Implementing and Troubleshooting Secure Voice/Video on Edge Devices

Paul Anholt, Technical Leader
Felipe Garrido, Technical Leader
BRKCOL-3224
Cisco Webex Teams

Questions?
Use Cisco Webex Teams (formerly Cisco Spark) to chat with the speaker after the session

How
1. Find this session in the Cisco Events App
2. Click “Join the Discussion”
3. Install Webex Teams or go directly to the team space
4. Enter messages/questions in the team space

Webex Teams will be moderated by the speaker until June 18, 2018.

cs.co/ciscolivebot#BRKCOL-3224
Agenda

- Introduction
- Where to Begin
- Implementing Secure Signaling and Media
  - Cisco Expressway
  - Cisco Unified Border Element (CUBE)
- Interoperability between Secure and Non-Secure Networks
- Troubleshooting
  - Methodology
  - Tools
  - Scenarios
Where to Begin
Session Prerequisites

- Working knowledge of
  - SIP
  - CUCM Phone and SIP trunk configuration
  - IOS/CUBE configuration
  - Expressway configuration
Deploying CA-Signed Certificates
Certificate Exchange – CA vs Self-signed

Self-Signed

- CUBE and CUCM generate self-signed certificates
- CUBE exports certificate to CUCM
- CUCM exports certificate to CUBE

CA-Signed

- Client generates a key-pair and sends a Certificate Signing Request (CSR) to the Certificate Authority (CA).
  - The CA signs it with its private key, creating an Identity Certificate
  - Client installs the list of trusted CA Root and Intermediary Certificates and Identity Certificate
Generating Certificate Signing Requests (CSR)
Generating a CSR in CUBE/IOS

Generating the Key-pair

For TLS 1.0, generate an RSA key-pair

ISR4KCUBE(config)#crypto key generate rsa general-key label rsakey modulus <360-4096>

For TLS 1.2, generate an EC key-pair

ISR4KCUBE(config)#crypto key generate ec keysiz 256/384 label eckey
Generating a CSR in CUBE/IOS

1. Create Trustpoint to store the root and identity certificates
   ISR4KCUBE(config)#crypto pki trustpoint caServer
   ISR4KCUBE(ca-trustpoint)#enrollment terminal
   ISR4KCUBE(ca-trustpoint)#subject-name CN=ISR4KCUBE.cisco-uc.com
   ISR4KCUBE(ca-trustpoint)#revocation-check none

1b. Specify the correct key pair to use
   ISR4KCUBE(ca-trustpoint)#eckeypair eckey
   or
   ISR4KCUBE(ca-trustpoint)#rsakeypair rsakey
   Required for TLS 1.2
2. Generate the CSR from the Trustpoint

ISR4KCUBE(config)#crypto pki enroll caServer

% Start certificate enrollment..

% The subject name in the certificate will include: CN=ISR4KCUBE.cisco-uc.com
% The subject name in the certificate will include: ISR4KCUBE.cisco-uc.com
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Display Certificate Request to terminal? [yes/no]: yes
Certificate Request follows:

MIIBezCBuwIBADA4MRwwGgYDVQQDExM0NDUxLUNVQkUuY2lzY28uY29tMRgwFgYJ
...snip...
ceZ+rgIhAN8AzHEJKdUnFovHCetMSed60qHkKjn/H+C1iOxaH4Yz

---End - This line not part of the certificate request---
Redisplay enrollment request? [yes/no]: 
Generating a CSR in Expressway

Maintenance > Security > Server certificate

Certificate signing request (CSR)

Certificate request: There is no certificate signing request in progress

Generate CSR

Notes

• Only one CSR may be in progress at a time.
• From version X8.10, you cannot select SHA-1
Importing the Certificate Trust Chain
Managing the Trusted CA certificates on Expressway

Maintenance > Security > Trusted CA certificate

View and manage trusted server certificates

View the CA trust file

Add one or more certificate to the CA trust file

View the CA trust file

### Trusted CA certificate

<table>
<thead>
<tr>
<th>Type</th>
<th>Issuer</th>
<th>Subject</th>
<th>Expiration date</th>
<th>Validity</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate</td>
<td>O=The Go Daddy Group, Inc., OU=Go Daddy Class 2 Certification Authority</td>
<td>Matches Issuer</td>
<td>Jun 20 2034</td>
<td>Valid</td>
<td>View (decoded)</td>
</tr>
<tr>
<td>Certificate</td>
<td>O=The Go Daddy Group, Inc., OU=Go Daddy Class 2 Certification Authority</td>
<td>O=GoDaddy.com, Inc., CN=Go Daddy Secure Certification Authority</td>
<td>Nov 15 2026</td>
<td>Valid</td>
<td>View (decoded)</td>
</tr>
<tr>
<td>Certificate</td>
<td>WIN-0H5N6IVO74T-CA</td>
<td>Matches Issuer</td>
<td>Sep 27 2026</td>
<td>Valid</td>
<td>View (decoded)</td>
</tr>
</tbody>
</table>
Importing the Root/Intermediate and Identity Certificates to CUBE

