Configuración de Radius DTLS en ISE y 9800 WLC

Contenido

- Introducción
- Background
- Prerequisites
  - Requirements
  - Componentes Utilizados
- Configurar
  - Overview
  - Opcional: cree un certificado de dispositivo DTLS RADIUS de WLC e ISE
    - Agregar secciones de configuración en el archivo openssl.cnf
    - Crear certificado de dispositivo WLC
    - Crear certificado de dispositivo ISE
  - Importar certificados a dispositivos
    - Importar certificados a ISE
    - Importar certificados a WLC
  - Configuración de RADIUS DTLS
    - Configuración de ISE
    - Configuración de WLC
- Verificación
  - Verificar información del certificado
  - Realizar autenticación de prueba
- Troubleshoot
  - CA desconocida informada por WLC
  - CA desconocida notificada por ISE
  - La comprobación de revocación está activa
  - Resolución de problemas de establecimiento de túnel DTLS en captura de paquetes

Introducción

Este documento describe un método para crear los certificados necesarios para configurar RADIUS DTLS entre ISE y el WLC 9800.

Background

RADIUS DTLS es una forma segura del protocolo RADIUS donde los mensajes RADIUS se envían a través de un túnel de seguridad de la capa de transporte (DTLS) de datos. Para crear este túnel entre el servidor de autenticación y el autenticador, se necesita un conjunto de certificados. Este conjunto de certificados requiere que se establezcan determinadas extensiones

de certificado de uso extendido de claves (EKU), en concreto, la autenticación de cliente en el certificado WLC y la autenticación de servidor, así como la autenticación de cliente para el certificado ISE.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Cómo configurar el 9800 WLC, el punto de acceso (AP) para el funcionamiento básico
- Cómo utilizar la aplicación OpenSSL
- Infraestructura de clave pública (PKI) y certificados digitales

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

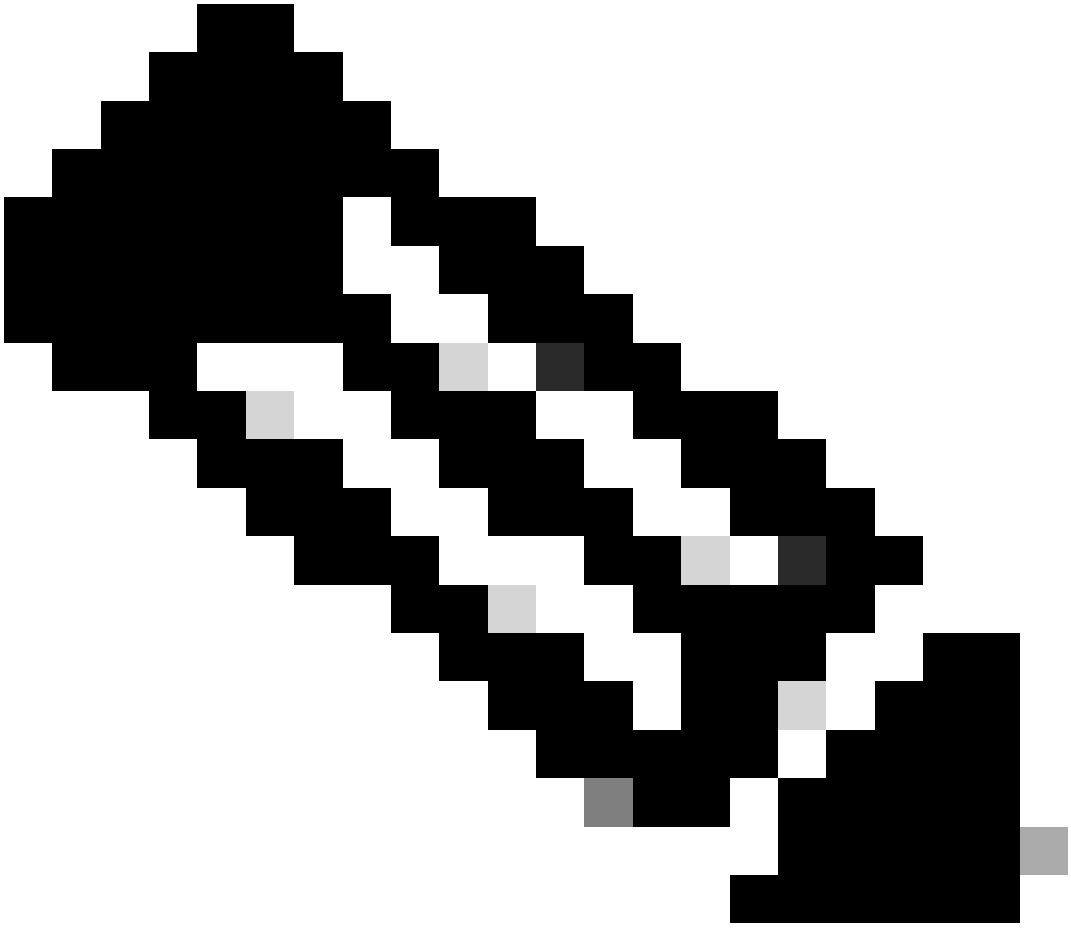
- Aplicación OpenSSL (versión 3.0.2).
- ISE (versión 3.1.0.518)
- 9800 WLC (versión 17.12.3)

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Configurar

Overview

El propósito es crear una entidad emisora de certificados de dos niveles con una CA raíz y una CA intermedia para firmar certificados de extremos. Una vez que se firman los certificados, se importan al WLC e ISE. Finalmente, los dispositivos se configuran para realizar la autenticación RADIUS DTLS con esos certificados.



Nota: Este documento utiliza comandos específicos de Linux para crear y organizar archivos. Los comandos se explican para que pueda realizar la misma acción en otros sistemas operativos donde OpenSSL esté disponible.

Opcional: cree un certificado de dispositivo DTLS RADIUS de WLC e ISE

El protocolo RADIUS DTLS necesita intercambiar certificados entre ISE y WLC para crear el túnel DTLS. Si aún no tiene certificados válidos, puede crear una CA local para generar los certificados, consulte Configuración de una Autoridad de Certificación Multinivel en OpenSSL para Generar Certificados Compatibles con Cisco IOS® XE y realice los pasos descritos en el documento desde el principio hasta el final del paso Crear certificado de CA intermedio.

Agregar secciones de configuración en el archivo openssl.cnf

Abra el archivo de configuración openssl.cnf y, en la parte inferior, copie y pegue las secciones de WLC e ISE utilizadas para generar una Solicitud de firma de certificado (CSR) válida.

Las secciones ISE_device_req_ext y WLC_device_req_ext señalan cada una a una lista de SAN que se incluirán en el 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
```

Como medida de seguridad, la CA reemplaza cualquier SAN presente en una CSR para firmarla de modo que los dispositivos no autorizados no puedan recibir un certificado válido para un nombre que no se les permite utilizar. Para volver a agregar las SAN al certificado firmado, utilice el parámetro subjectAltName para señalar a la misma lista de SAN que las utilizadas para la generación de CSR.

ISE requiere tanto serverAuth como clientAuth EKU presentes en el certificado, mientras que el WLC sólo necesita clientAuth. Se agregan al certificado firmado con el parámetro extendedKeyUsage.

Copie y pegue las secciones utilizadas para el certificado en la parte inferior del archivo 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
```

Crear certificado de dispositivo WLC

Cree una nueva carpeta para almacenar certificados WLC en la máquina que tiene OpenSSL instalado dentro de la carpeta de certificados de CA intermedia llamada IntermCA.db.certs. La nueva carpeta se llama WLC:

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

Modifique los parámetros DNS en la sección [WLC_alt_names] del archivo openssl.cnf. Cambie los nombres de ejemplo proporcionados para los valores deseados. Estos valores completan el campo SANs del certificado WLC:

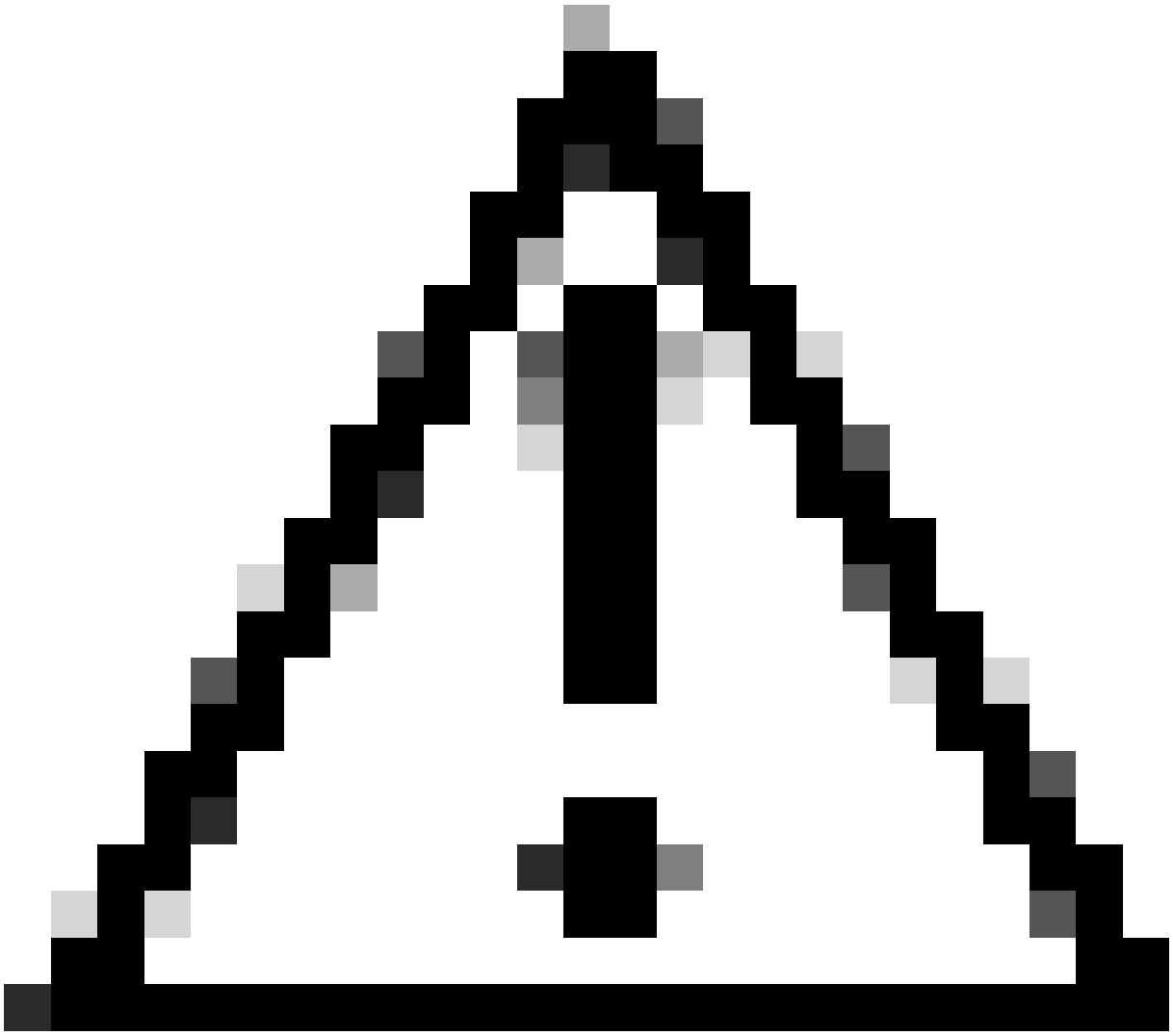
```
[WLC_alt_names]
DNS.1   = WLC.example.com    <-----Change the values after the equals sign
DNS.2   = WLC2.example.com   <-----Change the values after the equals sign
```

Cree la clave privada WLC y el WLC CSR con información de la sección WLC_device_req_ext para las SAN:

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

OpenSSL abre una solicitud interactiva para que introduzca los detalles del nombre distinguido (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
```



Precaución: el nombre común (CN) que proporcione en el mensaje interactivo debe ser idéntico a uno de los nombres de la sección [WLC_alt_names] del archivo openssl.cnf.

Utilice la CA denominada IntermCA para firmar el WLC CSR denominado WLC.csr con las extensiones definidas en [WLC_cert] y almacenar el certificado firmado dentro de ./IntermCA/IntermCA.db.certs/WLC. El certificado del dispositivo WLC se llama WLC.crt:

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

9800 WLC necesita que el certificado esté en formato pfx para importarlo. Cree un nuevo archivo que contenga la cadena de CAs que firmaron el certificado WLC, esto se llama un archivo cert:

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

Para crear su archivo .pfx ejecute uno de estos comandos según la versión del WLC.

Para versiones anteriores a 17.12.1:

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

Para la versión 17.12.1 o posterior:

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

Crear certificado de dispositivo ISE

Cree una nueva carpeta para almacenar certificados ISE en el equipo que tiene OpenSSL instalado dentro de la carpeta de certificados de CA intermedia denominada IntermCA.db.certs. La nueva carpeta se llama ISE:

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

Modifique los parámetros DNS en la sección [ISE_alt_names] del archivo openssl.cnf. Cambie los nombres de ejemplo proporcionados para sus valores deseados, estos valores completan el campo SANs del certificado 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
```

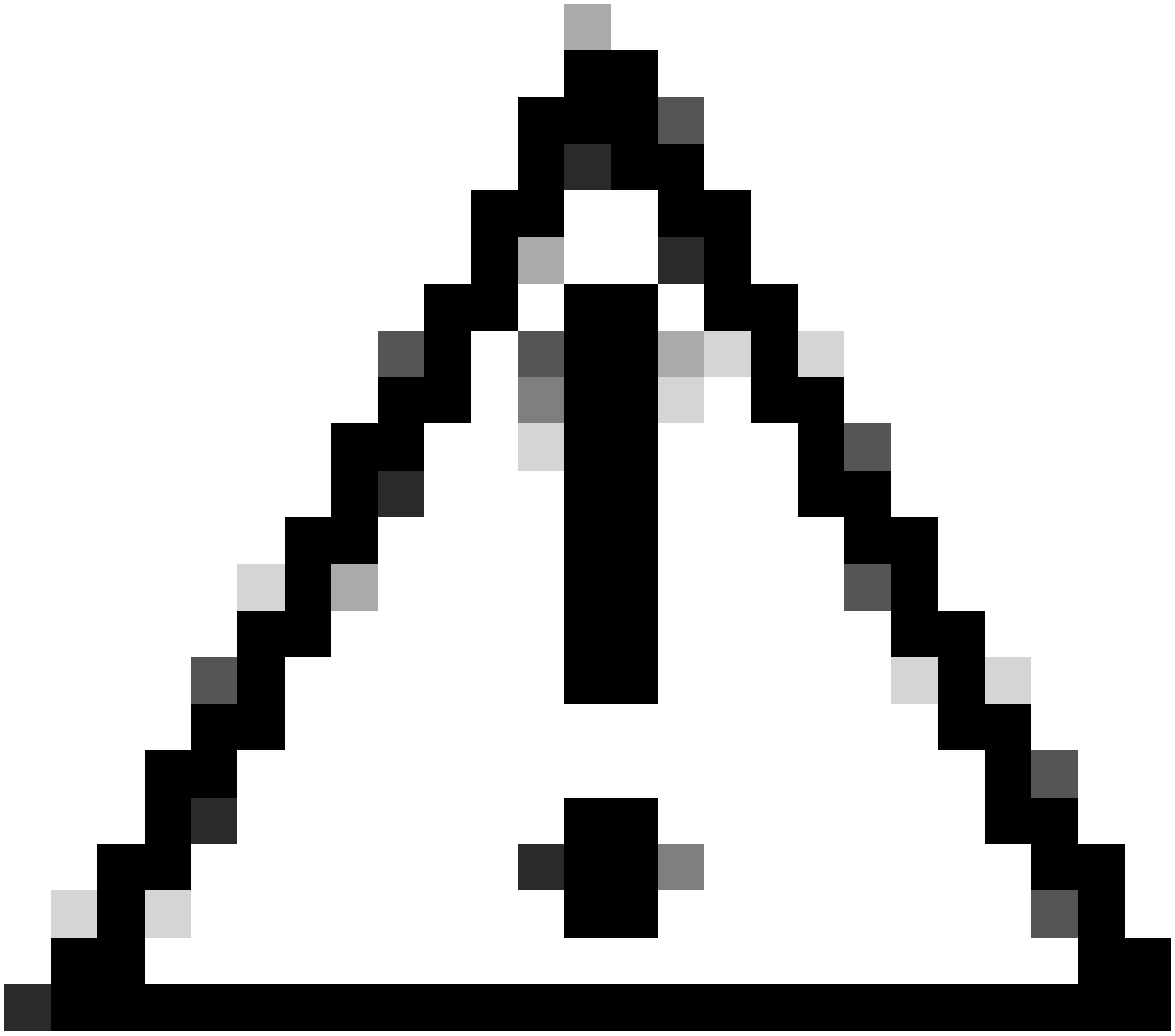
Cree la clave privada de ISE e ISE CSR con información de la sección ISE_device_req_ext para redes SAN:

