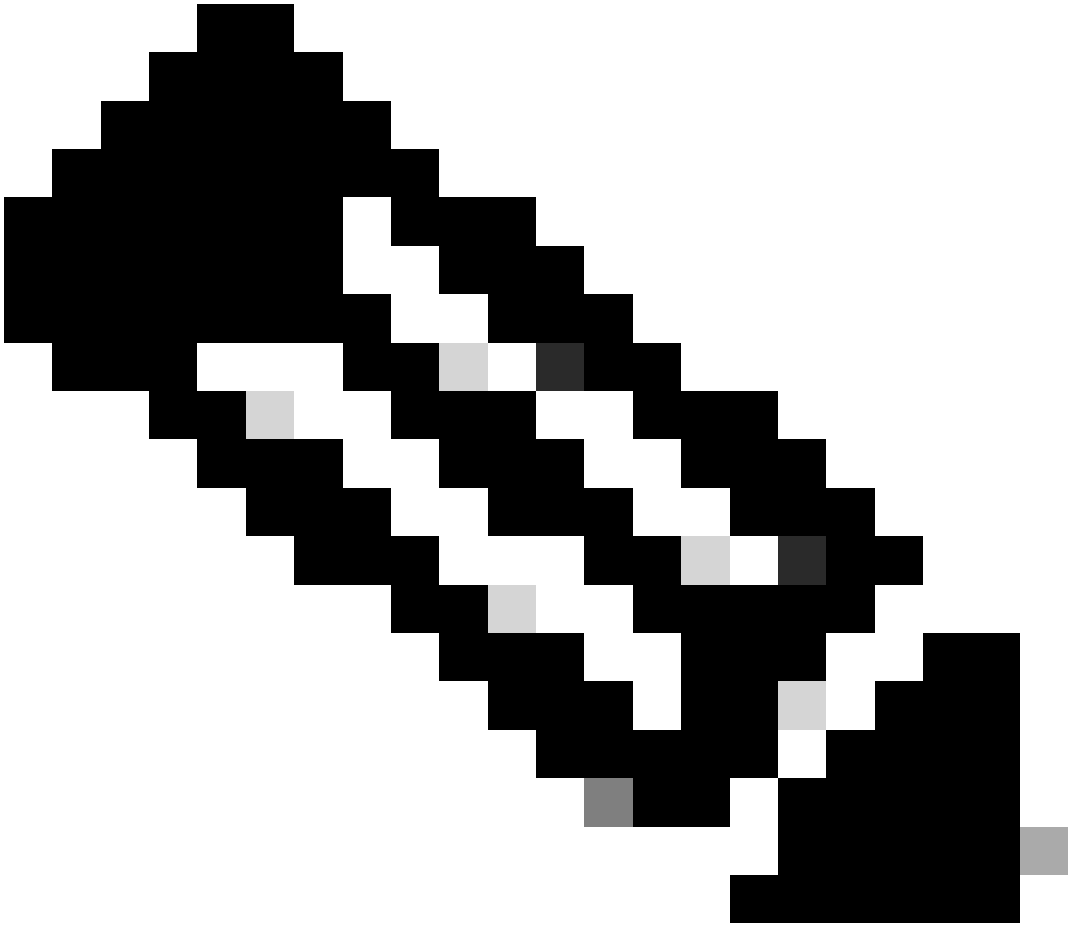
3. Klik op Bladeren en selecteer het bestand Root.crt.

4. Controleer het Vertrouwen voor verificatie binnen ISE evenals Vertrouwen voor cliëntauthentificatie en selectietekens Syslog en klik vervolgens op Indienen:

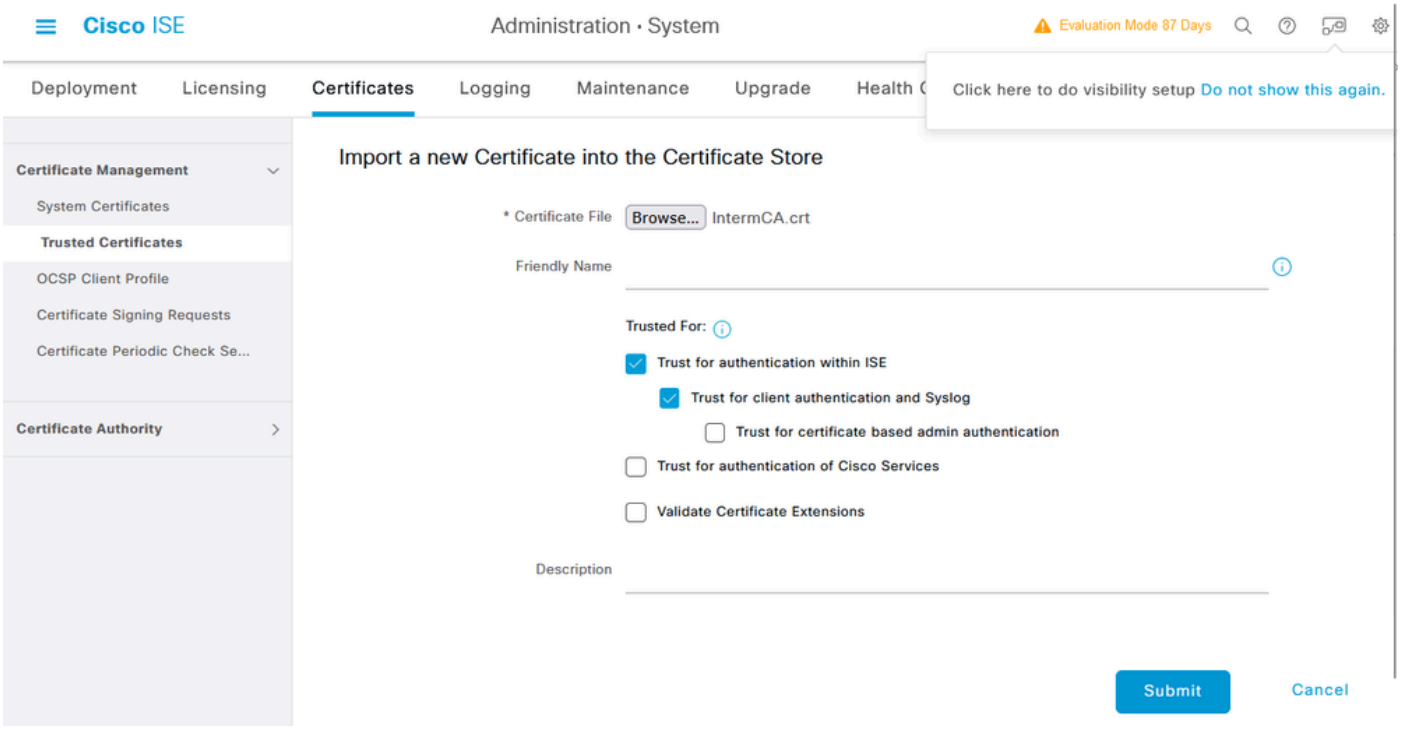


Dialoogvenster voor importeren van ISE-basiscertificaten

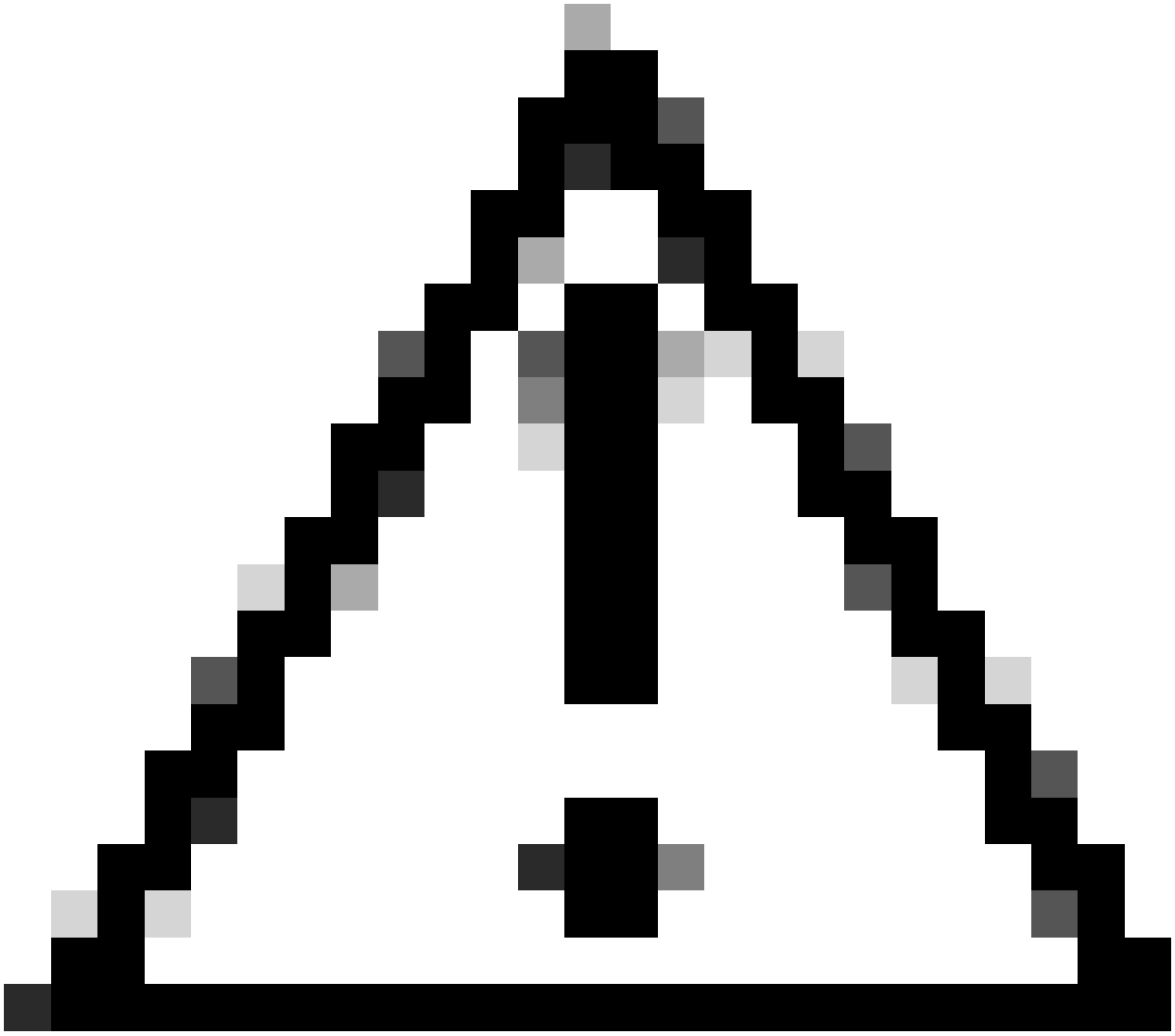
Doe hetzelfde voor het tussentijds certificaat, indien dit bestaat.



