Contents

[Introduction](#)
[Prerequisites](#)
[Requirements](#)
[Components Used](#)
[Background Information](#)
[Configure](#)
[Step 1. Use the Public CA or the Set Up CA on Windows Server 2003](#)
[Step 2. Verify Hostname and Settings](#)
[Step 3. Generate and Download the Certificate Signing Request (CSR)](#)
[Step 4. Sign the CSR with the Microsoft Windows 2003 Certificate Authority](#)
[Step 5. Get the Root Certificate from the CA](#)
[Step 6. Upload CA Root Certificate as CallManager Trust](#)
[Step 7. Upload CA sign CallManager CSR Certificate as CallManager certificate.](#)
[Step 8. Create SIP Trunk Security Profiles](#)
[Step 9. Create SIP Trunks](#)
[Step 10. Create Route Patterns](#)
[Verify](#)
[Troubleshoot](#)
[Collect Packet Capture on CUCM](#)
[Collect CUCM Traces](#)
[Related Cisco Support Community Discussions](#)

Introduction

This document describes a step by step process to configure the Session Initiation Protocol (SIP) Transport Layer Security (TLS) Trunk on Communications Manager with a Certificate Authority (

After following this document, SIP messages between two clusters will be encrypted using the TLS.

Prerequisites

Requirements

Cisco recommends that you have knowledge of:

- Cisco Unified Communications Manager (CUCM)
- SIP

Components Used

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

- CUCM Version 9.1(2)
- CUCM Version 10.5(2)
- Microsoft Windows Server 2003 as CA

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, make sure that you understand the potential impact of any command.

Background Information

As shown in this image, SSL Handshake using Certificates.

Configure

Step 1. Use the Public CA or the Set Up CA on Windows Server 2003

Refer to the link: Set Up CA on Windows 2003 Sever

Step 2. Verify Hostname and Settings

Certificates are based on names. Ensure that the names are correct before starting.

In order to change the hostname, refer to the link: Change Hostname on CUCM

Step 3. Generate and Download the Certificate Signing Request (CSR)

CUCM 9.1(2)

In order to generate the CSR, navigate to **OS Admin** > **Security** > **Certificate Management** > **Generate CSR**

In the **Certificate Name** field**,** select **CallManager** option from the drop-down list.

In order to download the CSR, navigate to **OS Admin > Security > Certificate Management > Download CSR**

In the **Certificate Name** field, select **CallManager** option from the drop-down list.

CUCM 10.5(2)

In order to generate the CSR**,** navigate to **OS Admin > Security > Certificate Management > Generate CSR**

1. In the **Certificate Purpose** field**,** select **CallManager** from the drop-down list.

2. In the **Key Length** field**,** select **1024** from the drop-down list**.**
3. In the **Hash Algorithm** field**,** select **SHA1** from the drop-down list.

In order to download the CSR, navigate to **OS Admin > Security > Certificate Management > Download CSR**

In the **Certificate Purpose** field**,** select **CallManager** option from the drop-down list.

> **Note**: The CallManager CSR is generated with the 1024 bit Rivest-Shamir-Addleman (RSA) Keys.

Step 4. Sign the CSR with the Microsoft Windows 2003 Certificate Authority

This is an optional information to sign the CSR with the Microsoft Windows 2003 CA.

1. Open the Certification Authority.

2. Right-click the **CA** icon and navigate to **All Tasks > Submit new request**

3. Select the CSR and click the **Open** option (Applicable in both the CSRs (CUCM 9.1(2) and CUCM 10.5(2))

4. All of the opened CSRs display in the Pending Requests Folder. Right-click each CSR and navigate to **All Tasks > Issue** in order to issue the certificates. (Applicable in both the CSRs (CUCM 9.1(2) and CUCM 10.5(2))

5. In order to download the certificate, choose **Issued Certificates** folder.

Right-click the certificate and click the **Open** option.

6. The certificate details are displayed. In order to download the certificate, select the **Details** tab and click the button **Copy to File…**

7. In the **Certificate Export Wizard** window, click the **Base-64 encoded X.509(.CER**) radio button.

8. Name the file accurately. This example uses **CUCM1052.cer** format.

For CUCM 9.1(2), follow same procedure.

Step 5. Get the Root Certificate from the CA

Open the **Certification Authority** window.

In order to download the root-CA

1. Right-click the CA icon and click the **Properties** option.

2. In general TAB, click **View Certificate.**

3. In the **Certificate** window, click the details TAB.

4. Click **Copy to File...**

Step 6. Upload CA Root Certificate as CallManager Trust

In order to upload the CA Root Certificate, login to **OS Admin > Security > Certificate Management > Upload Certificate/Certificate Chain**

Note: Perform these steps on both the CUCMs (CUCM 9.1(2) and CUCM 10.5(2))

Step 7. Upload CA sign CallManager CSR Certificate as CallManager certificate.