Import the Root Certificate via the Terminal using the `crypto pki authenticate` command:

```
ISR4KCUBE(config)#crypto pki authenticate caServer
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
-----BEGIN CERTIFICATE-----
...snip...
-----END CERTIFICATE-----
Certificate has the following attributes:
  Fingerprint MD5: 45F42000 781A427A 152E9DBD 7C438967
  Fingerprint SHA1: A8E57437 3AE1E33B 22768143 EE308B79 0A3C43E6

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
% Certificate successfully imported
```

Import the Identity Certificate via the Terminal using the `crypto pki import` command:

```
ISR4KCUBE(config)#crypto pki import caServer certificate
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
-----BEGIN CERTIFICATE-----
...snip...
-----END CERTIFICATE-----
% Router Certificate successfully imported
```
Certificate Verification

show crypto pki certificate

ISR4KCUBE#show crypto pki certificate

Certificate
Status: Available
Certificate Serial Number (hex): 10FCB16C000000000004
Certificate Usage: General Purpose

Issuer:
  cn=BRKUCC_CA
Subject:
  Name: ISR4KCUBE.cisco-uc.com
cn=ISR4KCUBE.cisco-uc.com

Validity Date:
  start date: 18:34:54 EDT May 29 2017
  end date: 18:44:54 EDT May 29 2019

Associated Trustpoints: caServer

(continued..)

CA Certificate
Status: Available
Certificate Serial Number (hex): 3A0E15FE9124C0B040801DC535DBA5C3
Certificate Usage: Signature

Issuer:
  cn=BRKUCC_CA
Subject:
  cn=BRKUCC_CA

Validity Date:
  start date: 17:44:42 EDT May 29 2017
  end date: 17:54:41 EDT May 29 2022

Associated Trustpoints: caServer
CA-signed Certificate Trust Chain
Implementing Secure Signaling
Do Unsecure calls work?
TLS Session Establishment

Client

TCP Established

Server

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certificate

Certif
SIP Session Establishment

TCP Established
TLS Established
Invite (Offer)
100 Trying
180 Ringing
200 Ok (Answer)
Ack
Media/RTP
MRA CUCM Phone Device Security Profile

The UCM Device Security Profile **Name** must be in the FQDN format with the enterprise domain.

This same FQDN must be present as a Subject Alternate Name (SAN) in the Expressway-C’s server certificate.

Use of a Universal Device Template is recommended to avoid having to regenerate the server certificate when new endpoint models are added.

Set the Device Security Mode to **Encrypted**

X509v3 extensions:

- X509v3 Basic Constraints: critical
  - CA:TRUE
- X509v3 Key Usage: critical
  - Digital Signature, Certificate Sign, CRL Sign
- X509v3 Extended Key Usage:
  - TLS Web Server Authentication, TLS Web Client Authentication
- X509v3 Subject Alternative Name: DNS:udt.rtp.ciscotac.net, DNS:conference-2-StandAloneCluster08c0c.rtp.ciscotac.net
- X509v3 Authority Key Identifier:

Status: Ready

**Phone Security Profile Information**

- **Product Type:** All
- **Device Protocol:** Protocol Not Specified
- **Name:** udt.rtp.ciscotac.net
- **Description:**
- **Device Security Mode:** Encrypted
  - TFTP Encrypted Config
Expressway Configuration

Enabling SIP TLS

Configuration > Protocols > SIP

<table>
<thead>
<tr>
<th>Configuration Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>TLS Mode/Port</strong> – Configure the <em>global</em> SIP TLS listening port</td>
</tr>
<tr>
<td>• <strong>MTLS Mode/Port</strong> – Configure the <em>global</em> SIP MTLS Listening Port</td>
</tr>
<tr>
<td>• <strong>TLS Handshake timeout</strong> – Configure the number of seconds before timing out the TLS Connection</td>
</tr>
<tr>
<td>• <strong>Certificate revocation checking mode</strong> – Enable or disable CRL</td>
</tr>
</tbody>
</table>

MTLS settings here may conflict with the following default zone settings
**Expressway Configuration**