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

OpenSSL abre una solicitud interactiva para que introduzca los detalles del nombre distinguido (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
```

Solicitud interactiva de nombre distintivo de certificado ISE



Precaución: el CN que proporcione en el mensaje interactivo debe ser exactamente el mismo que uno de los Nombres de la sección [ISE_alt_names] del archivo openssl.cnf.

Utilice la CA denominada IntermCA para firmar la CSR de ISE denominada ISE.csr con las extensiones definidas en [ISE_cert] y almacenar el certificado firmado dentro de ./IntermCA/IntermCA.db.certs/WLC. El certificado del dispositivo ISE se denomina ISE.crt:

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

Importar certificados a dispositivos

Importar certificados a ISE

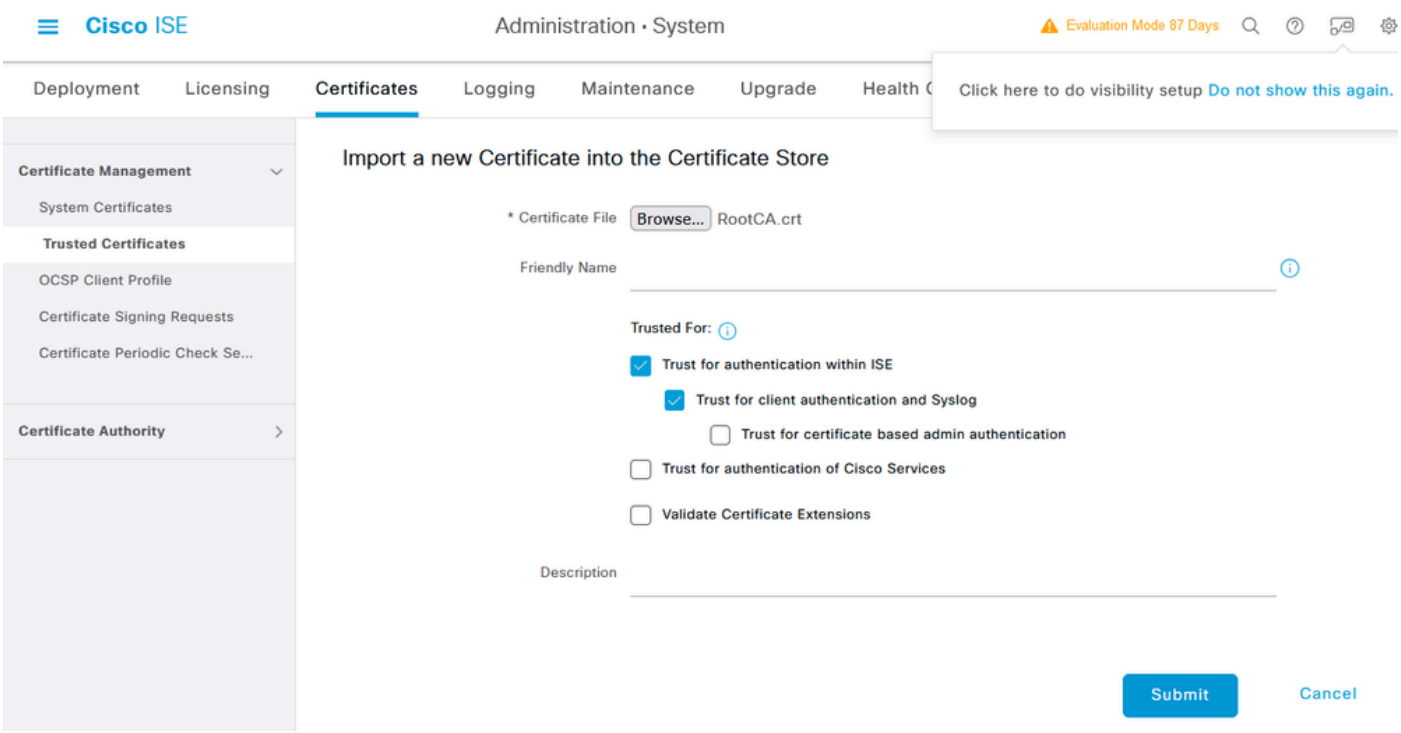
1. Importe el certificado de CA raíz de la cadena de certificados de ISE al almacén de certificados

de confianza.

2. Vaya a Administración>Sistema>Certificados>Certificados de confianza.

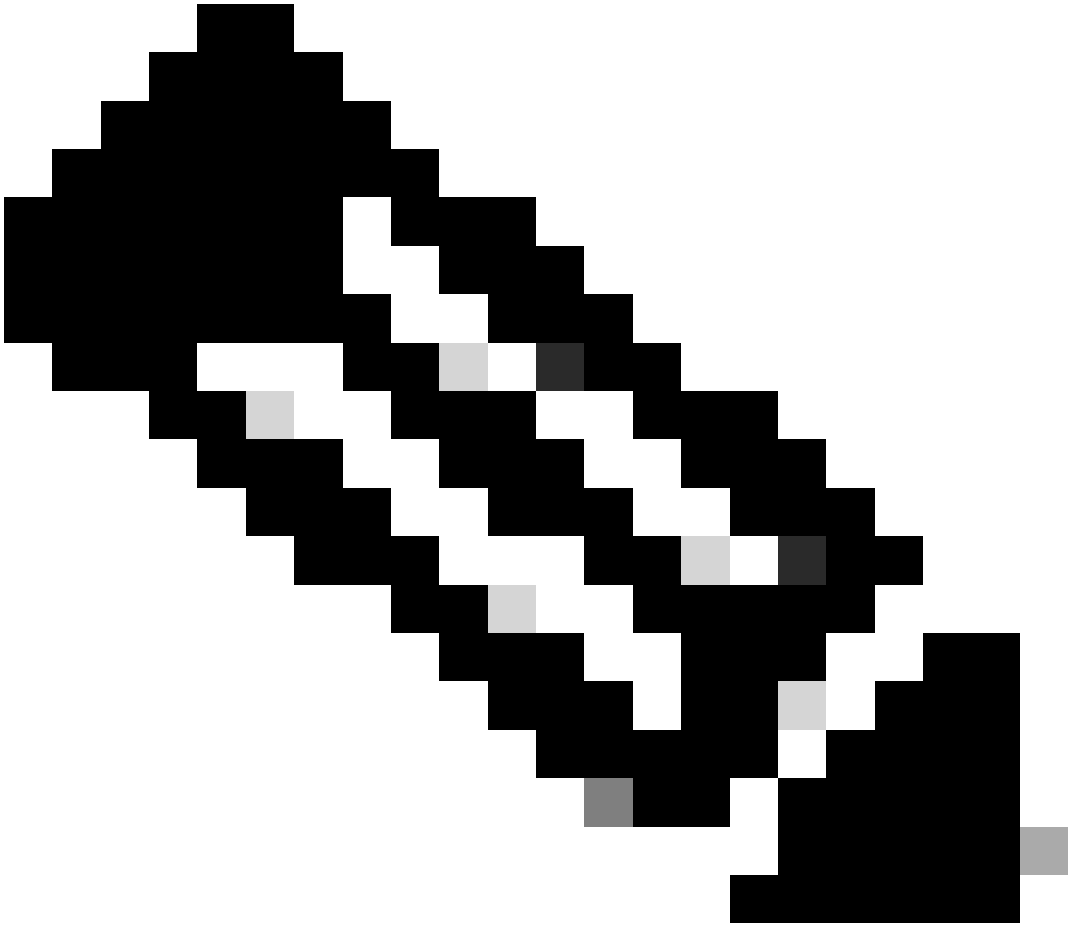
3. Haga clic en Examinar y seleccione el archivo Root.crt.

