Troubleshoot Certificate Installation on WLC

Contents

Introduction

This document describes the issues caused by the use of 3rd party certificates on the Wireless LAN Controller (WLC).

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Wireless LAN Controller (WLC)
- Public Key Infrastructure (PKI)
- X.509 Certificates

Components Used

The information in this document is based on these software and hardware versions:

- 3504 WLC with firmware version 8.10.105.0
- OpenSSL 1.0.2p for command line tool
- Windows 10 machine
- Certificate chain from private lab Certificate Authority (CA) with three certificates (leaf, Intermediate, Root)
- Trivial File Transfer Protocol (TFTP) Server for file transfer.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

On AireOS WLC, you can install 3rd party certificates to be used for WebAuth and WebAdmin. At installation, the WLC expects a single PEM (Privacy Enhanced Mail) formatted file with all certificates in the chain all the way to the Root CA certificate and the private key. Details about this procedure are documented in [Generate CSR for Third-Party Certificates and Download Chained Certificates to the WLC]().

This document expands and shows in more detail the most common installation errors with debug examples and resolution for each scenario. Debug outputs used throughout this document are from **debug transfer all enable** and **debug pm pki enable** enabled on the WLC. TFTP was used to transfer the certificates file.

Troubleshoot

Scenario 1. Password Provided to Decrypt the Private Key is Incorrect or no Password was Provided

<#root>

```
*TransferTask: Apr 21 03:51:20.737:

Add ID Cert: Adding certificate & private key using password check123


*TransferTask: Apr 21 03:51:20.737:

Add Cert to ID Table: Adding certificate (name: bsnSslWebauthCert) to ID table using password check123


*TransferTask: Apr 21 03:51:20.737: Add Cert to ID Table: Decoding PEM-encoded Certificate (verify: YES)
*TransferTask: Apr 21 03:51:20.737: Decode & Verify PEM Cert: Cert/Key Length was 0, so taking string le
*TransferTask: Apr 21 03:51:20.737: Decode & Verify PEM Cert: Cert/Key Length 6276 & VERIFY
*TransferTask: Apr 21 03:51:20.741: Decode & Verify PEM Cert: X509 Cert Verification return code: 1
*TransferTask: Apr 21 03:51:20.741: Decode & Verify PEM Cert: X509 Cert Verification result text: ok
*TransferTask: Apr 21 03:51:20.741:

Add Cert to ID Table: Decoding PEM-encoded Private Key using password check123


*TransferTask: Apr 21 03:51:20.799:

Decode PEM Private Key: Error reading Private Key from PEM-encoded PKCS12 bundle using password check12


*TransferTask: Apr 21 03:51:20.799: Add ID Cert: Error decoding / adding cert to ID cert table (verifyCh
*TransferTask: Apr 21 03:51:20.799: Add WebAuth Cert: Error adding ID cert
*TransferTask: Apr 21 03:51:20.799:

RESULT_STRING: Error installing certificate.
```

Solution: Ensure that the correct password is provided so that the WLC can decode it for installation.

Scenario 2. No Intermediate CA Certificate in the Chain

<#root>

```
*TransferTask: Apr 21 04:34:43.319: Add ID Cert: Adding certificate & private key using password Cisco12
*TransferTask: Apr 21 04:34:43.319: Add Cert to ID Table: Adding certificate (name: bsnSslWebauthCert) t
*TransferTask: Apr 21 04:34:43.319: Add Cert to ID Table: Decoding PEM-encoded Certificate (verify: YES)
```

```
*TransferTask: Apr 21 04:34:43.319: Decode & Verify PEM Cert: Cert/Key Length was 0, so taking string le
*TransferTask: Apr 21 04:34:43.319: Decode & Verify PEM Cert: Cert/Key Length 4840 & VERIFY
*TransferTask: Apr 21 04:34:43.321: Decode & Verify PEM Cert: X509 Cert Verification return code: 0
*TransferTask: Apr 21 04:34:43.321:

 Decode & Verify PEM Cert: X509 Cert Verification result text: unable to get local issuer certificate


*TransferTask: Apr 21 04:34:43.321:

Decode & Verify PEM Cert: Error in X509 Cert Verification at 0 depth: unable to get local issuer certifi


*TransferTask: Apr 21 04:34:43.321: Add Cert to ID Table: Error decoding (verify: YES) PEM certificate
*TransferTask: Apr 21 04:34:43.321: Add ID Cert: Error decoding / adding cert to ID cert table (verifyCh
*TransferTask: Apr 21 04:34:43.321: Add WebAuth Cert: Error adding ID cert
*TransferTask: Apr 21 04:34:43.321: RESULT_STRING: Error installing certificate.
```

Solution: Validate the **Issuer** and **X509v3 Authority Key Identifier** fields from the WLC certificate to validate the CA certificate that signed the certificate. If the Intermediate CA certificate was provided by the CA, that can be used to validate against. Otherwise, request the certificate to your CA.

