

LDAPSの正しい証明書の決定

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[証明書に問題があるかどうかを確認するには
使用する証明書/チェーンを決定します。](#)

概要

このドキュメントでは、セキュアなLightweight Directory Access Protocol(LDAP)の正しい証明書を決定する方法について説明します。

前提条件

要件

このドキュメントに特有の要件はありません。

使用するコンポーネント

このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

背景説明

Secure LDAPでは、Unified Computing System(UCS)ドメインに正しい証明書または証明書チェーンがトラステッドポイントとしてインストールされている必要があります。

誤った証明書（またはチェーン）が設定されている場合、または証明書が存在しない場合、認証は失敗します。

証明書に問題があるかどうかを確認するには

Secure LDAPに問題がある場合は、LDAPデバッグを使用して証明書が正しいかどうかを確認します。

```
[username]
[password]
connect nxos      *(make sure we are on the primary)
debug ldap all
term mon
```

次に、2番目のセッションを開き、セキュアLDAPクレデンシャルでログインを試みます。

デバッグが有効なセッションでは、試行されたログインがログに記録されます。ログインセッションでundebgコマンドを実行し、以降の出力を停止します。

```
undebg all
```

証明書に潜在的な問題があるかどうかを確認するには、次の行のデバッグ出力を調べます。

```
2018 Sep 25 10:10:29.144549 ldap: ldap_do_process_tls_resp: (user f-ucsapac-01) - ldap start TLS
sent succesfully;          Calling ldap_install_tls
2018 Sep 25 10:10:29.666311 ldap: ldap_do_process_tls_resp: (user f-ucsapac-01) - TLS START
failed
```

TLSが失敗すると、セキュアな接続を確立できず、認証が失敗します。

使用する証明書/チェーンを決定します。

セキュリティで保護された接続の確立に失敗したと判断したら、正しい証明書を決定します。

Ethanalyzerを使用して通信をキャプチャし、ファイルから証明書(またはチェーン)を抽出します。

デバッグセッションで次のコマンドを実行します。

```
ethanalyzer local interface mgmt capture-filter "host <address of controller/load balancer>"
limit-captured-frames 100 write volatile:ldap.pcap
```

次に、クレデンシャルを使用して別のログインを試みます。

デバッグセッションで新しい出力が表示されなくなったら、キャプチャを終了します。使用(ctrl + c)。

次のコマンドを使用して、ファブリックインターコネクト(FI)からパケットキャプチャを転送します。

```
copy volatile:ldap.pcap tftp:
```

ldap.pcapファイルを手に入れたら、Wiresharkでファイルを開き、TLS接続の初期化を開始するパケットを探します。

次の図に示すように、パケットの情報セクションに同様のメッセージが表示されます。

Server Hello, Certificate, Certificate Request, Server Hello Done

7	0.498834	SSLv2	190	Client Hello
8	0.753397	TCP	1514	[TCP segment of a reassembled PDU]
9	0.755902	TCP	1514	[TCP segment of a reassembled PDU]
10	0.755940	TCP	66	56328 → 3268 [ACK] Seq=156 Ack=2943 Win=11776 Len=0 TSval=1166916677 TSecr=112994803
11	1.005008	TLSv1	875	Server Hello, Certificate, Certificate Request, Server Hello Done
12	1.007214	TLSv1	73	Alert (Level: Fatal, Description: Unknown CA)

このパケットを選択して展開します。

Secure Sockets Layer

-->TLSv? Record Layer: Handshake Protocol: Multiple Handshake Messages

---->Handshake Protocol: Certificate

----->Certificates (xxxx bytes)

```
▶ [3 Reassembled TCP Segments (3705 bytes): #8(1448), #9(1448), #11(809)]
▼ Secure Sockets Layer
  ▼ TLSv1 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 3700
    ▼ Handshake Protocol: Server Hello
      Handshake Type: Server Hello (2)
      Length: 70
      Version: TLS 1.0 (0x0301)
      ▶ Random
        Session ID Length: 32
        Session ID: 8d34000098910c057c220a9a20684445399d6c37d95a0408...
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Compression Method: null (0)
    ▼ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 1695
      Certificates Length: 1692
      ▼ Certificates (1692 bytes)
        Certificate Length: 1689
        ▶ Certificate: 308206953082057da00302010202100ea240190f78560f7a... (id-at-commonName=)
```