4. Marque las casillas de verificación Trust for authentication inside ISE, así como Trust for client authentication and Syslog y, a continuación, haga clic en Submit:

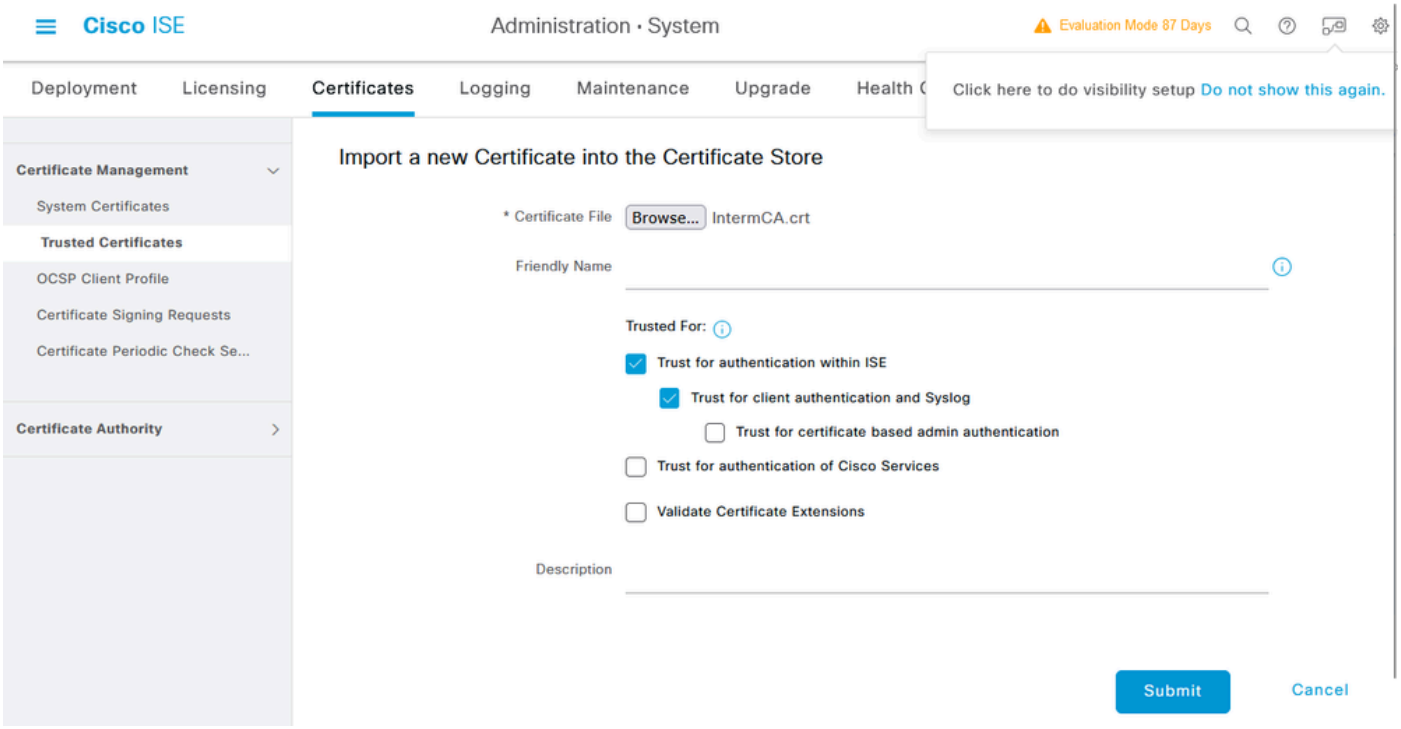


Cuadro de diálogo Importación de certificado de CA raíz de ISE

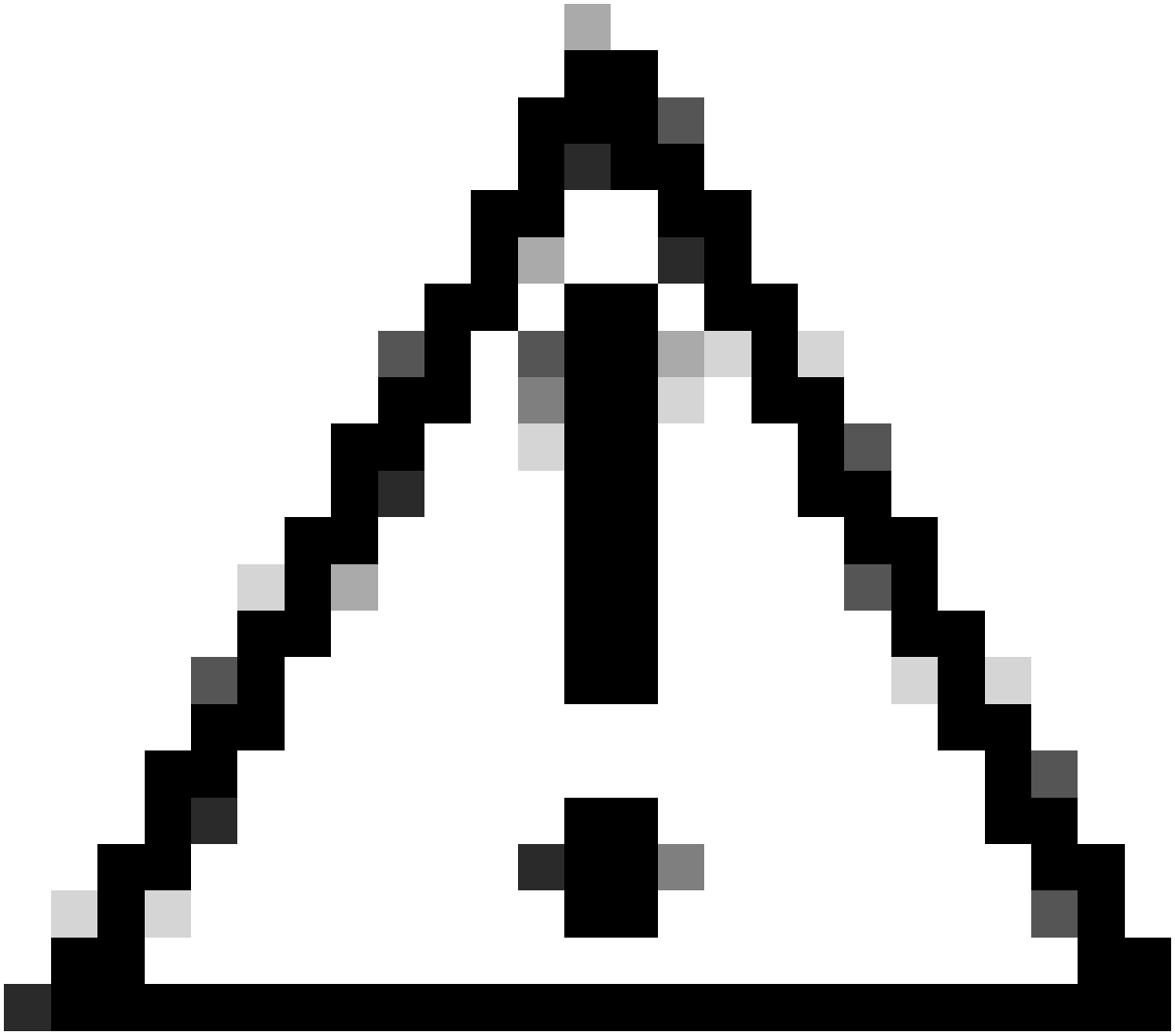
Haga lo mismo con el certificado intermedio si existe.



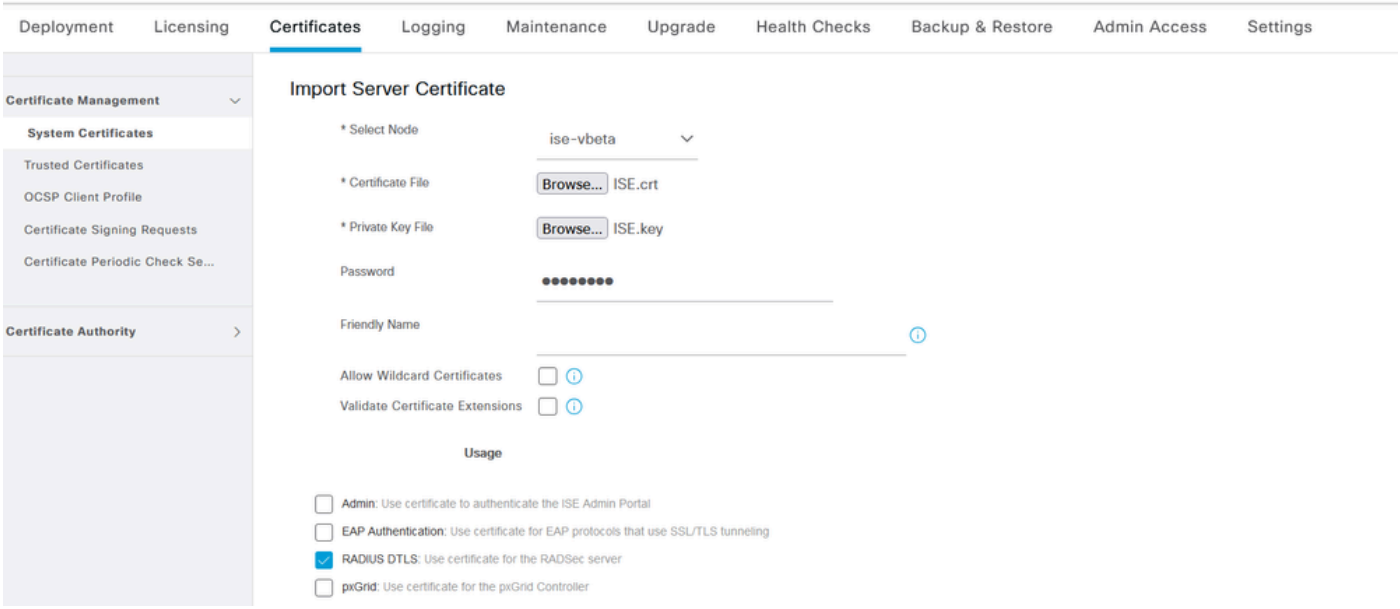
Nota: repita los pasos para cualquier certificado de CA que forme parte de la cadena de validación de certificados de ISE. Comience siempre con el certificado de CA raíz y termine con el certificado de CA intermedia más bajo de la cadena.



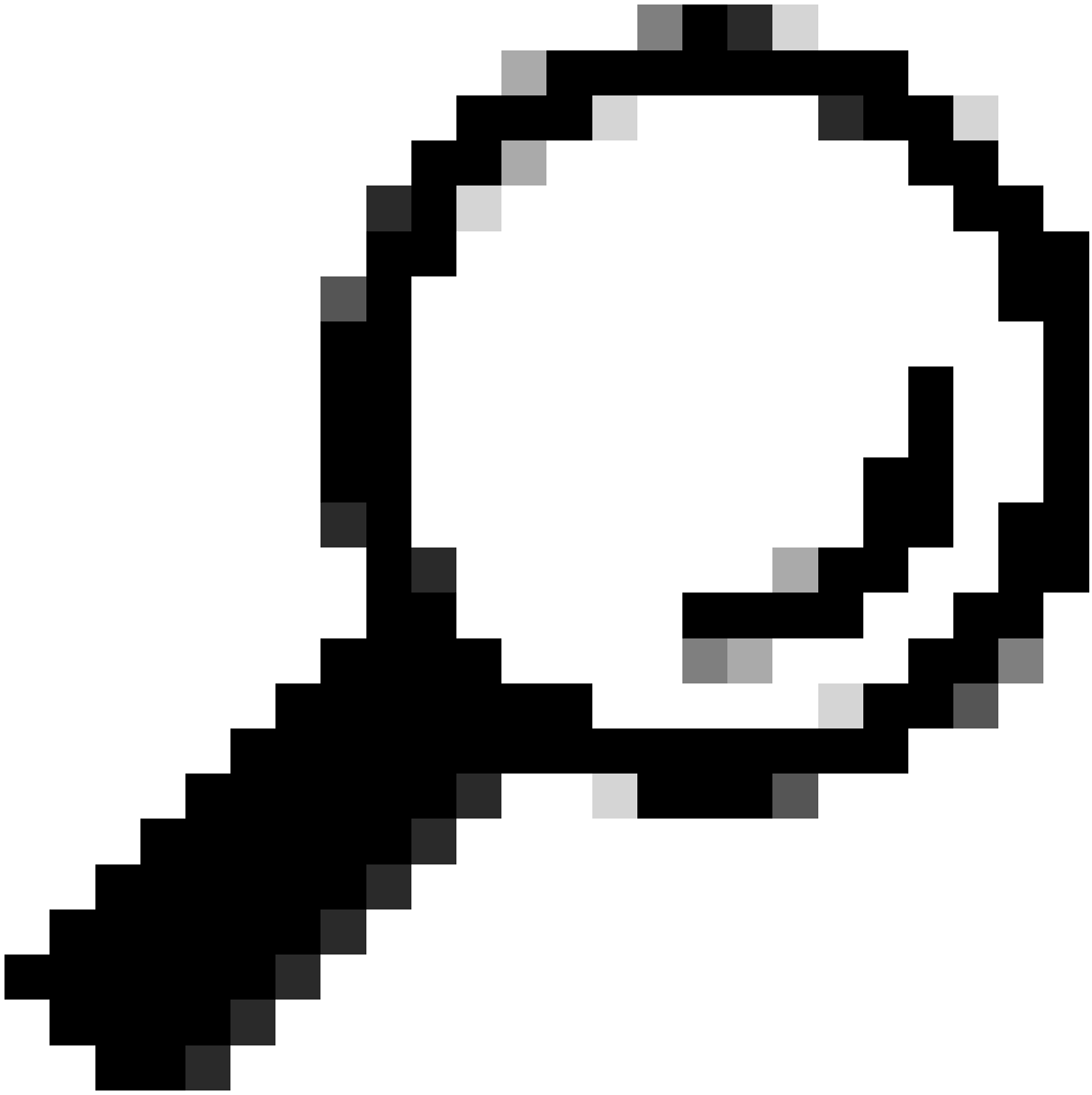
Cuadro de diálogo Importación de certificado de CA intermedia de ISE



Precaución: si el certificado ISE y el certificado WLC son emitidos por diferentes CA, debe importar también todos los certificados CA que pertenezcan a la cadena de certificados WLC. ISE no acepta el certificado WLC en el intercambio de certificados DTLS hasta que se importan dichos certificados de CA.



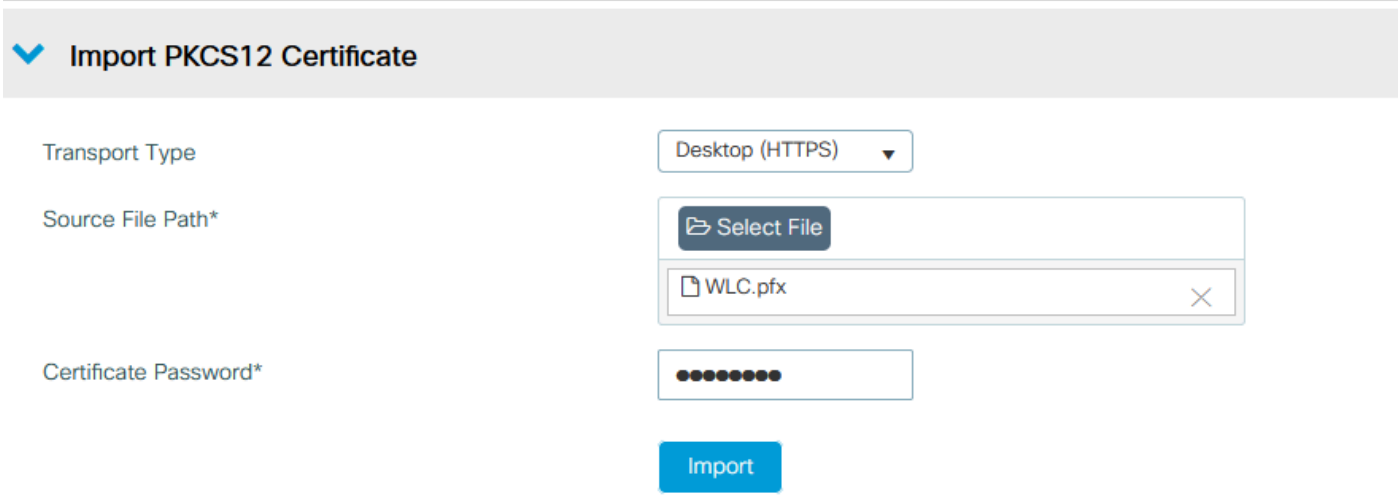
Menú de importación de certificados de dispositivos ISE



