

Configurazione di Radius DTLS su ISE e 9800 WLC

Sommario

[Introduzione](#)

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Configurazione](#)

[Panoramica](#)

[Facoltativo - Crea certificato dispositivo WLC e ISE RADIUS DTLS](#)

[Aggiungi sezioni di configurazione nel file openssl.cnf](#)

[Crea certificato dispositivo WLC](#)

[Crea certificato dispositivo ISE](#)

[Importa certificati nei dispositivi](#)

[Importazione di certificati ad ISE](#)

[Importa certificati in WLC](#)

[Configura DTLS RADIUS](#)

[Configurazione di ISE](#)

[Configurazione WLC](#)

[Verifica](#)

[Verifica informazioni certificato](#)

[Esegui test di autenticazione](#)

[Risoluzione dei problemi](#)

[CA sconosciuta segnalata dal WLC](#)

[CA sconosciuta segnalata da ISE](#)

[Verifica revoca in corso](#)

[Risoluzione dei problemi di definizione del tunnel DTLS sull'acquisizione dei pacchetti](#)

Introduzione

Questo documento descrive un metodo per creare i certificati necessari per configurare le DTLS RADIUS tra ISE e il WLC 9800.

Introduzione

RADIUS DTLS è un formato sicuro del protocollo RADIUS in cui i messaggi RADIUS vengono inviati tramite un tunnel DTLS (Data Transport Layer Security). Per creare questo tunnel tra il server di autenticazione e l'autenticatore, è necessario un set di certificati. Questo insieme di certificati richiede l'impostazione di determinate estensioni di certificati per l'utilizzo chiavi

avanzato (EKU), in particolare l'autenticazione client sul certificato WLC e l'autenticazione sia del server che del client per il certificato ISE.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Come configurare il WLC 9800, il punto di accesso (AP) per le operazioni di base
- Come utilizzare l'applicazione OpenSSL
- Infrastruttura a chiave pubblica (PKI) e certificati digitali

Componenti usati

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

- Applicazione OpenSSL (versione 3.0.2).
- ISE (versione 3.1.0.518)
- 9800 WLC (versione 17.12.3)

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

Configurazione

Panoramica

Lo scopo è creare un'Autorità di certificazione a due livelli con una CA radice e una CA intermedia per firmare i certificati dell'endpoint. Una volta firmati, i certificati vengono importati nel WLC e nell'ISE. Infine, i dispositivi sono configurati per eseguire l'autenticazione DTLS RADIUS con tali certificati.



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

Facoltativo - Crea certificato dispositivo WLC e ISE RADIUS DTLS

Il protocollo RADIUS DTLS deve scambiare certificati tra ISE e WLC per creare il tunnel DTLS. Se non si dispone ancora di certificati validi, è possibile creare una CA locale per generare i certificati. Per ulteriori informazioni, fare riferimento a [Configurazione di un'autorità di certificazione a più livelli su OpenSSL per generare certificati compatibili con Cisco IOS® XE](#) ed eseguire i passaggi descritti nel documento dall'inizio alla fine del passaggio Crea certificato CA intermedio.

Aggiungi sezioni di configurazione nel file openssl.cnf

Aprire il file di configurazione openssl.cnf e, in fondo, copiare e incollare le sezioni WLC e ISE utilizzate per generare una richiesta CSR (Certificate Sign Request) valida.

Entrambe le sezioni ISE_device_req_ext e WLC_device_req_ext fanno riferimento a un elenco di SAN da includere nel 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
```

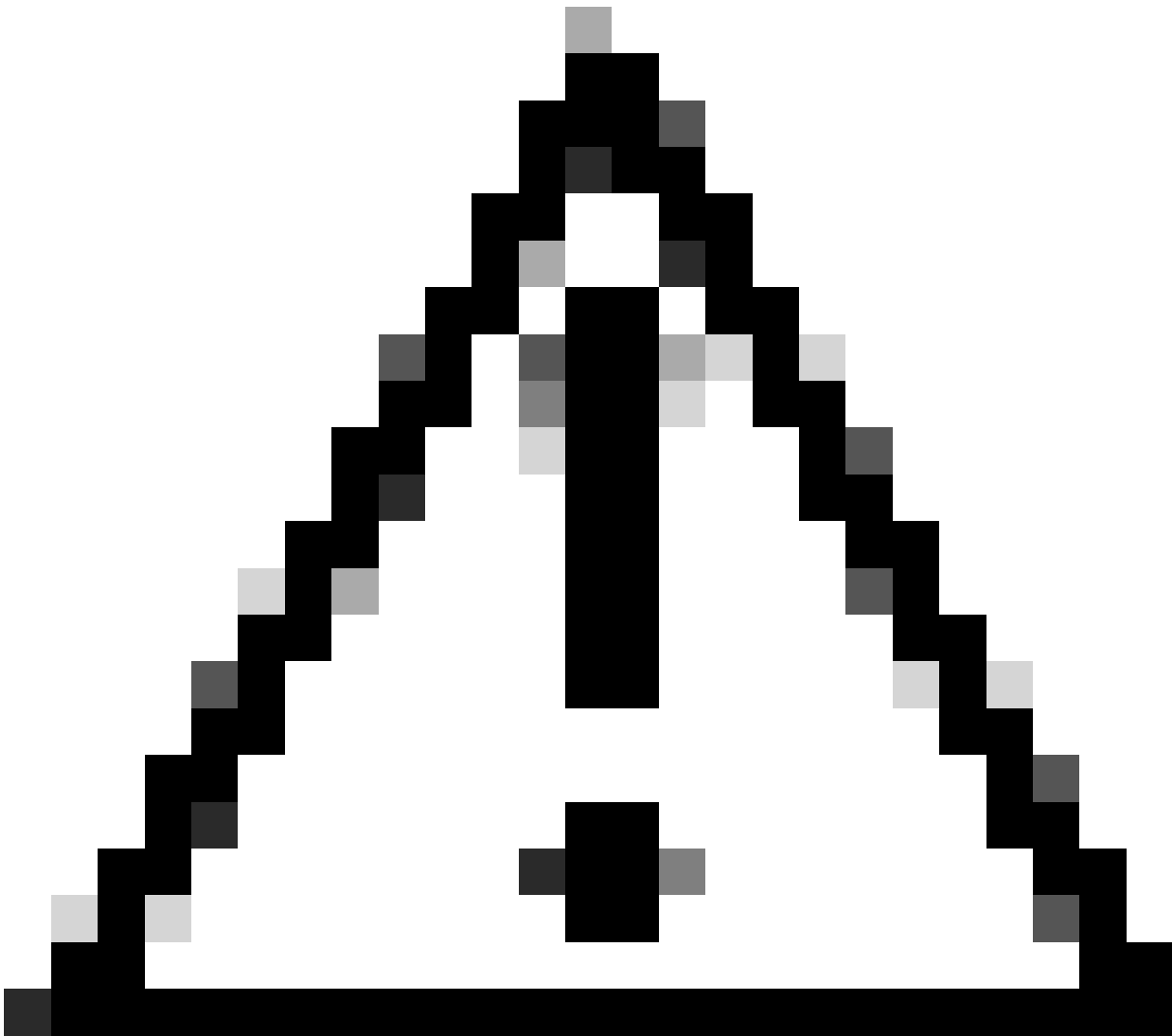
Come misura di sicurezza, l'autorità di certificazione ignora qualsiasi SAN presente in un CSR per firmarlo in modo che i dispositivi non autorizzati non possano ricevere un certificato valido per un nome che non sono autorizzati a utilizzare. Per aggiungere di nuovo le SAN al certificato firmato, utilizzare il parametro subjectAltName per puntare alle stesse SAN di elenco utilizzate per la generazione di CSR.

ISE richiede che sul certificato siano presenti gli EKU serverAuth e clientAuth, mentre il WLC richiede solo clientAuth. Vengono aggiunti al certificato firmato con il parametro extendedKeyUsage.

Copiare e incollare le sezioni utilizzate per la firma del certificato nella parte inferiore del file 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
```

