Configuration de Radius DTLS sur ISE et WLC 9800

Table des matières

Introduction

Ce document décrit une méthode pour créer les certificats nécessaires pour configurer RADIUS DTLS entre ISE et le WLC 9800.

Fond

RADIUS DTLS est une forme sécurisée du protocole RADIUS dans laquelle les messages RADIUS sont envoyés via un tunnel DTLS (Transport Layer Security) de données. Pour créer ce tunnel entre le serveur d'authentification et l'authentificateur, un ensemble de certificats est nécessaire. Cet ensemble de certificats nécessite la définition de certaines extensions de certificat

d'utilisation de clé étendue (EKU), en particulier l'authentification client sur le certificat WLC et l'authentification serveur ainsi que l'authentification client pour le certificat ISE.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Comment configurer le WLC 9800, le point d'accès (AP) pour le fonctionnement de base
- Comment utiliser l’application OpenSSL
- Infrastructure à clé publique (PKI) et certificats numériques

Composants utilisés

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

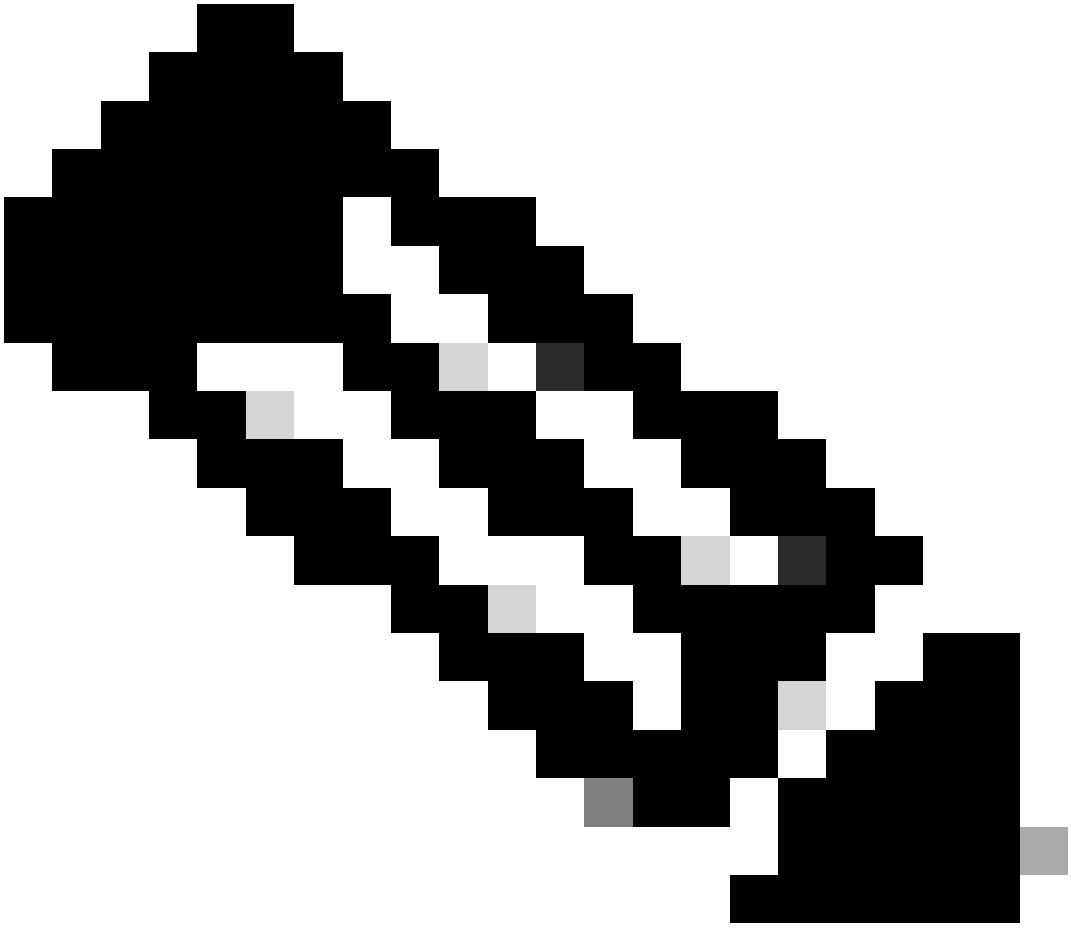
- Application OpenSSL (version 3.0.2).
- ISE (version 3.1.0.518)
- WLC 9800 (version 17.12.3)

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

Configurer

Aperçu

L'objectif est de créer une autorité de certification à deux niveaux avec une autorité de certification racine et une autorité de certification intermédiaire pour signer les certificats de point d'extrémité. Une fois les certificats signés, ils sont importés dans le WLC et ISE. Enfin, les périphériques sont configurés pour effectuer l'authentification RADIUS DTLS avec ces certificats.



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

Facultatif - Créer un certificat de périphérique WLC et ISE RADIUS DTLS

Le protocole RADIUS DTLS doit échanger des certificats entre ISE et WLC pour créer le tunnel DTLS. Si vous n'avez pas encore de certificats valides, vous pouvez créer une autorité de certification locale pour générer les certificats, référez-vous à Configurer une autorité de certification multiniveau sur OpenSSL pour générer des certificats compatibles avec Cisco IOS® XE et exécutez les étapes décrites sur le document du début à la fin de l'étape Créer un certificat CA intermédiaire.

Ajouter des sections de configuration sur le fichier openssl.cnf

Ouvrez votre fichier de configuration openssl.cnf et, au bas de celui-ci, copiez et collez les

sections WLC et ISE utilisées pour générer une demande de signature de certificat (CSR) valide.

Les sections ISE_device_req_ext et WLC_device_req_ext pointent chacune vers une liste de SAN à inclure dans le CSR :

```
#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
```

Par mesure de sécurité, l'autorité de certification remplace tous les SAN présents sur un CSR afin de le signer de sorte que les périphériques non autorisés ne puissent pas recevoir un certificat valide pour un nom qu'ils ne sont pas autorisés à utiliser. Afin de rajouter les SAN au certificat signé, utilisez le paramètre subjectAltName pour pointer vers la même liste de SAN que ceux utilisés pour la génération CSR.

ISE requiert à la fois les EKU serverAuth et clientAuth présents sur le certificat tandis que le WLC a seulement besoin de clientAuth. Ils sont ajoutés au certificat signé avec le paramètre extendedKeyUsage.

Copiez et collez les sections utilisées pour la signature de certificat au bas du fichier 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
```

Créer un certificat de périphérique WLC

Créez un nouveau dossier pour stocker les certificats WLC sur la machine sur laquelle OpenSSL est installé dans le dossier de certificats CA intermédiaire appelé IntermCA.db.certs. Le nouveau dossier est appelé WLC :

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

Modifiez les paramètres DNS sur la section [WLC_alt_names] du fichier openssl.cnf. Modifiez les noms d'exemple fournis pour les valeurs souhaitées. Ces valeurs renseignent le champ SAN du certificat WLC :

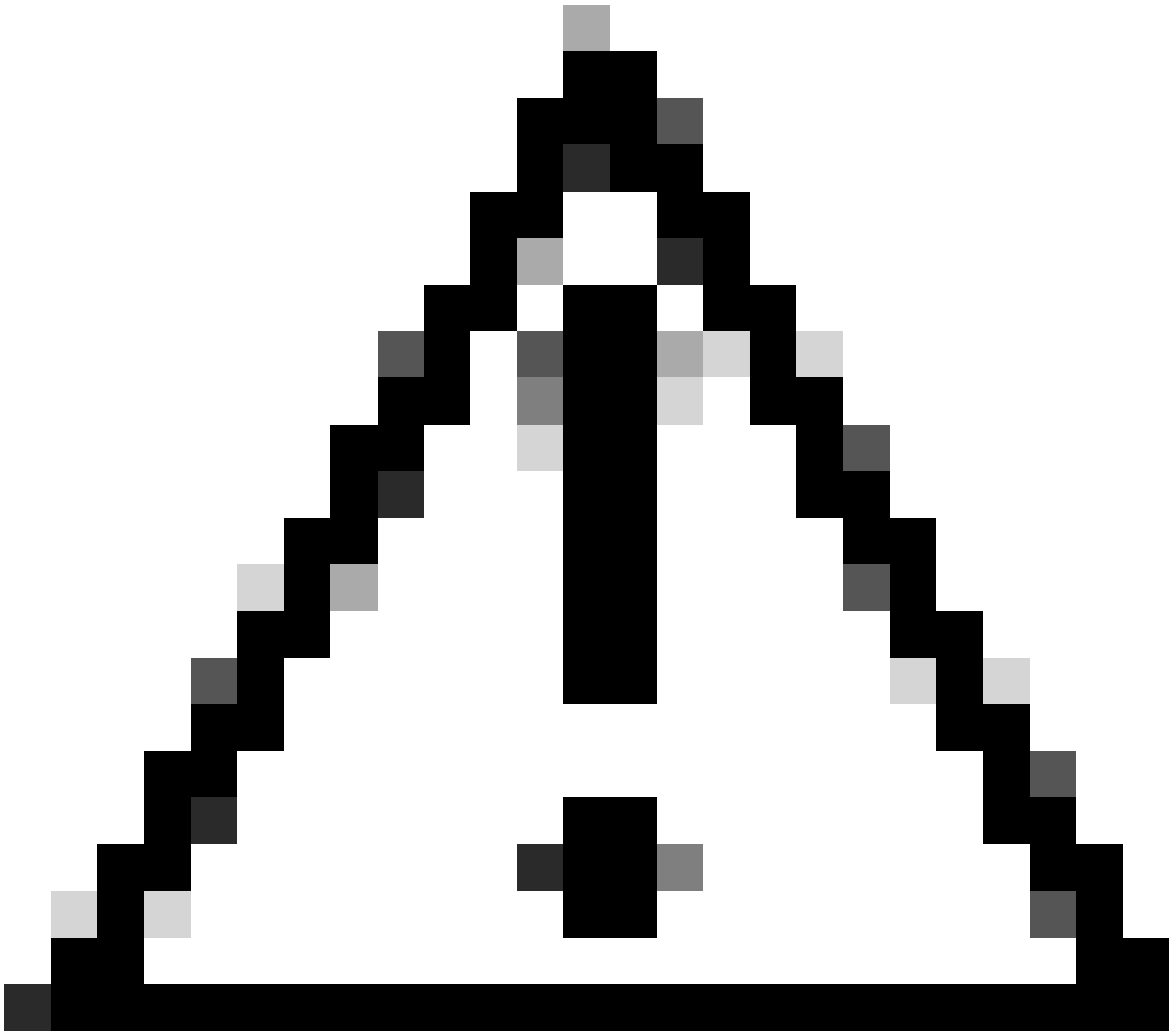
```
[WLC_alt_names]
DNS.1   = WLC.example.com    <-----Change the values after the equals sign
DNS.2   = WLC2.example.com   <-----Change the values after the equals sign
```

Créez la clé privée WLC et le CSR WLC avec les informations de la section WLC_device_req_ext pour les SAN :

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

OpenSSL ouvre une invite interactive vous invitant à saisir les détails du nom unique (DN) :