This OpenSSL command can be used to validate these details on each certificate:

<#root>

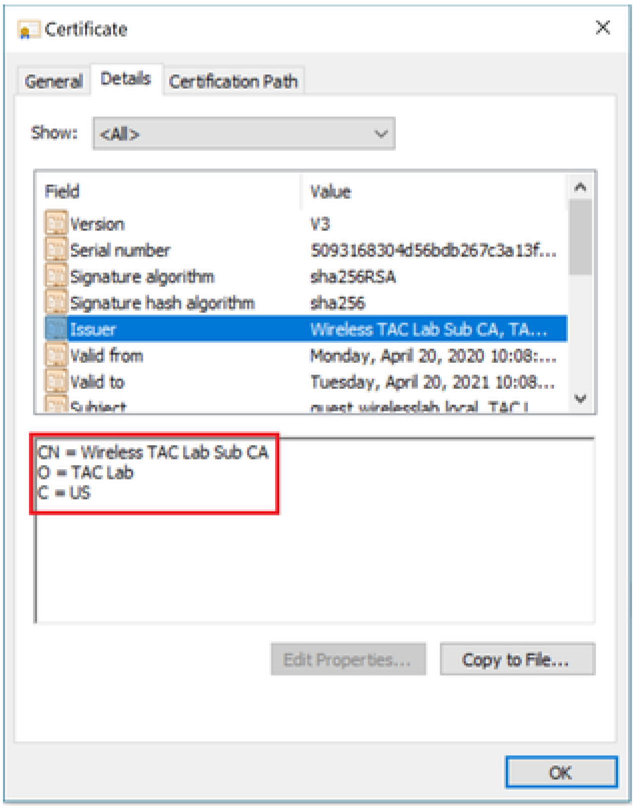
```
>

 openssl x509 -in

wlc.crt

-text -noout



Certificate:
Data:
Version: 3 (0x2)
Serial Number:
50:93:16:83:04:d5:6b:db:26:7c:3a:13:f3:95:32:7e
Signature Algorithm: sha256WithRSAEncryption

Issuer: C=US, O=TAC Lab, CN=Wireless TAC Lab Sub CA


Validity
Not Before: Apr 21 03:08:05 2020 GMT
Not After : Apr 21 03:08:05 2021 GMT
Subject: C=US, O=TAC Lab, CN=guest.wirelesslab.local

...

X509v3 extensions:

X509v3 Authority Key Identifier:



keyid:27:69:2E:C3:2F:20:5B:07:14:80:E1:86:36:7B:E0:92:08:4C:88:12
```

<#root>

```
>

 openssl x509 -in

int-ca.crt

 -text -noout



Certificate:
Data:
Version: 3 (0x2)
Serial Number:
d1:ec:26:0e:be:f1:aa:65:7b:4a:8f:c7:d5:7f:a4:97
Signature Algorithm: sha256WithRSAEncryption
Issuer: C=US, O=TAC Lab, CN=Wireless TAC Lab Root CA
Validity
Not Before: Apr 21 02:51:03 2020 GMT
Not After : Apr 19 02:51:03 2030 GMT

Subject: C=US, O=TAC Lab, CN=Wireless TAC Lab Sub CA



...


X509v3 Subject Key Identifier:



27:69:2E:C3:2F:20:5B:07:14:80:E1:86:36:7B:E0:92:08:4C:88:12
```

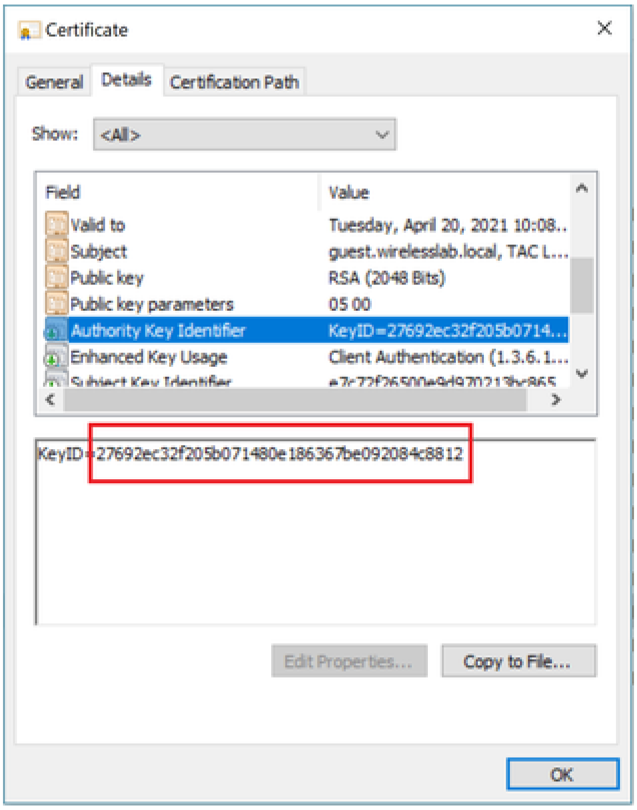
Alternatively if you use Windows, give the certificate a **.crt** extension and double click to validate these details.