Attenzione: il nome comune (CN) specificato sul prompt interattivo deve essere identico a uno dei nomi presenti nella sezione [WLC_alt_names] del file openssl.cnf.

Utilizzare la CA denominata IntermCA per firmare il CSR del WLC denominato WLC.csr con le estensioni definite in [WLC_cert] e archiviare il certificato firmato all'interno di `./IntermCA/IntermCA.db.certs/WLC`. Il certificato del dispositivo WLC è denominato `WLC.crt`:

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

9800 WLC richiede che il certificato sia in formato pfx per poterlo importare. Creare un nuovo file contenente la catena di CA che hanno firmato il certificato WLC, denominato certfile:

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

Per creare il file .pfx, eseguire uno di questi comandi in base alla versione WLC.

Per le versioni precedenti alla 17.12.1:

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

Per la versione 17.12.1 o successive:

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

Crea certificato dispositivo ISE

Creare una nuova cartella in cui archiviare i certificati ISE nel computer in cui è installato OpenSSL nella cartella dei certificati CA intermedi denominata IntermCA.db.certs. La nuova cartella si chiama ISE:

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

Modificare i parametri DNS nella sezione [ISE_alt_names] del file openssl.cnf. Per modificare i nomi di esempio forniti per i valori desiderati, questi valori popolano il campo SAN del certificato 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
```

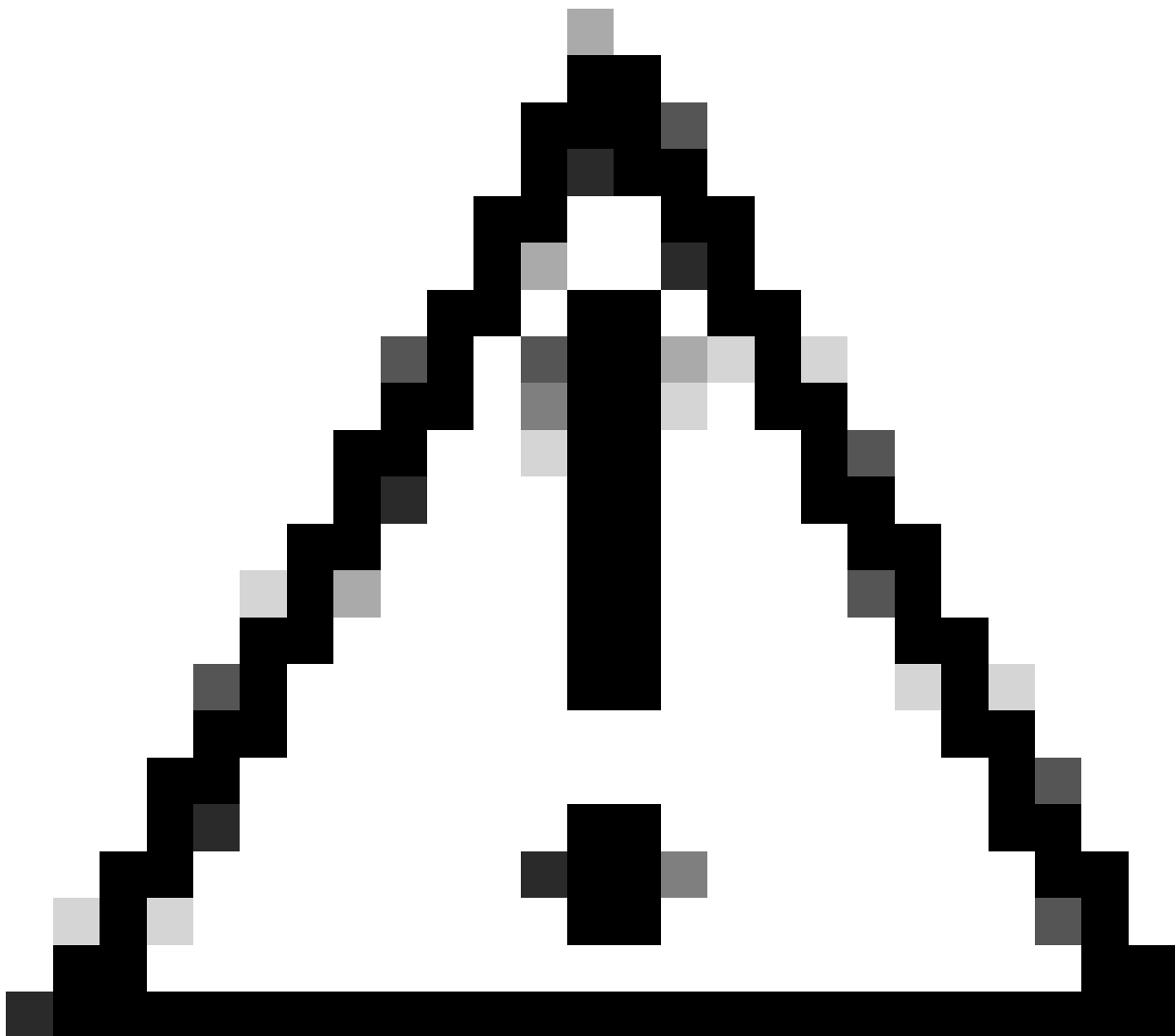
Creare la chiave privata ISE e ISE CSR con le informazioni della sezione ISE_device_req_ext per le SAN:

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

OpenSSL apre una richiesta interattiva per l'immissione dei dettagli del nome distinto (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
```

Prompt interattivo del nome distinto del certificato ISE



Attenzione: il CN specificato sul prompt interattivo deve essere esattamente uguale a uno dei Nomi nella sezione [ISE_alt_names] del file openssl.cnf.