```
.........+..+.......+..........+...+......+.+...........+..++++++++++++++
++++++
-----
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
```



Attention : le nom commun (CN) que vous fournissez à l'invite interactive doit être identique à l'un des noms de la section [WLC_alt_names] du fichier openssl.cnf.

Utilisez l'autorité de certification nommée IntermCA pour signer le CSR du WLC nommé WLC.csr avec les extensions définies sous [WLC_cert] et stockez le certificat signé dans ./IntermCA/IntermCA.db.certs/WLC. Le certificat de périphérique WLC est appelé WLC.crt :

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

Le WLC 9800 a besoin que le certificat soit au format pfx pour l'importer. Créez un nouveau fichier qui contient la chaîne d'autorités de certification qui ont signé le certificat WLC, ceci s'appelle un certfile :

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

Pour créer votre fichier .pfx, exécutez l'une de ces commandes selon la version du WLC.

Pour les versions antérieures à 17.12.1 :

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

Pour la version 17.12.1 ou ultérieure :

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

Créer un certificat de périphérique ISE

Créez un nouveau dossier pour stocker les certificats ISE sur l'ordinateur sur lequel OpenSSL est installé dans le dossier de certificats AC intermédiaire appelé IntermCA.db.certs. Le nouveau dossier s'appelle ISE :

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

Modifiez les paramètres DNS dans la section [ISE_alt_names] du fichier openssl.cnf. Modifiez les noms d'exemple fournis pour les valeurs souhaitées, ces valeurs remplissent le champ SAN du certificat WLC :

```
[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
```

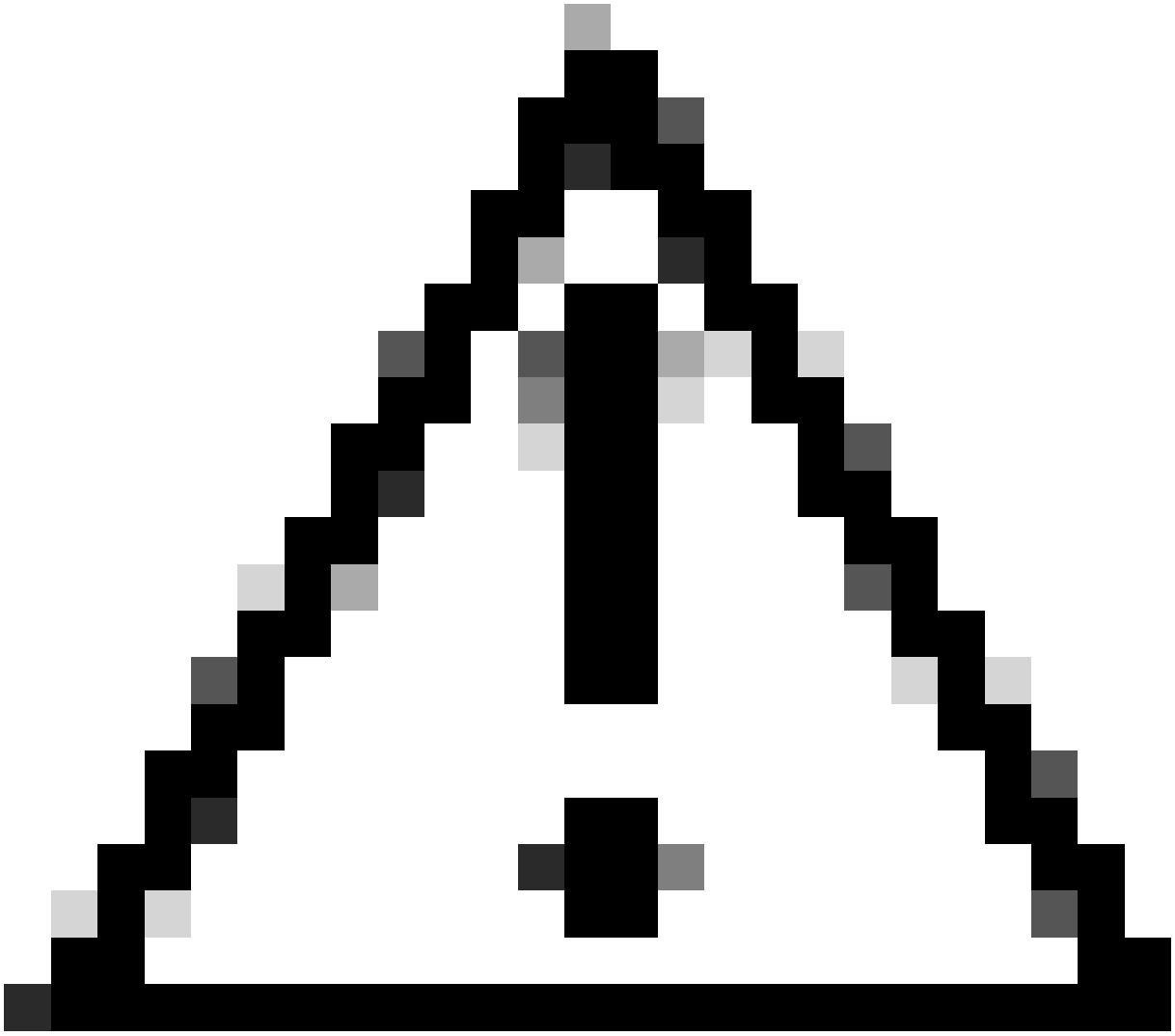
Créez la clé privée ISE et le CSR ISE avec les informations de la section ISE_device_req_ext pour les SAN :

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

OpenSSL ouvre une invite interactive vous invitant à saisir les détails du nom unique (DN) :