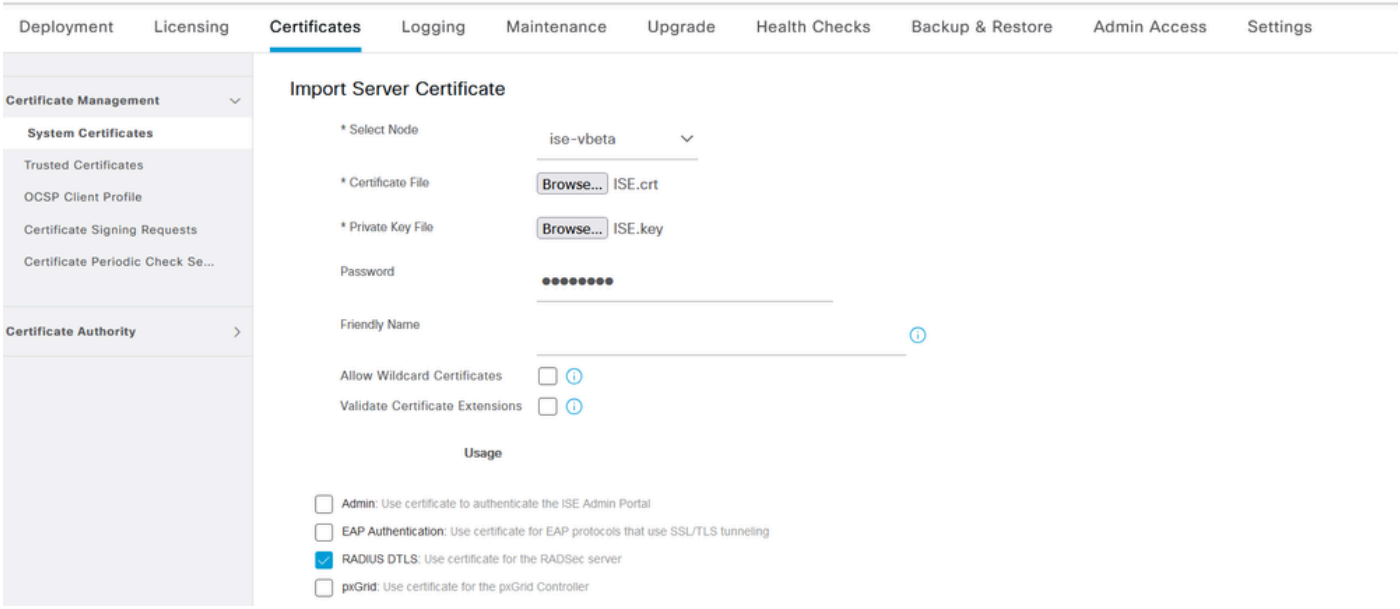
Opmerking: Herhaal de stappen voor elk CA-certificaat dat deel uitmaakt van de valideringsketen van het ISE-certificaat. Begin altijd met het Root CA-certificaat en eindig met het laagste Tussentijdse CA-certificaat van de keten.



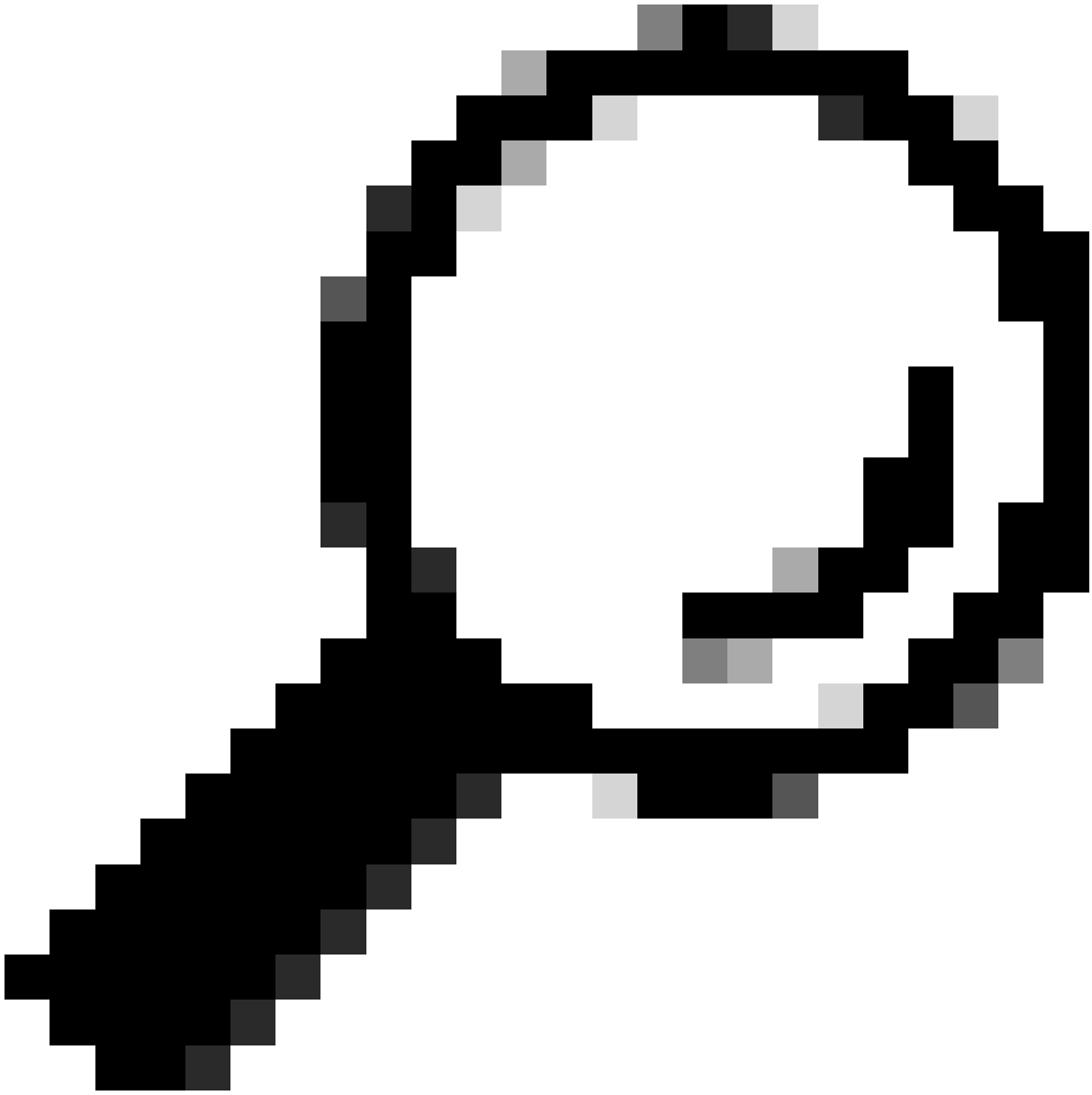
Dialoogvenster voor tussentijds CA-certificaat importeren



Waarschuwing: als het ISE-certificaat en het WLC-certificaat worden afgegeven door verschillende CA's, moet u ook alle CA-certificaten importeren die behoren tot de WLC-certificaatketen. ISE accepteert het WLC-certificaat niet op de DTLS-certificaatuitwisseling totdat u die CA-certificaten importeert.



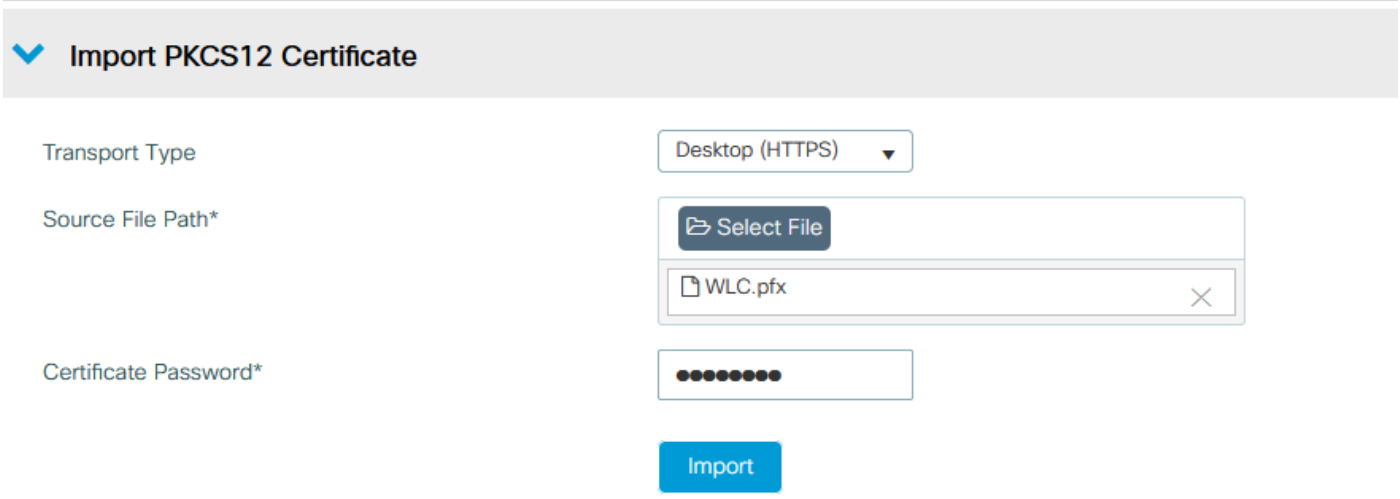
Menu Importeren van ISE-apparaatcertificaat



Tip: u hoeft bij deze stap alleen het ISE-apparaatcertificaat te importeren. Dit certificaat is de enige ISE-uitwisseling om de DTLS-tunnel in te stellen. Het is niet nodig om het WLC-apparaatcertificaat en de privésleutel te importeren, aangezien het WLC-certificaat wordt geverifieerd met het gebruik van de eerder geïmporteerde CA-certificaten.