WLC certificate:



Field	Value
Version	V3
Serial number	5093168304d56bdb267c3a13f...
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	Wireless TAC Lab Sub CA, TA...
Valid from	Monday, April 20, 2020 10:08:...
Valid to	Tuesday, April 20, 2021 10:08...
Subject	guest.wirelesslab.local, TAC L

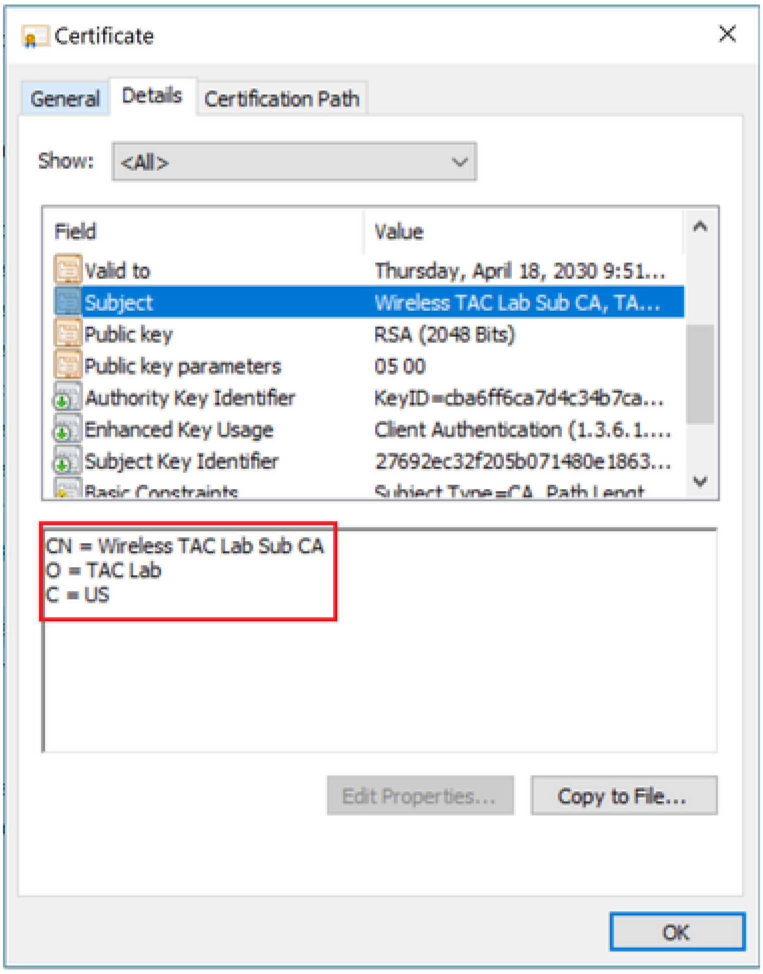
CN = Wireless TAC Lab Sub CA
O = TAC Lab
C = US



Field	Value
Valid to	Tuesday, April 20, 2021 10:08..
Subject	guest.wirelesslab.local, TAC L...
Public key	RSA (2048 Bits)
Public key parameters	05 00
Authority Key Identifier	KeyID=27692ec32f205b0714...
Enhanced Key Usage	Client Authentication (1.3.6.1...
Subject Key Identifier	e7c72f26500e9d970213bc865