Sugerencia: solo tiene que importar el certificado de dispositivo ISE en este paso. Este certificado es el que ISE intercambia para establecer el túnel DTLS. No es necesario importar el certificado de dispositivo WLC y la clave privada ya que el certificado WLC se verifica con el uso de los certificados CA importados previamente.

Importar certificados a WLC

1. Navegue hasta Configuration > Security > PKI Management en el WLC y vaya a la pestaña Add Certificate.
2. Haga clic en el menú desplegable Import PKCS12 Certificate y establezca el tipo de transporte como Desktop (HTTPS).
3. Haga clic en el botón Select File y seleccione el archivo .pfx que preparó anteriormente.
4. Escriba la contraseña de importación y, por último, haga clic en Import.

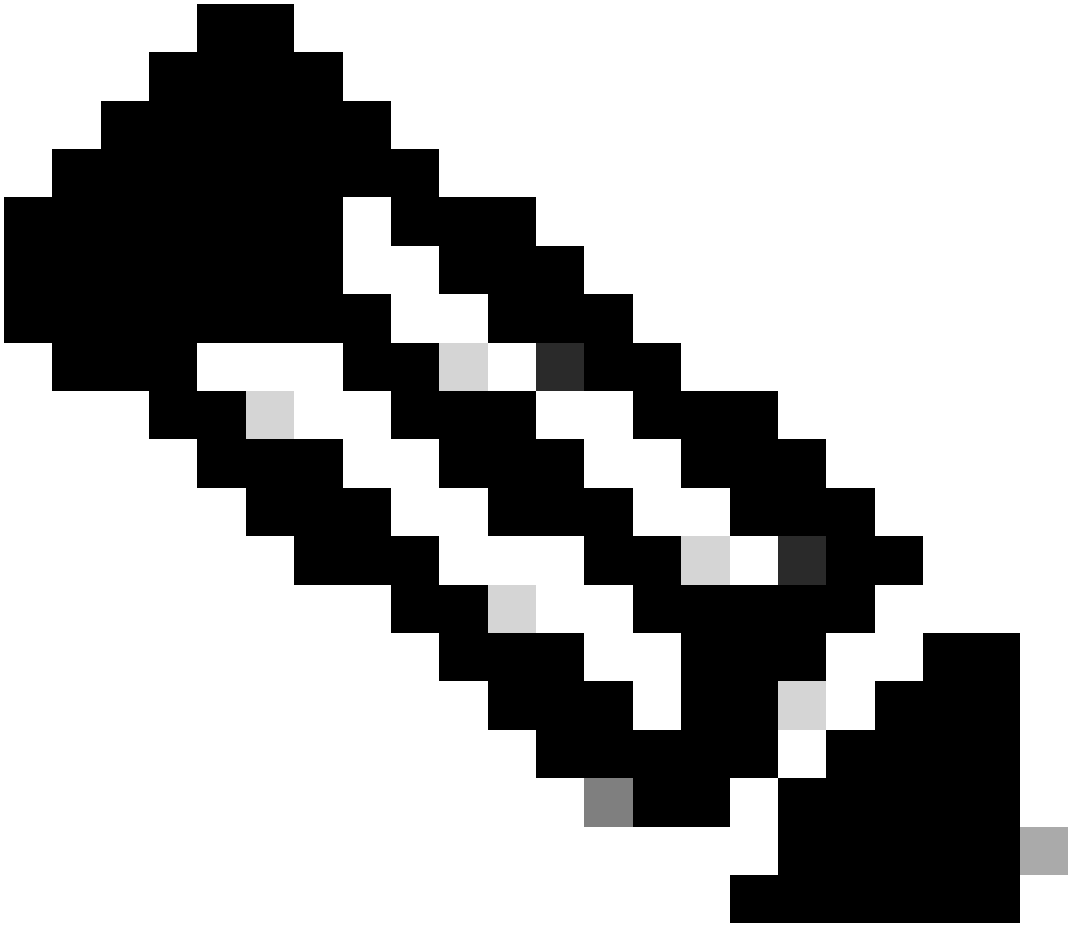


Cuadro de diálogo de importación de certificados WLC

Para obtener información detallada sobre el proceso de importación, consulte Generación y descarga de certificados CSR en WLC Catalyst 9800.

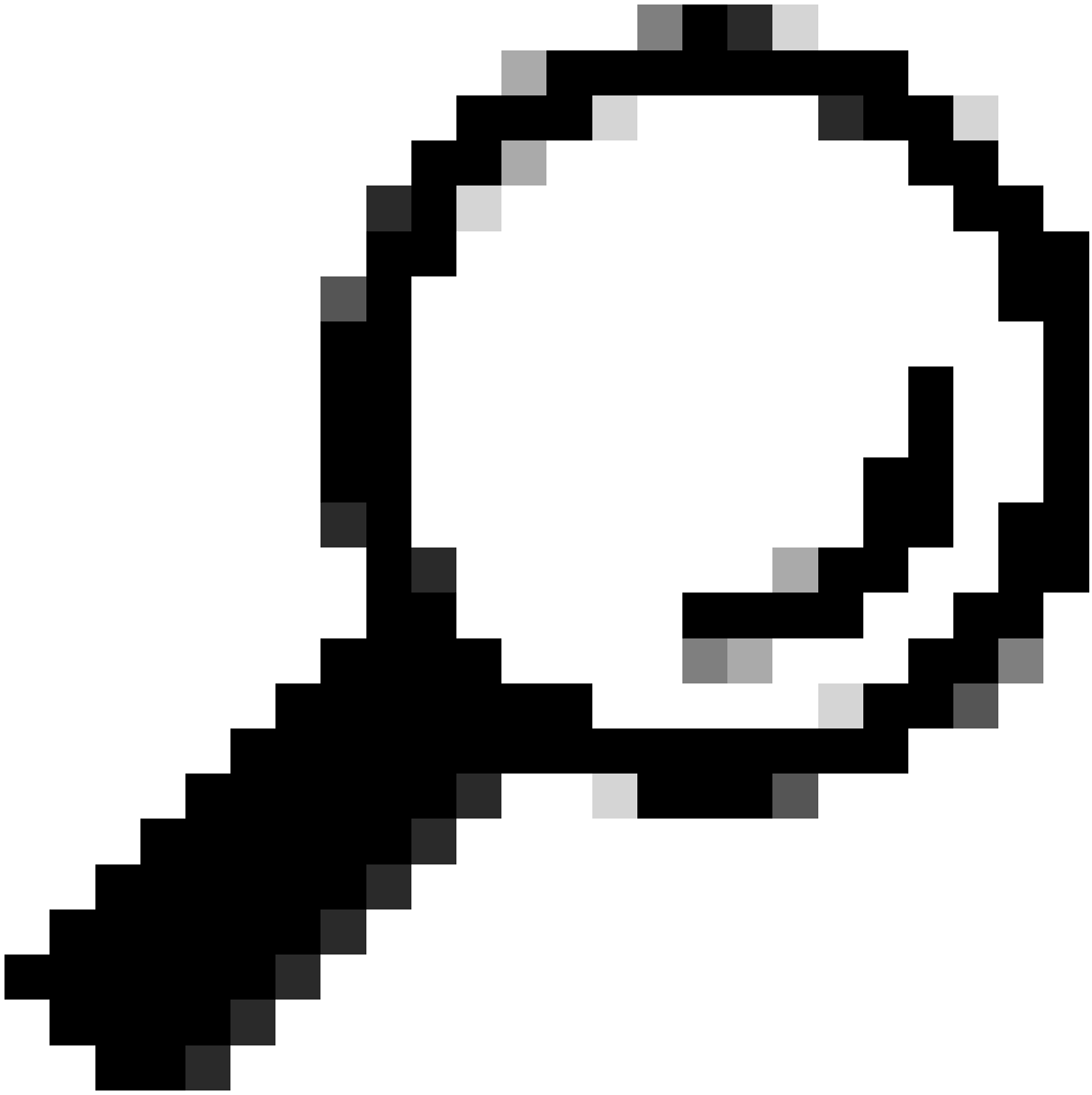
Inhabilite la comprobación de revocación dentro de cada punto de confianza creado automáticamente si el WLC no tiene ninguna lista de revocación de certificados que pueda comprobar a través de la red:

```
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
```



Nota: Si creó una CA multinivel en OpenSSL con el documento Configurar CA multinivel en OpenSSL para generar certificados Cisco IOS XE, debe inhabilitar la comprobación de revocación ya que no se crea ningún servidor CRL.

La importación automatizada crea los puntos de confianza necesarios para contener el certificado WLC y sus certificados CA.



Sugerencia: si los certificados WLC fueron emitidos por la misma CA que los certificados ISE, puede utilizar los mismos puntos de confianza creados automáticamente a partir de la importación de certificados WLC. No es necesario importar los certificados ISE por separado.

Si el certificado de WLC es emitido por una CA diferente del certificado de ISE, también debe importar los certificados de CA de ISE al WLC para que el WLC confíe en el certificado de dispositivo de ISE.
Cree un nuevo punto de confianza para la CA raíz e importe la CA raíz de 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-----
```

Importe el siguiente certificado de CA intermedia en la cadena de CA de ISE, es decir, el certificado de CA emitido por la CA raíz:

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

Cada CA adicional de la cadena requiere un punto de confianza independiente. Cada punto de confianza de la cadena debe hacer referencia al punto de confianza que contiene el certificado del emisor del certificado que desea importar con el comando chain-validation continue **<**nombre del punto de confianza del emisor**>**.

Importe tantos certificados de CA como contenga la cadena de CA. Ha finalizado la importación de la CA emisora del certificado de dispositivo ISE. Tome nota del nombre de este punto de confianza.

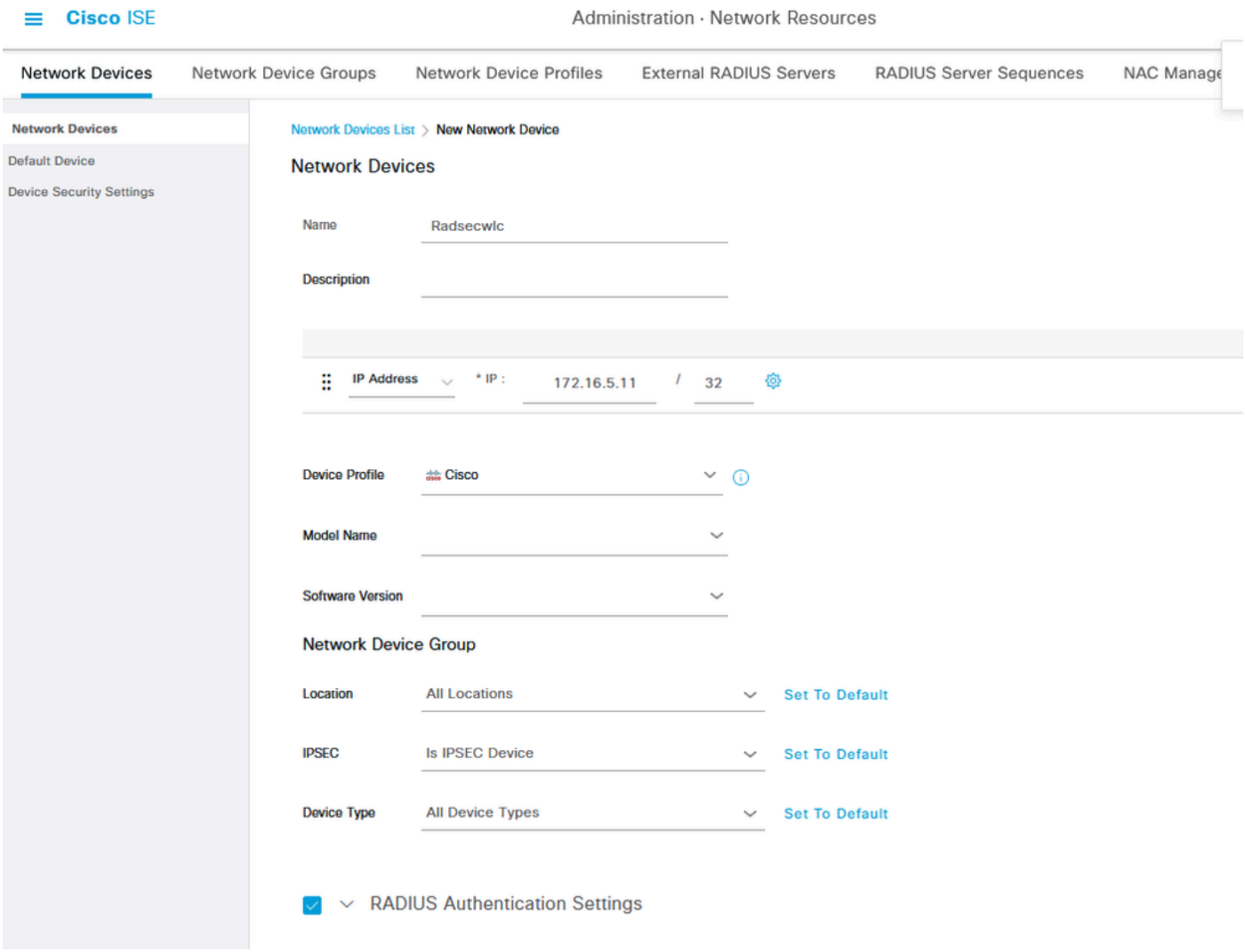
No es necesario importar el certificado del dispositivo ISE en el WLC para que RADIUS DTLS funcione.