Certificateという行を選択します。

この行を右クリックし、[Export Packet Bytes]を選択し、ファイルを.derファイルとして保存します。

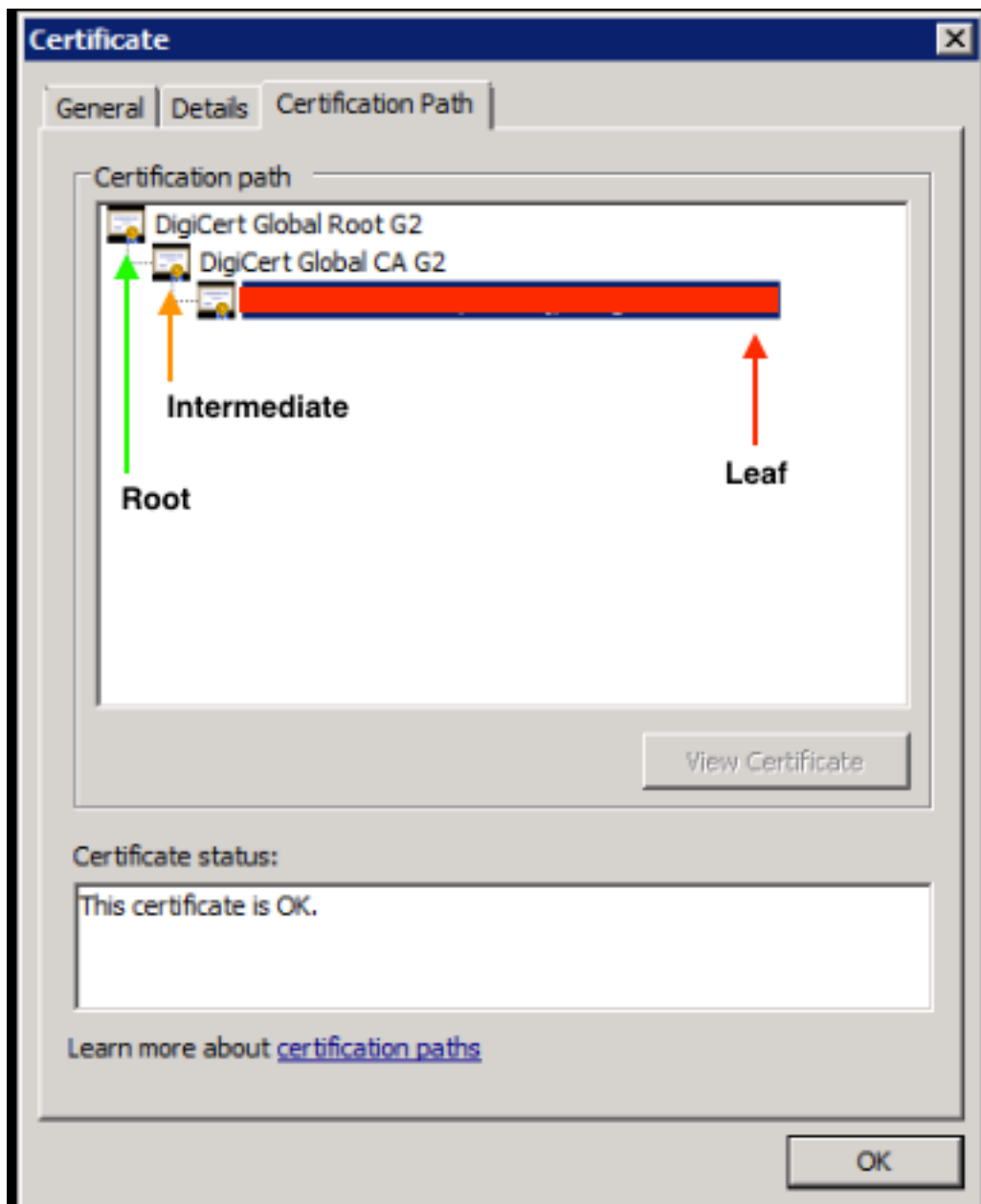
Windowsで証明書を開き、[証明書のパス]タブに移動します。

これは、ルート証明書からリーフ(エンドホスト)へのフルパスを示しています。リーフ以外のすべてのノードに対して、次の操作を行います。

Select the node

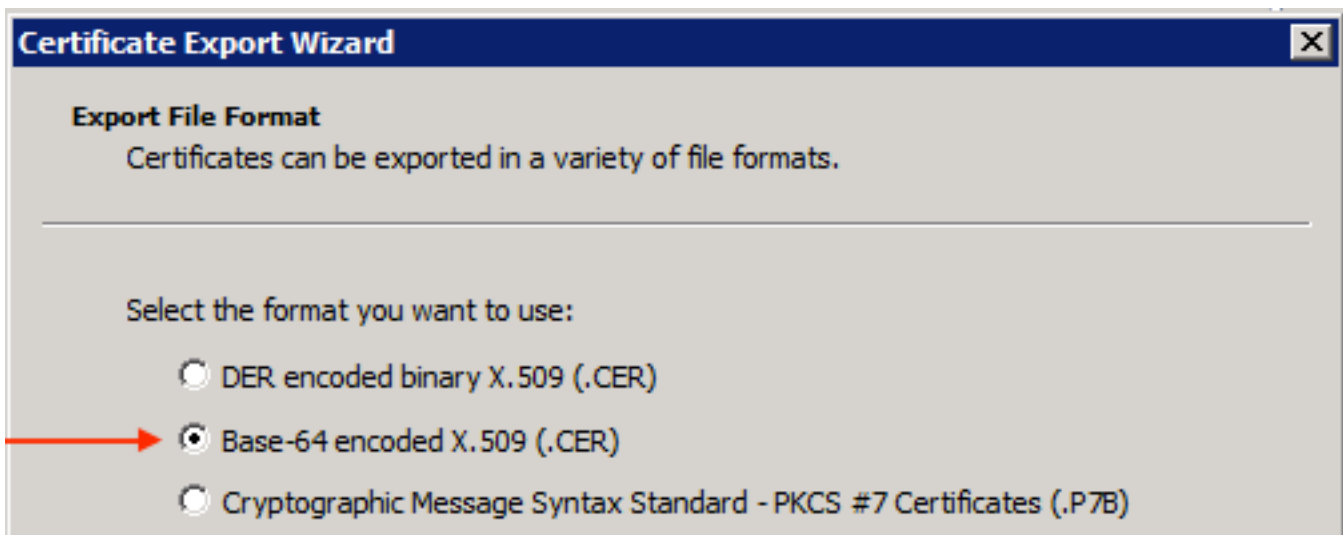
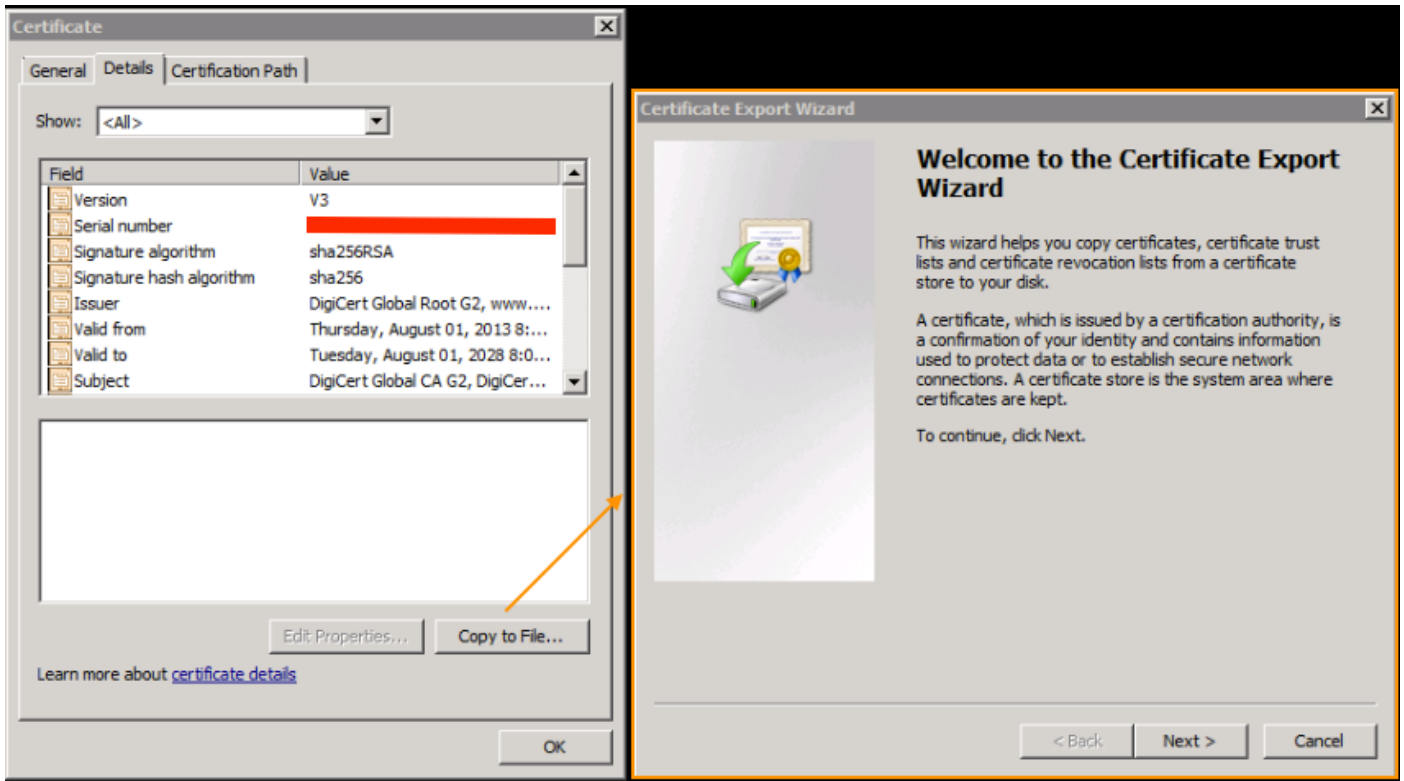
-->Select 'View Certificate'

----->Select the 'Details' tab



[ファイルにコピー]オプションを選択し、[証明書のエクスポートウィザード]に従います (Base-64エンコード形式を使用してください)。

これにより、リスト内の各ノードに対する.cerファイルが作成され、完了します。



これらのファイルをメモ帳、メモ帳++、Sublimeなどで開き、ハッシュされた証明書を表示します。

チェーンが存在する場合は、新しいドキュメントを開き、最後のノードのハッシュされた証明書を貼り付けます。

ハッシュされた各証明書をルートCAで終えて貼り付けるリストを作成してください。

ルートCA(チェーンがない場合)または生成したチェーン全体をトラステッドポイントに貼り付けます。