KeyID=27692ec32f205b071480e186367be092084c8812

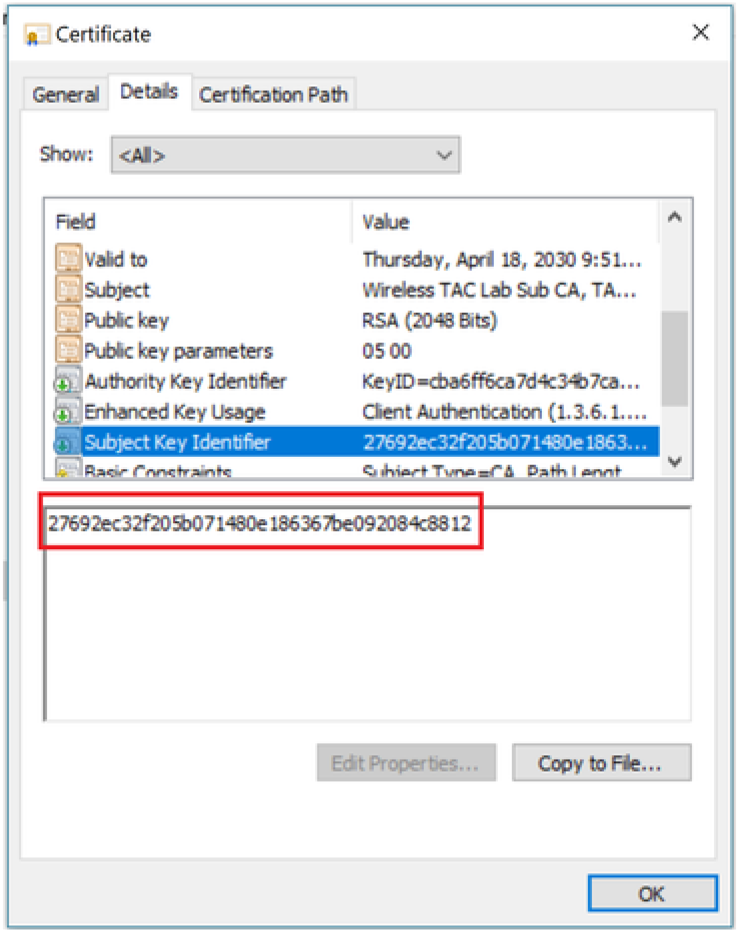
Intermediate CA certificate:



Show: <All>

Field	Value
Valid to	Thursday, April 18, 2030 9:51...
Subject	Wireless TAC Lab Sub CA, TA...
Public key	RSA (2048 Bits)
Public key parameters	05 00
Authority Key Identifier	KeyID=cba6ff6ca7d4c34b7ca...
Enhanced Key Usage	Client Authentication (1.3.6.1....
Subject Key Identifier	27692ec32f205b071480e1863...
Basic Constraints	Subject Type=CA, Path Lengt

```
CN = Wireless TAC Lab Sub CA
O = TAC Lab
C = US
```



Show: <All>

Field	Value
Valid to	Thursday, April 18, 2030 9:51...
Subject	Wireless TAC Lab Sub CA, TA...
Public key	RSA (2048 Bits)
Public key parameters	05 00
Authority Key Identifier	KeyID=cba6ff6ca7d4c34b7ca...
Enhanced Key Usage	Client Authentication (1.3.6.1....
Subject Key Identifier	27692ec32f205b071480e1863...
Basic Constraints	Subject Type=CA, Path Lengt

```
27692ec32f205b071480e186367be092084c8812
```

Once the the Intermediate CA certificate is identified, proceed with the chain accordingly and reinstall.

Scenario 3. No Root CA Certificate in the Chain

<#root>

```
*TransferTask: Apr 21 04:28:09.643: Add ID Cert: Adding certificate & private key using password Cisco12
*TransferTask: Apr 21 04:28:09.643: Add Cert to ID Table: Adding certificate (name: bsnSslWebauthCert) t
*TransferTask: Apr 21 04:28:09.643: Add Cert to ID Table: Decoding PEM-encoded Certificate (verify: YES)
*TransferTask: Apr 21 04:28:09.643: Decode & Verify PEM Cert: Cert/Key Length was 0, so taking string le
*TransferTask: Apr 21 04:28:09.643: Decode & Verify PEM Cert: Cert/Key Length 4929 & VERIFY
*TransferTask: Apr 21 04:28:09.645: Decode & Verify PEM Cert: X509 Cert Verification return code: 0
*TransferTask: Apr 21 04:28:09.645:

Decode & Verify PEM Cert: X509 Cert Verification result text: unable to get issuer certificate


*TransferTask: Apr 21 04:28:09.645:

Decode & Verify PEM Cert: Error in X509 Cert Verification at 1 depth: unable to get issuer certificate


*TransferTask: Apr 21 04:28:09.646: Add Cert to ID Table: Error decoding (verify: YES) PEM certificate
*TransferTask: Apr 21 04:28:09.646: Add ID Cert: Error decoding / adding cert to ID cert table (verifyCh
```

Solution: This scenario is similar to scenario 2, but this time against the intermediate certificate when you validate the issuer (Root CA). The same instructions can be followed with the **Issuer** and **X509v3 Authority Key Identifier** fields verification on the intermediate CA certificate to validate the Root CA.