Configuración de RADIUS DTLS

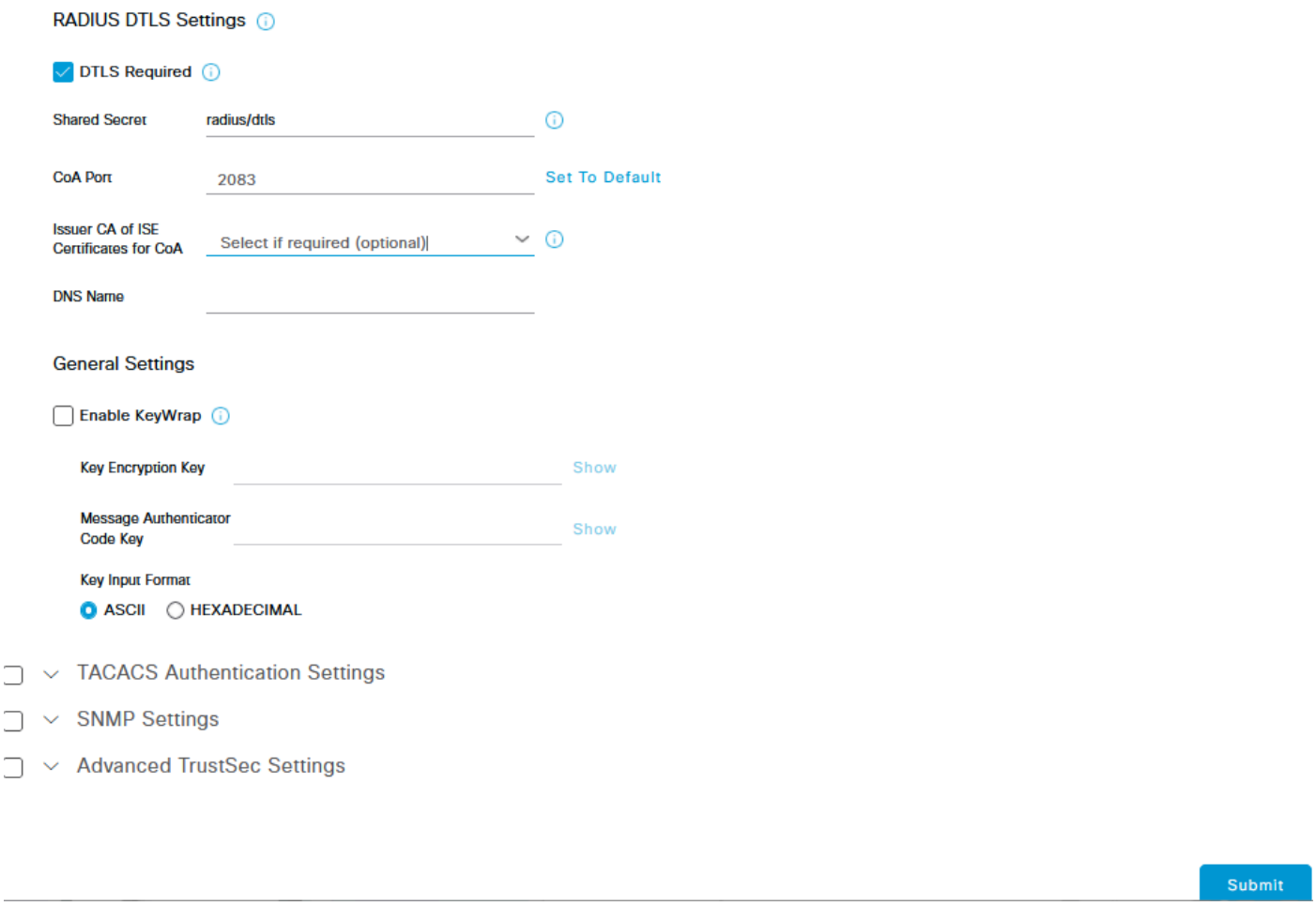
Configuración de ISE

Agregue el WLC como un dispositivo de red en ISE. Para ello, navegue hasta Administración**>**Recursos de red**>**Dispositivos de red**>**Agregar
Introduzca el nombre del dispositivo y la IP de la interfaz WLC que origina el tráfico RADIUS.

Normalmente, la IP de la interfaz de administración inalámbrica. Desplácese hacia abajo y verifique RADIUS Authentication Settings así como DTLS Required y haga clic en Submit:



Nueva configuración de dispositivo de red



Configuración de RADIUS DTLS para el dispositivo de red en ISE

Configuración de WLC

Defina un nuevo servidor Radius junto con la dirección IP de ISE y el puerto predeterminado para Radius DTLS. Esta configuración sólo está disponible en la CLI:

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

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

Radius DTLS debe utilizar el radius/dtls secreto compartido, el WLC 9800 ignora cualquier clave configurada que no sea esta:

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

Utilice el`dtls trustpoint client` comando para configurar el punto de confianza que contiene el certificado del dispositivo WLC para intercambiar para el túnel DTLS.
Utilice el`dtls trustpoint server` comando para configurar el punto de confianza que contiene la CA del emisor para el certificado del dispositivo ISE.

Los nombres de punto de confianza del cliente y del servidor son los mismos sólo si los certificados WLC e ISE son emitidos por la misma CA:

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

Configure el WLC para verificar si hay alguno de los nombres alternativos de sujeto (SAN) presentes en el certificado de ISE. Esta configuración debe coincidir exactamente con una de las SAN presentes en el campo SAN del certificado.

El WLC 9800 no realiza una coincidencia basada en expresiones regulares para el campo SAN. Esto significa, por ejemplo, que el comando `dtls match-server-identity hostname *.example.com` para un certificado comodín que tiene *.example.com en su campo SAN es correcto pero el mismo comando para un certificado que contiene www.example.com en el campo SAN no lo es.

El WLC no verifica este nombre contra cualquier servidor de nombre:

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

Cree un nuevo grupo de servidores para utilizar el nuevo Radius DTLS para la autenticación:

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

A partir de este punto, puede utilizar este grupo de servidores como cualquier otro grupo de

servidores en el WLC. Consulte Configuración de la Autenticación 802.1X en Catalyst 9800 Wireless Controller Series para utilizar este servidor para la autenticación de clientes inalámbricos.

Verificación

Verificar información del certificado

Para verificar la información del certificado para los certificados creados, en el terminal Linux ejecute el comando:

```
openssl x509 -in

      -text -noout
```

Muestra la información completa del certificado. Esto es útil para determinar la CA emisora de un certificado dado o si los certificados contienen las EKU y SAN requeridas:

```
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:
```

Información del certificado del dispositivo Cisco IOS XE tal como la muestra OpenSSL

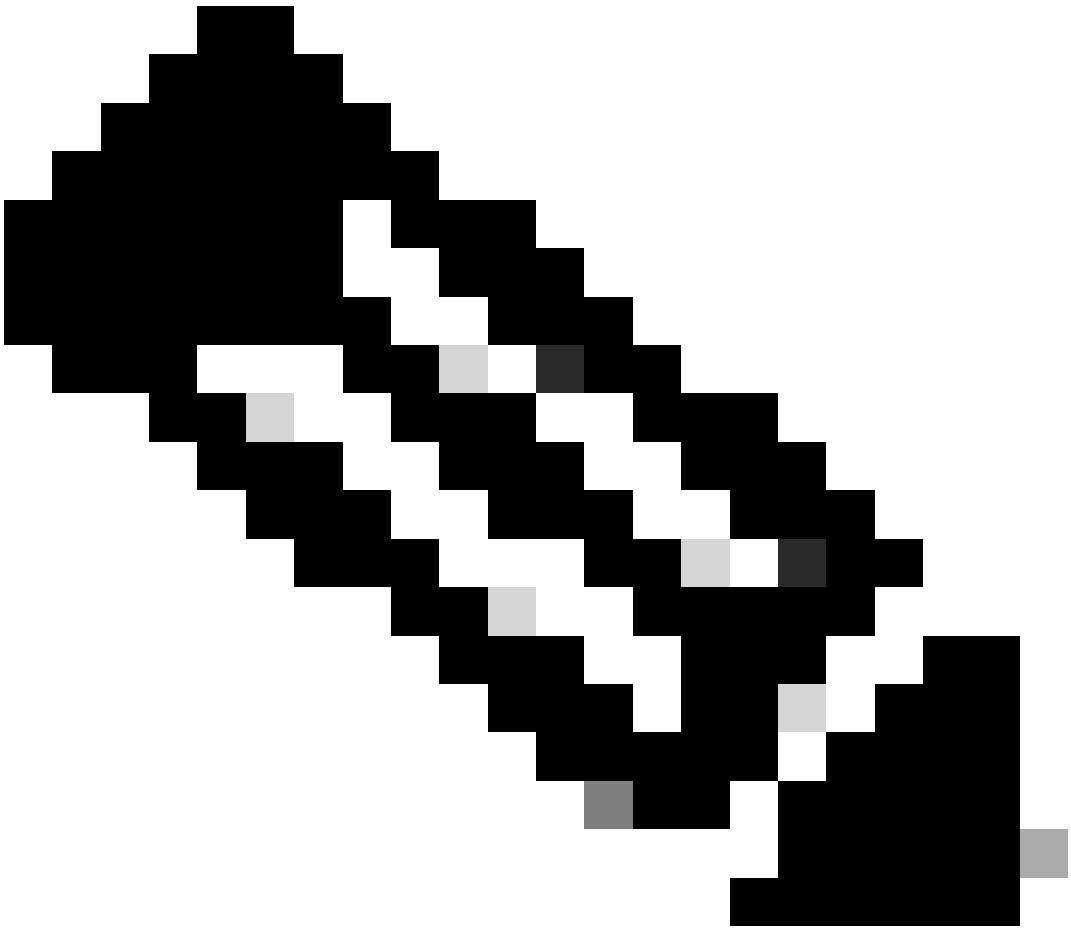
Realizar autenticación de prueba

Desde el WLC puede probar la funcionalidad de Radius DTLS con el comando test aaa group

new-code

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

```
USER ATTRIBUTES

username              0   "testuser"
```



> Nota: Una salida de rechazo de acceso en el comando de prueba significa que el WLC recibió un mensaje RADIUS de rechazo de acceso, en cuyo caso RADIUS DTLS está funcionando. Sin embargo, también puede indicar una falla al establecer el túnel DTLS. El comando test no diferencia ambos escenarios, vea la sección troubleshooting para identificar si hay un problema.

Troubleshoot

Para revisar la causa de una autenticación fallida, puede habilitar estos comandos antes de realizar una autenticación de prueba.