```
-----
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
```

Invite interactive du nom unique du certificat ISE



Attention : le CN que vous fournissez à l'invite interactive doit être exactement identique à l'un des noms de la section [ISE_alt_names] du fichier openssl.cnf.

Utilisez l'autorité de certification nommée IntermCA pour signer le CSR ISE nommé ISE.csr avec les extensions définies sous [ISE_cert] et stockez le certificat signé dans ./IntermCA/IntermCA.db.certs/WLC. Le certificat de périphérique ISE est appelé ISE.crt :

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

Importer des certificats vers des périphériques

Importer des certificats dans ISE

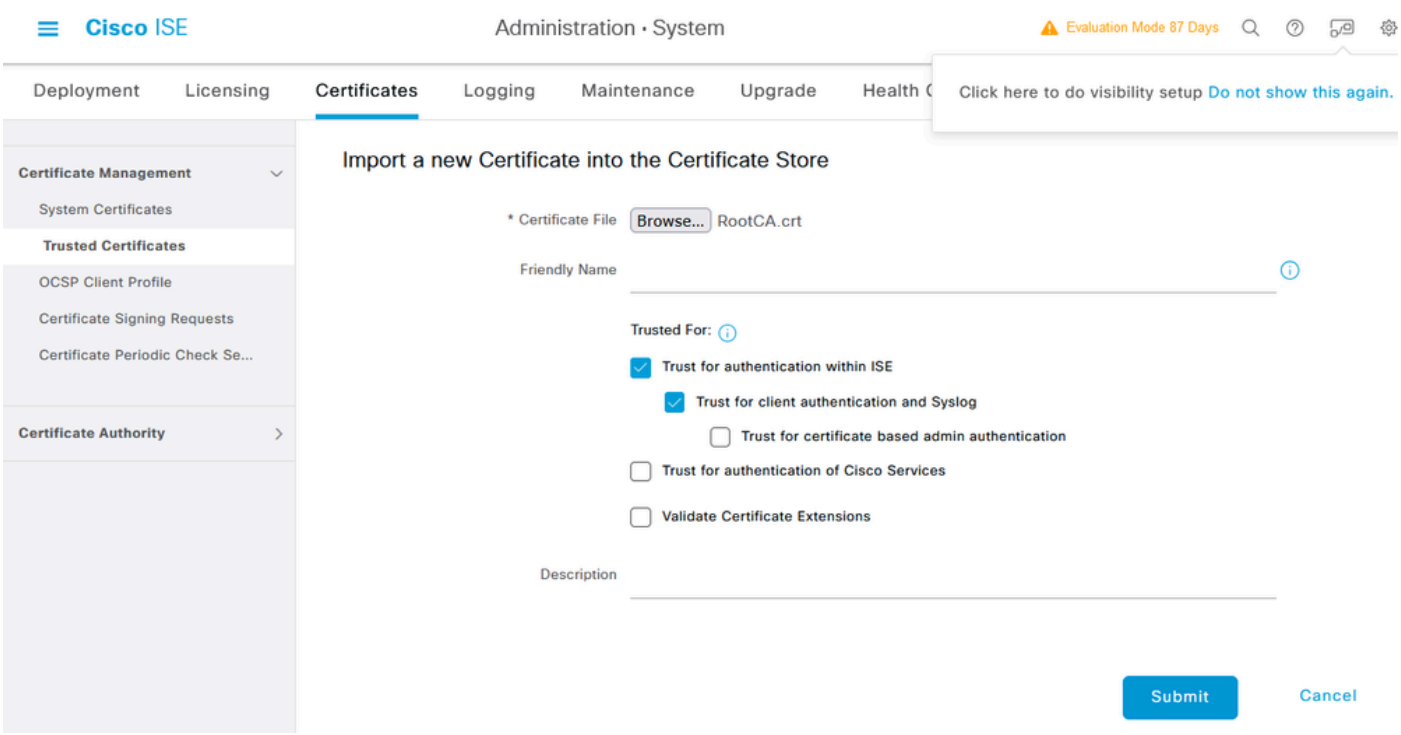
1. Importez le certificat d'autorité de certification racine de la chaîne de certificats ISE vers le

magasin de certificats de confiance.

2. Accédez à Administration>Système>Certificats>Certificats approuvés.

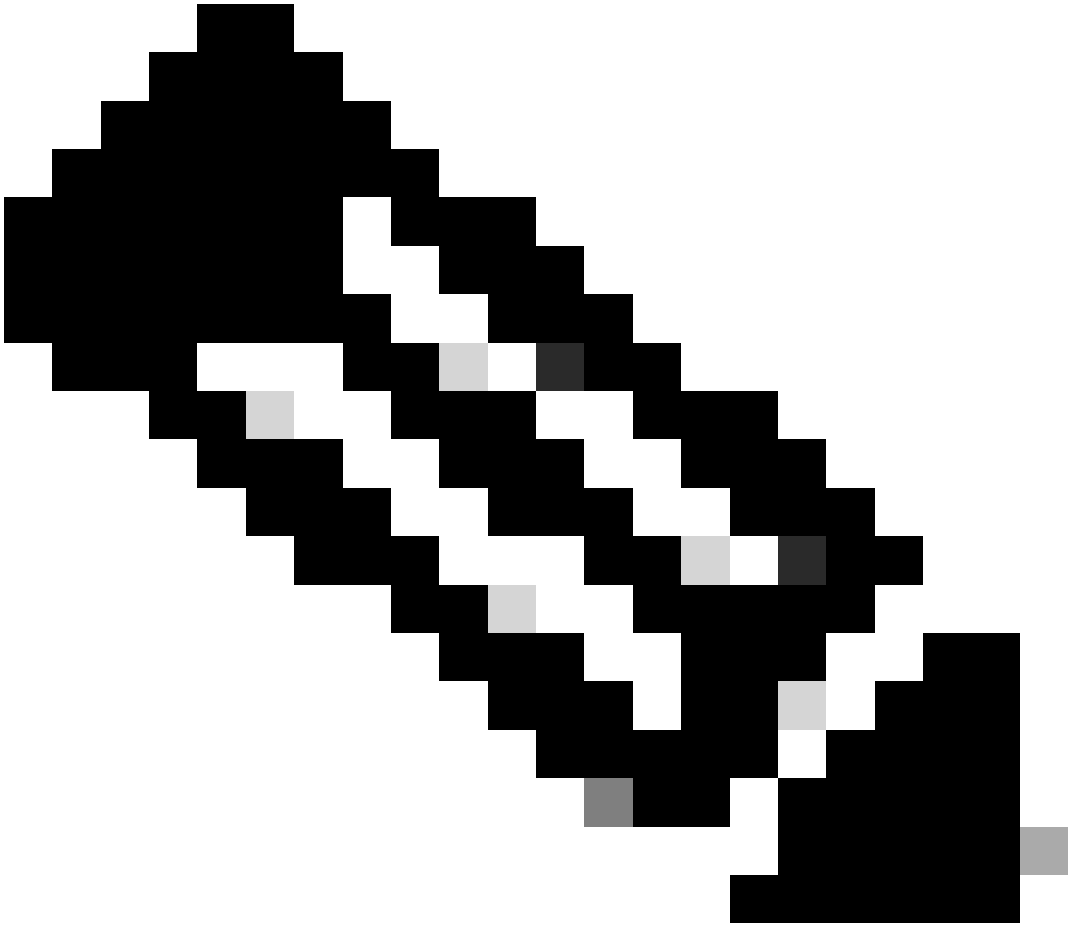
3. Cliquez sur Parcourir et sélectionnez le fichier Root.crt.

4. Cochez les cases Trust for authentication within ISE ainsi que Trust for client authentication et Syslog, puis cliquez sur Submit :