Importeer certificaten naar WLC

1. Navigeer naar Configuration > Security > PKI Management op de WLC en ga naar het tabblad Certificaat toevoegen.
2. Klik op de vervolgkeuzelijst PKCS12-certificaat importeren en stel het transporttype in als bureaublad (HTTPS).
3. Klik op de knop Bestand selecteren en selecteer het bestand .pfx dat u eerder hebt gemaakt.
4. Typ het invoerwachtwoord en klik tot slot op Importeren.

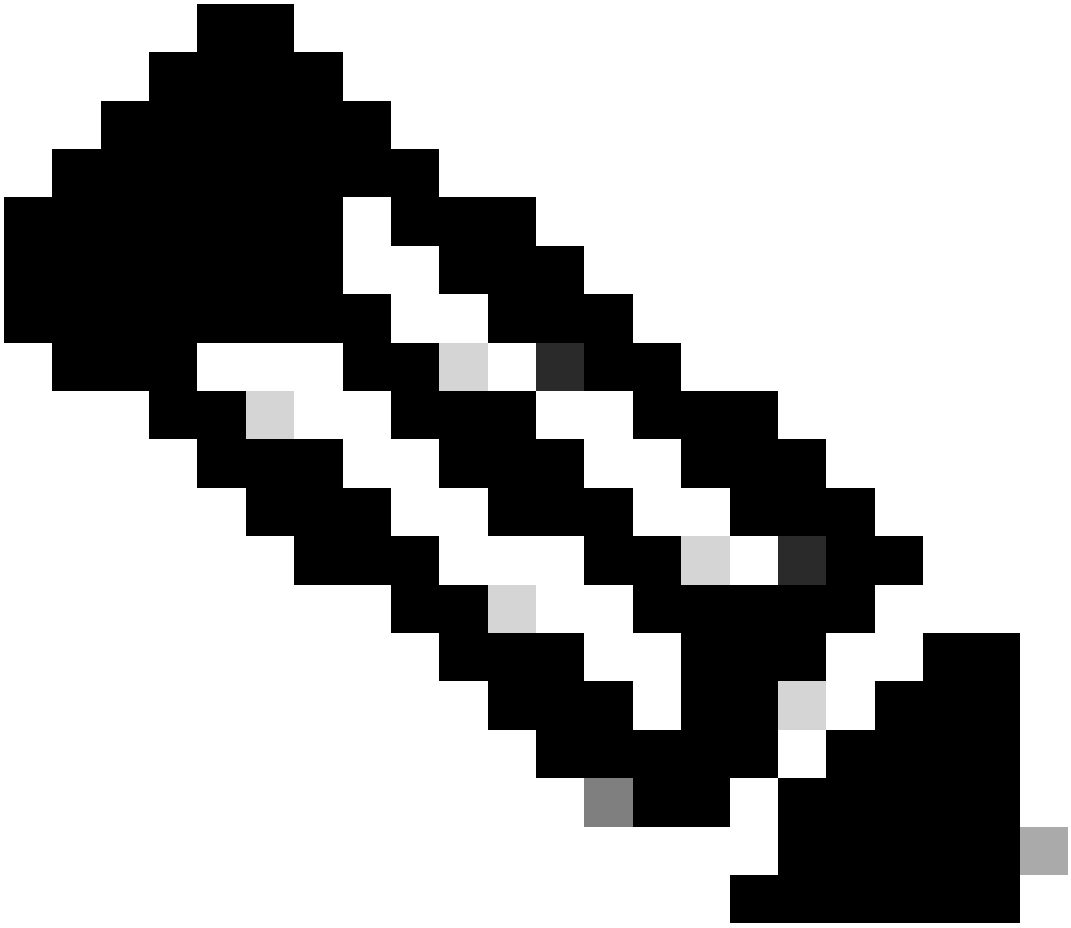


Dialoogvenster WLC-certificaat importeren

Raadpleeg voor meer informatie over het importproces CSR-certificaten genereren en downloaden op Catalyst 9800 WLC's.

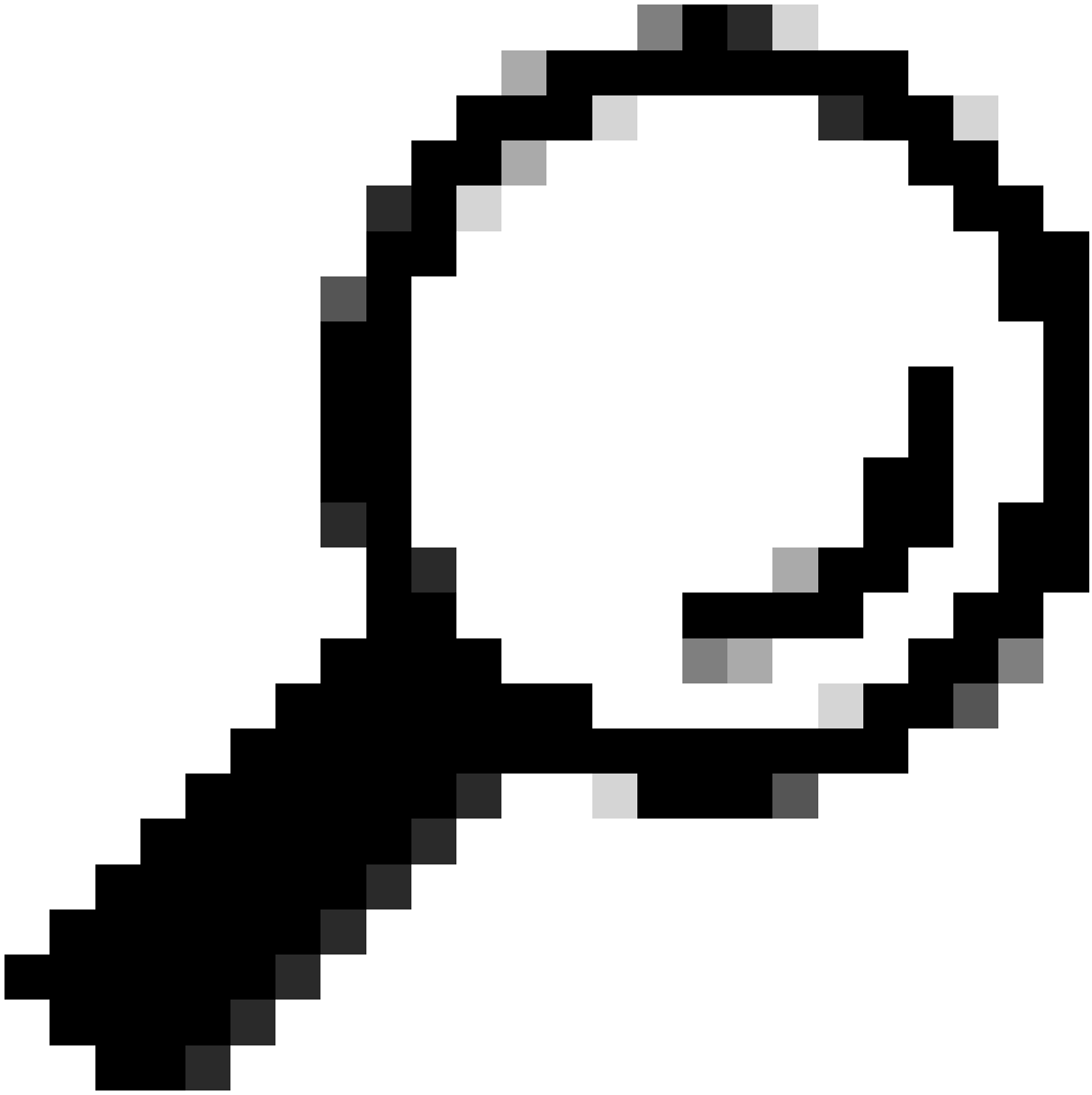
Schakel de herroepingscontrole binnen elk automatisch gemaakt trustpoint uit als de WLC geen certificaatherroepingslijst heeft die het kan controleren door het netwerk:

```
9800#configure terminal

9800(config)#crypto pki trustpoint WLC.pfx
9800(config)#revocation-check none
9800(config)#exit

9800(config)#crypto pki trustpoint WLC.pfx-rrr1
9800(config)#revocation-check none
9800(config)#exit
```



Opmerking: als u een CA op meerdere niveaus hebt gemaakt op OpenSSL met de CA op meerdere niveaus configureren op OpenSSL om document met Cisco IOS XE-certificaten te genereren, moet u de herroepingscontrole uitschakelen aangezien er geen CRL-server is gemaakt.

De geautomatiseerde import creëert de nodige vertrouwenspunten om het WLC-certificaat en de bijbehorende CA-certificaten te bevatten.



Tip: Als de WLC-certificaten zijn uitgegeven door dezelfde CA als de ISE-certificaten, kunt u dezelfde vertrouwde punten gebruiken die automatisch zijn gemaakt bij de WLC-certificaatimport. Het is niet nodig om de ISE-certificaten afzonderlijk in te voeren.

