

在ISE和9800 WLC上配置Radius DTLS

目錄

[簡介](#)

[背景](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[設定](#)

[概觀](#)

[可選-建立WLC和ISE RADIUS DTLS裝置證書](#)

[在openssl.cnf檔案上新增組態區段](#)

[建立WLC裝置證書](#)

[建立ISE裝置證書](#)

[將憑證匯入裝置](#)

[將證書導入到ISE](#)

[將憑證匯入WLC](#)

[配置RADIUS DTLS](#)

[ISE 組態](#)

[WLC配置](#)

[驗證](#)

[驗證憑證資訊](#)

[執行測試驗證](#)

[疑難排解](#)

[WLC報告的未知CA](#)

[ISE報告的未知CA](#)

[撤銷檢查已就緒](#)

[對資料包捕獲上的DTLS隧道建立進行故障排除](#)

簡介

本文檔介紹建立在ISE和9800 WLC之間配置RADIUS DTLS所需的證書的方法。

背景

RADIUS DTLS是RADIUS通訊協定的安全形式，其中RADIUS訊息是透過資料傳輸層安全(DTLS)通道傳送。要在身份驗證伺服器 and 身份驗證器之間建立此隧道，需要一組證書。這組憑證需要設定特定的延伸金鑰使用(EKU)憑證擴充功能，特別是WLC憑證上的使用者端驗證，以及ISE憑證的伺服器驗證和使用者端驗證。

必要條件

需求

思科建議您瞭解以下主題：

- 如何設定9800 WLC(存取點(AP))的基本操作
- 如何使用 OpenSSL 應用程式
- 公開金鑰基礎架構(PKI)和數位憑證

採用元件

本文中的資訊係根據以下軟體和硬體版本：

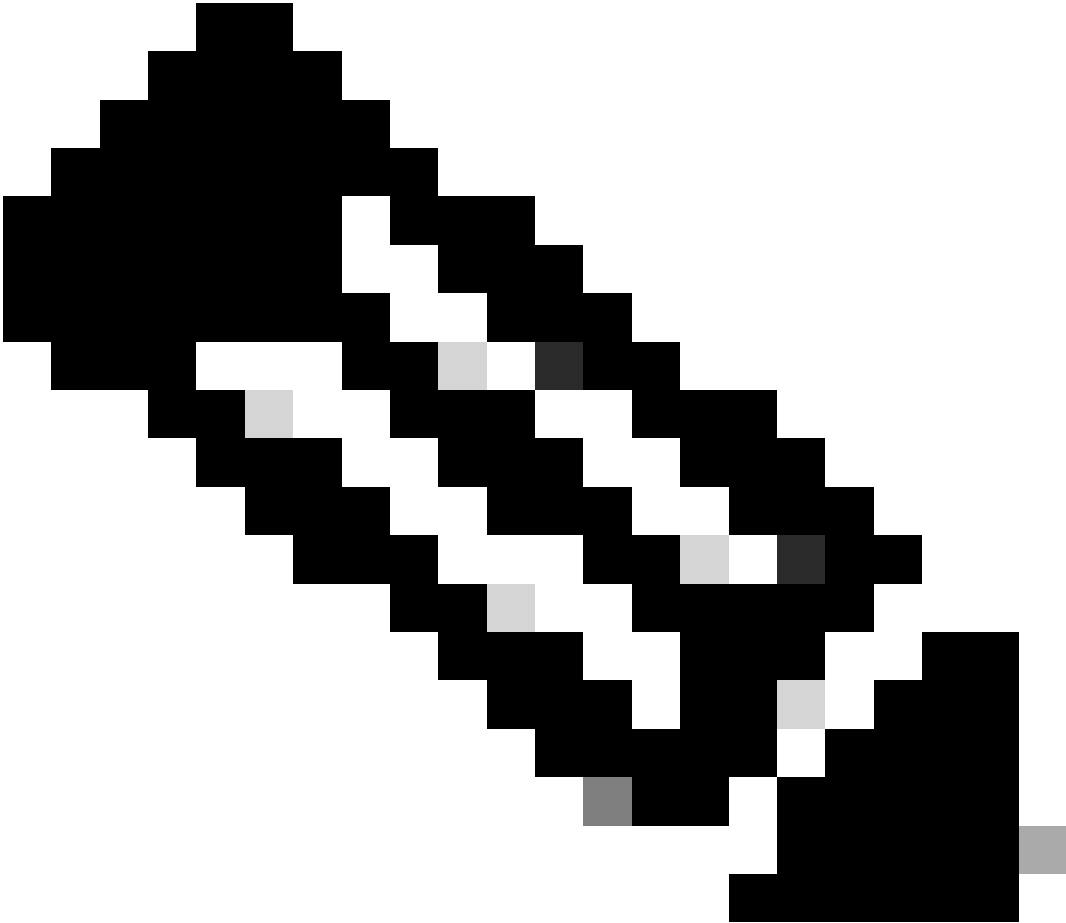
- OpenSSL應用程式 (版本3.0.2)。
- ISE (版本3.1.0.518)
- 9800 WLC (版本17.12.3)

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除 (預設) 的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

設定

概觀

其目的是建立一個具有根CA和中間CA的二級證書頒發機構來簽署終端證書。一旦簽署憑證，就會匯入到WLC和ISE。最後，裝置配置為使用這些證書執行RADIUS DTLS身份驗證。



注意：本文檔使用Linux特定命令來建立和排列檔案。這些指令會詳細說明，如此您便可在其他可使用OpenSSL的作業系統上執行相同的動作。

可選-建立WLC和ISE RADIUS DTLS裝置證書

RADIUS DTLS協定需要在ISE和WLC之間交換證書以建立DTLS隧道。如果您還沒有有效的證書，您可以建立一個本地CA來生成證書，請參閱[在OpenSSL上配置多層證書頒發機構以生成Cisco IOS® XE相容證書](#)，並從開始到步驟結束時執行文檔中概述的步驟 建立中繼CA憑證。

在openssl.cnf檔案上新增組態區段

打開您的openssl.cnf配置檔案，在配置檔案底部，複製並貼上用於生成有效證書簽名請求(CSR)的WLC和ISE部分。

ISE_device_req_ext和WLC_device_req_ext部分都指向將包含在CSR上的SAN清單：

```

#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

```

作為一項安全措施，CA會覆蓋CSR上存在的任何SAN以進行簽名，使未經授權的裝置無法接收其不允許使用的名稱的有效證書。要將SAN重新增加到簽名證書中，請使用subjectAltName引數指向與用於生成CSR的SAN相同的清單SAN。

ISE要求證書上同時存在serverAuth和clientAuth EKU，而WLC僅需要clientAuth。它們將使用extendedKeyUsage引數增加到簽名證書中。