This OpenSSL command can be used to validate these details on each certificate:

<#root>

```
>

openssl x509 -in

int-ca.crt

 -text -noout



Certificate:
Data:
Version: 3 (0x2)
Serial Number:
d1:ec:26:0e:be:f1:aa:65:7b:4a:8f:c7:d5:7f:a4:97
Signature Algorithm: sha256WithRSAEncryption

Issuer: C=US, O=TAC Lab, CN=Wireless TAC Lab Root CA


Validity
Not Before: Apr 21 02:51:03 2020 GMT
Not After : Apr 19 02:51:03 2030 GMT
Subject: C=US, O=TAC Lab, CN=Wireless TAC Lab Sub CA

...
```

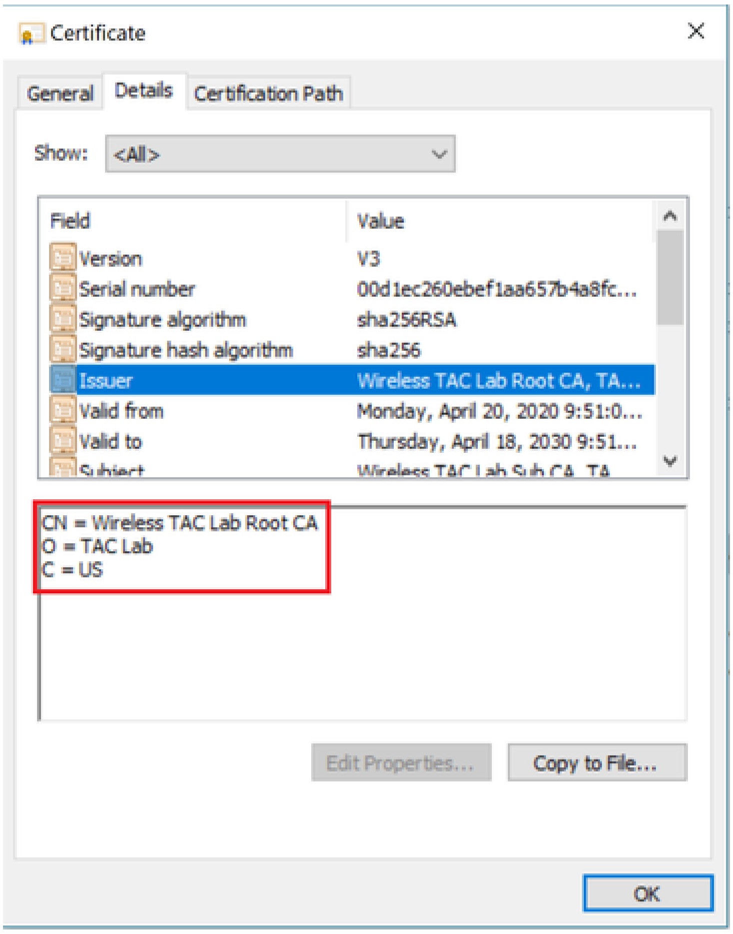
```
X509v3 extensions:
X509v3 Authority Key Identifier:

keyid:CB:A6:FF:6C:A7:D4:C3:4B:7C:A3:A9:A3:14:C3:90:8D:9B:04:A0:32
```

<#root>

```
>
openssl x509 -in
root-ca.crt
-text -noout
```

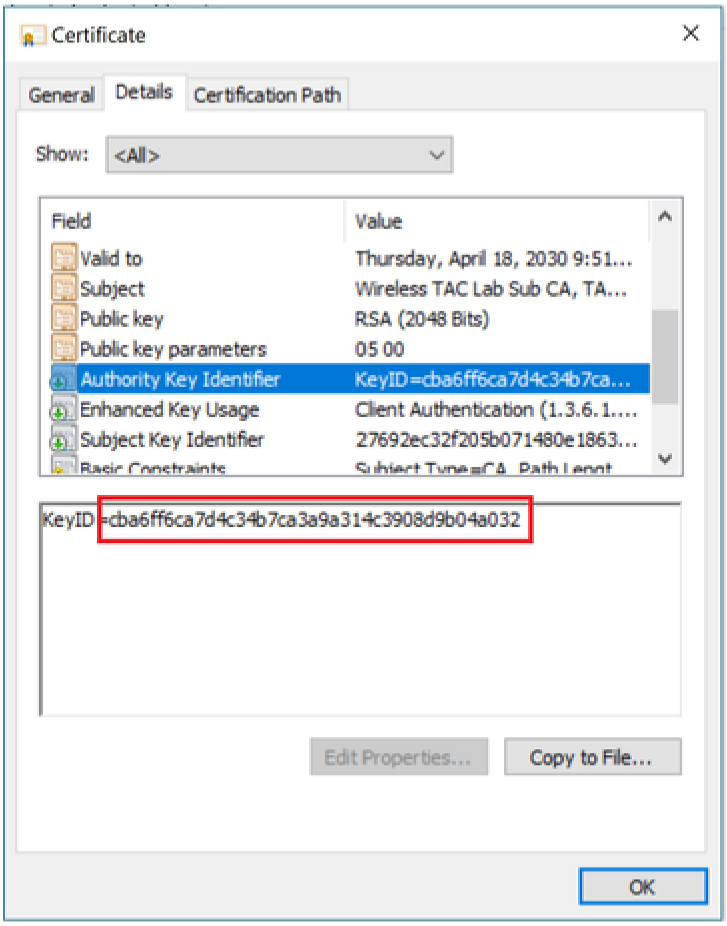
```
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
d1:ec:26:0e:be:f1:aa:65:7b:4a:8f:c7:d5:7f:a4:96
Signature Algorithm: sha256WithRSAEncryption
Issuer: C=US, O=TAC Lab, CN=Wireless TAC Lab Root CA

Validity
Not Before: Apr 21 02:40:24 2020 GMT
Not After : Apr 19 02:40:24 2030 GMT
Subject: C=US, O=TAC Lab, CN=Wireless TAC Lab Root CA

...
X509v3 Subject Key Identifier:

CB:A6:FF:6C:A7:D4:C3:4B:7C:A3:A9:A3:14:C3:90:8D:9B:04:A0:32
```