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

Ésta es la salida de una autenticación exitosa con depuraciones habilitadas:

```
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 desconocida informada por WLC

Cuando el WLC no puede validar los certificados proporcionados por ISE, no puede crear el túnel DTLS y las autenticaciones fallan.
Este es un ejemplo de los mensajes de debug presentados cuando este es el caso:

```
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:
```

Para corregirlo, asegúrese de que la identidad configurada en el WLC coincida exactamente con una de las SAN incluidas en el certificado ISE:

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

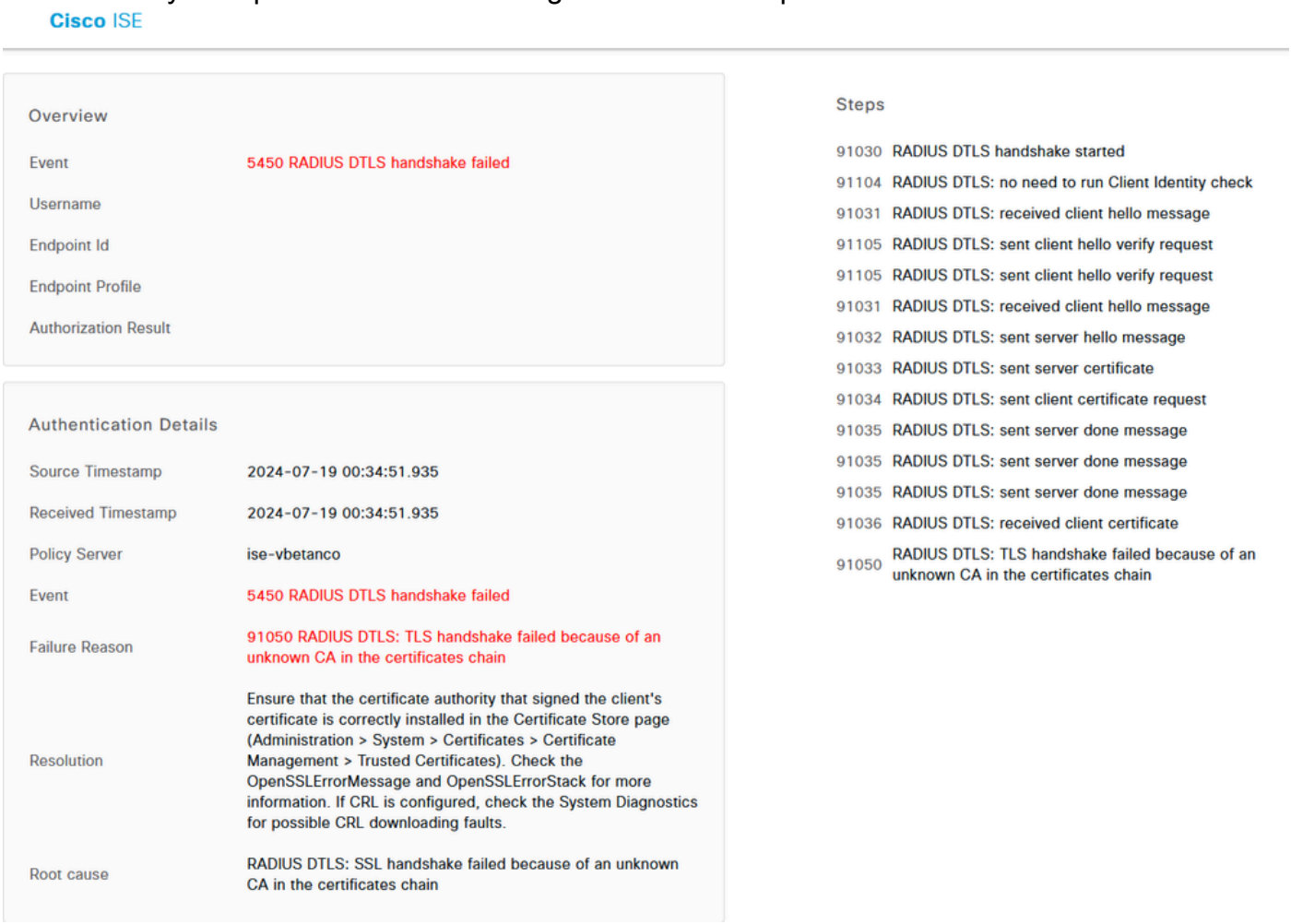
```
      9800(config)#dtls match-server-identity hostname
```

Asegúrese de que la cadena de certificados de la CA se importe correctamente en el controlador y de que el **dtls trustpoint server**

configuration uses the Issuer CA trustpoint.

CA desconocida notificada por ISE

Cuando ISE no puede validar los certificados proporcionados por el WLC, no puede crear el túnel DTLS y las autenticaciones fallan. Esto se muestra como un error en los registros en directo de RADIUS. Vaya a Operaciones>Radio>Registros en directo para verificarlos.



ISE Live Log informa de un fallo de enlace DTLS debido a una CA desconocida

Para corregirlo, asegúrese de que tanto el certificado intermedio como el certificado raíz, seleccione las casillas de verificación Trust for client authentication and Syslog en Administration>System>Certificates>Trusted certificates.

La comprobación de revocación está activa

Cuando los certificados se importan al WLC, los trustpoints recién creados tienen habilitada la comprobación de revocación. Esto hace que el WLC intente buscar una lista de revocación de certificados que no está disponible o accesible y no supera la verificación del certificado. Asegúrese de que cada punto de confianza en la ruta de verificación de los certificados contenga el comando revocation-check none .

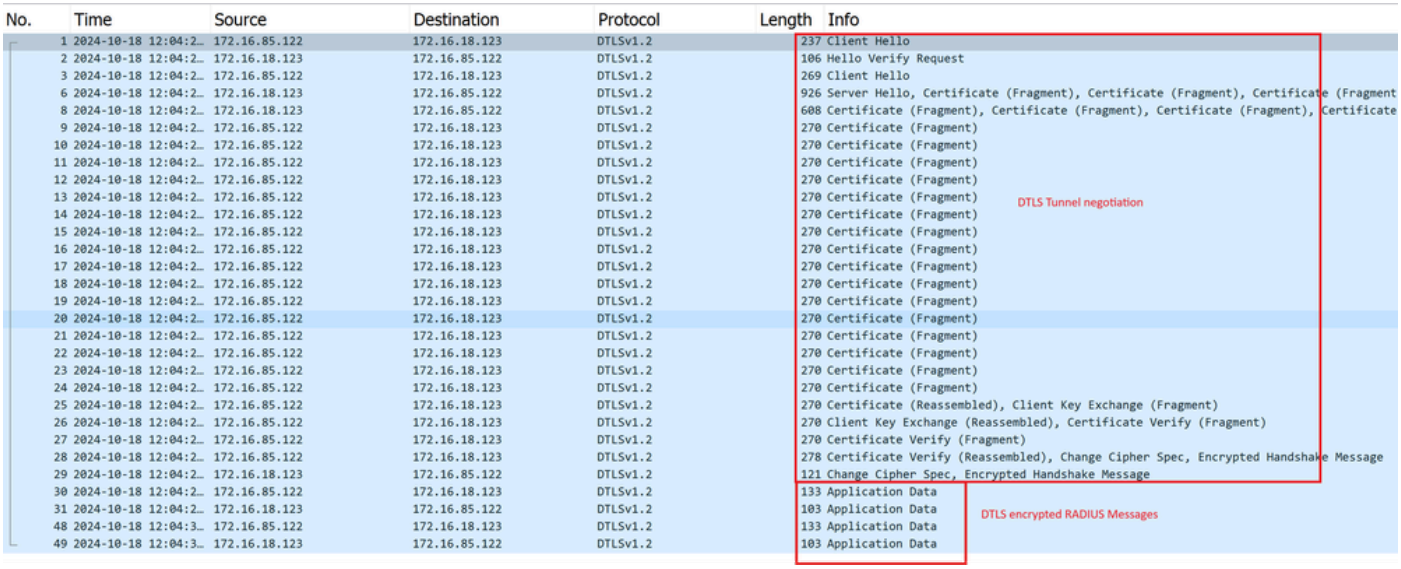
```
Jul 17 21:50:39.064: RADIUS_RADSEC_HASH_KEY_MATCH: hashkey1(0) matches hashkey2(0) TRUE
Jul 17 21:50:39.064: RADIUS_RADSEC_HASH_KEY_GET_CTX: radius radsec  sock_ctx(0x780FB0715978:0) get for
Jul 17 21:50:39.064: RADIUS_RADSEC_PROCESS_SOCK_EVENT: Handle socket event for TLS handshake(172.16.18.
Jul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

Resolución de problemas de establecimiento de túnel DTLS en captura de paquetes

El 9800 WLC ofrece la captura de paquetes incorporada (EPC) característica que le permite capturar todo el tráfico enviado y recibido para una interfaz dada. ISE ofrece una función similar denominada volcado de TCP para supervisar el tráfico entrante y saliente. Cuando se utilizan al mismo tiempo, le permiten analizar el tráfico de establecimiento de sesión DTLS desde la perspectiva de ambos dispositivos.

Consulte la Guía del administrador de Cisco Identity Services Engine para obtener pasos detallados para configurar el volcado de TCP en ISE. Consulte también Troubleshooting Catalyst 9800 Wireless LAN Controllers para obtener información para configurar la función EPC en el WLC.

Este es un ejemplo de establecimiento exitoso de un túnel DTLS.



Captura de paquetes de una negociación de túnel RADIUS DTLS y mensajes cifrados

Las capturas de paquetes muestran cómo ocurre el establecimiento del túnel DTLS. Si hay un problema con la negociación, desde tráfico perdido entre dispositivos o paquetes de alerta cifrados DTLS, la captura de paquetes le ayuda a identificar el problema.

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).