將用於證書簽名的部分複製並貼上到openssl.cnf檔案底部：

```

#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

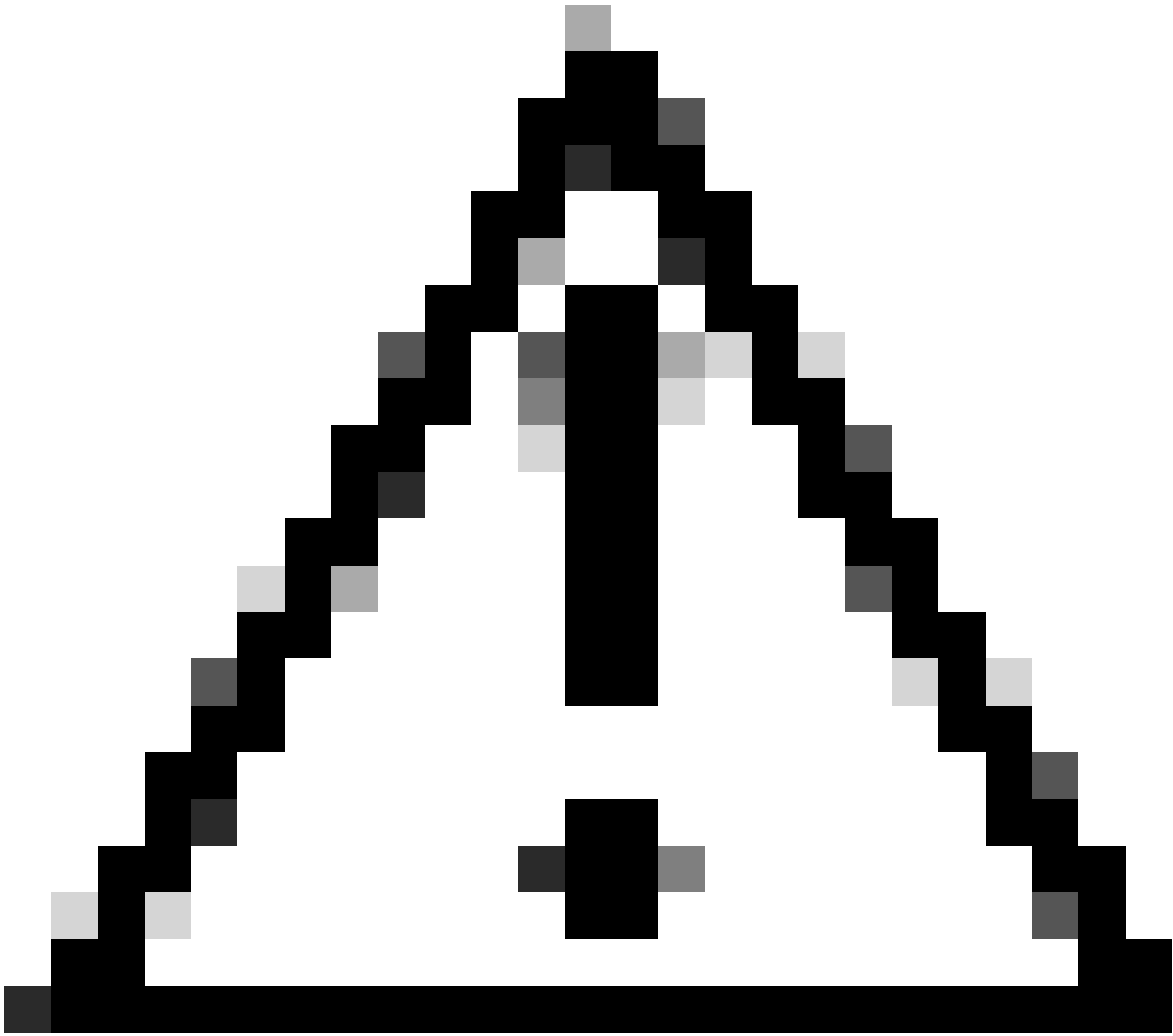
[ WLC_cert ]
basicConstraints=CA:FALSE
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always
#EKU client is needed for RADIUS DTLS on WLC
extendedKeyUsage = clientAuth
subjectAltName = @WLC_alt_names

```

建立WLC裝置證書

在已將OpenSSL安裝在名為IntermCA.db.certs的中間CA證書資料夾中的電腦上建立新資料夾以儲存WLC證書。新資料夾稱為WLC：

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

注意：您在互動式提示上提供的「一般名稱」(CN)必須與openssl.cnf檔案的 [WLC_alt_names]區段上的其中一個「名稱」相同。

使用名為IntermCA的CA對名為WLC.csr的WLC CSR進行簽名，並使用在[WLC_cert] 下定義的副檔名對該CSR進行簽名，並將簽名證書儲存在./IntermCA/IntermCA.db.certs/WLC中。WLC裝置證書稱為WLC.crt：

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

9800 WLC需要使用pfx格式的證書才能導入該證書。建立包含簽署WLC憑證的CA鏈結的新檔案，這稱為certfile：

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

要建立.pfx檔案，請根據WLC版本運行以下命令之一。

對於早於17.12.1的版本：

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

對於版本17.12.1或更高版本：

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

建立ISE裝置證書

建立一個新資料夾，以便在已安裝有OpenSSL的中間CA證書資料夾(名為IntermCA.db.certs)的電腦上儲存ISE證書。新資料夾稱為ISE：

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

在openssl.cnf檔案的[ISE_alt_names]部分修改DNS引數。更改為您所需值提供的示例名稱，這些值將填充WLC證書的SAN欄位：

```
[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
```