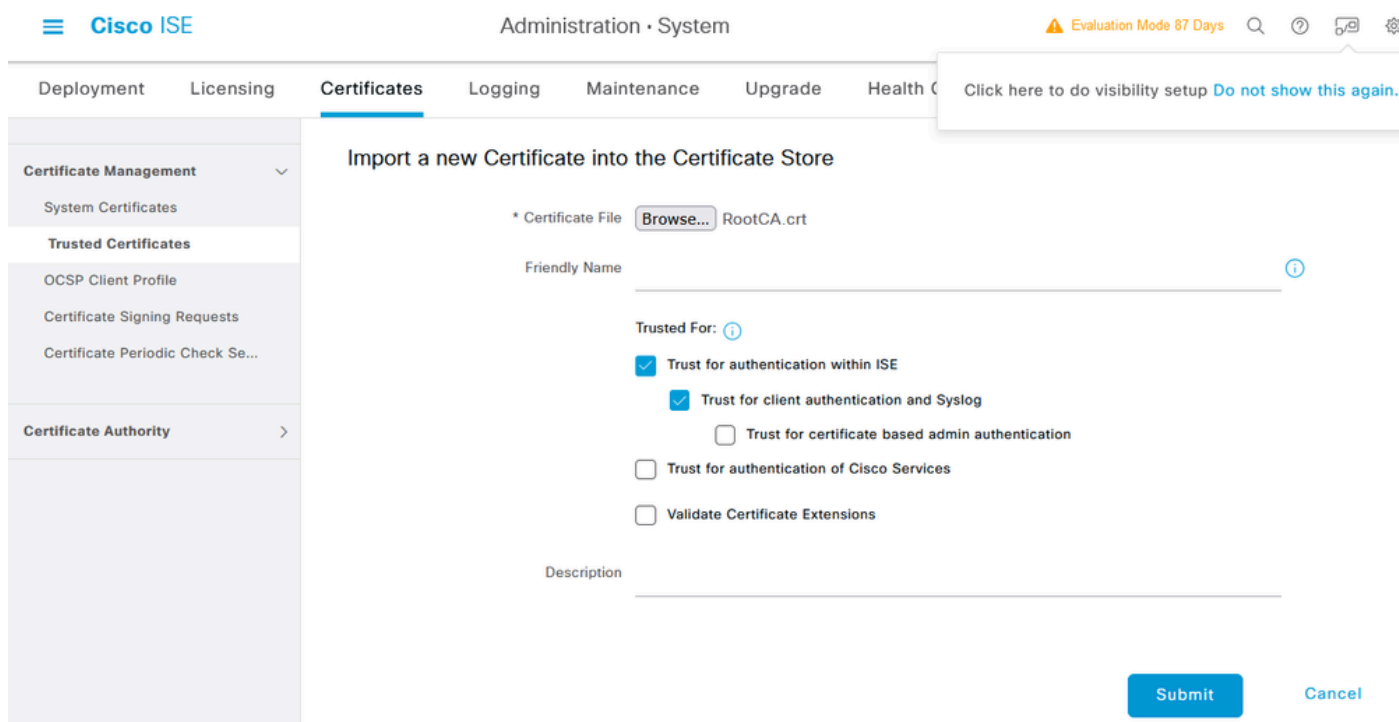
Utilizzare l'autorità di certificazione IntermCA per firmare l'ISE CSR denominata ISE.csr con le estensioni definite in [ISE_cert] e archiviare il certificato firmato in ./IntermCA/IntermCA.db.certs/WLC. Il certificato del dispositivo ISE è denominato ISE.crt:

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

Importa certificati nei dispositivi

Importazione di certificati ad ISE

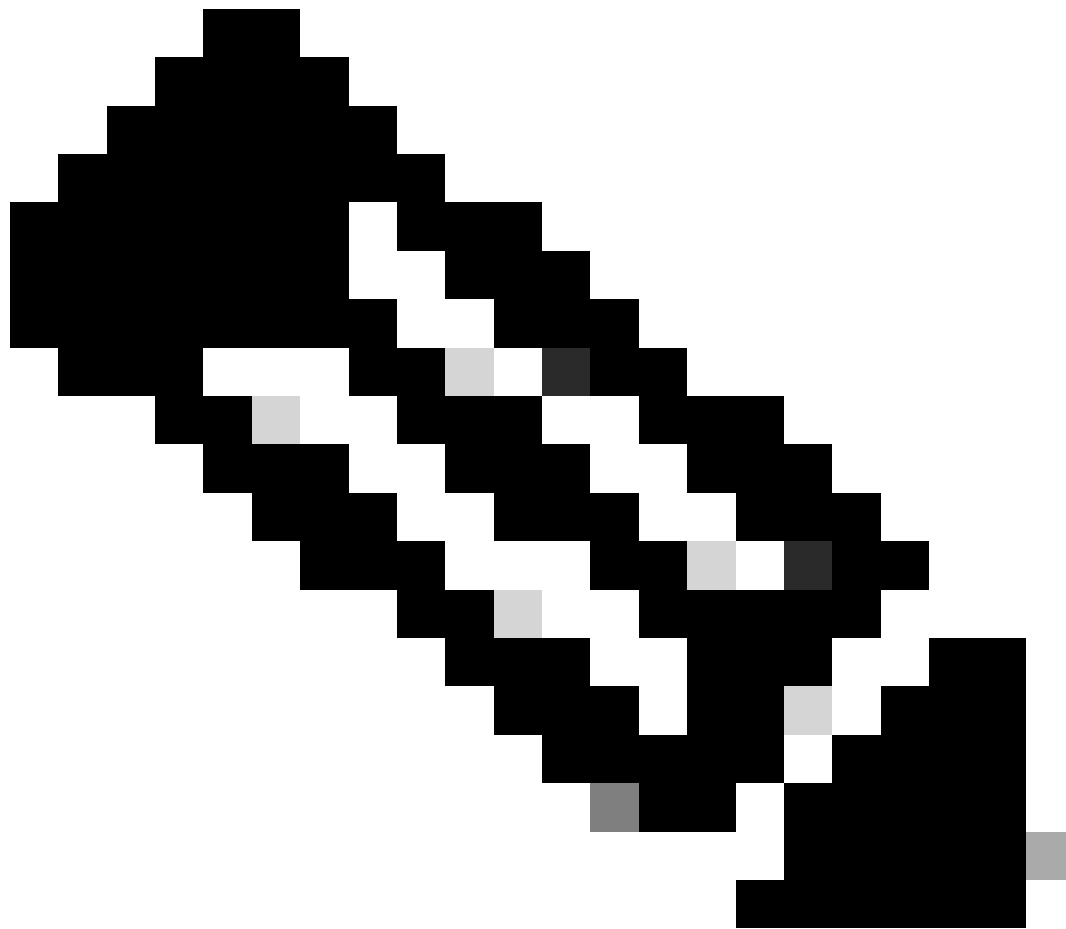
1. Importare il certificato CA radice dalla catena di certificati ISE nell'archivio dei certificati protetti.
2. Passare a Amministrazione>Sistema>Certificati>Certificati attendibili.
3. Fare clic su Browse (Sfogliare) e selezionare il file Root.crt.
4. Selezionare le caselle di controllo Trust for authentication within ISE and Trust for client authentication and Syslog, quindi fare clic su Submit:



The screenshot shows the Cisco ISE Administration console interface. The top navigation bar includes 'Cisco ISE', 'Administration · System', and a notification for 'Evaluation Mode 87 Days'. The main menu on the left is expanded to 'Certificates'. The central panel displays the 'Import a new Certificate into the Certificate Store' form. The form includes a 'Certificate File' field with a 'Browse...' button and the filename 'RootCA.crt'. Below this is a 'Friendly Name' field. The 'Trusted For' section contains several checkboxes: 'Trust for authentication within ISE' (checked), 'Trust for client authentication and Syslog' (checked), 'Trust for certificate based admin authentication' (unchecked), 'Trust for authentication of Cisco Services' (unchecked), and 'Validate Certificate Extensions' (unchecked). A 'Description' field is at the bottom. The form concludes with 'Submit' and 'Cancel' buttons.

Finestra di dialogo Importazione certificato CA radice ISE

Eseguire la stessa operazione per il certificato intermedio, se esistente.



Nota: ripetere i passaggi per ogni certificato CA che fa parte della catena di convalida dei certificati ISE. Inizia sempre con il certificato CA radice e termina con il certificato CA intermedio più basso della catena.

Click here to do visibility setup [Do not show this again.](#)

- Certificate Management
 - System Certificates
 - Trusted Certificates**
 - OCSP Client Profile
 - Certificate Signing Requests
 - Certificate Periodic Check Se...
- Certificate Authority

Import a new Certificate into the Certificate Store

* Certificate File IntermCA.crt

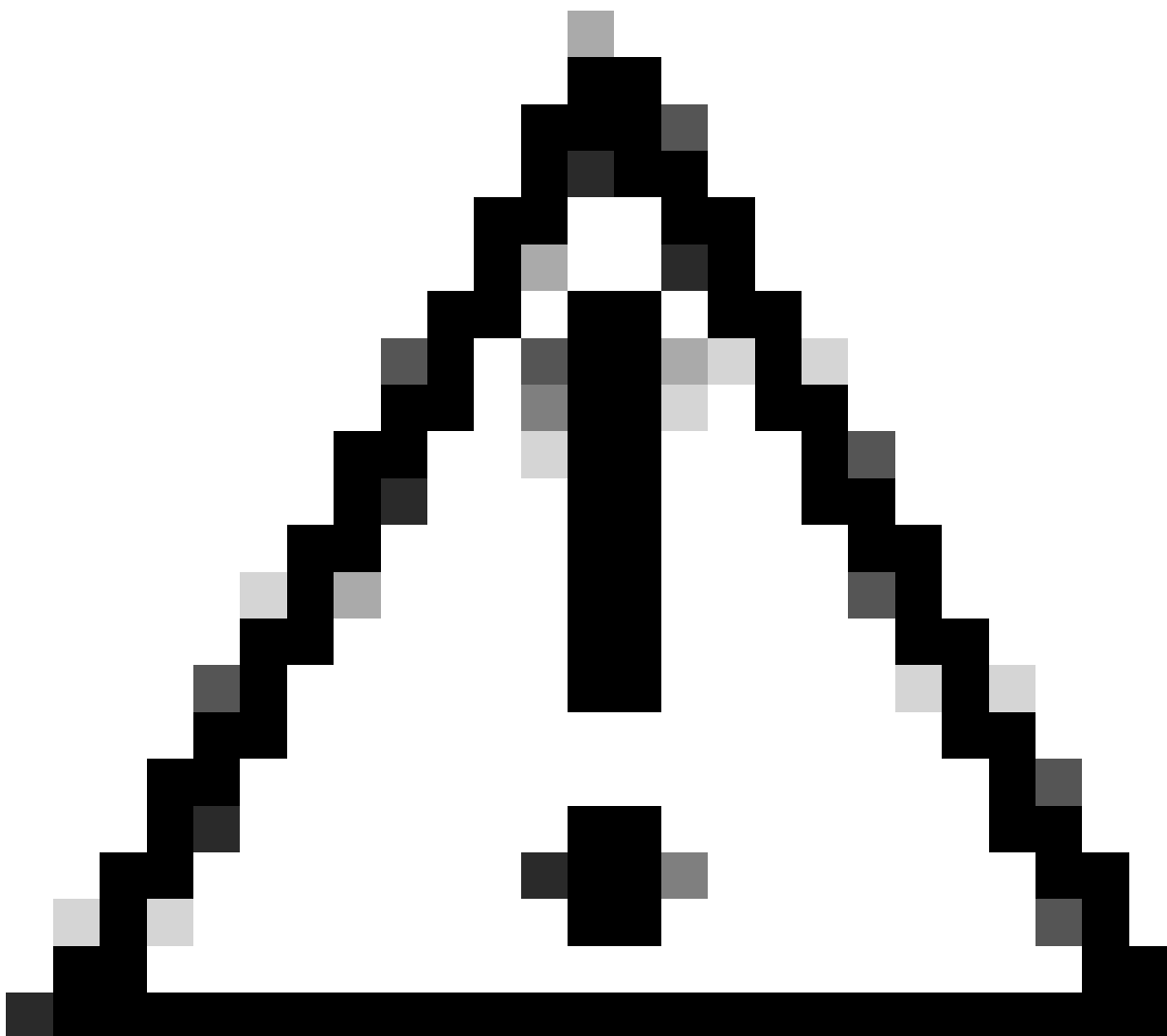
Friendly Name

Trusted For: ⓘ

- Trust for authentication within ISE
 - Trust for client authentication and Syslog
 - Trust for certificate based admin authentication
- Trust for authentication of Cisco Services
- Validate Certificate Extensions

Description

Finestra di dialogo ISE Intermediate CA Certificate Import



Attenzione: se il certificato ISE e il certificato WLC sono emessi da CA diverse, è necessario importare anche tutti i certificati CA che appartengono alla catena di certificati WLC. ISE non accetta il certificato WLC sullo scambio di certificati DTLS finché non vengono importati tali certificati CA.