**Enabling SIP TLS on Zones**

**Configuration Settings**

**SIP**
- **Transport** – (TLS) Set the zone’s *outbound* transport type to TLS
- **TLS Port** – *Peer’s* SIP TLS Listening port
- **TLS Verify Mode:**
  - On – Match the peers x.509 subject name to the zone’s peer
  - Off – Accept any trusted x.509 certificate chain

**Location**
- **Peer Address** – FQDN or IP address of peer
### Expressway Configuration

**Cipher Suites**

#### Maintenance > Security > Ciphers

<table>
<thead>
<tr>
<th>Ciphers</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward proxy TLS ciphers</td>
<td><img src="https://example.com/icon.png" alt="Icon" /></td>
</tr>
<tr>
<td>Forward proxy minimum TLS version</td>
<td><img src="https://example.com/icon.png" alt="Icon" /></td>
</tr>
<tr>
<td>HTTPS ciphers</td>
<td><img src="https://example.com/icon.png" alt="Icon" /></td>
</tr>
<tr>
<td>HTTPS minimum TLS version</td>
<td><img src="https://example.com/icon.png" alt="Icon" /></td>
</tr>
<tr>
<td>Reverse proxy TLS ciphers</td>
<td><img src="https://example.com/icon.png" alt="Icon" /></td>
</tr>
<tr>
<td>Reverse proxy minimum TLS version</td>
<td><img src="https://example.com/icon.png" alt="Icon" /></td>
</tr>
<tr>
<td>SIP TLS ciphers</td>
<td><img src="https://example.com/icon.png" alt="Icon" /></td>
</tr>
<tr>
<td>SIP minimum TLS version</td>
<td><img src="https://example.com/icon.png" alt="Icon" /></td>
</tr>
<tr>
<td>UC server discovery TLS ciphers</td>
<td><img src="https://example.com/icon.png" alt="Icon" /></td>
</tr>
<tr>
<td>UC server discovery minimum TLS version</td>
<td><img src="https://example.com/icon.png" alt="Icon" /></td>
</tr>
<tr>
<td>XMPP TLS ciphers</td>
<td><img src="https://example.com/icon.png" alt="Icon" /></td>
</tr>
<tr>
<td>XMPP minimum TLS version</td>
<td><img src="https://example.com/icon.png" alt="Icon" /></td>
</tr>
</tbody>
</table>

⚠ **TLS 1.0 is supported in X8.10 but not recommended**

#### xconfig //cipher

- **xConfiguration Ciphers ForwardProxyTLS Ciphers Value:** "HIGH!MD5:RC4"  
- **xConfiguration Ciphers ForwardProxyTLS Protocol Value:** "minTLSv1.0"  
- **xConfiguration Ciphers HTTPS Ciphers Value:** "HIGH!EXP!MD5:RC4"  
- **xConfiguration Ciphers HTTPS Protocol Value:** "minTLSv1.0"  
- **xConfiguration Ciphers ReverseProxyTLS Ciphers Value:** "HIGH!MD5:RC4"  
- **xConfiguration Ciphers ReverseProxyTLS Protocol Value:** "minTLSv1.0"  
- **xConfiguration Ciphers SIP TLS Ciphers Value:** "ALL!EXP!LOW!MD5:@"  
- **xConfiguration Ciphers UcClientTLS Ciphers Value:** "ALL"  
- **xConfiguration Ciphers UcClientTLS Protocol Value:** "minTLSv1.0"  
- **xConfiguration Ciphers XCPTLS Ciphers Value:** "ALL!ADH!LOW!EXP!MD..."  
- **xConfiguration Ciphers XCPTLS Protocol Value:** "minTLSv1.0"  
- **xConfiguration Ciphers sshd_ciphers Value:** "aes256-gcm@openssh.com..."  
- **xConfiguration Ciphers sshd_sshd_key Value:** "ecdh-sha2-nistp521,ecdh-sha2..."  
- **xConfiguration Ciphers sshd_macs Value:** "hmac-sha2-512,hmac-sha2..."  
- **xConfiguration Ciphers sshd_pfwd_ciphers Value:** "aes256-ctr"  
- **xConfiguration Authentication ADS Cipher Suite:** "HIGH!MEDIUM!ADH..."
IOS Configuration Prerequisites

General Unsecure Voice Configuration

1. Global CUBE Configuration

- voice service voip
- address-hiding
- mode border-element license capacity 1000
- allow-connection sip to sip

2. Dial-peer Configuration – LAN side

- dial-peer voice 1 voip
description to CUCM Sub – unSecure preference 1
destination-pattern 418110....
session protocol sipv2
session target ipv4:14.50.248.103
session transport tcp
incoming called-number 418.......
dtmf-relay rtp-nte
codec g711ulaw
no vad