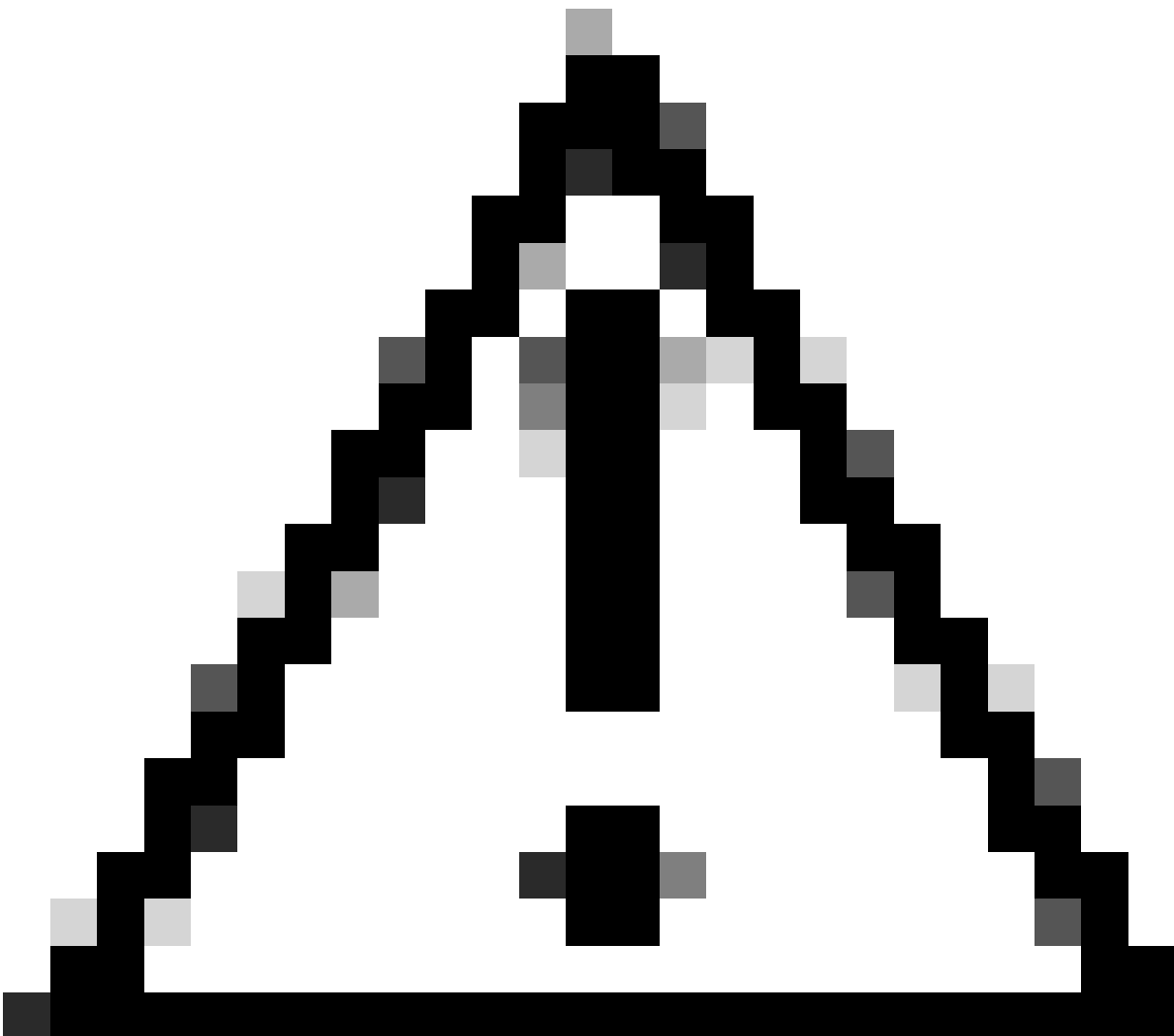
使用ISE_device_req_ext部分的資訊為SAN建立ISE私鑰和ISE CSR：

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

OpenSSL會開啟互動式提示，提示您輸入辨別名稱(DN)詳細資訊：

```
-----  
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
```

ISE證書可分辨名稱互動式提示



注意：您在互動式提示中提供的CN必須與openssl.cnf檔案的[ISE_alt_names]部分中的一個Names完全相同。

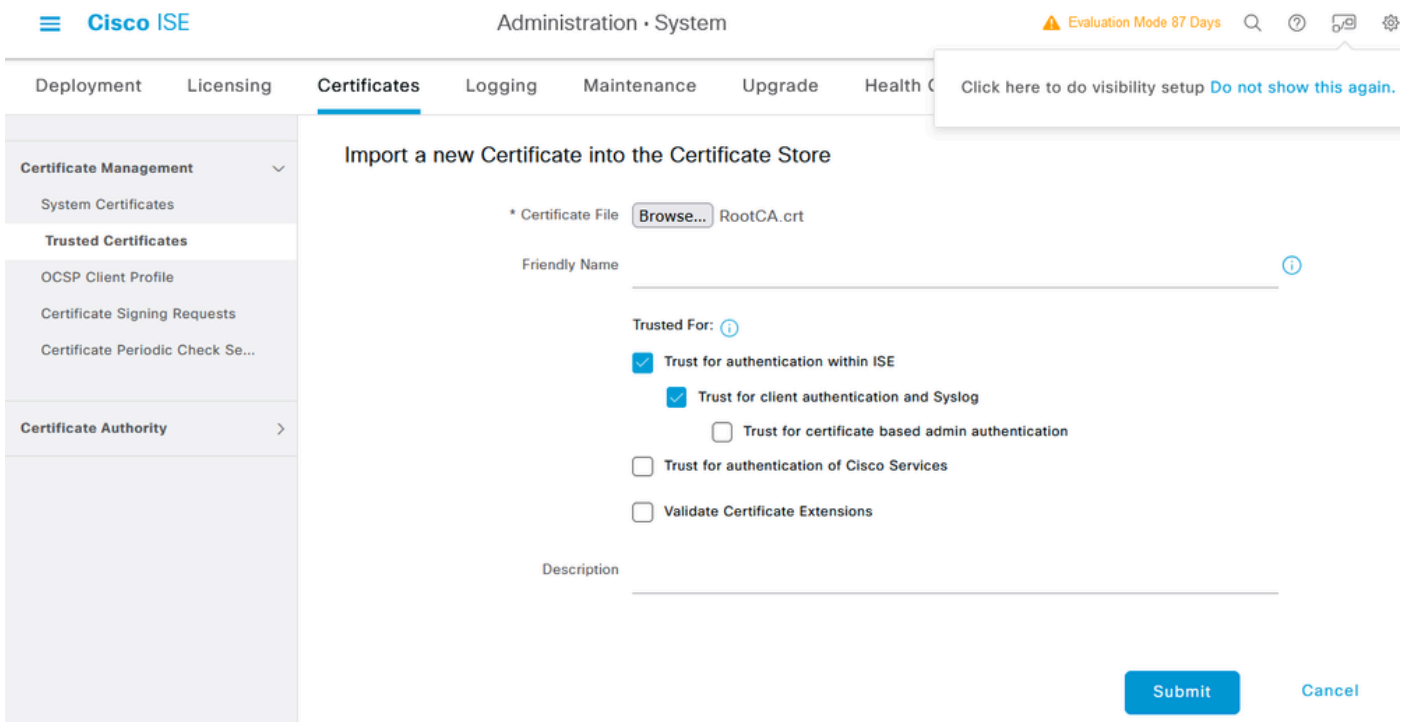
使用名為IntermCA的CA對名為ISE.csr的ISE CSR進行簽名，使用的擴展在[ISE_cert] 下，並將簽名證書儲存在./IntermCA/IntermCA.db.certs/WLC中。ISE裝置證書稱為ISE.crt：