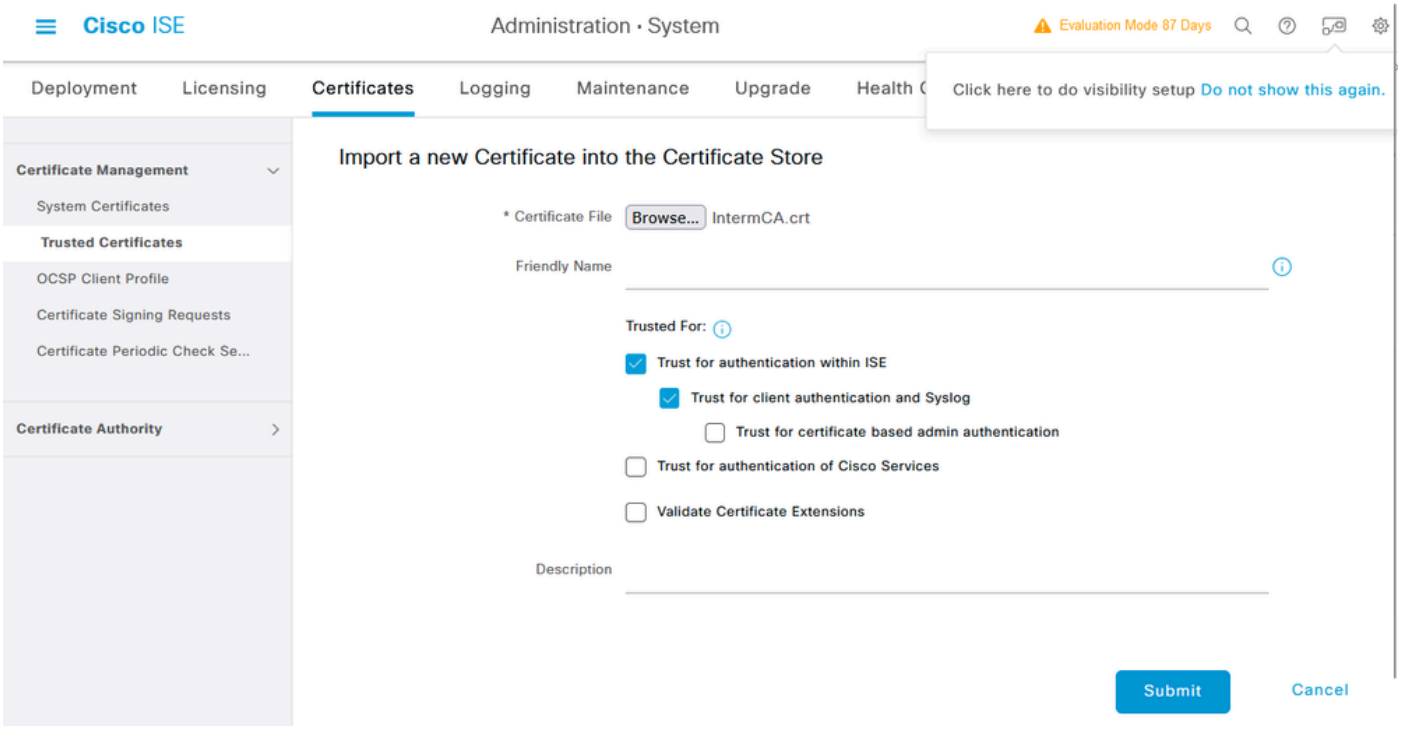


Boîte de dialogue Importation de certificat CA racine ISE

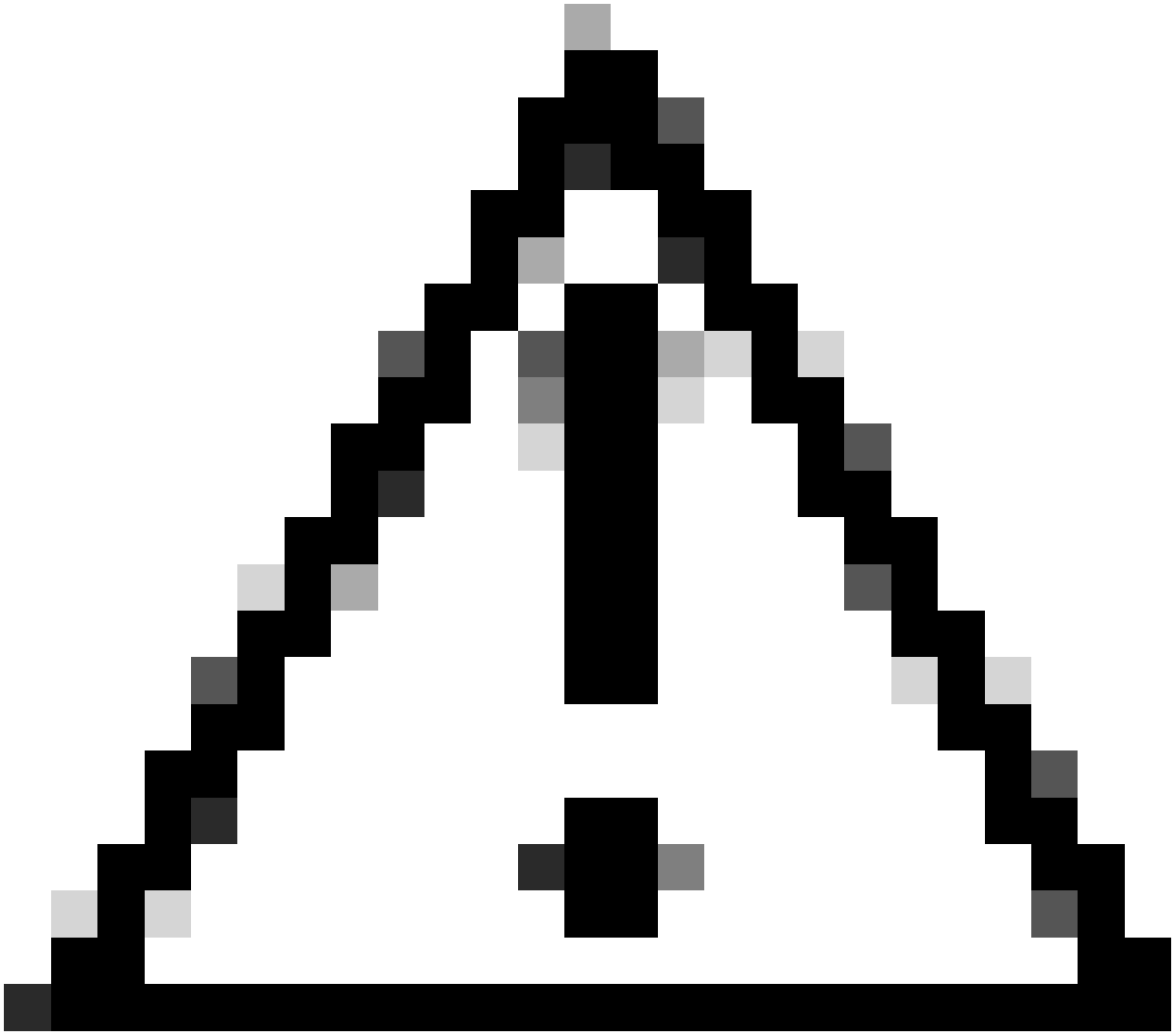
Faites de même pour le certificat intermédiaire s'il existe.



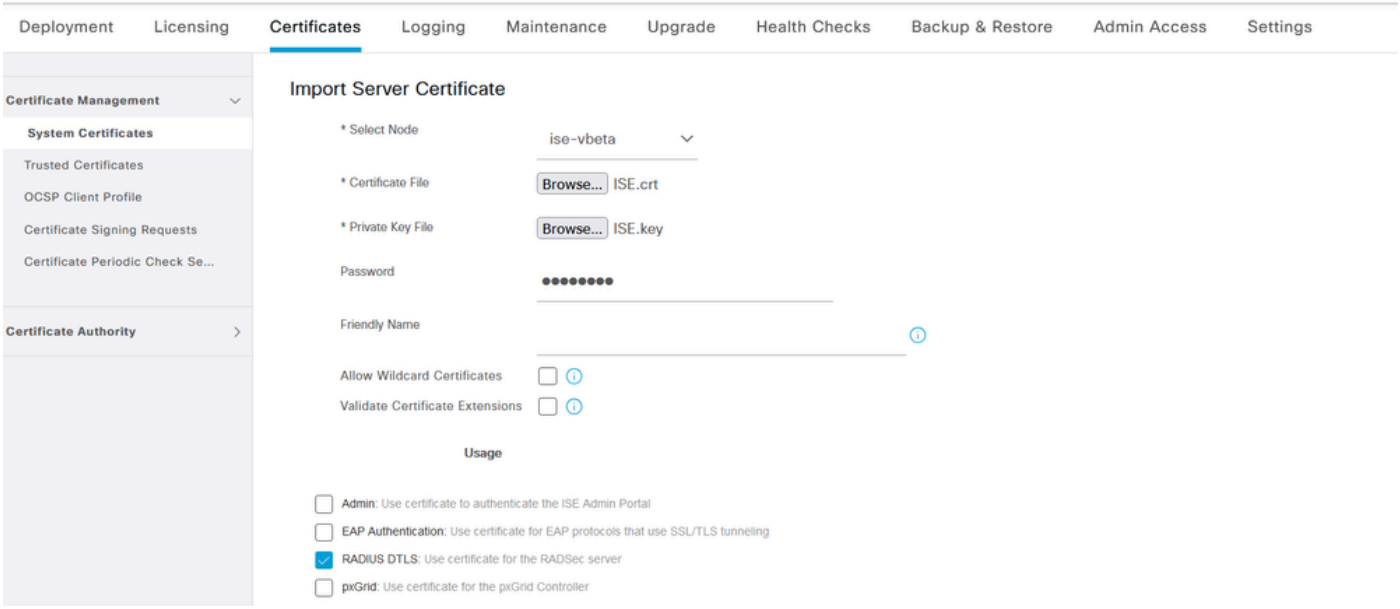
Remarque : répétez les étapes pour tout certificat CA faisant partie de la chaîne de validation de certificat ISE. Commencez toujours par le certificat d'autorité de certification racine et terminez toujours par le certificat d'autorité de certification intermédiaire le plus bas de la chaîne.



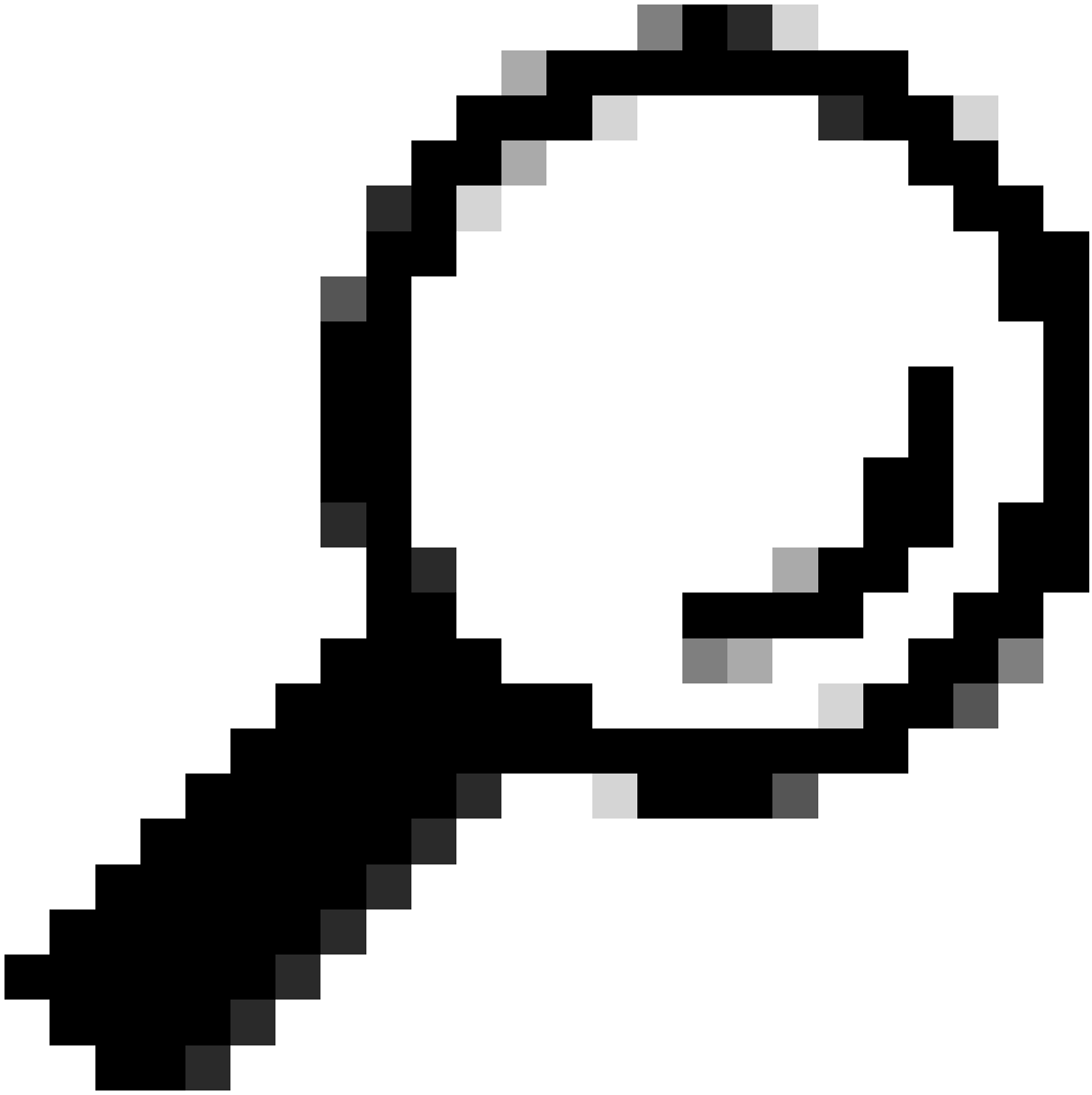
Boîte de dialogue Importation de certificat CA intermédiaire ISE



Attention : si le certificat ISE et le certificat WLC sont émis par des CA différentes, vous devez également importer tous les certificats CA qui appartiennent à la chaîne de certificats WLC. ISE n'accepte pas le certificat WLC sur l'échange de certificats DTLS tant que vous n'avez pas importé ces certificats CA.



Menu d'importation de certificat de périphérique ISE

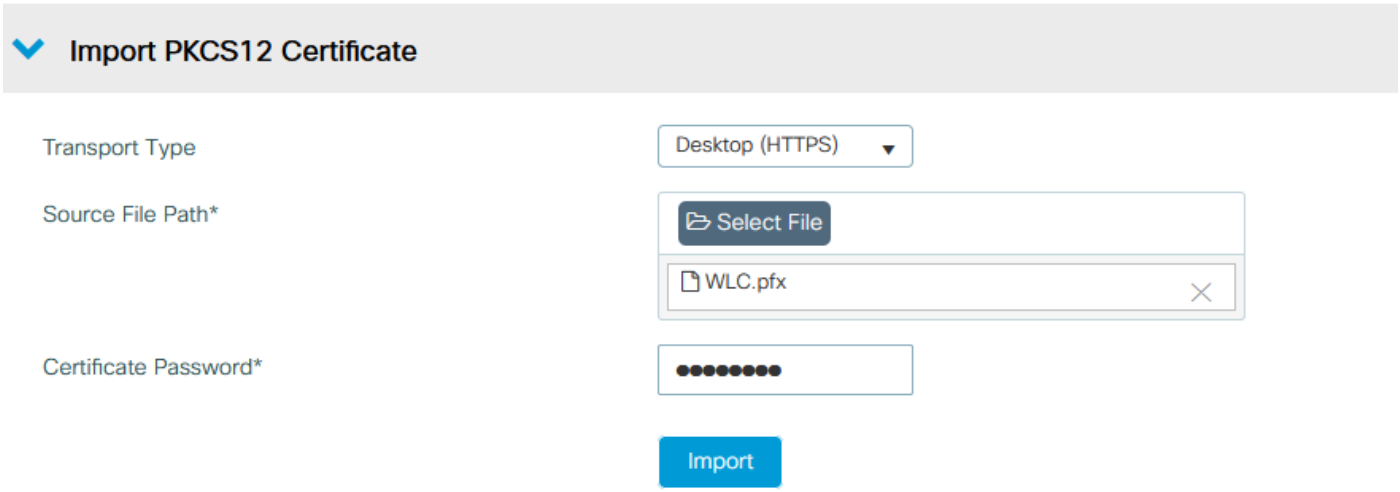