Intermediate CA certificate



Field	Value
Version	V3
Serial number	00d1ec260ebef1aa657b4a8fc...
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	Wireless TAC Lab Root CA, TA...
Valid from	Monday, April 20, 2020 9:51:0...
Valid to	Thursday, April 18, 2030 9:51...
Subject	Wireless TAC Lab Sub CA, TA

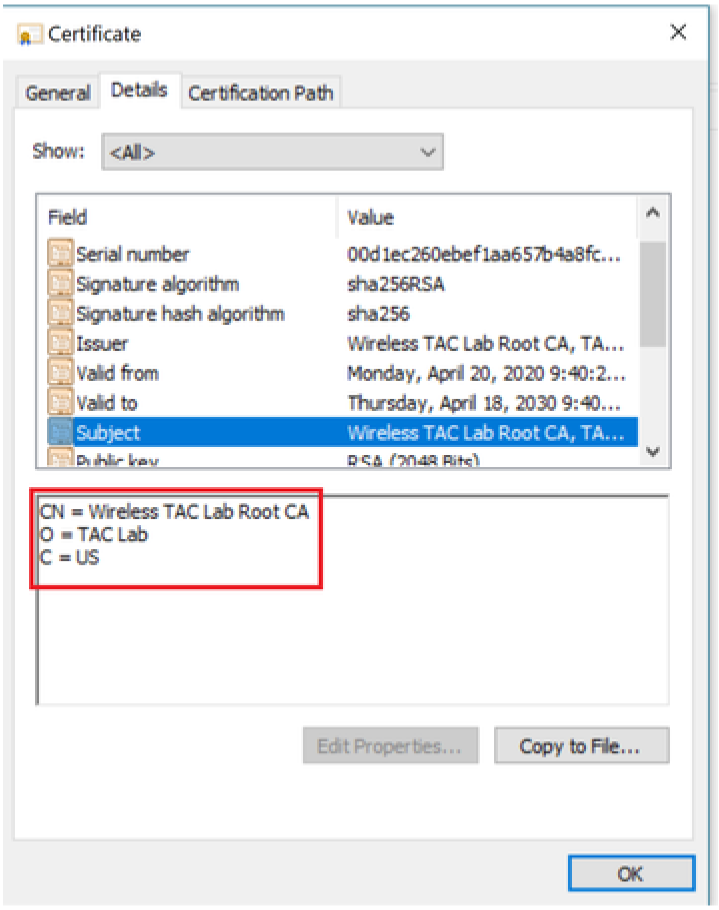
CN = Wireless TAC Lab Root CA
O = TAC Lab
C = US



Field	Value
Valid to	Thursday, April 18, 2030 9:51...
Subject	Wireless TAC Lab Sub CA, TA...
Public key	RSA (2048 Bits)
Public key parameters	05 00
Authority Key Identifier	KeyID=cba6ff6ca7d4c34b7ca...
Enhanced Key Usage	Client Authentication (1.3.6.1....
Subject Key Identifier	27692ec32f205b071480e1863...
Basic Constraints	Subject Type=CA, Path Lengt