Certificate Management ▾

System Certificates

- Trusted Certificates
- OCSP Client Profile
- Certificate Signing Requests
- Certificate Periodic Check Se...

Certificate Authority >

Import Server Certificate

* Select Node ▾

* Certificate File ISE.crt

* Private Key File ISE.key

Password

Friendly Name

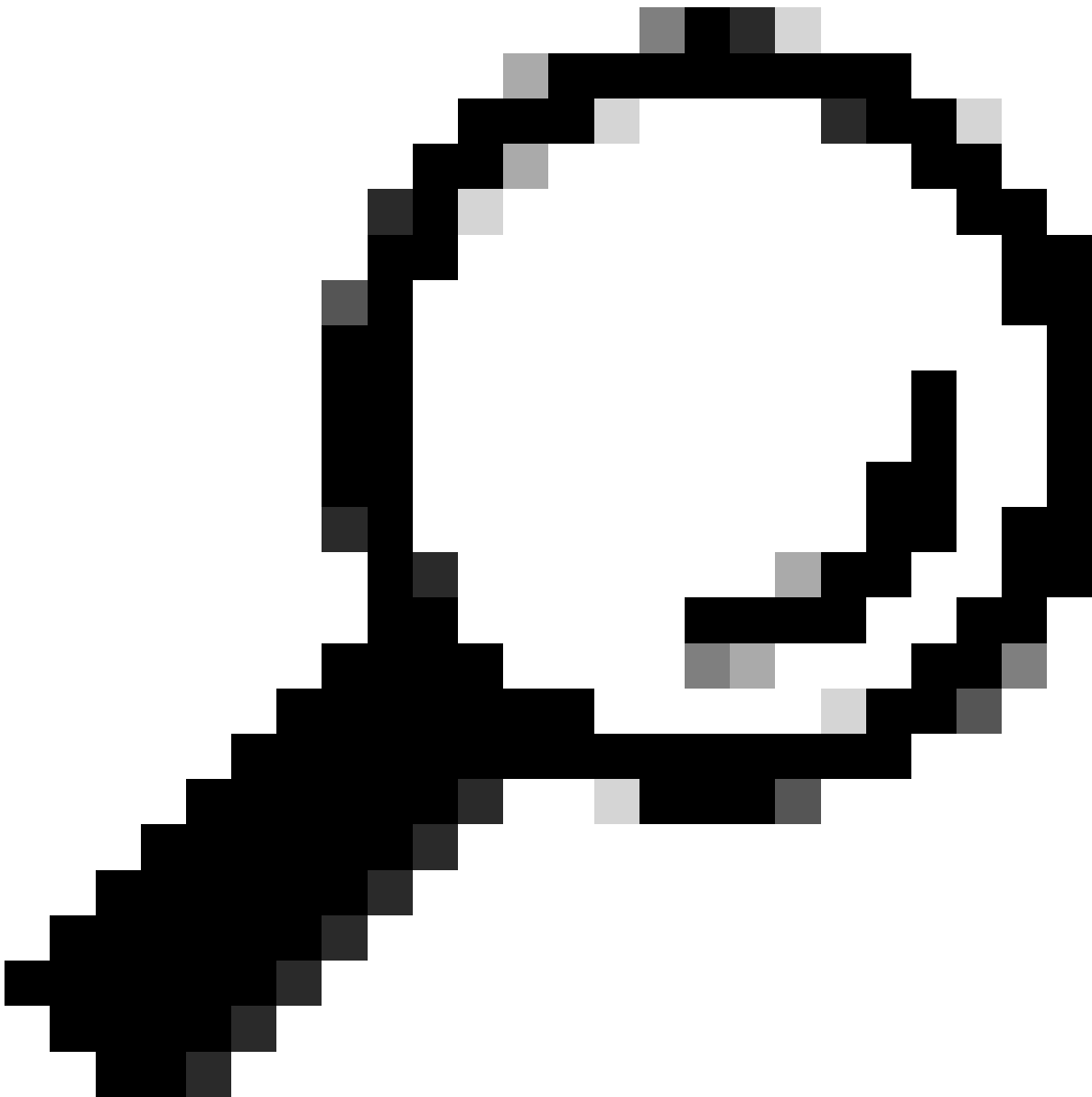
Allow Wildcard Certificates ⓘ

Validate Certificate Extensions ⓘ

Usage

- Admin:** Use certificate to authenticate the ISE Admin Portal
- EAP Authentication:** Use certificate for EAP protocols that use SSL/TLS tunneling
- RADIUS DTLS:** Use certificate for the RADSec server
- pxGrid:** Use certificate for the pxGrid Controller

Menu di importazione dei certificati dei dispositivi ISE



Suggerimento: in questo passaggio è sufficiente importare il certificato del dispositivo ISE. Questo certificato è uno scambio ISE per stabilire il tunnel DTLS. Non è necessario importare il certificato del dispositivo WLC e la chiave privata, in quanto il certificato WLC viene verificato con l'utilizzo dei certificati CA importati in precedenza.

Importa certificati in WLC

1. Passare a Configurazione > Sicurezza > Gestione PKI sul WLC e passare alla scheda Aggiungi certificato.
2. Fare clic sull'elenco a discesa Import PKCS12 Certificate (Importa certificato PKCS12) e impostare il tipo di trasporto su Desktop (HTTPS).
3. Fare clic sul pulsante Seleziona file e selezionare il file .pfx preparato in precedenza.
4. Digitare la password per l'importazione e fare clic su Import (Importa).

Import PKCS12 Certificate

Transport Type

Desktop (HTTPS) ▼

Source File Path*

Select File

WLC.pfx

Certificate Password*

••••••••

Import

Finestra di dialogo Importazione certificato WLC

Per informazioni dettagliate sul processo di importazione, consultare il documento sulla [generazione e il download dei certificati CSR sui WLC di Catalyst 9800](#).