Conseil : vous n'avez besoin d'importer le certificat de périphérique ISE qu'à cette étape. Ce certificat est celui qu'ISE échange pour établir le tunnel DTLS. Il n'est pas nécessaire d'importer le certificat de périphérique WLC et la clé privée car le certificat WLC est vérifié avec l'utilisation des certificats CA importés précédemment.

Importer des certificats vers WLC

1. Accédez à Configuration > Security > PKI Management sur le WLC et allez à l'onglet Add Certificate.
2. Cliquez sur la liste déroulante Import PKCS12 Certificate et définissez le type de transport sur Desktop (HTTPS).
3. Cliquez sur le bouton Select File et sélectionnez le fichier .pfx que vous avez préparé précédemment.

4. Tapez le mot de passe d'importation et enfin cliquez sur Importer.

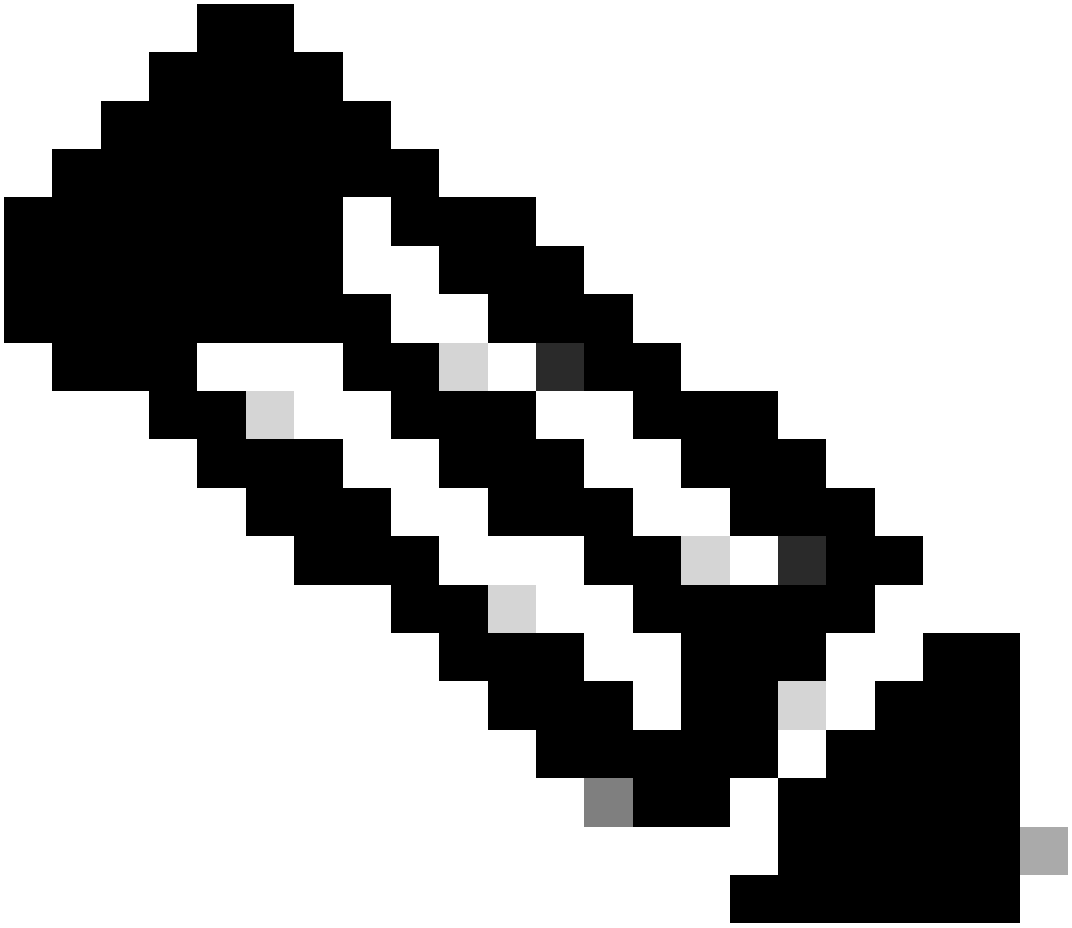


Boîte de dialogue Importation de certificat WLC

Pour des informations détaillées sur le processus d'importation, référez-vous à Générer et télécharger des certificats CSR sur des WLC Catalyst 9800.

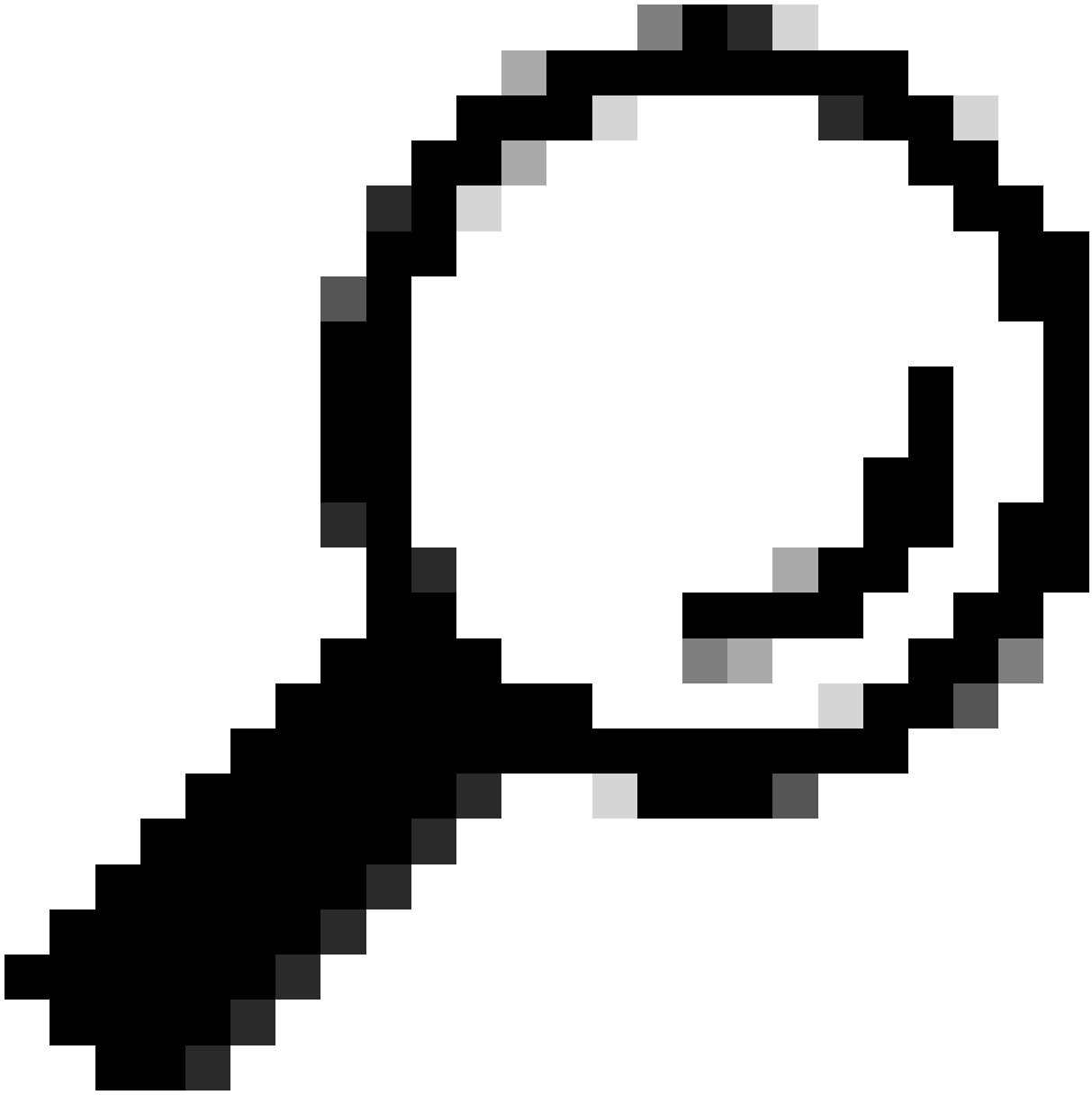
Désactivez la vérification de révocation dans chaque point de confiance créé automatiquement si le WLC n'a pas de liste de révocation de certificats qu'il peut vérifier sur le réseau :