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

將憑證匯入裝置

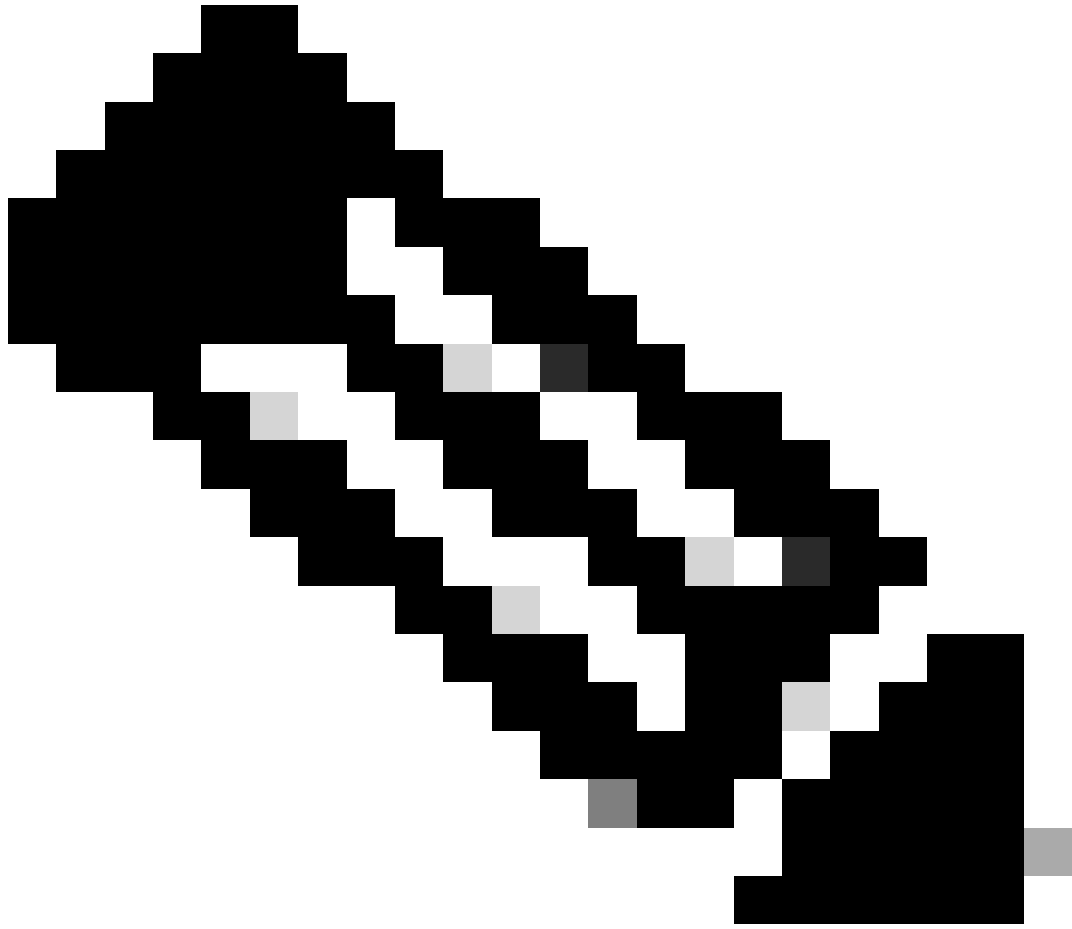
將證書導入到ISE

1. 將根CA證書從ISE證書鏈導入到受信任的證書儲存中。
2. 導航到管理>系統>證書>受信任證書。
3. 按一下「瀏覽」並選擇Root.crt檔案。
4. 選中Trust for authentication within ISE 以及Trust for client authentication and Syslog 覈取方塊，然後按一下Submit：



ISE根CA證書導入對話方塊

如果中間憑證存在，請執行相同的動作。



注意：對屬於ISE證書驗證鏈的任何CA證書重複這些步驟。請一律從根CA憑證開始，並以鏈結中最低的中間CA憑證結束。

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: ⓘ

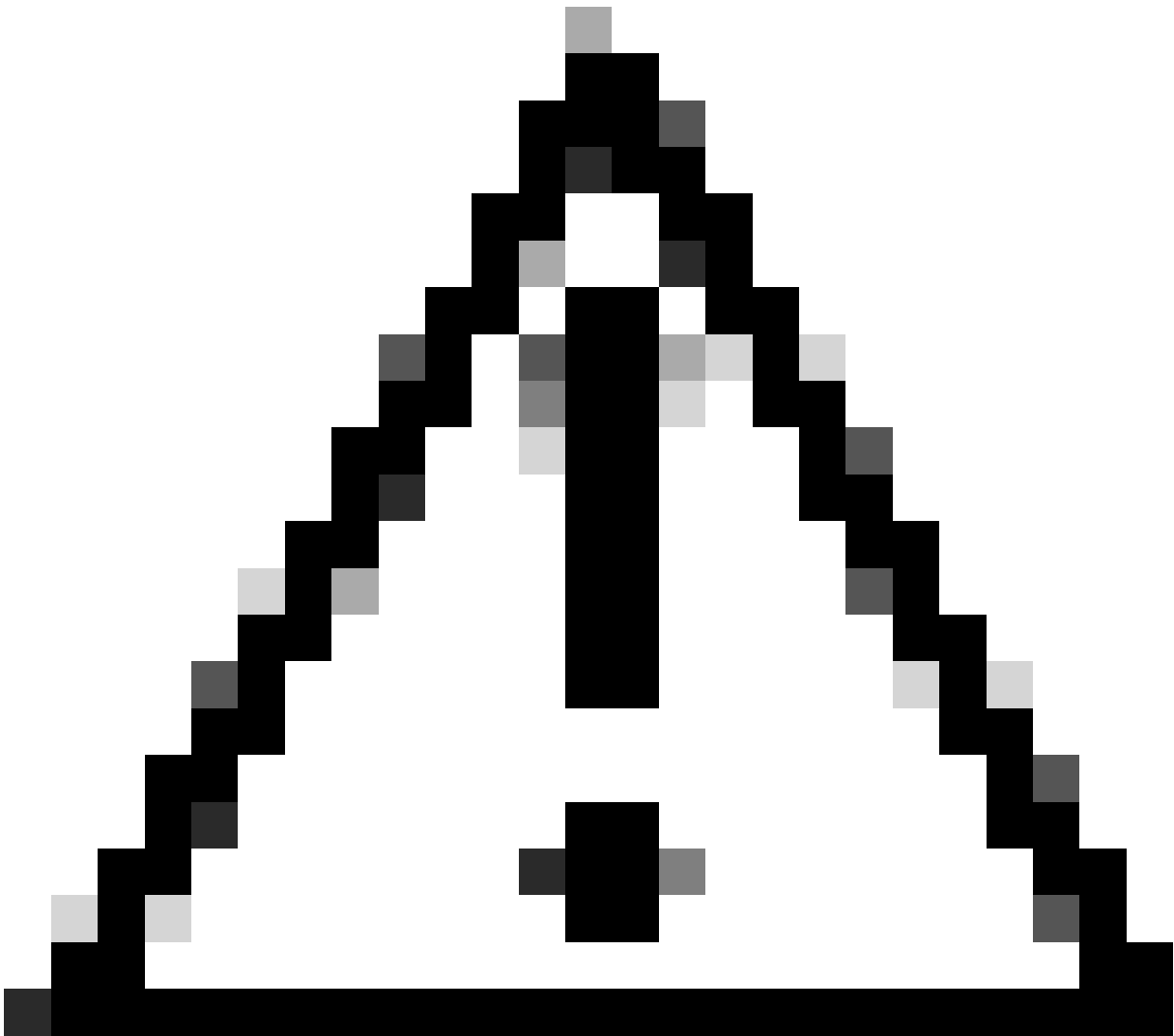
- Trust for authentication within ISE
 - Trust for client authentication and Syslog
 - Trust for certificate based admin authentication
- Trust for authentication of Cisco Services
- Validate Certificate Extensions