Disabilita il controllo delle revoche all'interno di ogni trust point creato automaticamente se il WLC non dispone di un elenco di revoche di certificati che possa controllare attraverso la rete:

```
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
```

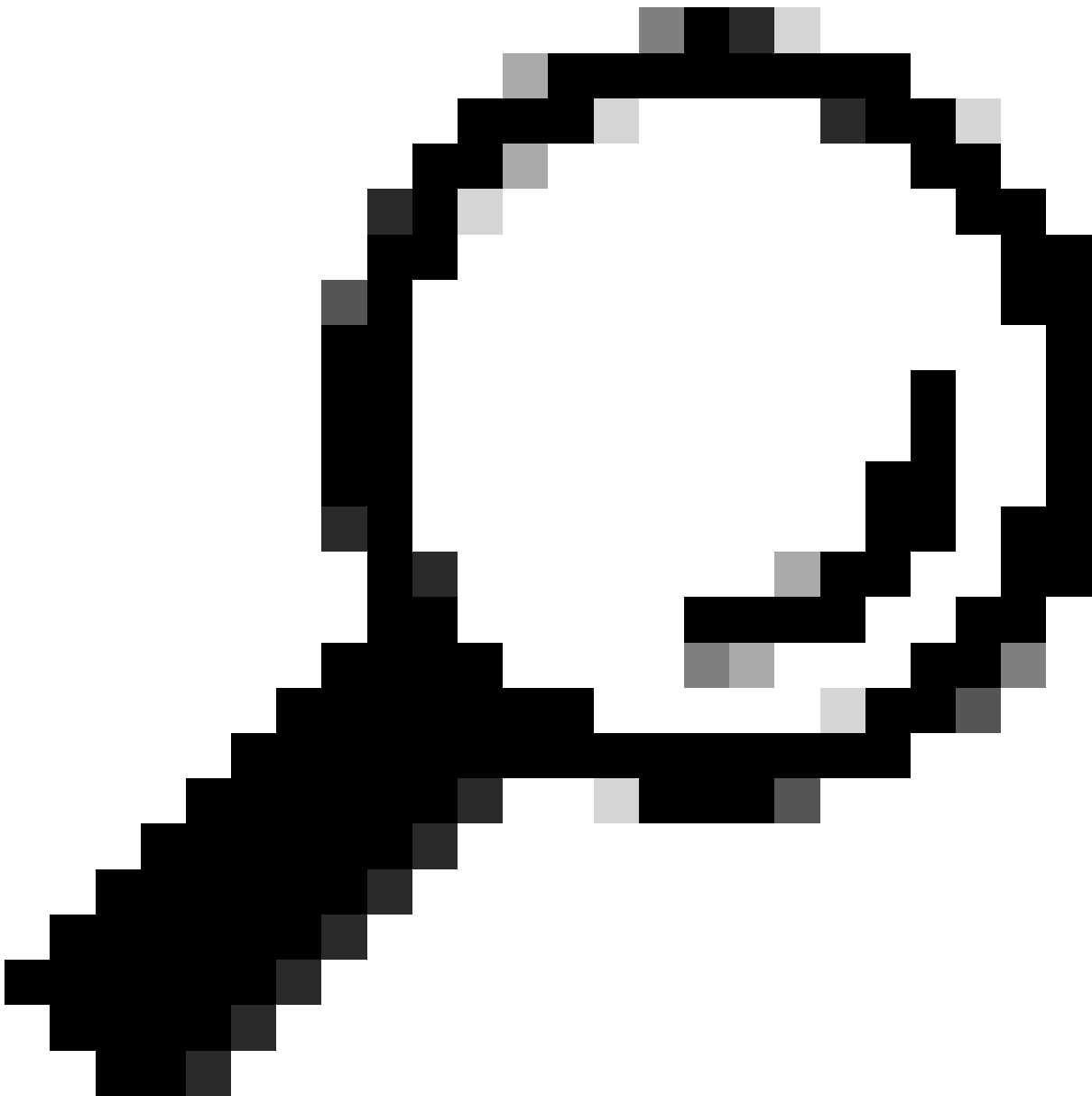
```
9800(config)#revocation-check none
```

```
9800(config)#exit
```



Nota: se è stata creata una CA multilivello su OpenSSL con il documento Configura CA multilivello su OpenSSL per generare certificati Cisco IOS XE, è necessario disabilitare il controllo di revoca perché non viene creato alcun server CRL.

L'importazione automatica crea i trust point necessari per contenere il certificato WLC e i relativi certificati CA.



Suggerimento: se i certificati WLC sono stati emessi dalla stessa CA dei certificati ISE, è possibile utilizzare gli stessi trust creati automaticamente dall'importazione del certificato WLC. Non è necessario importare i certificati ISE separatamente.

Se il certificato WLC è rilasciato da una CA diversa dal certificato ISE, è necessario importare anche i certificati ISE CA nel WLC in modo che il WLC possa considerare attendibile il certificato del dispositivo ISE.

Creare un nuovo trust point per la CA radice e importare la CA radice 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-----

Importare il successivo certificato CA intermedio sulla catena di CA ISE, in altre parole, il certificato CA rilasciato dalla CA radice:

```
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-----

Ogni CA aggiuntiva della catena richiede un trust point separato. Ogni trust point nella catena deve fare riferimento al trust point che contiene il certificato dell'autorità di certificazione del certificato che si desidera importare con il comando chain-validation continue <nome trustpoint autorità di certificazione>.

Importa tutti i certificati CA contenuti nella catena di CA. Dopo aver importato la CA dell'autorità di certificazione del certificato del dispositivo ISE, prendere nota del nome di questo trust point.

non è necessario importare il certificato del dispositivo ISE sul WLC perché le DTLS RADIUS funzionino.

Configura DTLS RADIUS

Configurazione di ISE

Aggiungere il WLC come dispositivo di rete in ISE, a tale scopo, selezionare Amministrazione>Risorse di rete>Dispositivi di rete>Aggiungi

Immettere il nome del dispositivo e l'indirizzo IP dell'interfaccia WLC da cui proviene il traffico RADIUS. In genere, l'indirizzo IP dell'interfaccia di gestione wireless. Scorrere verso il basso e

selezionare RADIUS Authentication Settings (Impostazioni di autenticazione RADIUS) e DTLS Required (DTLS richiesto), quindi fare clic su Submit (Invia):

Cisco ISE Administration · Network Resources

Network Devices | Network Device Groups | Network Device Profiles | External RADIUS Servers | RADIUS Server Sequences | NAC Management

Network Devices List > New Network Device

Network Devices

Name

Description

IP Address * IP: /

Device Profile

Model Name

Software Version

Network Device Group

Location [Set To Default](#)

IPSEC [Set To Default](#)

Device Type [Set To Default](#)

RADIUS Authentication Settings

Nuova configurazione dispositivo di rete

RADIUS DTLS Settings ⓘ

DTLS Required ⓘ

Shared Secret ⓘ

CoA Port [Set To Default](#)

Issuer CA of ISE Certificates for CoA ⓘ

DNS Name

General Settings

Enable KeyWrap ⓘ

Key Encryption Key [Show](#)

Message Authenticator Code Key [Show](#)

Key Input Format

ASCII HEXADECIMAL

TACACS Authentication Settings

SNMP Settings

Advanced TrustSec Settings

Submit

Impostazioni Radius DTLS per il dispositivo di rete su ISE

Configurazione WLC

Definire un nuovo server Radius insieme all'indirizzo IP ISE e alla porta predefinita per le DTLS Radius. Questa configurazione è disponibile solo dalla CLI:

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

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

Le DTLS Radius devono utilizzare il segreto condiviso radius/dtls, il WLC 9800 ignora qualsiasi chiave configurata diversa da questa:

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

Usare il comando `dtls trustpoint client`

per configurare il trust point che contiene il certificato del dispositivo WLC da scambiare per il tunnel DTLS.

Utilizzare il comando `dtls trustpoint server`

per configurare il trust point che contiene l'autorità di certificazione dell'autorità di certificazione per il certificato del dispositivo ISE.

I nomi dei trust point client e server sono gli stessi solo se i certificati WLC e ISE sono emessi dalla stessa CA:

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

Configurare il WLC in modo da verificare la presenza di uno dei nomi alternativi del soggetto (SAN) presenti nel certificato ISE. Questa configurazione deve corrispondere esattamente a una delle SAN presenti nel campo SAN del certificato.

Il WLC 9800 non esegue una corrispondenza basata su espressioni regolari per il campo SAN. Ciò significa, ad esempio, che il comando `dtls match-server-identity hostname *.example.com` per un certificato con caratteri jolly che ha *.example.com sul suo campo SAN è corretto, ma lo stesso comando per un certificato che contiene www.example.com sul campo SAN non lo è.

Il WLC non confronta questo nome con alcun server dei nomi:

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

Creare un nuovo gruppo di server per utilizzare il nuovo DTLS Radius per l'autenticazione:

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

Da questo momento in poi è possibile utilizzare questo gruppo di server come qualsiasi altro

gruppo di server sul WLC. Per ulteriori informazioni, fare riferimento a [Configurazione dell'autenticazione 802.1X sui controller wireless Catalyst serie 9800](#) per l'utilizzo di questo server per l'autenticazione dei client wireless.

Verifica

Verifica informazioni certificato

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