KeyID=cba6ff6ca7d4c34b7ca3a9a314c3908d9b04a032

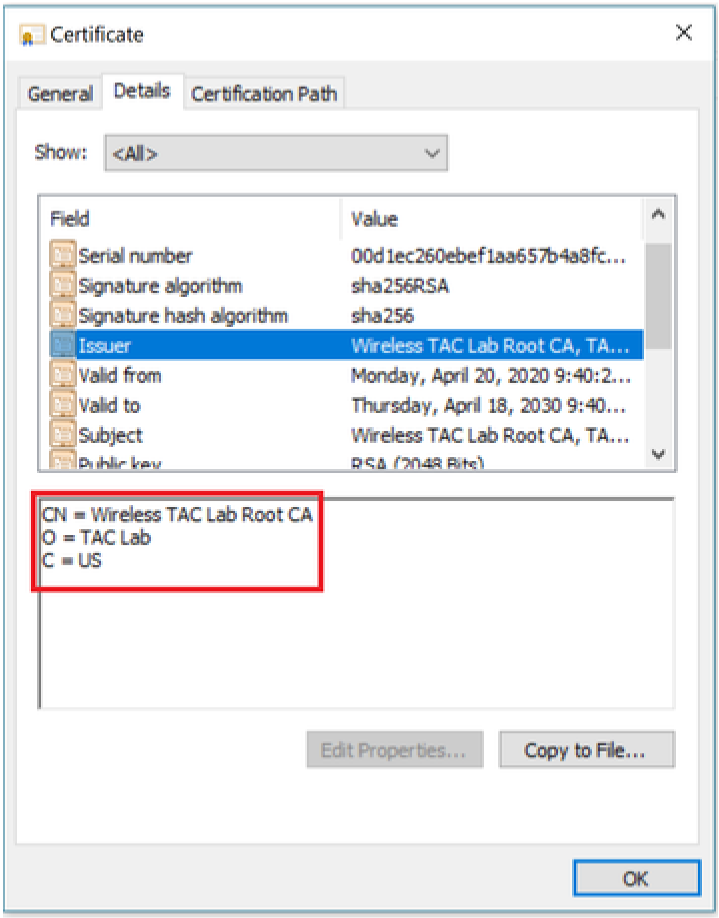
Root CA certificate:



Show: <All>

Field	Value
Serial number	00d1ec260ebef1aa657b4a8fc...
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	Wireless TAC Lab Root CA, TA...
Valid from	Monday, April 20, 2020 9:40:2...
Valid to	Thursday, April 18, 2030 9:40...
Subject	Wireless TAC Lab Root CA, TA...
Public key	RSA (2048 Bits)

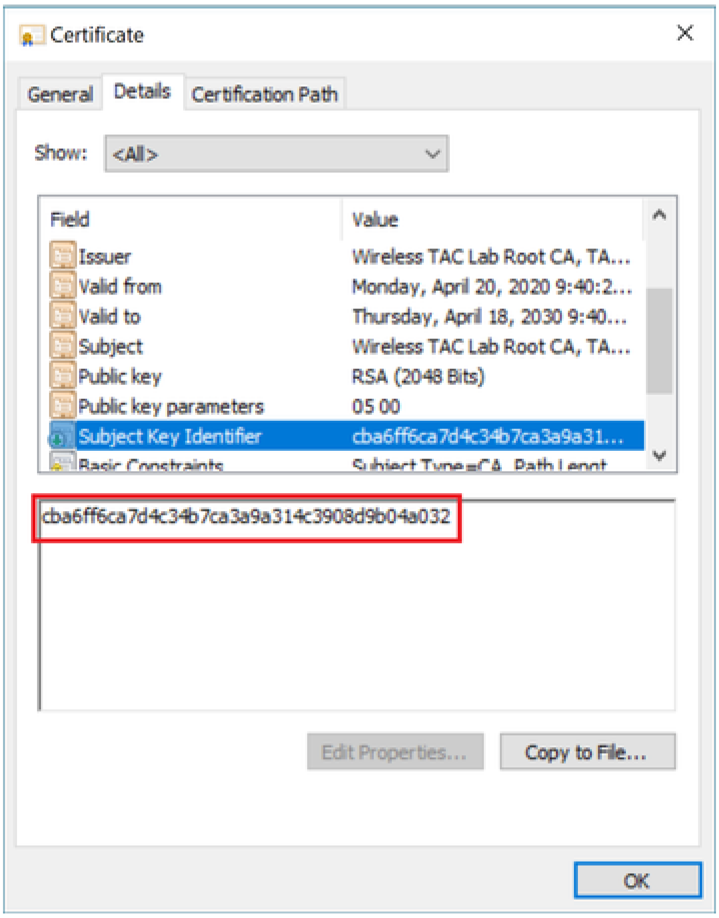
```
CN = Wireless TAC Lab Root CA
O = TAC Lab
C = US
```



Show: <All>

Field	Value
Serial number	00d1ec260ebef1aa657b4a8fc...
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	Wireless TAC Lab Root CA, TA...
Valid from	Monday, April 20, 2020 9:40:2...
Valid to	Thursday, April 18, 2030 9:40...
Subject	Wireless TAC Lab Root CA, TA...
Public key	RSA (2048 Bits)

```
CN = Wireless TAC Lab Root CA
O = TAC Lab
C = US
```