In order to upload the CA sign CallManager CSR, login to **OS Admin > Security > Certificate Management > Upload Certificate/Certificate Chain**

Note: Perform these steps on both the CUCMs (CUCM 9.1(2) and CUCM 10.5(2))

Step 8. Create SIP Trunk Security Profiles

CUCM 9.1(2)

In order to create the SIP Trunk Security Profile, navigate to **System > Security > SIP Trunk Security Profile**.

Copy the existing Non Secure SIP Trunk Profile and give it a new name. In the example, Non Secure SIP Trunk Profile has been renamed with Secure SIP Trunk Profile TLS.

In X.509 Subject Name use the Common Name (CN) of the CUCM 10.5(2) (CA signed certificate) as shown in this image.

CUCM 10.5(2)

Navigate to **System > Security > SIP Trunk Security Profile.**

Copy the existing Non Secure SIP Trunk Profile and give it a new name. In the example, Non Secure SIP Trunk Profile was renamed with Secure SIP Trunk Profile TLS.

In X.509 Subject Name use the CN of the CUCM 9.1(2) (CA signed certificate) as highlighted:

Both the SIP Trunk Security Profiles set an incoming port of 5061, in which each cluster listens on the TCP port 5061 for the new inbound SIP TLS calls.

Step 9. Create SIP Trunks

After the Security Profiles are created, create the SIP Trunks and make the changes for the below configuration parameter on the SIP Trunk.

CUCM 9.1(2)

1. On the SIP **Trunk Configuration** window, check the configuration parameter **SRTP Allowed** checkbox.

This secures the Real-time Transport Protocol (RTP) to be used for the calls over this trunk. This box must only be checked when you use SIP TLS because the keys for Secure Real-time Transport Protocol (SRTP) are exchanged in the body of the SIP message. The SIP signaling must be secured by TLS, otherwise anyone with the non-secure SIP signaling could decrypt the corresponding SRTP stream over the trunk.

1. On the **SIP Information** section of the SIP **Trunk Configuration** window, add the **Destination Address**, **Destination Port**, and **SIP Trunk Security Profile.**

CUCM 10.5(2)

1. On the SIP **Trunk Configuration** window, check the configuration parameter **SRTP Allowed** checkbox.

This allows SRTP to be used for calls over this trunk. This box must only be checked when using SIP TLS, because the keys for SRTP are exchanged in the body of the SIP message. The SIP signaling must be secured by the TLS because anyone with a non-secure SIP signaling could decrypt the corresponding Secure RTP stream over the trunk.

1. On the **SIP Information** section of the SIP **Trunk Configuration** window, add the **Destination IP Address**, **Destination Port**, and **Security Profile**

Step 10. Create Route Patterns

The simplest method is to create a Route Pattern on each cluster, pointing directly to the SIP Trunk. Route Groups and Route Lists could also be used.

CUCM 9.1(2) points to **Route Pattern** 9898 via the TLS SIP Trunk to the CUCM 10.5(2)

The CUCM 10.5(2) points to **Route Pattern** 1018 via the TLS SIP Trunk to the CUCM 9.1(2)

Verify

There is currently no verification procedure available for this configuration.

Troubleshoot

The SIP TLS call can be debugged with these steps.

Collect Packet Capture on CUCM

In order to check the connectivity between the CUCM 9.1(2) and the CUCM 10.5(2), take a packet capture on the CUCM servers and watch for the SIP TLS traffic.

The SIP TLS traffic is transmitted on the TCP port 5061, seen as sip-tls.

In the following example there is an SSH CLI session established to the CUCM 9.1(2)

1. CLI Packet Capture on Screen

This CLI prints the output on the screen for the SIP TLS traffic.

```
admin:utils network capture host ip 10.106.95.200
Executing command with options:
interface=eth0
ip=10.106.95.200
19:04:13.410944 IP CUCMA.42387 > 10.106.95.200.sip-tls: P 790302485:790303631(1146) ack
3661485150 win 182 <nop,nop,timestamp 2864697196 5629758>
19:04:13.450507 IP 10.106.95.200.sip-tls > CUCMA.42387: . ack 1146 win 249 <nop,nop,timestamp
6072188 2864697196>
19:04:13.465388 IP 10.106.95.200.sip-tls > CUCMA.42387: P 1:427(426) ack 1146 win 249
<nop,nop,timestamp 6072201 2864697196>
```

2. CLI Captures to File

This CLI does the packet capture based on the host and creates a file named packets.

```
admin:utils network capture eth0 file packets count 100000 size all host ip 10.106.95.200
```

Restart the SIP trunk on CUCM 9.1(2) and make the call from the extension 1018 (CUCM 9.1(2)) to the extension 9898 (CUCM 10.5(2))

In order to download the file from the CLI, run this command:

```
admin:file get activelog platform/cli/packets.cap
```

The capture is done in the standard .cap format. This example uses Wireshark to open packets.cap file but any packet capture display tool can be used.