Als het WLC-certificaat wordt afgegeven door een andere CA dan het ISE-certificaat, moet u ook de ISE CA-certificaten importeren naar de WLC voor de WLC om het ISE-apparaatcertificaat te vertrouwen.
Maak een nieuw vertrouwingspunt voor de Root CA en voer de ISE Root CA in:

```
9800(config)#crypto pki trustpoint ISEroot
9800(ca-trustpoint)#revocation-check none
```

```
9800(ca-trustpoint)#enrollment terminal
9800(ca-trustpoint)#chain-validation stop
9800(ca-trustpoint)#exit
9800(config)#crypto pki authenticate ISEroot

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

------Paste the ISE root CA-----
```

Importeer het volgende tussenliggende CA-certificaat op de ISE CA-keten, met andere woorden het CA-certificaat dat is afgegeven door de Root CA:

```
hamariomed1(config)#crypto pki trustpoint ISEintermediate
hamariomed1(ca-trustpoint)#revocation-check none
hamariomed1(ca-trustpoint)#chain-validation continue ISErootCA
hamariomed1(ca-trustpoint)#enrollment terminal
hamariomed1(ca-trustpoint)#exit

hamariomed1(config)#crypto pki authenticate ISEintermediate

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

------Paste the ISE intermediate CA-------
```

Elke bijkomende CA in de keten vereist een afzonderlijk trustpoint. Elk trustpoint in de keten moet verwijzen naar het trustpoint dat het emittentencertificaat bevat van het certificaat dat u wilt importeren met de opdrachtkettingvalidatie <Emittent trustpoint name>.

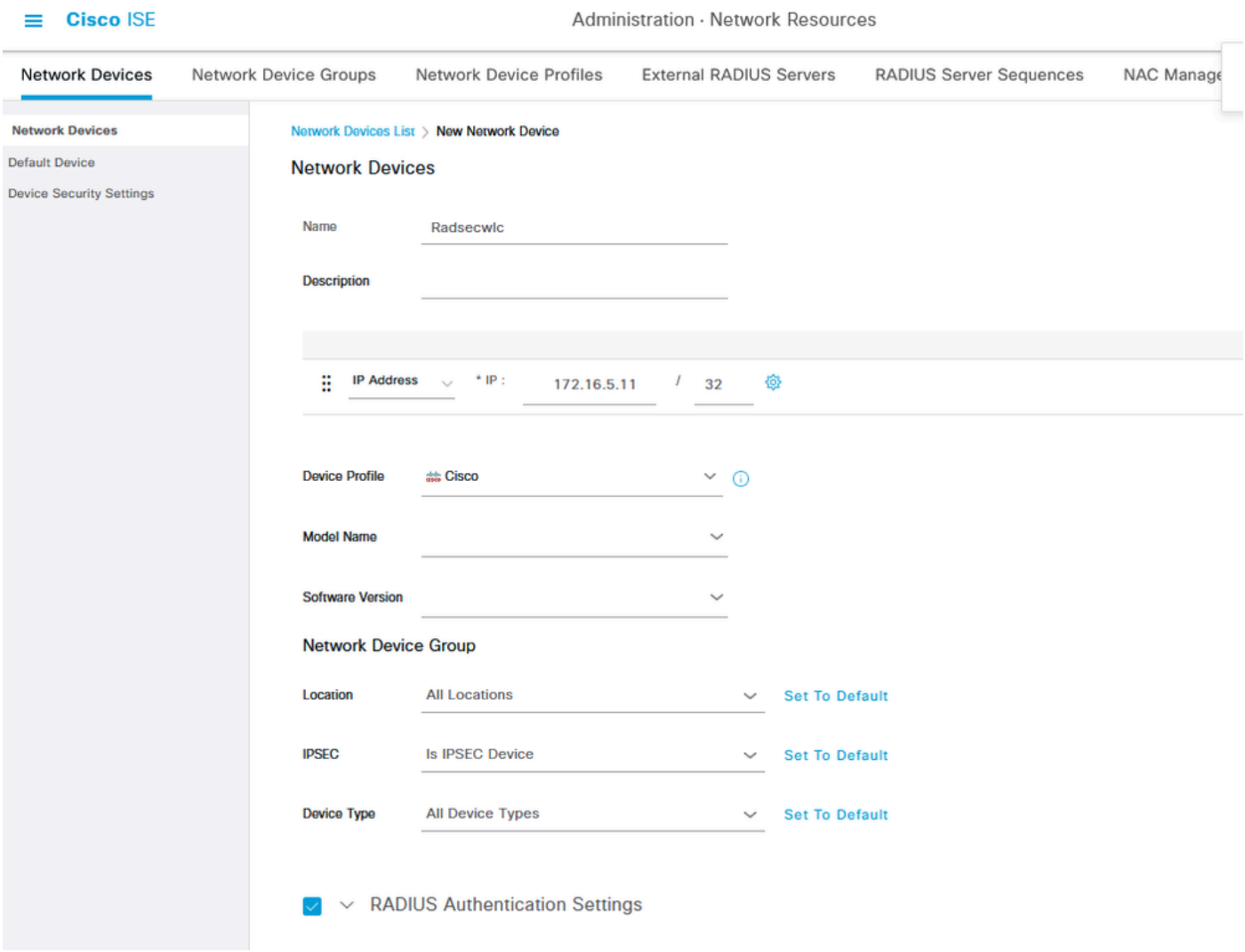
Importeer zoveel CA-certificaten als uw CA-keten bevat. U bent klaar nadat u de emittent CA van het ISE-apparaatcertificaat importeert, noteer de naam van dit trustpoint.

U hoeft het ISE-apparaatcertificaat niet op de WLC te importeren om RADIUS DTLS te kunnen gebruiken.

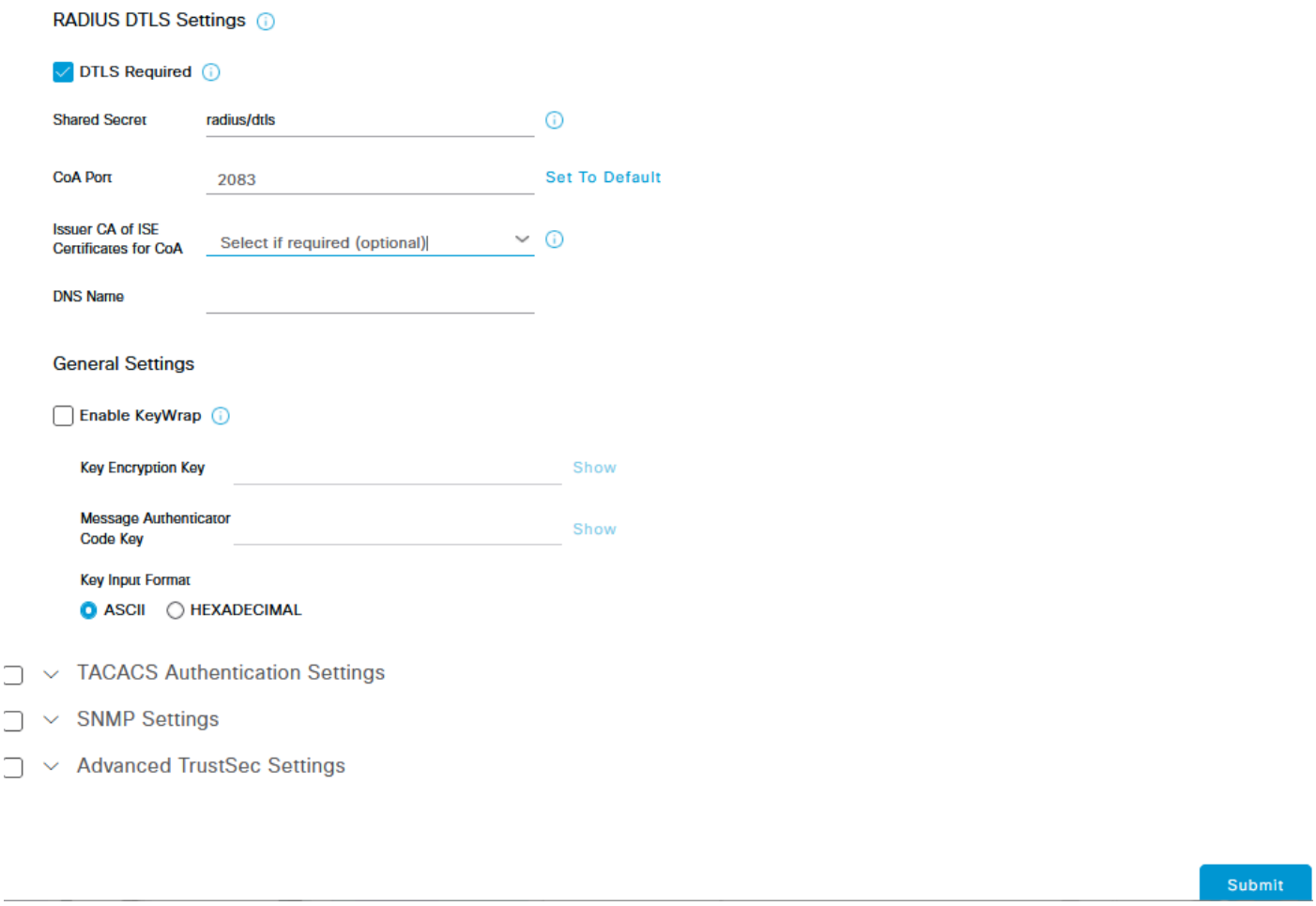
RADIUS-DTLS configureren

ISE-configuratie

Voeg de WLC als netwerkapparaat toe aan ISE, om dit te doen, navigeer naar Beheer>Netwerkbronnen>Netwerkapparaten>Toevoegen
Voer de naam van het apparaat en de IP van de WLC-interface in waarmee het RADIUS-verkeer wordt gegenereerd. Meestal de draadloze beheerinterface voor IP. Scroll naar beneden en controleer RADIUS-verificatie-instellingen en DTLS vereist en klik op Indienen:



Nieuwe configuratie van netwerkapparaten



RADIUS DTLS-instellingen voor het netwerkapparaat op ISE

WLC-configuratie

Definieer een nieuwe Radius-server samen met het ISE IP-adres en de standaardpoort voor Radius DTLS. Deze configuratie is alleen beschikbaar op de CLI:

```
9800#configure terminal
9800(config)#radius server ISE
9800(config-radius-server)#address ipv4
```

```
9800(config-radius-server)#dtls port 2083
```

Straal DTLS moet de gedeelde geheime straal/dtls gebruiken, de 9800 WLC negeert elke geconfigureerde toets anders dan deze:

```
9800(config-radius-server)#key radius/dtls
```

Gebruik het`dtls trustpoint client`
bevel om trustpoint te vormen dat het WLC apparatencertificaat bevat om voor de tunnel DTLS te ruilen.
Gebruik de opdracht om het`dtls trustpoint server`
trustpoint te configureren dat de emittent CA bevat voor het ISE-apparaatcertificaat.