Show: <All>

Field	Value
Issuer	Wireless TAC Lab Root CA, TA...
Valid from	Monday, April 20, 2020 9:40:2...
Valid to	Thursday, April 18, 2030 9:40...
Subject	Wireless TAC Lab Root CA, TA...
Public key	RSA (2048 Bits)
Public key parameters	05 00
Subject Key Identifier	cba6ff6ca7d4c34b7ca3a9a31...
Basic Constraints	Subject Type=CA, Path Lengt

cba6ff6ca7d4c34b7ca3a9a314c3908d9b04a032

Once the Root CA certificate is identified (both Issuer and Subject are the same), proceed with the chain accordingly and reinstall.

Note: This document uses three certificate chain (leaf, Intermediate CA, Root CA), which is the most common scenario. There can be scenarios when 2 Intermediate CA certificates are involved. The same guideline from this scenario can be used until the Root CA certificate is found.

Scenario 4. No CA Certificates in the Chain

<#root>

```
*TransferTask: Apr 21 04:56:50.272: Add ID Cert: Adding certificate & private key using password Cisco12
*TransferTask: Apr 21 04:56:50.272: Add Cert to ID Table: Adding certificate (name: bsnSslWebauthCert) t
*TransferTask: Apr 21 04:56:50.272: Add Cert to ID Table: Decoding PEM-encoded Certificate (verify: YES)
*TransferTask: Apr 21 04:56:50.272: Decode & Verify PEM Cert: Cert/Key Length was 0, so taking string le
*TransferTask: Apr 21 04:56:50.272: Decode & Verify PEM Cert: Cert/Key Length 3493 & VERIFY
*TransferTask: Apr 21 04:56:50.273: Decode & Verify PEM Cert: X509 Cert Verification return code: 0
*TransferTask: Apr 21 04:56:50.273:
```

`Decode & Verify PEM Cert: Error in X509 Cert Verification at 0 depth: unable to get local issuer certifi`

```
*TransferTask: Apr 21 04:56:50.274: Add Cert to ID Table: Error decoding (verify: YES) PEM certificate
*TransferTask: Apr 21 04:56:50.274: Add WebAuth Cert: Error adding ID cert
*TransferTask: Apr 21 04:56:50.274: RESULT_STRING: Error installing certificate.
```

Solution: With no other certificate in the file other than the WLC certificate, validation fails at **Verification**

at 0 depth. The file can be opened in a text editor to be validated. Guidelines from Scenario 2 and 3 can be followed to identify the chain all the way to the Root CA and re-chain accordingly and re-install.

Scenario 5. No Private Key

<#root>

```
*TransferTask: Apr 21 05:02:34.764: Add WebAuth Cert: Adding certificate & private key using password
*TransferTask: Apr 21 05:02:34.764: Add ID Cert: Adding certificate & private key using password
*TransferTask: Apr 21 05:02:34.764: Add Cert to ID Table: Adding certificate (name: bsnSslWebauthCert) t
*TransferTask: Apr 21 05:02:34.764: Add Cert to ID Table: Decoding PEM-encoded Certificate (verify: YES)
*TransferTask: Apr 21 05:02:34.764: Decode & Verify PEM Cert: Cert/Key Length was 0, so taking string le
*TransferTask: Apr 21 05:02:34.764: Decode & Verify PEM Cert: Cert/Key Length 3918 & VERIFY
*TransferTask: Apr 21 05:02:34.767: Decode & Verify PEM Cert: X509 Cert Verification return code: 1
*TransferTask: Apr 21 05:02:34.767: Decode & Verify PEM Cert: X509 Cert Verification result text: ok
*TransferTask: Apr 21 05:02:34.768: Add Cert to ID Table: Decoding PEM-encoded Private Key using passwor
*TransferTask: Apr 21 05:02:34.768:

Retrieve CSR Key: can't open private key file for ssl cert.



*TransferTask: Apr 21 05:02:34.768:

Add Cert to ID Table: No Private Key



*TransferTask: Apr 21 05:02:34.768: Add ID Cert: Error decoding / adding cert to ID cert table (verifyCh
*TransferTask: Apr 21 05:02:34.768: Add WebAuth Cert: Error adding ID cert
*TransferTask: Apr 21 05:02:34.768: RESULT_STRING: Error installing certificate.
```

Solution: The WLC expects the private key to be included in the file if Certificate Signing Request (CSR) was generated externally and needs to be chained in the file. In the case that the CSR was generated in the WLC, ensure that the WLC is not reloaded before the installation, otherwise the private key gets lost.

Related Information

- **Cisco Technical Support & Downloads**