Description

Submit

Cancel

ISE中間CA證書導入對話方塊



注意：如果ISE證書和WLC證書是由不同的CA頒發的，則還必須導入屬於WLC證書鏈的所有CA證書。在您導入這些CA證書之前，ISE不會接受DTLS證書交換上的WLC證書。

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

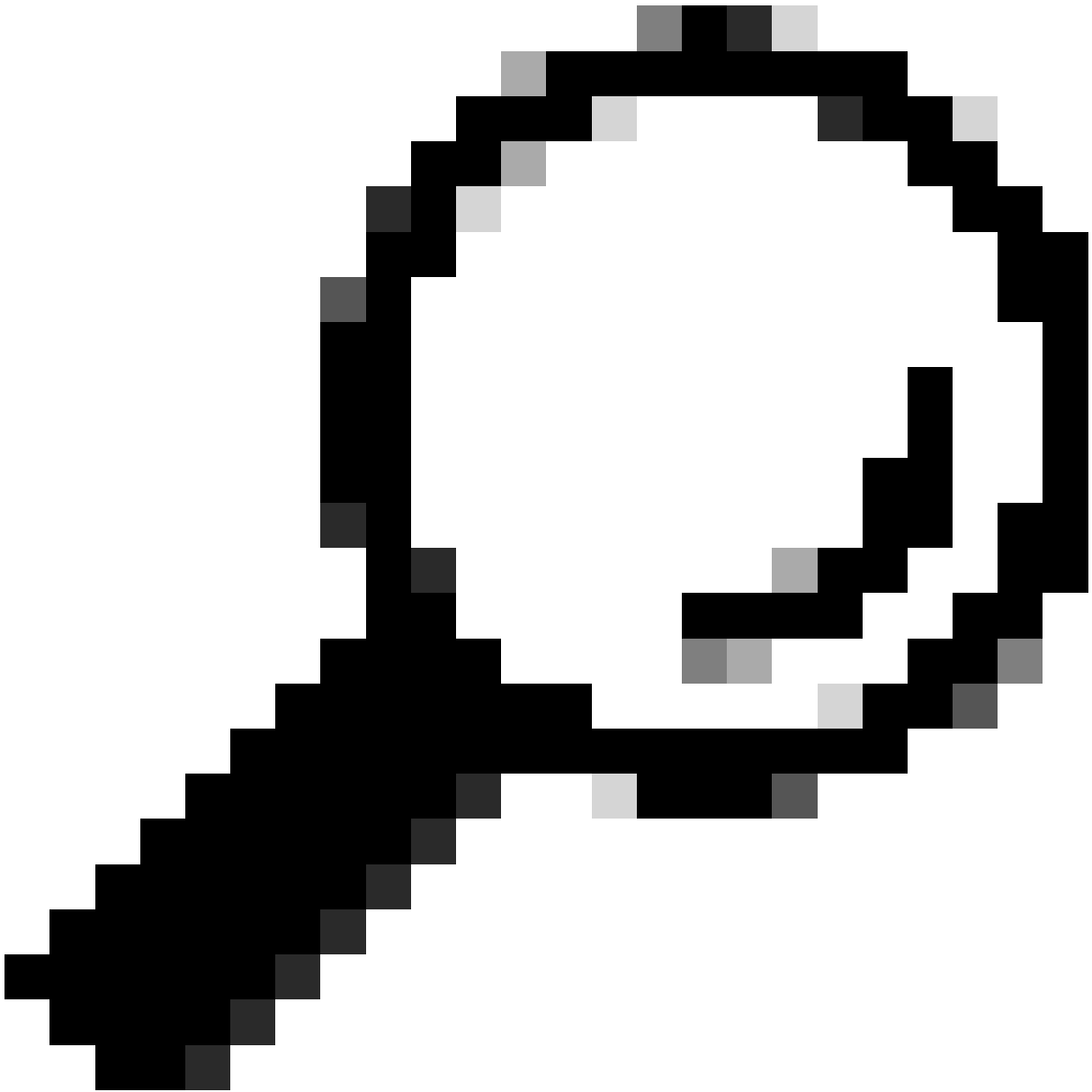
Allow Wildcard Certificates ⓘ

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

ISE裝置證書導入選單



提示：在此步驟中，您只需導入ISE裝置證書。此證書是建立DTLS隧道的ISE交換。沒有必要匯入WLC裝置憑證和私密金鑰，因為會使用之前匯入的CA憑證來驗證WLC憑證。

將憑證匯入WLC

1. 導航到WLC上的Configuration > Security > PKI Management，然後轉到Add Certificate頁籤。
2. 按一下Import PKCS12 Certificate 下拉選單，將傳輸型別設定為Desktop (HTTPS)。
3. 按一下Select File 按鈕，然後選擇您之前準備的.pfx檔案。
4. 鍵入導入口令，然後最後按一下Import。

Import PKCS12 Certificate

Transport Type	Desktop (HTTPS) ▾
Source File Path*	<div>➤ Select File</div> <div>WLC.pfx ✕</div>
Certificate Password*	●●●●●●●●
<div>Import</div>	