Zowel de client- als de server trustpoint namen zijn alleen hetzelfde als de WLC- en ISE-certificaten worden afgegeven door dezelfde CA:

```
9800(config-radius-server)#dtls trustpoint client WLC.pfx
9800(config-radius-server)#dtls trustpoint server WLC.pfx
```

Configureer de WLC om te controleren of er een van de alternatieve onderwerpnamen (SAN's) aanwezig is op het ISE-certificaat. Deze configuratie moet exact overeenkomen met een van de SAN's die in het veld SAN's van het certificaat aanwezig zijn.

De 9800 WLC voert geen reguliere expressie-gebaseerde match uit voor het SAN-veld. Dit betekent bijvoorbeeld dat de opdracht `dtls match-server-identity hostname *.example.com` voor een wildcard-certificaat met *.voorbeeld.com in het SAN-veld correct is, maar dezelfde opdracht voor een certificaat met www.example.com in het SAN-veld niet.

WLC controleert deze naam niet tegen enige naamserver:

```
9800(config-radius-server)#dtls match-server-identity hostname ISE.example.com
9800(config-radius-server)#exit
```

Maak een nieuwe servergroep om de nieuwe RADIUS DTLS te gebruiken voor verificatie:

```
9800(config)#aaa group server radius Radsec
9800(config-sg-radius)#server name ISE
9800(config-sg-radius)#exit
```

Vanaf dit punt kunt u deze servergroep gebruiken als elke andere servergroep op de WLC. Raadpleeg 802.1X-verificatie configureren op Catalyst 9800 draadloze controllerserie om deze

server te gebruiken voor draadloze clientverificatie.

Verifiëren

Controleer de certificaatinformatie

Om de certificaatinformatie voor de gemaakte certificaten te verifiëren, voert u op de Linux-terminal de opdracht uit:

```
openssl x509 -in

      -text -noout
```

Het toont de volledige certificaatinformatie. Dit is nuttig om de CA van de emittent van een bepaald certificaat te bepalen of indien de certificaten de vereiste EKU's en SAN's bevatten:

```
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-apparaatcertificaatinformatie zoals weergegeven door OpenSSL

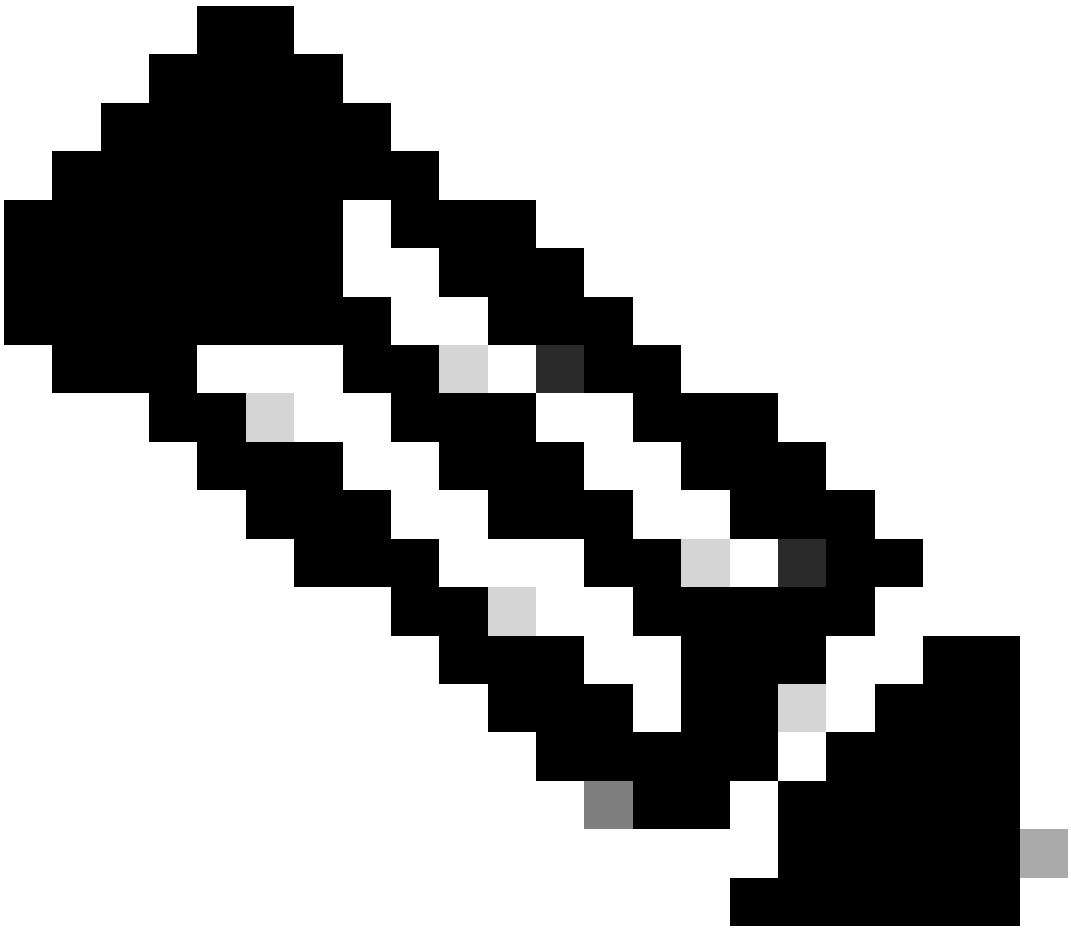
Testverificatie uitvoeren

Vanuit de WLC kunt u de Radius DTLS-functionaliteit testen met de opdracht test aaa group

new-code

```
9800#test aaa group Radsec testuser Cisco123 new-code
User successfully authenticated
```

```
USER ATTRIBUTES

username              0   "testuser"
```



Opmerking: Een toegangsweigering-uitvoer op de testopdracht betekent dat de WLC een Access-Reject RADIUS-bericht heeft ontvangen, in welk geval RADIUS DTLS werkt. Het kan echter ook wijzen op het niet tot stand brengen van de DTLS-tunnel. Het testbevel maakt geen onderscheid tussen beide scenario's, zie de sectie van het oplossen van problemen om te identificeren als er een probleem is.

Problemen oplossen

Om de oorzaak van een mislukte verificatie te bekijken, kunt u deze opdrachten inschakelen voordat u een testverificatie uitvoert.

```
9800#debug radius
9800#debug radius radsec
9800#terminal monitor
```

Dit is de uitvoer van een succesvolle verificatie waarbij debugs ingeschakeld is:

```
9800#test aaa group Radsec testuser Cisco123 new-code
User successfully authenticated

USER ATTRIBUTES

username             0   "testuser"
9800#
Jul 18 21:24:38.301: %PARSER-5-HIDDEN: Warning!!! ' test platform-aaa group server-group Radsec user-na
Jul 18 21:24:38.313: RADIUS/ENCODE(00000000):Orig. component type = Invalid
Jul 18 21:24:38.313: RADIUS/ENCODE(00000000): dropping service type, "radius-server attribute 6 on-for-
Jul 18 21:24:38.313: RADIUS(00000000): Config NAS IP: 0.0.0.0
Jul 18 21:24:38.313: vrfid: [65535]  ipv6 tableid : [0]
Jul 18 21:24:38.313: idb is NULL
Jul 18 21:24:38.313: RADIUS(00000000): Config NAS IPv6: ::
Jul 18 21:24:38.313: RADIUS(00000000): sending
Jul 18 21:24:38.313: RADIUS/DECODE(00000000): There is no General DB. Want server details may not be sp
Jul 18 21:24:38.313: RADSEC: DTLS default secret
Jul 18 21:24:38.313: RADIUS/ENCODE: Best Local IP-Address 172.16.5.11 for Radius-Server 172.16.18.123
Jul 18 21:24:38.313: RADSEC: DTLS default secret
Jul 18 21:24:38.313: RADIUS(00000000): Send Access-Request to 172.16.18.123:2083 id 53808/10, len 54
RADIUS:  authenticator C3 4E 34 0A 91 EF 42 53 - 7E C8 BB 50 F3 98 B3 14
Jul 18 21:24:38.313: RADIUS:  User-Password       [2]   18  *
Jul 18 21:24:38.313: RADIUS:  User-Name           [1]   10  "testuser"
Jul 18 21:24:38.313: RADIUS:  NAS-IP-Address      [4]   6   172.16.5.11
Jul 18 21:24:38.313: RADIUS_RADSEC_ENQ_WAIT_Q: Success Server(172.16.18.123)/Id(10)
Jul 18 21:24:38.313: RADIUS_RADSEC_CLIENT_PROCESS: Got DATA SEND MSG
Jul 18 21:24:38.313: RADIUS_RADSEC_SOCK_SET:  0 Success
Jul 18 21:24:38.313: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(0) generated for sock(0)
Jul 18 21:24:38.313: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(0) generated for sock(0)
Jul 18 21:24:38.313: RADIUS_RADSEC_HASH_KEY_ADD_CTX: add [radius_radsec ctx(0x7522CE91BAC0)] succeedd f
Jul 18 21:24:38.313: RADIUS_RADSEC_GET_SOURCE_ADDR: Success
Jul 18 21:24:38.313: RADIUS_RADSEC_GET_SOCK_ADDR: Success
Jul 18 21:24:38.313: RADIUS_RADSEC_SET_LOCAL_SOCK: Success
Jul 18 21:24:38.313: RADIUS_RADSEC_SOCK_SET: Success
Jul 18 21:24:38.314: RADIUS_RADSEC_BIND_SOCKET: Success
Jul 18 21:24:38.314: RADIUS_RADSEC_CONN_SET_LPORT: Success
Jul 18 21:24:38.314: RADIUS_RADSEC_CONN_SET_SERVER_PORT: Success
Jul 18 21:24:38.314: RADIUS_RADSEC_CLIENT_HS_START: local port = 54509
Jul 18 21:24:38.314: RADIUS_RADSEC_SOCKET_CONNECT: Success
Jul 18 21:24:38.315: RADIUS_RADSEC_UPDATE_SVR_REF_CNT: Got radsec_data
Jul 18 21:24:38.315: RADIUS_RADSEC_UPDATE_SVR_REF_CNT: Got valid rctx, with server_handle B0000019
Jul 18 21:24:38.316: RADIUS_RADSEC_CLIENT_HS_START: TLS handshake in progress...(172.16.18.123/2083)
Jul 18 21:24:38.316: RADIUS_RADSEC_START_CONN_TIMER: Started (172.16.18.123/2083) for 5 secs
Jul 18 21:24:38.316: RADIUS_RADSEC_CONN_STATE_UPDATE: Success - State = 2
Jul 18 21:24:38.318: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
Jul 18 21:24:38.318: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(0) generated for sock(0)
Jul 18 21:24:38.318: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(0) generated for sock(0)
Jul 18 21:24:38.318: RADIUS_RADSEC_HASH_KEY_MATCH: hashkey1(0) matches hashkey2(0) TRUE
Jul 18 21:24:38.318: RADIUS_RADSEC_HASH_KEY_GET_CTX: radius radsec  sock_ctx(0x7522CE91BAC0:0) get for
Jul 18 21:24:38.318: RADIUS_RADSEC_PROCESS_SOCK_EVENT: Handle socket event for TLS handshake(172.16.18.
Jul 18 21:24:38.318: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Jul 18 21:24:38.318: RADIUS_RADSEC_START_CONN_TIMER: Started (172.16.18.123/2083) for 5 secs
```