```
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
```



Remarque : si vous avez créé une autorité de certification multiniveau sur OpenSSL avec le document Configure Multi-level CA on OpenSSL to Generate Cisco IOS XE Certificates, vous devez désactiver la vérification de révocation car aucun serveur CRL n'est créé.

L'importation automatisée crée les points de confiance nécessaires pour contenir le certificat WLC et ses certificats CA.



Conseil : Si les certificats WLC ont été émis par la même CA que les certificats ISE, vous pouvez utiliser les mêmes points de confiance créés automatiquement à partir de l'importation de certificat WLC. Il n'est pas nécessaire d'importer les certificats ISE séparément.

Si le certificat WLC est émis par une autorité de certification différente du certificat ISE, vous devez également importer les certificats d'autorité de certification ISE dans le WLC pour que le WLC puisse faire confiance au certificat de périphérique ISE.
Créez un nouveau point de confiance pour l'autorité de certification racine et importez l'autorité de certification racine ISE :

```
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-----
```

Importez le certificat d'autorité de certification intermédiaire suivant sur la chaîne d'autorité de certification ISE, en d'autres termes, le certificat d'autorité de certification émis par l'autorité de certification racine :

```
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-------
```

Chaque autorité de certification supplémentaire de la chaîne nécessite un point de confiance distinct. Chaque point de confiance de la chaîne doit faire référence au point de confiance qui contient le certificat de l'émetteur du certificat que vous souhaitez importer à l'aide de la commande chain-validation continue **<**Nom du point de confiance de l'émetteur**>**.

Importez autant de certificats CA que votre chaîne CA en contient. Vous avez terminé après avoir importé l'autorité de certification de l'émetteur du certificat de périphérique ISE, prenez note du nom de ce point de confiance.

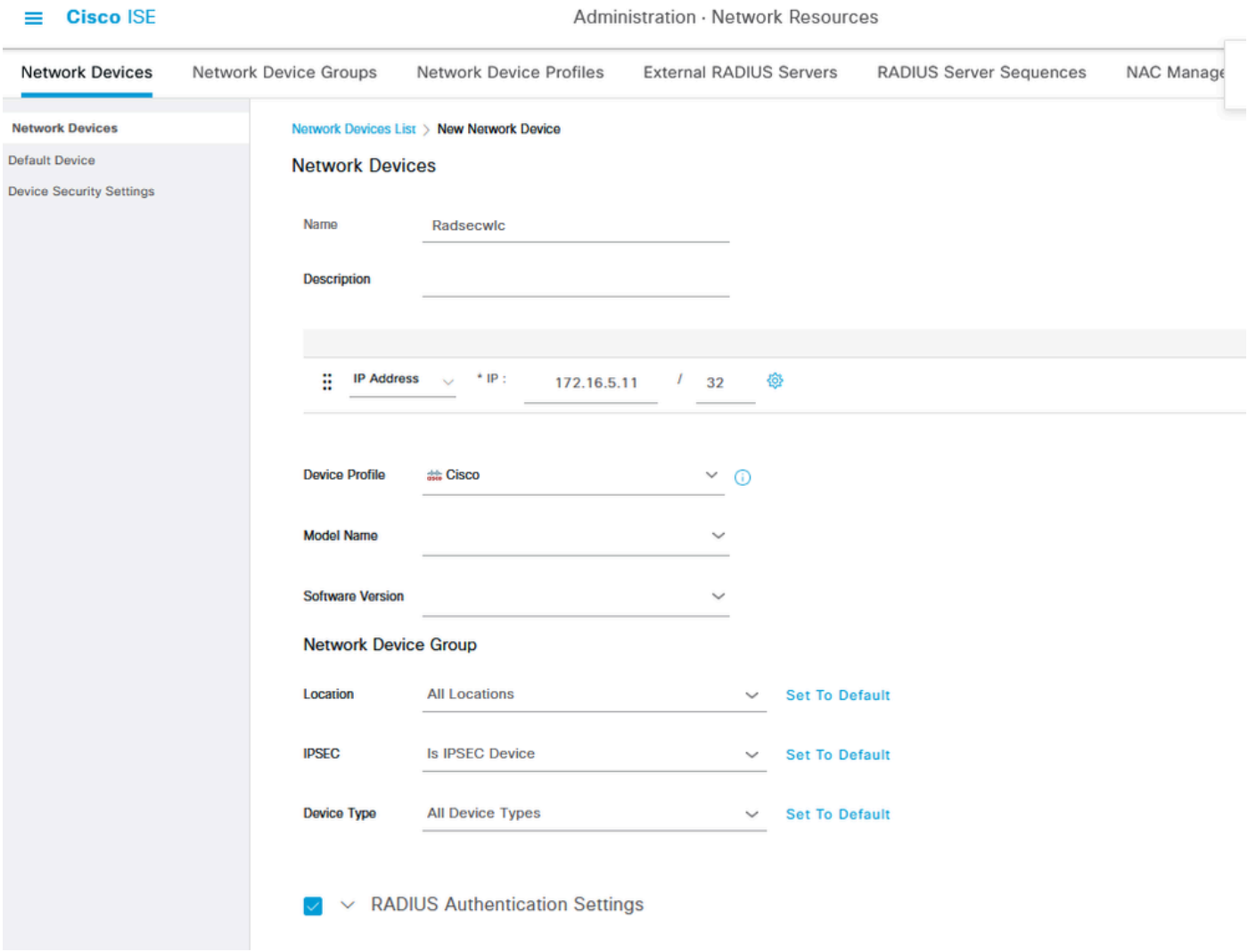
Vous n'avez pas besoin d'importer le certificat de périphérique ISE sur le WLC pour que RADIUS DTLS fonctionne.

Configuration de RADIUS DTLS

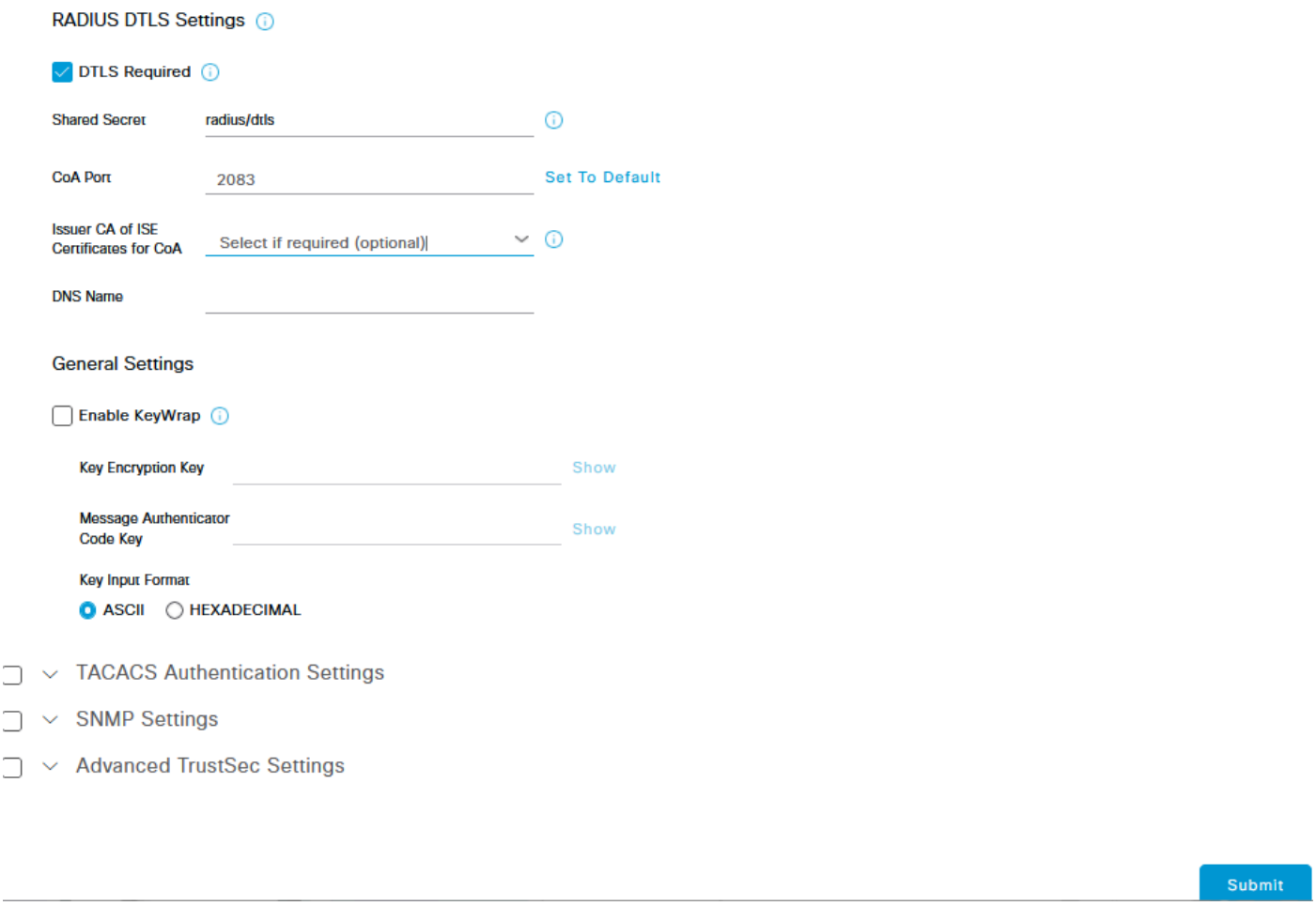
Configuration ISE

Ajoutez le WLC en tant que périphérique réseau dans ISE, pour ce faire, accédez à

Administration>Ressources réseau>Périphériques réseau>Ajouter
Saisissez le nom du périphérique et l'adresse IP de l'interface WLC qui génère le trafic RADIUS. En général, l'adresse IP de l'interface de gestion sans fil. Faites défiler vers le bas et vérifiez RADIUS Authentication Settings ainsi que DTLS Required et cliquez sur Submit :



Nouvelle configuration de périphérique réseau



Paramètres Radius DTLS pour le périphérique réseau sur ISE

Configuration WLC

Définissez un nouveau serveur Radius avec l'adresse IP ISE et le port par défaut pour Radius DTLS. Cette configuration est disponible sur l'interface de ligne de commande uniquement :

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

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

Radius DTLS doit utiliser le secret partagé radius/dtls, le WLC 9800 ignore toute clé configurée autre que celle-ci :

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

Utilisez la commande`dtls trustpoint client`
pour configurer le point de confiance qui contient le certificat de périphérique WLC à échanger pour le tunnel DTLS.
Utilisez la commande`dtls trustpoint server`
pour configurer le point de confiance qui contient l'autorité de certification de l'émetteur pour le certificat de périphérique ISE.

Les noms des points de confiance du client et du serveur sont identiques uniquement si les certificats WLC et ISE sont émis par la même autorité de certification :

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

Configurez le WLC pour vérifier l'un des noms de substitution de sujet (SAN) qui est présent sur le certificat ISE. Cette configuration doit correspondre exactement à l'un des SAN présents dans le champ SAN du certificat.

Le WLC 9800 n'effectue pas de correspondance basée sur une expression régulière pour le champ SAN. Cela signifie, par exemple, que la commande `dtls match-server-identity hostname *.example.com` pour un certificat générique qui a *.example.com sur son champ SAN est correcte, mais la même commande pour un certificat qui contient www.example.com sur le champ SAN ne l'est pas.

Le WLC ne vérifie pas ce nom par rapport à un serveur de noms :

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

Créez un nouveau groupe de serveurs pour utiliser le nouveau Radius DTLS pour l'authentification :

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

À partir de ce moment, vous pouvez utiliser ce groupe de serveurs comme vous utilisez n'importe quel autre groupe de serveurs sur le WLC. Référez-vous à Configurer l'authentification 802.1X sur la gamme de contrôleurs sans fil Catalyst 9800 pour utiliser ce serveur pour l'authentification du client sans fil.

Vérifier

Vérifier les informations de certificat

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