WLC證書導入對話方塊

有關導入過程的詳細資訊，請參閱[在Catalyst 9800 WLC上生成和下載CSR證書](#)。

如果WLC沒有可透過網路檢查的證書撤銷清單，請停用每個自動建立的信任點內的撤銷檢查：

```
9800#configure terminal
```

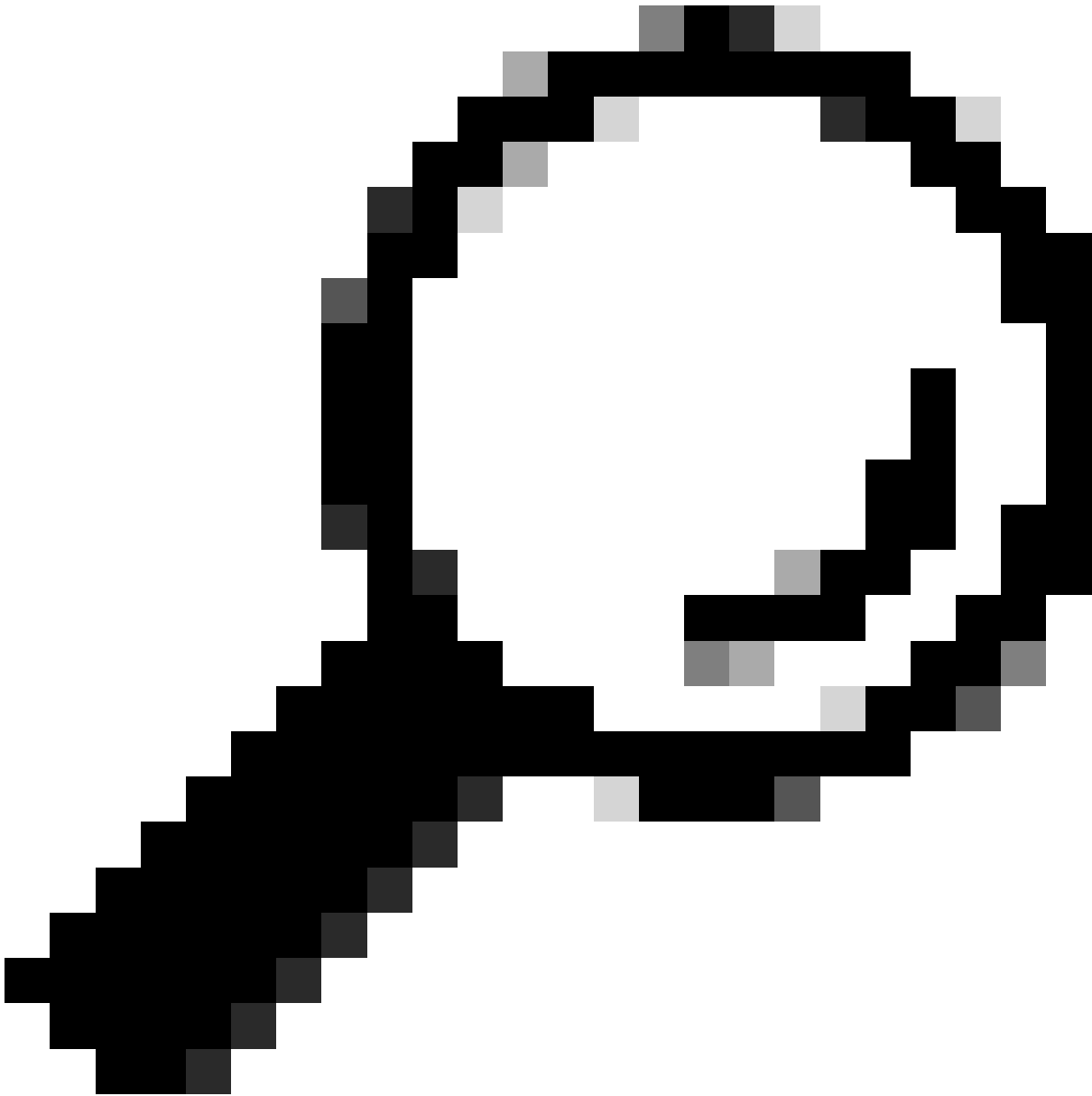
```
9800(config)#crypto pki trustpoint WLC.pfx
9800(config)#revocation-check none
9800(config)#exit
```

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



注意：如果用在OpenSSL上配置多級CA以生成Cisco IOS XE證書文檔在OpenSSL上建立多級CA，則必須停用撤銷檢查，因為未建立CRL伺服器。

自動導入會建立包含WLC證書及其CA證書的必要信任點。



提示：如果WLC證書由與ISE證書相同的CA頒發，您可以使用從WLC證書導入中自動建立的相同信任點。無需單獨導入ISE證書。

如果WLC證書由ISE證書以外的其他CA頒發，您還需要將ISE CA證書導入WLC以使WLC信任ISE裝置證書。

為根CA建立新的信任點並導入ISE根CA：

```
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-----
```

導入ISE CA鏈上的下一個中間CA證書，即根CA頒發的CA證書：

```
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-----
```

鏈上的每個其他CA都需要一個單獨的信任點。鏈中的每個信任點都必須引用包含要使用chain-validation continue <Issuer trustpoint name>命令導入的證書的頒發者證書的信任點。

導入與您的CA鏈包含數量相匹配的CA證書。在導入ISE裝置證書的頒發者CA之後，請記下此信任點的名稱。

您無需在WLC上導入ISE裝置證書，RADIUS DTLS即可正常工作。

配置RADIUS DTLS

ISE 組態

將WLC作為網路裝置增加到ISE，為此，請導航到管理>網路資源>網路裝置>增加輸入裝置名稱和來源RADIUS流量的WLC介面的IP。通常是無線管理介面IP。向下滾動並選中RADIUS Authentication Settings以及DTLS Required 並按一下Submit：

Network Devices

Default Device

Device Security Settings

Network Devices List > New Network Device

Network Devices

Name Radsecwlc

Description

IP Address * IP : 172.16.5.11 / 32

Device Profile Cisco

Model Name

Software Version

Network Device Group

Location All Locations Set To Default

IPSEC Is IPSEC Device Set To Default

Device Type All Device Types Set To Default

RADIUS Authentication Settings

新網路裝置配置

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

ISE上網路裝置的RADIUS DTLS設定

WLC配置

定義新的RADIUS伺服器以及ISE IP地址和預設埠RADIUS DTLS。此配置僅適用於CLI：

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

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

Radius DTLS必須使用共用金鑰radius/dtls，9800 WLC將忽略此金鑰以外的任何已配置金鑰：

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

使用 `dtls trustpoint client`

命令配置信任點，該信任點包含DTLS隧道要交換的WLC裝置證書。

使用 `dtls trustpoint server`

命令配置信任點，該信任點包含ISE裝置證書的頒發者CA。

只有當WLC和ISE證書由同一CA頒發時，客戶端和伺服器信任點名稱才相同：

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

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

配置WLC以檢查ISE證書上是否存在一個主題備用名稱(SAN)。此配置必須與證書的SAN欄位中存在的一個SAN完全匹配。

9800 WLC不會對SAN欄位執行基於正規表示式的匹配。例如，這表示萬用字元憑證的命令 `dtls match-server-identity hostname *.example.com`(其SAN欄位中包含`*.example.com`)是正確的，但SAN欄位中包含 www.example.com的證書的相同命令不正確。

WLC不會針對任何名稱伺服器檢查此名稱：

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

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

建立新的伺服器群組，以使用新的RADIUS DTLS進行驗證：

```
9800(config)#aaa group server radius Radsec
```

```
9800(config-sg-radius)#server name ISE
```

```
9800(config-sg-radius)#exit
```

從此以後，您可以像使用WLC上的任何其他伺服器組一樣使用此伺服器組。請參閱[在Catalyst 9800無線控制器系列上配置802.1X身份驗證](#)，以便使用此伺服器進行無線客戶端身份驗證。