```
Jul 18 21:24:38.318: RADIUS_RADSEC_HS_CONTINUE: TLS handshake in progress...(172.16.18.123/2083)
Jul 18 21:24:38.318: RADIUS_RADSEC_SOCK_TLS_EVENT_HANDLE: Success
Jul 18 21:24:38.318: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
Jul 18 21:24:38.327: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
Jul 18 21:24:38.327: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(0) generated for sock(0)
Jul 18 21:24:38.327: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(0) generated for sock(0)
Jul 18 21:24:38.327: RADIUS_RADSEC_HASH_KEY_MATCH: hashkey1(0) matches hashkey2(0) TRUE
Jul 18 21:24:38.327: RADIUS_RADSEC_HASH_KEY_GET_CTX: radius radsec  sock_ctx(0x7522CE91BAC0:0) get for
Jul 18 21:24:38.327: RADIUS_RADSEC_PROCESS_SOCK_EVENT: Handle socket event for TLS handshake(172.16.18.
Jul 18 21:24:38.327: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Jul 18 21:24:38.391: RADIUS_RADSEC_START_CONN_TIMER: Started (172.16.18.123/2083) for 5 secs
Jul 18 21:24:38.391: RADIUS_RADSEC_HS_CONTINUE: TLS handshake in progress...(172.16.18.123/2083)
Jul 18 21:24:38.391: RADIUS_RADSEC_SOCK_TLS_EVENT_HANDLE: Success
Jul 18 21:24:38.391: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
Jul 18 21:24:38.397: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
Jul 18 21:24:38.397: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(0) generated for sock(0)
Jul 18 21:24:38.397: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(0) generated for sock(0)
Jul 18 21:24:38.397: RADIUS_RADSEC_HASH_KEY_MATCH: hashkey1(0) matches hashkey2(0) TRUE
Jul 18 21:24:38.397: RADIUS_RADSEC_HASH_KEY_GET_CTX: radius radsec  sock_ctx(0x7522CE91BAC0:0) get for
Jul 18 21:24:38.397: RADIUS_RADSEC_PROCESS_SOCK_EVENT: Handle socket event for TLS handshake(172.16.18.
Jul 18 21:24:38.397: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Jul 18 21:24:38.397: RADIUS_RADSEC_HS_CONTINUE: TLS handshake success!(172.16.18.123/2083) <-------- TL
Jul 18 21:24:38.397: RADIUS_RADSEC_CONN_STATE_UPDATE: Success - State = 3
Jul 18 21:24:38.397: RADIUS_RADSEC_UPDATE_SVR_REF_CNT: Got radsec_data
Jul 18 21:24:38.397: RADIUS_RADSEC_UPDATE_SVR_REF_CNT: Got valid rctx, with server_handle B0000019
Jul 18 21:24:38.397: RADIUS-RADSEC-HS-SUCCESS: Negotiated Cipher is ECDHE-RSA-AES256-GCM-SHA384
Jul 18 21:24:38.397: RADIUS_RADSEC_START_DATA_SEND: RADSEC HS Done, Start data send (172.16.18.123/2083
Jul 18 21:24:38.397: RADIUS_RADSEC_UNQUEUE_WAIT_Q: Success Server(172.16.18.123)/Id(10)
Jul 18 21:24:38.397: RADIUS_RADSEC_MSG_SEND: RADSEC Write SUCCESS(id=10)
Jul 18 21:24:38.397: RADIUS(00000000): Started 5 sec timeout
Jul 18 21:24:38.397: RADIUS_RADSEC_UNQUEUE_WAIT_Q: Empty Server(172.16.18.123)/Id(-1)
Jul 18 21:24:38.397: RADIUS_RADSEC_START_DATA_SEND: no more data available
Jul 18 21:24:38.397: RADIUS_RADSEC_IDLE_TIMER: Started (172.16.18.123/2083)
Jul 18 21:24:38.397: RADIUS-RADSEC-HS-SUCCESS: Success
Jul 18 21:24:38.397: RADIUS_RADSEC_SOCK_TLS_EVENT_HANDLE: Success
Jul 18 21:24:38.397: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
Jul 18 21:24:38.453: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
Jul 18 21:24:38.453: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(0) generated for sock(0)
Jul 18 21:24:38.453: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(0) generated for sock(0)
Jul 18 21:24:38.453: RADIUS_RADSEC_HASH_KEY_MATCH: hashkey1(0) matches hashkey2(0) TRUE
Jul 18 21:24:38.453: RADIUS_RADSEC_HASH_KEY_GET_CTX: radius radsec  sock_ctx(0x7522CE91BAC0:0) get for
Jul 18 21:24:38.453: RADIUS_RADSEC_MSG_RECV: RADSEC Bytes read= 20, Err= 0
Jul 18 21:24:38.453: RADIUS_RADSEC_SOCK_READ_EVENT_HANDLE: Radius length is 113
Jul 18 21:24:38.453: RADIUS_RADSEC_SOCK_READ_EVENT_HANDLE: Going to read rest 93 bytes
Jul 18 21:24:38.453: RADIUS_RADSEC_MSG_RECV: RADSEC Bytes read= 93, Err= 0
Jul 18 21:24:38.453: RADIUS_RADSEC_SOCK_READ_EVENT_HANDLE: linktype = 7 - src port = 2083 - dest port =
Jul 18 21:24:38.453: RADIUS: Received from id 54509/10 172.16.18.123:2083, Access-Accept, len 113 <----
RADIUS:  authenticator 4E CE 96 63 41 4B 43 04 - C7 A2 B5 05 C2 78 A7 0D
Jul 18 21:24:38.453: RADIUS:  User-Name           [1]   10  "testuser"
Jul 18 21:24:38.453: RADIUS:  Class               [25]  83
RADIUS:   43 41 43 53 3A 61 63 31 30 31 32 37 62 64 38 74  [CACS:ac10127bd8t]
RADIUS:   47 58 50 47 4E 63 6C 57 76 2F 39 67 44 66 51 67  [GXPGNclWv/9gDfQg]
RADIUS:   63 4A 76 6C 35 47 72 33 71 71 47 36 4C 66 35 59  [cJvl5Gr3qqG6Lf5Y]
RADIUS:   52 42 2F 7A 57 55 39 59 3A 69 73 65 2D 76 62 65  [RB/zWU9Y:ise-vbe]
RADIUS:   74 61 6E 63 6F 2F 35 31 30 34 33 39 38 32 36 2F  [tanco/510439826/]
RADIUS:   39                  [ 9]
Jul 18 21:24:38.453: RADSEC: DTLS default secret
Jul 18 21:24:38.453: RADIUS/DECODE(00000000): There is no General DB. Reply server details may not be r
Jul 18 21:24:38.453: RADIUS(00000000): Received from id 54509/10
```

Onbekende CA gerapporteerd door WLC

Wanneer de WLC geen certificaten kan valideren die worden geleverd door ISE, slaagt het er niet in om de DTLS-tunnel te maken en worden verificaties mislukt.
Dit is een voorbeeld van de debug-berichten die worden weergegeven wanneer dit het geval is:

```
9800#test aaa group Radsec testuser Cisco123 new-code

Jul 19 00:59:09.695: %PARSER-5-HIDDEN: Warning!!! ' test platform-aaa group server-group Radsec user-na
Jul 19 00:59:09.706: RADIUS/ENCODE(00000000):Orig. component type = Invalid
Jul 19 00:59:09.707: RADIUS/ENCODE(00000000): dropping service type, "radius-server attribute 6 on-for-
Jul 19 00:59:09.707: RADIUS(00000000): Config NAS IP: 0.0.0.0
Jul 19 00:59:09.707: vrfid: [65535]  ipv6 tableid : [0]
Jul 19 00:59:09.707: idb is NULL
Jul 19 00:59:09.707: RADIUS(00000000): Config NAS IPv6: ::
Jul 19 00:59:09.707: RADIUS(00000000): sending
Jul 19 00:59:09.707: RADIUS/DECODE(00000000): There is no General DB. Want server details may not be sp
Jul 19 00:59:09.707: RADSEC: DTLS default secret
Jul 19 00:59:09.707: RADIUS/ENCODE: Best Local IP-Address 172.16.5.11 for Radius-Server 172.16.18.123
Jul 19 00:59:09.707: RADSEC: DTLS default secret
Jul 19 00:59:09.707: RADIUS(00000000): Send Access-Request to 172.16.18.123:2083 id 52764/13, len 54
RADIUS:  authenticator E8 09 1D B0 72 50 17 E6 - B4 27 F6 E3 18 25 16 64
Jul 19 00:59:09.707: RADIUS:  User-Password       [2]   18  *
Jul 19 00:59:09.707: RADIUS:  User-Name           [1]   10  "testuser"
Jul 19 00:59:09.707: RADIUS:  NAS-IP-Address      [4]   6   172.16.5.11
Jul 19 00:59:09.707: RADIUS_RADSEC_ENQ_WAIT_Q: Success Server(172.16.18.123)/Id(13)
Jul 19 00:59:09.707: RADIUS_RADSEC_CLIENT_PROCESS: Got DATA SEND MSG
Jul 19 00:59:09.707: RADIUS_RADSEC_SOCK_SET:  0 Success
Jul 19 00:59:09.707: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(0) generated for sock(0)
Jul 19 00:59:09.707: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(0) generated for sock(0)
Jul 19 00:59:09.707: RADIUS_RADSEC_HASH_KEY_ADD_CTX: add [radius_radsec ctx(0x7522CE91BAC0)] succeedd f
Jul 19 00:59:09.707: RADIUS_RADSEC_GET_SOURCE_ADDR: Success
Jul 19 00:59:09.707: RADIUS_RADSEC_GET_SOCK_ADDR: Success
Jul 19 00:59:09.707: RADIUS_RADSEC_SET_LOCAL_SOCK: Success
Jul 19 00:59:09.707: RADIUS_RADSEC_SOCK_SET: Success
Jul 19 00:59:09.707: RADIUS_RADSEC_BIND_SOCKET: Success
Jul 19 00:59:09.707: RADIUS_RADSEC_CONN_SET_LPORT: Success
Jul 19 00:59:09.707: RADIUS_RADSEC_CONN_SET_SERVER_PORT: Success
Jul 19 00:59:09.707: RADIUS_RADSEC_CLIENT_HS_START: local port = 49556
Jul 19 00:59:09.707: RADIUS_RADSEC_SOCKET_CONNECT: Success
Jul 19 00:59:09.709: RADIUS_RADSEC_UPDATE_SVR_REF_CNT: Got radsec_data
Jul 19 00:59:09.709: RADIUS_RADSEC_UPDATE_SVR_REF_CNT: Got valid rctx, with server_handle B0000019
Jul 19 00:59:09.709: RADIUS_RADSEC_CLIENT_HS_START: TLS handshake in progress...(172.16.18.123/2083)
Jul 19 00:59:09.709: RADIUS_RADSEC_START_CONN_TIMER: Started (172.16.18.123/2083) for 5 secsUser reject

uwu-9800#
Jul 19 00:59:09.709: RADIUS_RADSEC_CONN_STATE_UPDATE: Success - State = 2
Jul 19 00:59:09.711: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
Jul 19 00:59:09.711: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(0) generated for sock(0)
Jul 19 00:59:09.711: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(0) generated for sock(0)
Jul 19 00:59:09.711: RADIUS_RADSEC_HASH_KEY_MATCH: hashkey1(0) matches hashkey2(0) TRUE
Jul 19 00:59:09.711: RADIUS_RADSEC_HASH_KEY_GET_CTX: radius radsec  sock_ctx(0x7522CE91BAC0:0) get for
Jul 19 00:59:09.711: RADIUS_RADSEC_PROCESS_SOCK_EVENT: Handle socket event for TLS handshake(172.16.18.
Jul 19 00:59:09.711: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Jul 19 00:59:09.711: RADIUS_RADSEC_START_CONN_TIMER: Started (172.16.18.123/2083) for 5 secs
Jul 19 00:59:09.711: RADIUS_RADSEC_HS_CONTINUE: TLS handshake in progress...(172.16.18.123/2083)
Jul 19 00:59:09.711: RADIUS_RADSEC_SOCK_TLS_EVENT_HANDLE: Success
Jul 19 00:59:09.713: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
```

```
Jul 19 00:59:09.720: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
Jul 19 00:59:09.720: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(0) generated for sock(0)
Jul 19 00:59:09.720: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(0) generated for sock(0)
Jul 19 00:59:09.720: RADIUS_RADSEC_HASH_KEY_MATCH: hashkey1(0) matches hashkey2(0) TRUE
Jul 19 00:59:09.720: RADIUS_RADSEC_HASH_KEY_GET_CTX: radius radsec  sock_ctx(0x7522CE91BAC0:0) get for
Jul 19 00:59:09.720: RADIUS_RADSEC_PROCESS_SOCK_EVENT: Handle socket event for TLS handshake(172.16.18.
Jul 19 00:59:09.720: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Jul 19 00:59:09.722: RADIUS_RADSEC_HS_CONTINUE: TLS handshake failed!
Jul 19 00:59:09.722: RADIUS_RADSEC_UNQUEUE_WAIT_Q: Success Server(172.16.18.123)/Id(13)
Jul 19 00:59:09.722: RADIUS_RADSEC_FAILOVER_HANDLER:Failng-over to new server = 0x0
Jul 19 00:59:09.722: RADIUS_RADSEC_UNQUEUE_WAIT_Q: Empty Server(172.16.18.123)/Id(-1)
Jul 19 00:59:09.722: RADIUS_RADSEC_FAILOVER_HANDLER: no more data available
Jul 19 00:59:09.722: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Jul 19 00:59:09.722: RADIUS_RADSEC_CONN_CLOSE: Cleaned up timers for Radius RADSEC ctx
Jul 19 00:59:09.722: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(0) generated for sock(0)
Jul 19 00:59:09.722: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(0) generated for sock(0)
Jul 19 00:59:09.722: RADIUS_RADSEC_HASH_KEY_DEL_CTX: remove [radius_radsec  ctx(0x7522CE91BAC0)] succee
Jul 19 00:59:09.722: RADIUS_RADSEC_CONN_CLOSE: Hash table entry removed for RADSEC sock ctx
Jul 19 00:59:09.723: RADIUS_RADSEC_CONN_CLOSE: Success
Jul 19 00:59:09.723: RADIUS_RADSEC_SOCK_TLS_EVENT_HANDLE: Failed to complete TLS handshake <----------D
Jul 19 00:59:09.723: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Jul 19 00:59:09.723: RADIUS_RADSEC_CONN_CLOSE: Cleaned up timers for Radius RADSEC ctx
Jul 19 00:59:09.723: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(-1) generated for sock(-1)
Jul 19 00:59:09.723: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(-1) generated for sock(-1)
uwu-9800#
Jul 19 00:59:09.723: RADIUS_RADSEC_HASH_KEY_DEL_CTX: remove [radius_radsec  ctx(0x7522CE91BAC0)] succee
Jul 19 00:59:09.723: RADIUS_RADSEC_CONN_CLOSE: Hash table entry removed for RADSEC sock ctx
Jul 19 00:59:09.723: RADIUS_RADSEC_CONN_CLOSE: Success
Jul 19 00:59:09.723: RADIUS_RADSEC_SOCK_TLS_EVENT_HANDLE: Error
Jul 19 00:59:09.723: RADIUS_RADSEC_PROCESS_SOCK_EVENT: failed to hanlde radsec hs event
Jul 19 00:59:09.723: RADIUS/DECODE: No response from radius-server; parse response; FAIL
Jul 19 00:59:09.723: RADIUS/DECODE: Case error(no response/ bad packet/ op decode);parse response; FAIL
Jul 19 00:59:09.723: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
Jul 19 00:59:10.718: %RADSEC_AUDIT_MESSAGE-3-FIPS_AUDIT_FCS_RADSEC_SERVER_CERTIFICATE_VALIDATION_FAILUR
Jul 19 00:59:10.718: %RADSEC_AUDIT_MESSAGE-3-FIPS_AUDIT_FCS_RADSEC_SERVER_IDENTITY_CHECK_FAILURE: Chass
Jul 19 00:59:10.718: %RADSEC_AUDIT_MESSAGE-6-FIPS_AUDIT_FCS_DTLSC_DTLS_SESSION_CLOSED: Chassis 1 R0/0:
```

Om dit te corrigeren, dient u ervoor te zorgen dat de identiteit die is geconfigureerd op de WLC exact overeenkomt met een van de SAN's die in het ISE-certificaat zijn opgenomen:

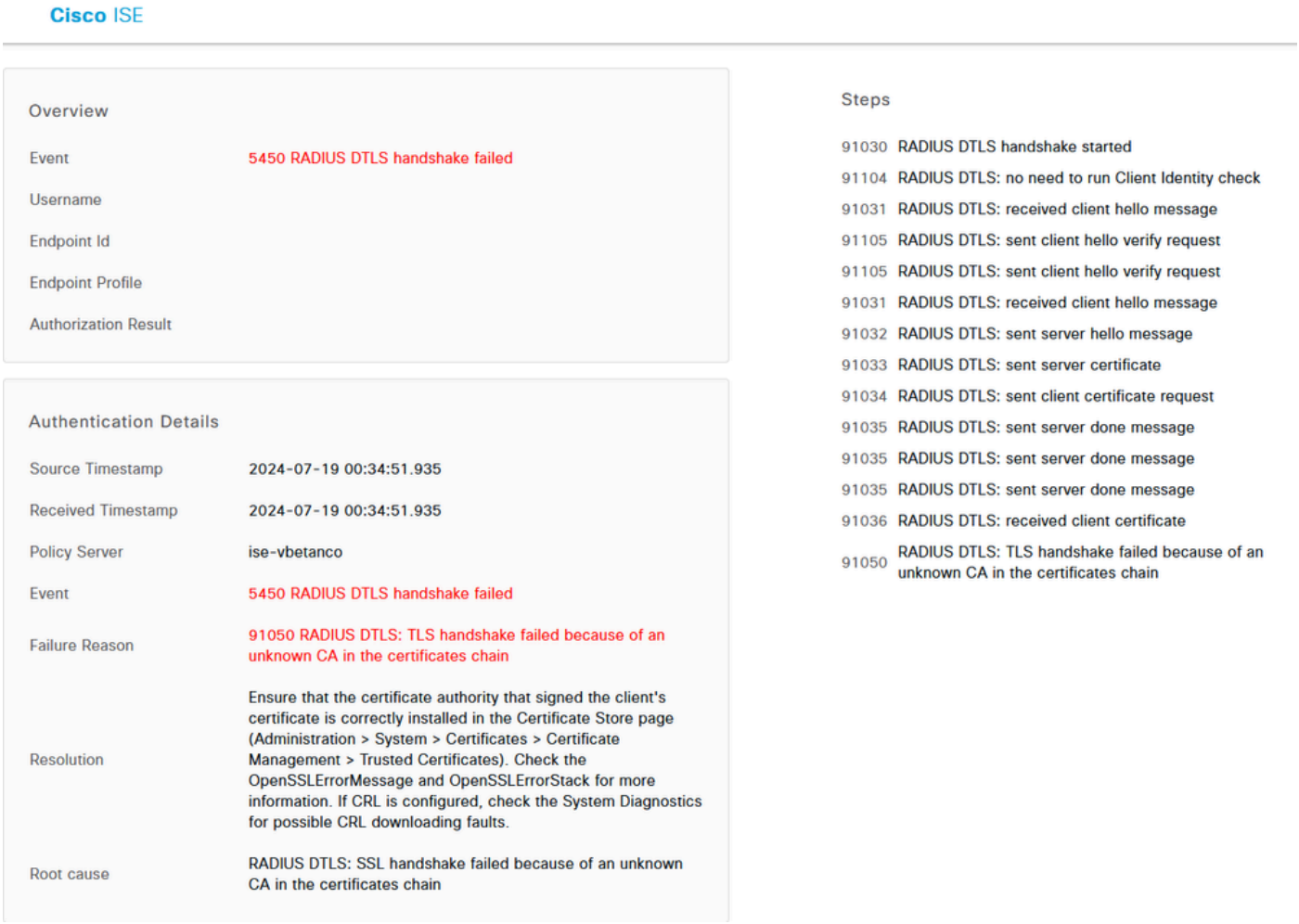
```
9800(config)#radius server
```