1. The Transmission Control Protocol (TCP) Synchronize (SYN) to establish the TCP communication between the CUCM 9.1(2)(Client) and the CUCM 10.5(2)(Server).
2. The CUCM 9.1(2) sends the Client Hello to start the TLS session.
3. The CUCM 10.5(2) sends The Server Hello, Server Certificate, and Certificate Request to start the certificate exchange process.
4. The Certificate that the client CUCM 9.1(2) sends to complete the certificate exchange.
5. The Application Data that is encrypted SIP signaling, shows that the TLS session has been established.

Further check whether the correct certificates are exchanged. After Server Hello, the server CUCM 10.5(2) sends its certificate to the client CUCM 9.1(2).

The serial number and the subject information that the server CUCM 10.5(2) has, is presented to the client CUCM 9.1(2).The serial number, subject, issuer, and validity dates are all compared to the information on the OS Admin Certificate Management page.

The server CUCM 10.5(2) presents its own certificate for verification, now it checks the certificate of the client CUCM 9.1(2). The verification happens in both directions.

If there is a mismatch between the certificates in the packet capture and the certificates in the OS Admin Web Page, then the correct certificates are not uploaded.

The correct certificates must be uploaded onto the OS Admin Cert page.

Collect CUCM Traces

The CUCM traces can also be helpful to determine what messages are exchanged between the CUCM 9.1(2) and the CUCM 10.5(2) servers and whether or not the SSL session is properly established.

In the example, the traces from the CUCM 9.1(2) have been collected.

Call Flow:

Ext 1018 > CUCM 9.1(2) > SIP TLS TRUNK > CUCM 10.5(2) > Ext 9898

++ Digit Analysis

```
04530161.009 |19:59:21.185 |AppInfo |Digit analysis: match(pi="2", fqcn="1018",
```

cn="1018",plv="5", pss="", TodFilteredPss="", **dd="9898"**,dac="0")
04530161.010 |19:59:21.185 |AppInfo |Digit analysis: analysis results
04530161.011 |19:59:21.185 |AppInfo ||PretransformCallingPartyNumber=1018
|CallingPartyNumber=1018
|DialingPartition=
|DialingPattern=9898
|FullyQualifiedCalledPartyNumber=9898

++ SIP TLS is being used on the port 5061 for this call.

04530191.034 |19:59:21.189 |AppInfo |//SIP/SIPHandler/ccbId=0/scbId=0/SIP_PROCESS_ENQUEUE:
createConnMsg tls_security=3
04530204.002 |19:59:21.224 |AppInfo
|//SIP/Stack/Transport/0x0/sipConnectionManagerProcessConnCreated: gConnTab=0xb444c150,
addr=10.106.95.200, port=5061, connid=12, **transport=TLS Over TCP**
04530208.001 |19:59:21.224 |AppInfo |SIPTcp - wait_SdlSPISignal: Outgoing SIP TCP message to
10.106.95.200 on port **5061** index 12
[131,NET]
INVITE sip:9898@10.106.95.200:**5061** SIP/2.0
Via: SIP/2.0/TLS 10.106.95.203:5061;branch=z9hG4bK144f49a43a
From: <sip:1018@10.106.95.203>;tag=34~4bd244e4-0988-4929-9df2-2824063695f5-19024196
To: <sip:9898@10.106.95.200>
Call-ID: 94fffc00-57415541-7-cb5f6a0a@10.106.95.203
User-Agent: Cisco-CUCM9.1

++ Signal Distribution Layer (SDL) message SIPCertificateInd provides details about Subject CN and connection information.

04530218.000 |19:59:21.323 |SdlSig |**SIPCertificateInd** |wait
|SIPHandler(1,100,72,1) |SIPTcp(1,100,64,1)
|1,100,17,11.3^*^* |[T:N-H:0,N:1,L:0,V:0,Z:0,D:0] connIdx= 12 --
remoteIP=**10.106.95.200** --remotePort = 5061 --**X509SubjectName**
/C=IN/ST=cisco/L=cisco/O=cisco/OU=cisco/CN=CUCM10 --Cipher AES128-SHA --SubjectAltname =
04530219.000 |19:59:21.324 |SdlSig |SIPCertificateInd
|restart0 |SIPD(1,100,74,16)
|SIPHandler(1,100,72,1) |1,100,17,11.3^*^* |[R:N-
H:0,N:0,L:0,V:0,Z:0,D:0] connIdx= 12 --remoteIP=10.106.95.200 --remotePort = 5061 --
X509SubjectName /C=IN/ST=cisco/L=cisco/O=cisco/OU=cisco/CN=CUCM10 --Cipher AES128-SHA --
SubjectAltname =