驗證

驗證憑證資訊

要驗證已建立證書的證書資訊，請在Linux終端上運行命令：

```
openssl x509 -in
```

```
-text -noout
```

它會顯示完整的憑證資訊。這對於確定給定證書的頒發者CA或證書是否包含所需的EKU和SAN非常有用：

```

Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 2 (0x2)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C = MX, ST = CDMX, L = CDMX, O = Cisco lab, OU = Cisco Wireless, CN = Intermediate.example.com
    Validity
      Not Before: Jul 18 19:14:57 2024 GMT
      Not After : Apr 14 19:14:57 2027 GMT
    Subject: C = MX, ST = CDMX, L = CDMX, O = Cisco lab, OU = Cisco Wireless, CN = WLC.example.com
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      Public-Key: (2048 bit)
      Modulus:
        00:b1:10:7d:6c:6c:14:2f:18:a6:0b:69:d9:60:03:
        56:2d:48:22:f0:42:10:65:44:24:3b:54:e1:4b:87:
        b8:ab:c5:5f:f6:a1:a3:5e:f6:3c:c5:45:cc:01:6d:
        df:e8:a7:81:28:50:44:54:4c:af:a0:56:cf:06:be:
        10:7e:e2:46:42:ea:3c:b9:d4:03:75:08:84:70:36:
        bb:3d:95:3b:e2:86:e6:f7:d9:4d:00:28:c4:3c:cb:
        f8:6d:37:5c:89:28:c1:75:b1:7e:fa:bd:91:cf:8e:
        5c:a2:37:4f:71:da:6a:04:ee:ba:68:bf:4d:f2:d3:
        ae:aa:13:42:3b:ff:a0:b3:65:c9:ff:f6:9a:06:d7:
        6c:08:10:e0:b9:d8:ca:93:2d:e5:5d:7b:74:cd:93:
        68:b1:46:c7:35:d7:6b:0f:a6:ae:34:e6:23:d1:c8:
        d3:bf:c0:85:ab:2d:02:a8:dd:54:77:e3:32:61:4e:
        33:58:b0:62:12:82:42:ae:2b:69:f0:5f:0c:90:c7:
        9c:ef:b9:9c:fc:29:e2:2c:cb:b4:a9:01:fa:5d:3c:
        97:11:67:cc:25:96:01:3d:26:1a:43:34:bd:43:b0:
        a0:f1:ec:a0:c7:98:ad:32:32:99:9c:6b:61:af:57:
        53:ee:20:cc:d5:ed:db:1c:5c:65:51:42:8c:28:bf:
        62:bf
      Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Basic Constraints:
        CA:FALSE
      X509v3 Subject Key Identifier:
        87:89:CA:28:06:95:D5:CE:7C:66:B4:75:81:AA:D4:19:EC:43:01:BB
      X509v3 Authority Key Identifier:
        keyid:2B:08:D8:4C:23:72:5B:62:03:EA:44:F6:9E:D9:F7:75:2E:64:97:DE
        DirName:/C=MX/ST=CDMX/L=CDMX/O=Cisco lab/OU=Cisco Wireless/CN=RootCA
        serial:01
      X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication
      X509v3 Subject Alternative Name:
        DNS:WLC.example.com, DNS:WLC2.example.com
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:

```

Cisco IOS XE裝置證書資訊，如OpenSSL所示

執行測試驗證

從WLC中，您可以使用命令測試Radius DTLS功能 `test aaa group`

new-code

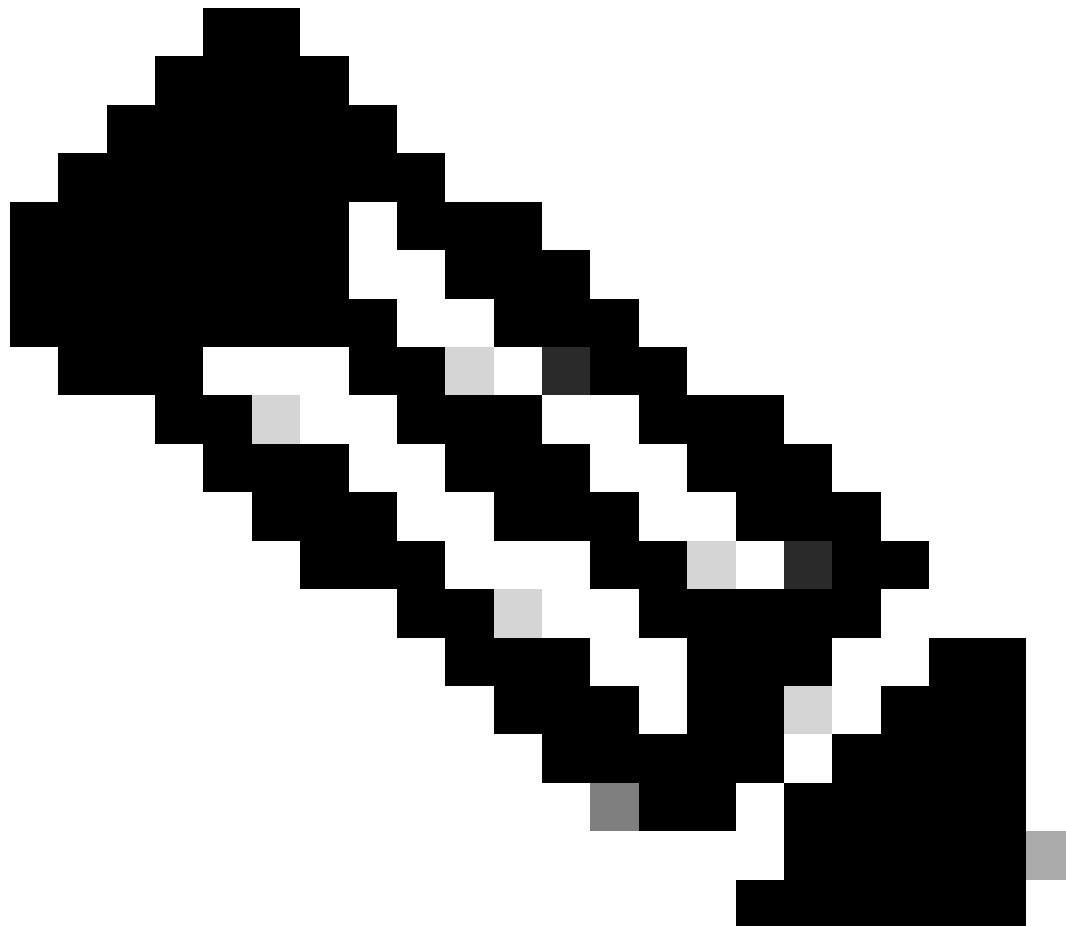
```

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

```

USER ATTRIBUTES

```
username          0  "testuser"
```



注意：test命令上的訪問拒絕輸出表示WLC收到Access-Reject RADIUS消息，在這種情況下，RADIUS DTLS正在運行。但是，這也可能表示無法建立DTLS隧道。test命令無法區分這兩種情況，請參閱故障排除部分以確定是否存在問題。

疑難排解

要檢查身份驗證失敗的原因，可以在執行測試身份驗證之前啟用這些命令。

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


以下是在啟用調試的情況下成功進行身份驗證的輸出：

```
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_SOCKET_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 [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