```
openssl x509 -in
```

```
-text -noout
```

Visualizza le informazioni complete sul certificato. Questa opzione è utile per determinare l'autorità di certificazione di un determinato certificato o se i certificati contengono gli ECU e le SAN richiesti:

```

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:

```

Informazioni sul certificato del dispositivo Cisco IOS XE mostrate da OpenSSL

Esegui test di autenticazione

Dal WLC è possibile verificare la funzionalità DTLS Radius con il comando `test aaa group`

new-code

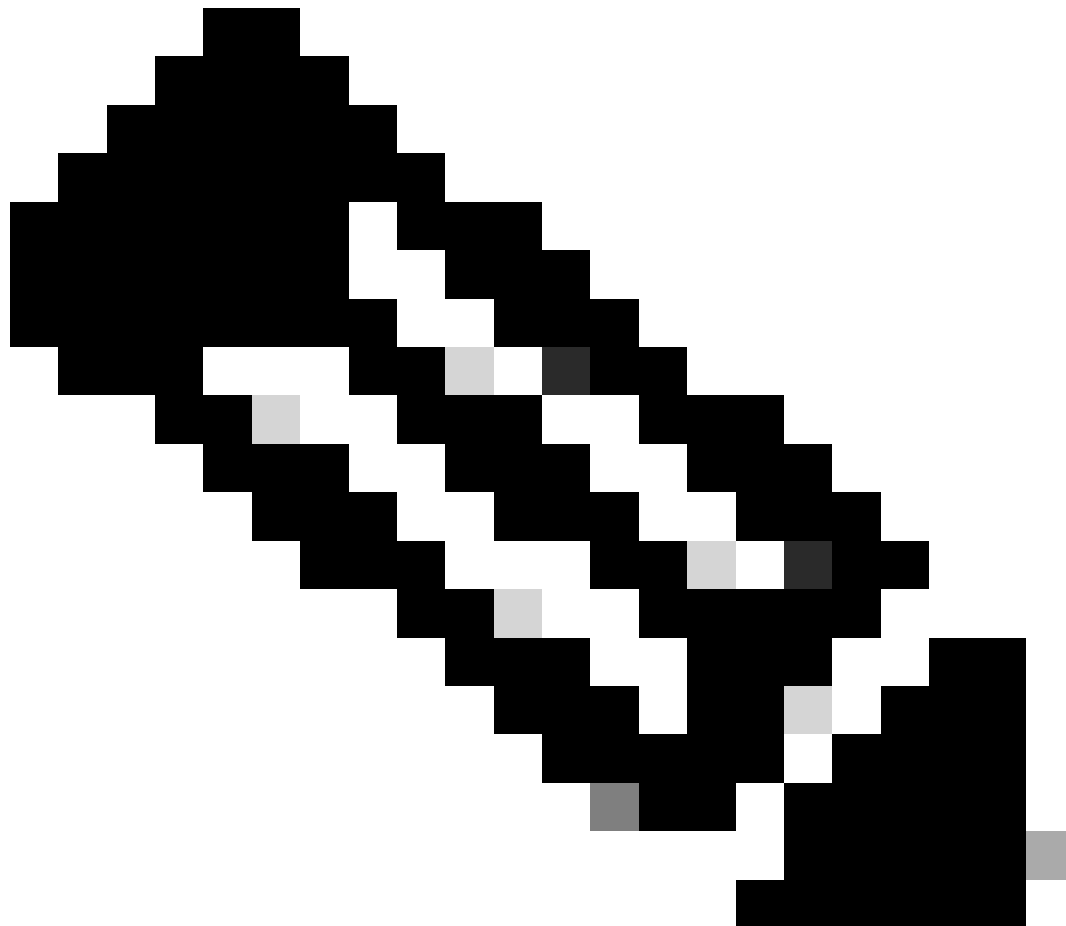
```

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

```

USER ATTRIBUTES

username 0 "testuser"



Nota: un output di rifiuto di accesso sul comando test indica che il WLC ha ricevuto un messaggio RADIUS Access-Reject, nel qual caso il DTLS RADIUS funziona. Tuttavia, può anche indicare che non è stato possibile stabilire il tunnel DTLS. Il comando di test non distingue entrambi gli scenari. Vedere la sezione relativa alla risoluzione dei problemi per determinare se si è verificato un problema.

Risoluzione dei problemi

Per verificare la causa di un errore di autenticazione, è possibile attivare questi comandi prima di eseguire un test di autenticazione.


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

Questo è l'output di un'autenticazione riuscita con debug abilitati:

```
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_SOCKET_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_SOCKET_ADDR: Success
Jul 18 21:24:38.313: RADIUS_RADSEC_SET_LOCAL_SOCKET: Success
Jul 18 21:24:38.313: RADIUS_RADSEC_SOCKET_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_SOCKET_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(172.16.18.123/2083)
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(172.16.18.123/2083)
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 [GXPGNc1Wv/9gDfQg]
RADIUS: 63 4A 76 6C 35 47 72 33 71 71 47 36 4C 66 35 59 [cJv15Gr3qqG6Lf5Y]
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 sconosciuta segnalata dal WLC

Quando il WLC non può convalidare i certificati forniti da ISE, non riesce a creare il tunnel DTLS e le autenticazioni non riescono.