```
openssl x509 -in


      -text -noout
```

Elle affiche les informations complètes du certificat. Cela est utile pour déterminer l'autorité de certification émettrice d'un certificat donné ou si les certificats contiennent les unités EKU et SAN requises :

```
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:
```

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

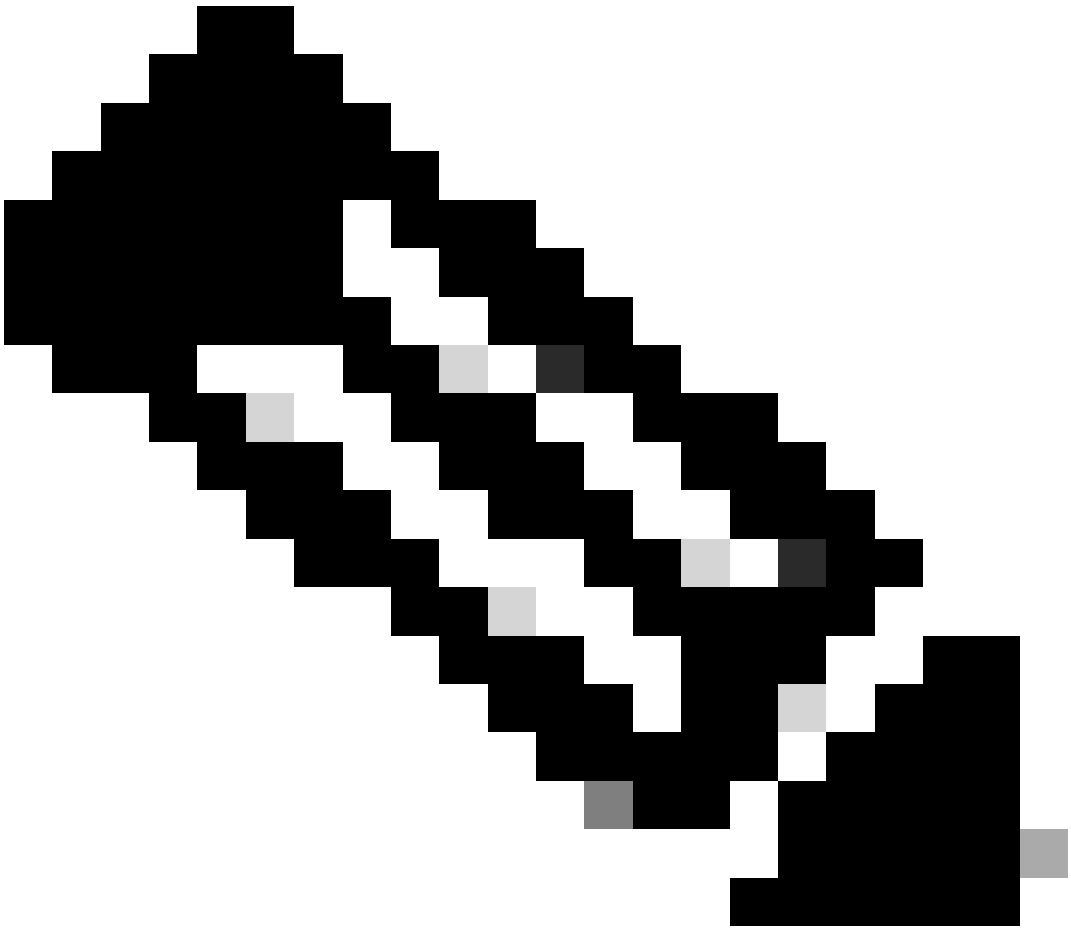
Effectuer une authentification de test

À partir du WLC, vous pouvez tester la fonctionnalité Radius DTLS avec la commande `test aaa group`

new-code

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

```
USER ATTRIBUTES

username              0   "testuser"
```



Remarque : une sortie de rejet d'accès sur la commande de test signifie que le WLC a reçu un message RADIUS de rejet d'accès, auquel cas RADIUS DTLS fonctionne. Cependant, il peut également indiquer une défaillance de l'établissement du tunnel DTLS. La commande test ne distingue pas les deux scénarios. Reportez-vous à la section Dépannage pour identifier un problème.

Dépannage

Pour examiner la cause d'un échec d'authentification, vous pouvez activer ces commandes avant d'effectuer un test d'authentification.

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

Voici le résultat d'une authentification réussie avec des débogages activés :

```
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
```

CA inconnue signalée par le WLC

Lorsque le WLC ne peut pas valider les certificats fournis par ISE, il ne parvient pas à créer le tunnel DTLS et les authentifications échouent.
Voici un exemple des messages de débogage présentés lorsque c'est le cas :