```

WLC報告的未知CA

當WLC無法驗證ISE提供的證書時，它無法建立DTLS隧道，身份驗證失敗。

以下是出現這種情況時的調試消息示例：

```
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_SOCKET_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_SOCKET_ADDR: Success
Jul 19 00:59:09.707: RADIUS_RADSEC_SET_LOCAL_SOCKET: Success
Jul 19 00:59:09.707: RADIUS_RADSEC_SOCKET_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_SOCKET_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_SOCKET_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_SOCKET_EVENT: Handle socket event for TLS handshake(172.16.18.
```

```
Ju1 19 00:59:09.720: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Ju1 19 00:59:09.722: RADIUS_RADSEC_HS_CONTINUE: TLS handshake failed!
Ju1 19 00:59:09.722: RADIUS_RADSEC_UNQUEUE_WAIT_Q: Success Server(172.16.18.123)/Id(13)
Ju1 19 00:59:09.722: RADIUS_RADSEC_FAILOVER_HANDLER:Failng-over to new server = 0x0
Ju1 19 00:59:09.722: RADIUS_RADSEC_UNQUEUE_WAIT_Q: Empty Server(172.16.18.123)/Id(-1)
Ju1 19 00:59:09.722: RADIUS_RADSEC_FAILOVER_HANDLER: no more data available
Ju1 19 00:59:09.722: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Ju1 19 00:59:09.722: RADIUS_RADSEC_CONN_CLOSE: Cleaned up timers for Radius RADSEC ctx
Ju1 19 00:59:09.722: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(0) generated for sock(0)
Ju1 19 00:59:09.722: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(0) generated for sock(0)
Ju1 19 00:59:09.722: RADIUS_RADSEC_HASH_KEY_DEL_CTX: remove [radius_radsec ctx(0x7522CE91BAC0)] succee
Ju1 19 00:59:09.722: RADIUS_RADSEC_CONN_CLOSE: Hash table entry removed for RADSEC sock ctx
Ju1 19 00:59:09.723: RADIUS_RADSEC_CONN_CLOSE: Success
Ju1 19 00:59:09.723: RADIUS_RADSEC SOCK_TLS_EVENT_HANDLE: Failed to complete TLS handshake <-----D
Ju1 19 00:59:09.723: RADIUS_RADSEC_STOP_TIMER: Stopped (172.16.18.123/2083)
Ju1 19 00:59:09.723: RADIUS_RADSEC_CONN_CLOSE: Cleaned up timers for Radius RADSEC ctx
Ju1 19 00:59:09.723: RADIUS_RADSEC_GENERATE_HASHKEY: hash key(-1) generated for sock(-1)
Ju1 19 00:59:09.723: RADIUS_RADSEC_GENERATE_HASHBUCKET: hash bucket(-1) generated for sock(-1)
uwu-9800#
Ju1 19 00:59:09.723: RADIUS_RADSEC_HASH_KEY_DEL_CTX: remove [radius_radsec ctx(0x7522CE91BAC0)] succee
Ju1 19 00:59:09.723: RADIUS_RADSEC_CONN_CLOSE: Hash table entry removed for RADSEC sock ctx
Ju1 19 00:59:09.723: RADIUS_RADSEC_CONN_CLOSE: Success
Ju1 19 00:59:09.723: RADIUS_RADSEC SOCK_TLS_EVENT_HANDLE: Error
Ju1 19 00:59:09.723: RADIUS_RADSEC_PROCESS SOCK_EVENT: failed to hanlde radsec hs event
Ju1 19 00:59:09.723: RADIUS/DECODE: No response from radius-server; parse response; FAIL
Ju1 19 00:59:09.723: RADIUS/DECODE: Case error(no response/ bad packet/ op decode);parse response; FAIL
Ju1 19 00:59:09.723: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event
Ju1 19 00:59:10.718: %RADSEC_AUDIT_MESSAGE-3-FIPS_AUDIT_FCS_RADSEC_SERVER_CERTIFICATE_VALIDATION_FAILURE
Ju1 19 00:59:10.718: %RADSEC_AUDIT_MESSAGE-3-FIPS_AUDIT_FCS_RADSEC_SERVER_IDENTITY_CHECK_FAILURE: Chassis
Ju1 19 00:59:10.718: %RADSEC_AUDIT_MESSAGE-6-FIPS_AUDIT_FCS_DTLS_SESSION_CLOSED: Chassis 1 R0/0:
```

要更正此問題，請確保WLC上配置的身份與ISE證書中包含的其中一個SAN完全匹配：

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

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

請確定已在控制器上正確匯入CA憑證鏈結， dtls trustpoint server

configuration uses the Issuer CA trustpoint.

ISE報告的未知CA

當ISE無法驗證WLC提供的證書時，它無法建立DTLS隧道，身份驗證失敗。這會在RADIUS即時日誌中顯示為錯誤。導航到操作>Radius>即時日誌進行驗證。

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即時日誌報告由於未知CA導致的DTLS握手失敗

要更正此問題，請確保「Intermediate Certificates」和「Root Certificates」，並在「Administration>System>Certificates>Trusted certificates」下選中Trust for client authentication and Syslog覈取方塊。

撤銷檢查已就緒

當證書導入到WLC時，新建立的信任點啟用了撤銷檢查。這會使WLC嘗試搜尋無法使用或無法連線的憑證撤銷清單，而且憑證驗證失敗。

確保證書驗證路徑中的每個信任點都包含命令 `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_SOCKET_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)] success
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)] success
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 handle radsec hs event
Jul 17 21:50:39.070: RADIUS_RADSEC_CLIENT_PROCESS: Got Socket Event

```

對資料包捕獲上的DTLS隧道建立進行故障排除

9800 WLC提供嵌入式封包擷取(EPC)功能，此功能可讓您擷取指定介面的所有已傳送和已接收流量。ISE提供稱為TCP轉儲的類似功能來監控傳入和傳出流量。同時使用時，它們允許您從兩個裝置的角度分析DTLS會話建立流量。

有關在ISE上配置TCP轉儲的詳細步驟，請參閱[Cisco身份服務引擎管理員指南](#)。有關在WLC上配置EPC功能的資訊，另請參閱[Catalyst 9800無線LAN控制器故障排除](#)。

這是一個成功建立DTLS隧道的示例。

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	172.16.85.122	172.16.18.123	DTLSv1.2	608	Certificate (Fragment), Certificate (Fragment), Certificate (Fragment), Certificate (Fragment)
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	172.16.85.122	172.16.18.123	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
48	2024-10-18 12:04:3	172.16.85.122	172.16.18.123	DTLSv1.2	133	Application Data DTLS encrypted RADIUS Messages
49	2024-10-18 12:04:3	172.16.18.123	172.16.85.122	DTLSv1.2	103	Application Data

RADIUS DTLS通道交涉和加密訊息的封包擷取

資料包捕獲顯示DTLS隧道建立過程。如果協商有問題，例如裝置之間流量丟失或DTLS加密警報資料包丟失，資料包捕獲將幫助您辨識問題。

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。