Di seguito è riportato un esempio dei messaggi di debug presentati:

```
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_DTLS_SESSION_CLOSED: Chassis 1 R0/0:

```

Per correggerlo, verificare che l'identità configurata sul WLC corrisponda esattamente a una delle SAN incluse nel certificato ISE:

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

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

Verificare che la catena di certificati CA sia importata correttamente nel controller e che `dtls trustpoint server`

configuration uses the Issuer CA trustpoint.

CA sconosciuta segnalata da ISE

Quando ISE non può convalidare i certificati forniti dal WLC, non riesce a creare il tunnel DTLS e le autenticazioni non riescono. Nei registri attivi RADIUS viene visualizzato un errore. Passare a Operazioni>Raggio>Live Log per verificare.

Cisco ISE

Overview	Steps
Event 5450 RADIUS DTLS handshake failed	91030 RADIUS DTLS handshake started
Username	91104 RADIUS DTLS: no need to run Client Identity check
Endpoint Id	91031 RADIUS DTLS: received client hello message
Endpoint Profile	91105 RADIUS DTLS: sent client hello verify request
Authorization Result	91105 RADIUS DTLS: sent client hello verify request
	91031 RADIUS DTLS: received client hello message
	91032 RADIUS DTLS: sent server hello message
	91033 RADIUS DTLS: sent server certificate
	91034 RADIUS DTLS: sent client certificate request
	91035 RADIUS DTLS: sent server done message
	91035 RADIUS DTLS: sent server done message
	91035 RADIUS DTLS: sent server done message
	91036 RADIUS DTLS: received client certificate
	91050 RADIUS DTLS: TLS handshake failed because of an unknown CA in the certificates chain

Authentication Details	
Source Timestamp	2024-07-19 00:34:51.935
Received Timestamp	2024-07-19 00:34:51.935
Policy Server	ise-vbetanco
Event	5450 RADIUS DTLS handshake failed
Failure Reason	91050 RADIUS DTLS: TLS handshake failed because of an unknown CA in the certificates chain
Resolution	Ensure that the certificate authority that signed the client's certificate is correctly installed in the Certificate Store page (Administration > System > Certificates > Certificate Management > Trusted Certificates). Check the OpenSSLErrorMessage and OpenSSLErrorStack for more information. If CRL is configured, check the System Diagnostics for possible CRL downloading faults.
Root cause	RADIUS DTLS: SSL handshake failed because of an unknown CA in the certificates chain

ISE Live Log segnala un errore di handshake DTLS causato da CA sconosciuta

Per correggere il problema, verificare che sia il Certificato intermedio che il Certificato principale, selezionare le caselle di controllo Considera attendibile l'autenticazione del client e Syslog in Amministrazione>Sistema>Certificati>Certificati attendibili.

Verifica revoca in corso

Quando i certificati vengono importati nel WLC, per i nuovi trust point creati il controllo di revoca è abilitato. In questo modo il WLC tenta di cercare un elenco di revoche di certificati non disponibile o raggiungibile e la verifica del certificato non riesce.

Verificare che ogni trust point nel percorso di verifica dei certificati contenga il 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 c
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

```

Risoluzione dei problemi di definizione del tunnel DTLS sull'acquisizione dei pacchetti

Lo switch 9800 WLC offre la funzionalità Embedded Packet Capture (EPC) che consente di acquisire tutto il traffico inviato e ricevuto per una determinata interfaccia. ISE offre una funzione simile, chiamata dump TCP per monitorare il traffico in entrata e in uscita. Se utilizzati contemporaneamente, consentono di analizzare il traffico di impostazione della sessione DTLS dal punto di vista di entrambi i dispositivi.

Per i dettagli sulla configurazione del dump TCP su ISE, consultare la [Cisco Identity Services Engine Administrator Guide](#). Per informazioni sulla configurazione della funzione EPC sul WLC, fare riferimento anche alla sezione [Risoluzione dei problemi dei controller LAN wireless Catalyst 9800](#).

Questo è un esempio di come il tunnel DTLS è stato creato correttamente.

No.	Time	Source	Destination	Protocol	Length	Info
1	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	237	Client Hello
2	2024-10-18 12:04:2...	172.16.18.123	172.16.85.122	DTLSv1.2	106	Hello Verify Request
3	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	269	Client Hello
6	2024-10-18 12:04:2...	172.16.18.123	172.16.85.122	DTLSv1.2	926	Server Hello, Certificate (Fragment), Certificate (Fragment), Certificate (Fragment)
8	2024-10-18 12:04:2...	172.16.18.123	172.16.85.122	DTLSv1.2	608	Certificate (Fragment), Certificate (Fragment), Certificate (Fragment), Certificate
9	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
10	2024-10-18 12:04:2...	172.16.18.123	172.16.85.122	DTLSv1.2	270	Certificate (Fragment)
11	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
12	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
13	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
14	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment) DTLS Tunnel negotiation
15	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
16	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
17	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
18	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
19	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
20	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
21	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
22	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
23	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
24	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Fragment)
25	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate (Reassembled), Client Key Exchange (Fragment)
26	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Client Key Exchange (Reassembled), Certificate Verify (Fragment)
27	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	270	Certificate Verify (Fragment)
28	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	278	Certificate Verify (Reassembled), Change Cipher Spec, Encrypted Handshake Message
29	2024-10-18 12:04:2...	172.16.18.123	172.16.85.122	DTLSv1.2	121	Change Cipher Spec, Encrypted Handshake Message
30	2024-10-18 12:04:2...	172.16.85.122	172.16.18.123	DTLSv1.2	133	Application Data
31	2024-10-18 12:04:2...	172.16.18.123	172.16.85.122	DTLSv1.2	103	Application Data DTLS encrypted RADIUS Messages
48	2024-10-18 12:04:3...	172.16.85.122	172.16.18.123	DTLSv1.2	133	Application Data
49	2024-10-18 12:04:3...	172.16.18.123	172.16.85.122	DTLSv1.2	103	Application Data

Acquisizione di pacchetti di una negoziazione del tunnel DTLS RADIUS e di messaggi crittografati

Le acquisizioni dei pacchetti mostrano come viene stabilito il tunnel DTLS. Se si verifica un problema con la negoziazione, causato da traffico perso tra dispositivi o pacchetti di avviso crittografati DTLS, l'acquisizione del pacchetto consente di identificare il problema.

Informazioni su questa traduzione

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