```
      9800(config)#dtls match-server-identity hostname
```

Zorg ervoor dat de CA-certificaatketen correct op de controller is geïmporteerd en dat de **dtls trustpoint server** configuration uses the Issuer CA trustpoint.

Onbekende CA gerapporteerd door ISE

Wanneer ISE geen certificaten kan valideren die worden geleverd door de WLC, kan de DTLS-tunnel niet worden gemaakt en kunnen verificaties niet worden uitgevoerd. Dit verschijnt als een fout in de actieve logboeken van RADIUS. Navigeer naar Operations>Radius>Live-logbestanden om te verifiëren.



ISE Live Log meldt DTLS Handshake falen door onbekend CA

Als u dit wilt corrigeren, controleert u zowel het tussentijds als het basiscertificaat. Selecteer vervolgens de selectievakjes Vertrouwen voor clientverificatie en Syslog onder Beheer>Systeem>Certificaten>Betrouwbare certificaten.

Herroepingscontrole is uitgevoerd

Wanneer de certificaten in WLC worden geïmporteerd, hebben de nieuw gecreëerde trustpoints herroepingscontrole ingeschakeld. Hierdoor probeert de WLC te zoeken naar een certificaatintrekkingslijst die niet beschikbaar of bereikbaar is en de certificaatverificatie niet doorstaat.
Zorg ervoor dat elk trustpoint in het verificatiepad voor de certificaten de opdracht bevat revocation-

check none .

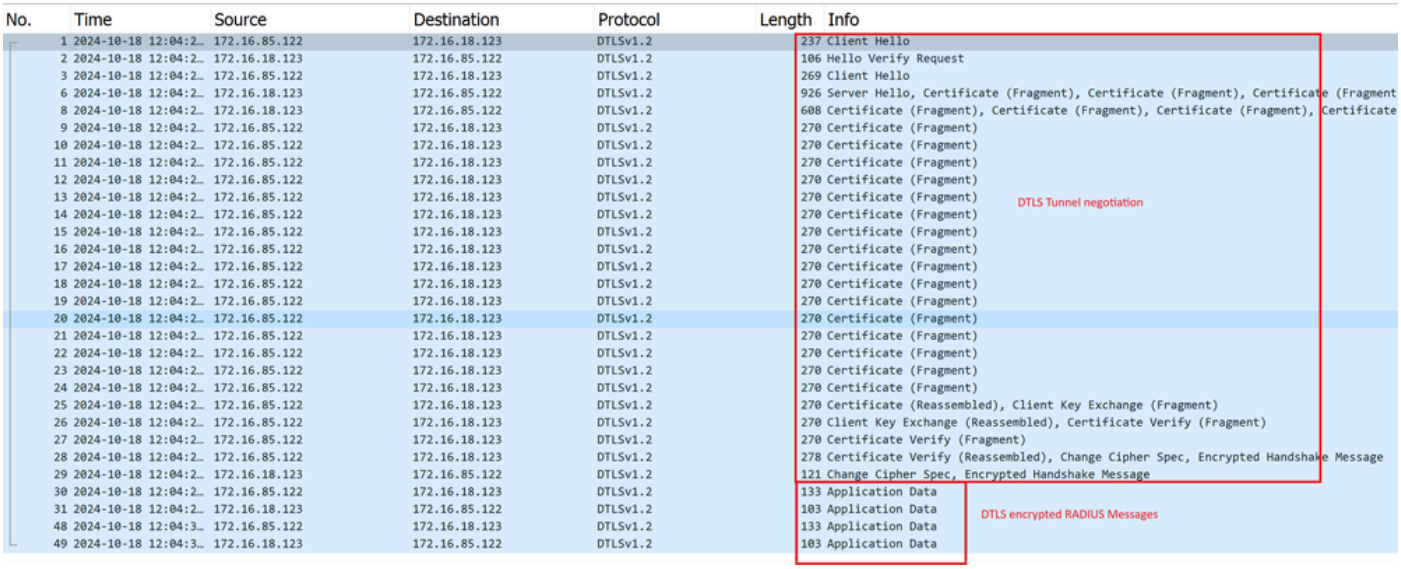
```
Jul 17 21:50:39.064: RADIUS_RADSEC_HASH_KEY_MATCH: hashkey1(0) matches hashkey2(0) TRUE
Jul 17 21:50:39.064: RADIUS_RADSEC_HASH_KEY_GET_CTX: radius radsec  sock_ctx(0x780FB0715978:0) get for
Jul 17 21:50:39.064: RADIUS_RADSEC_PROCESS_SOCK_EVENT: Handle socket event for TLS handshake(172.16.18.
Jul 17 21:50:39.064: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Jul 17 21:50:39.068: %PKI-3-CRL_FETCH_FAIL: CRL fetch for trustpoint WLC1.pfx failed
                      Reason : Enrollment URL not configured. <------ WLC tries to perform revocation ch
Jul 17 21:50:39.070: RADIUS_RADSEC_HS_CONTINUE: TLS handshake failed!
Jul 17 21:50:39.070: RADIUS_RADSEC_UNQUEUE_WAIT_Q: Success Server(172.16.18.123)/Id(2)
Jul 17 21:50:39.070: RADIUS_RADSEC_FAILOVER_HANDLER:Failng-over to new server = 0x0
Jul 17 21:50:39.070: RADIUS_RADSEC_UNQUEUE_WAIT_Q: Empty Server(172.16.18.123)/Id(-1)
Jul 17 21:50:39.070: RADIUS_RADSEC_FAILOVER_HANDLER: no more data available
Jul 17 21:50:39.070: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Jul 17 21:50:39.070: RADIUS_RADSEC_CONN_CLOSE: Cleaned up timers for Radius RADSEC ctx
Jul 17 21:50:39.070: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(0) generated for sock(0)
Jul 17 21:50:39.070: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(0) generated for sock(0)
Jul 17 21:50:39.070: RADIUS_RADSEC_HASH_KEY_DEL_CTX: remove [radius_radsec  ctx(0x780FB0715978)] succee
Jul 17 21:50:39.070: RADIUS_RADSEC_CONN_CLOSE: Hash table entry removed for RADSEC sock ctx
Jul 17 21:50:39.070: RADIUS_RADSEC_CONN_CLOSE: Success
Jul 17 21:50:39.070: RADIUS_RADSEC_SOCK_TLS_EVENT_HANDLE: Failed to complete TLS handshake
Jul 17 21:50:39.070: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Jul 17 21:50:39.070: RADIUS_RADSEC_CONN_CLOSE: Cleaned up timers for Radius RADSEC ctx
Jul 17 21:50:39.070: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(-1) generated for sock(-1)
Jul 17 21:50:39.070: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(-1) generated for sock(-1)
Jul 17 21:50:39.070: RADIUS_RADSEC_HASH_KEY_DEL_CTX: remove [radius_radsec  ctx(0x780FB0715978)] succee
Jul 17 21:50:39.070: RADIUS_RADSEC_CONN_CLOSE: Hash table entry removed for RADSEC sock ctx
Jul 17 21:50:39.070: RADIUS_RADSEC_CONN_CLOSE: Success
Jul 17 21:50:39.070: RADIUS_RADSEC_SOCK_TLS_EVENT_HANDLE: Error
Jul 17 21:50:39.070: RADIUS_RADSEC_PROCESS_SOCK_EVENT: failed to hanlde radsec hs event
Jul 17 21:50:39.070: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
```

Probleemoplossing voor DTLS-tunnelinstelling bij pakketvastlegging

De 9800 WLC biedt de functie Embedded Packet Capture (EPC) waarmee u al het verzonden en ontvangen verkeer voor een bepaalde interface kunt opnemen. ISE biedt een vergelijkbare functie genaamd TCP dump om inkomend en uitgaand verkeer te monitoren. Wanneer ze tegelijkertijd worden gebruikt, kunt u het DTLS-sessiegedefinieerde verkeer analyseren vanuit het perspectief van beide apparaten.

Raadpleeg de beheerdershandleiding van Cisco Identity Services Engine voor gedetailleerde stappen om TCP bij ISE te configureren. Raadpleeg ook Catalyst 9800 draadloze LAN-controllers voor probleemoplossing voor informatie over het configureren van de EPC-functie op de WLC.

Dit is een voorbeeld van een succesvolle DTLS-tunnelinrichting.



Packet Capture van een RADIUS DTLS-tunnelonderhandeling en versleutelde berichten

Packet-opnamen tonen hoe de DTLS-tunnelinrichting verloopt. Als er een probleem is met de onderhandeling, van verloren verkeer tussen apparaten of DTLS-versleutelde waarschuwingspakketten, helpt de pakketopname u het probleem te identificeren.

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.