```
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:
```

Pour le corriger, assurez-vous que l'identité configurée sur le WLC correspond exactement à l'un des SAN inclus sur le certificat ISE :

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

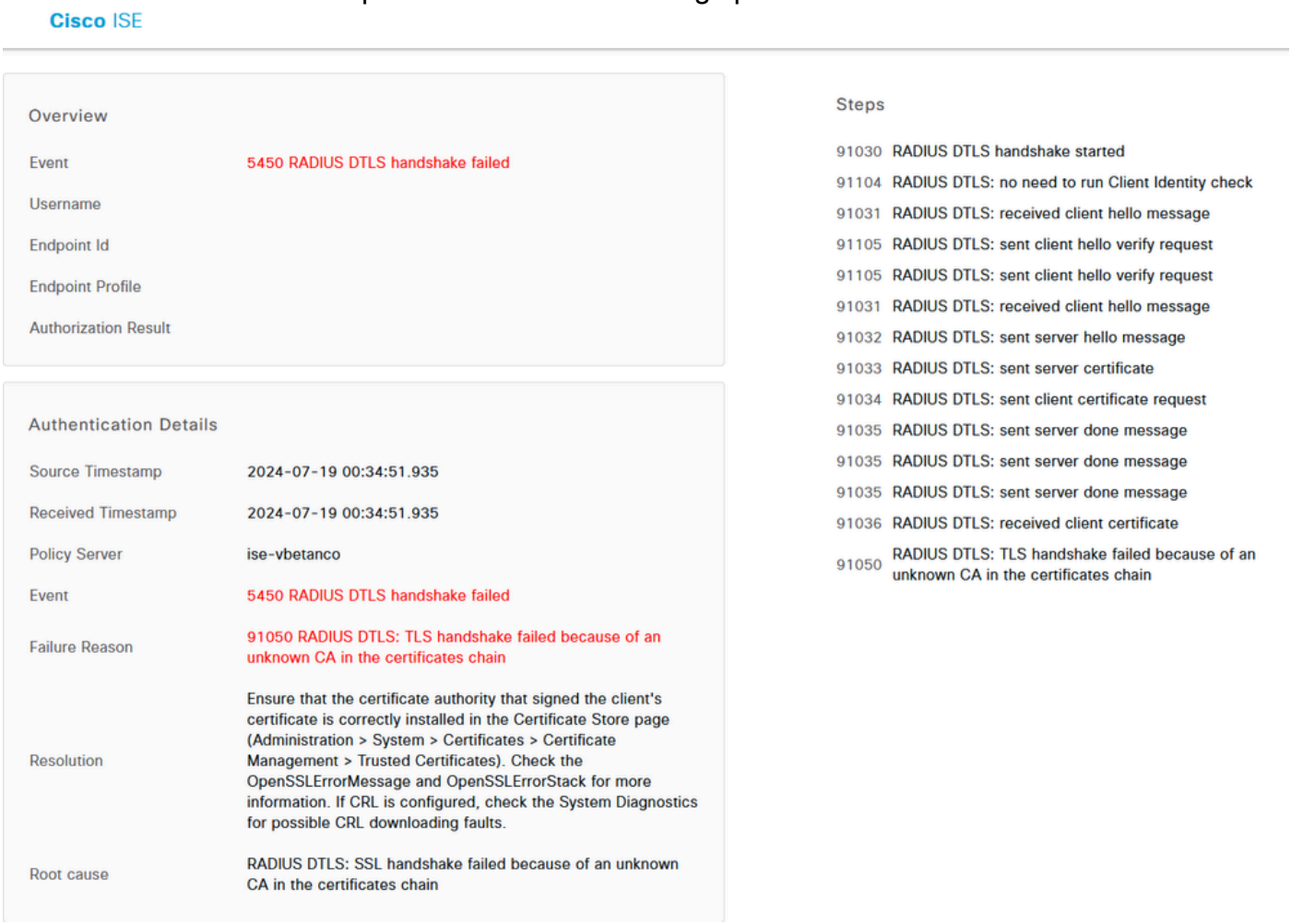
```
      9800(config)#dtls match-server-identity hostname
```

Assurez-vous que la chaîne de certificats de l'autorité de certification est correctement importée sur le contrôleur et que le **dtls trustpoint server**

configuration uses the Issuer CA trustpoint.

Autorité de certification inconnue signalée par ISE

Lorsque ISE ne peut pas valider les certificats fournis par le WLC, il ne parvient pas à créer le tunnel DTLS et les authentifications échouent. Ceci apparaît comme une erreur dans les journaux RADIUS Live. Accédez à Operations>Radius>Live logs pour vérifier.



Le journal en direct ISE signale un échec de connexion DTLS dû à une CA inconnue

Pour le corriger, assurez-vous que les certificats intermédiaires et racine sont tous deux activés, cochez les cases Approuver pour l'authentification client et Syslog sous Administration>Système>Certificats>Certificats approuvés.

La vérification de révocation est en place

Lorsque les certificats sont importés dans le WLC, le contrôle de révocation est activé pour les points de confiance nouvellement créés. Cela fait que le WLC essaie de rechercher une liste de révocation de certificats qui n'est pas disponible ou accessible et qui échoue à la vérification du certificat.
Assurez-vous que chaque point de confiance du chemin de vérification des certificats contient la commande revocation-check none .

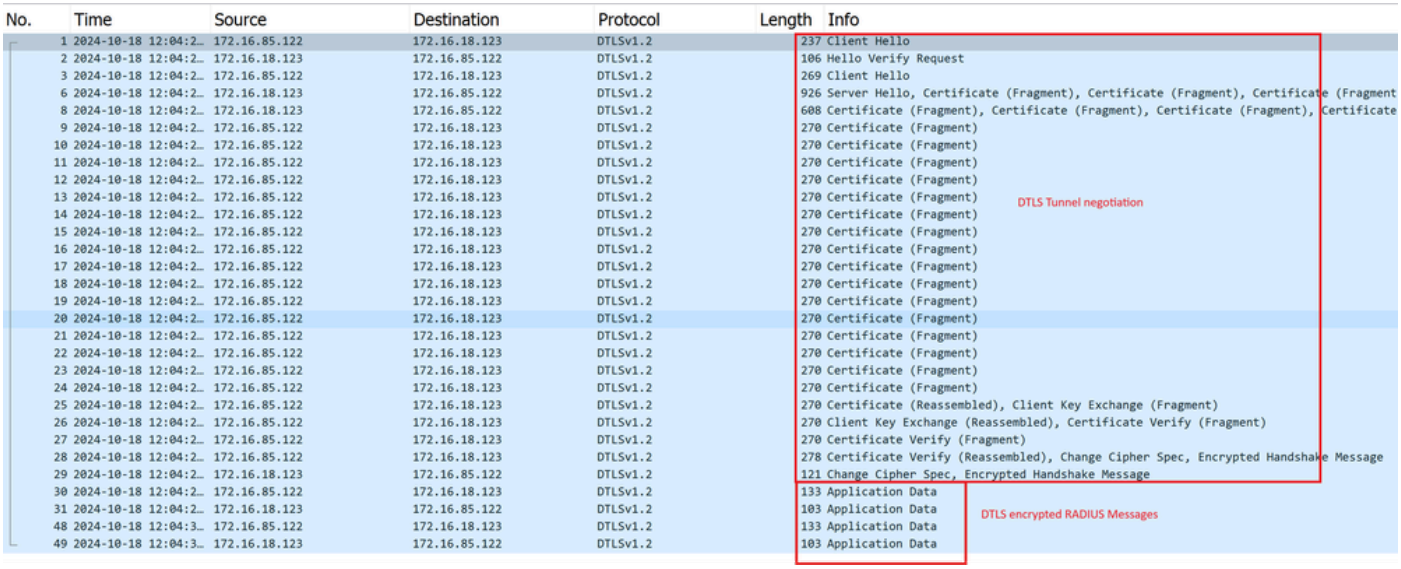
```
Jul 17 21:50:39.064: RADIUS_RADSEC_HASH_KEY_MATCH: hashkey1(0) matches hashkey2(0) TRUE
Jul 17 21:50:39.064: RADIUS_RADSEC_HASH_KEY_GET_CTX: radius radsec  sock_ctx(0x780FB0715978:0) get for
Jul 17 21:50:39.064: RADIUS_RADSEC_PROCESS_SOCK_EVENT: Handle socket event for TLS handshake(172.16.18.
Jul 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
```

Dépannage de l’établissement du tunnel DTLS sur la capture de paquets

Le WLC 9800 offre la fonctionnalité Embedded Packet Capture (EPC) qui vous permet de capturer tout le trafic envoyé et reçu pour une interface donnée. ISE offre une fonctionnalité similaire appelée TCP dump pour surveiller le trafic entrant et sortant. Utilisés en même temps, ils vous permettent d'analyser le trafic d'établissement de session DTLS du point de vue des deux périphériques.

Reportez-vous au Guide de l'administrateur de Cisco Identity Services Engine pour connaître les étapes détaillées de configuration du vidage TCP sur ISE. Reportez-vous également à la section Dépannage des contrôleurs LAN sans fil Catalyst 9800 pour des informations sur la configuration de la fonctionnalité EPC sur le WLC.

Voici un exemple d'établissement réussi d'un tunnel DTLS.



Capture de paquets d'une négociation de tunnel DTLS RADIUS et messages chiffrés

Les captures de paquets montrent comment l’établissement du tunnel DTLS se produit. En cas de problème avec la négociation, dû à la perte de trafic entre les périphériques ou aux paquets d'alerte chiffrés DTLS, la capture de paquets vous aide à identifier le problème.

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

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