3. Dial-peer Configuration – WAN (PSTN) side

- dial-peer voice 10 voip
description to PSTN – unSecure preference 1
destination-pattern [9]1T
session protocol sipv2
session target ipv4:14.50.211.31
session transport tcp
incoming called-number 418.......
dtmf-relay rtp-nte
codec g711ulaw
no vad

TCP is required to enable TLS!
IOS Configuration – Enabling TLS

1. Associate trustpoint with IOS voice process

```
ISR4KCUBE(config)#sip-ua
ISR4KCUBE(config-sip-ua)#crypto signaling default trustpoint caServer
```

2. Enable TLS transport on dial-peer

```
dial-peer voice 1 voip
description to CUCM Sub - Secure Signaling
preference 1
destination-pattern 418110....
session protocol sipv2
session target ipv4:14.50.248.101
session transport tcp tls
incoming called-number 9.T
dtmf-relay rtp-nte
codec g711ulaw
no vad
```

2. Enable TLS transport globally

```
voice service voip
  sip
  session transport tcp tls
```

or

```
```
```
# IOS Configuration – crypto signaling

## Enabling Secure Signaling

### Associate CUBE trustpoint with voice process

```plaintext
sip-ua
crypto signaling remote-addr 14.50.248.100 255.255.255.255
trustpoint caServer
```

### Table: Crypto Signaling Configuration

<table>
<thead>
<tr>
<th>Base command</th>
<th>Peer IP address/network association</th>
<th>trustpoint association</th>
<th>cipher selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto signaling</td>
<td>default</td>
<td>trustpoint &lt;name&gt;</td>
<td>&lt;enter&gt; (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ecdsa-cipher</td>
</tr>
<tr>
<td>crypto signaling</td>
<td>remote-addr &lt;ip.address&gt; &lt;mask&gt;</td>
<td>trustpoint &lt;name&gt;</td>
<td>&lt;enter&gt; (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ecdsa-cipher</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>strict-cipher</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# IOS Signaling Cipher Suites

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Cipher Suites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Cipher</td>
<td>TLS_RSA_WITH_RC4_128_MD5&lt;br&gt;TLS_RSA_WITH_AES_128_CBC_SHA&lt;br&gt;TLS_DHE_RSA_WITH_AES_128_CBC_SHA1&lt;br&gt;TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256&lt;br&gt;TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256&lt;br&gt;TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256&lt;br&gt;TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384&lt;br&gt;TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384</td>
</tr>
<tr>
<td>Strict Cipher</td>
<td>TLS_RSA_WITH_AES_128_CBC_SHA&lt;br&gt;TLS_DHE_RSA_WITH_AES_128_CBC_SHA1&lt;br&gt;TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256&lt;br&gt;TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384</td>
</tr>
<tr>
<td>ECDSA Cipher</td>
<td>TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256&lt;br&gt;TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384</td>
</tr>
</tbody>
</table>
Configuring the SIP Trunk Security Profile
CUCM SIP Trunk Security Profile Configuration

Ensure Subject-name matches between IOS and CUCM SIP trunk

```
crypto pki trustpoint caServer
  enrollment terminal
  subject-name CN=ISR4KCUBE.cisco-uc.com
  revocation-check none
  rsakeypair rsakey
```
CUCM Configuring the IOS SIP Trunk

**SIP Information**

**Destination**
- Destination Address is an SRV
- 1* 14.50.248.31
  - Destination Address IPv6
  - 5061

**MTP Preferred Originating Codec**
- 711ulaw

**BLF Presence Group**
- Standard Presence group

**SIP Trunk Security Profile**
- Secure SIP Trunk Profile ISR4KCL

**Rerouting Calling Search Space**
- <None>

**Out-Of-Dialog Refer Calling Search Space**
- <None>

**SUBSCRIBE Calling Search Space**
- <None>

**SIP Profile**
- Standard SIP Profile Early-Offer
  - View Details

**DTMF Signaling Method**
- No Preference
Are Our Calls Completely Secure Now?
Implementing Secure Media
What’s Secure RTP?

As per RFC 3711, SRTP is “a profile of the Real-time Transport Protocol (RTP), which can provide confidentiality, message authentication, and replay protection to the RTP traffic”

SDP for RTP

```
m=audio 8256 RTP/AVP 0
c=IN IP4 14.50.248.31
a=rtpmap:0 PCMU/8000
```

SDP for SRTP

```
m=audio 8264 RTP/SAVP 0
c=IN IP4 14.50.248.31
a=rtpmap:0 PCMU/8000
a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline: L5+zq2AXJxLk+058lu/XRQWJZiK0c0D0
```
Secure RTP – SDP Security Descriptions (SDES)

Breaking down the crypto m-line

a=crypto:1 AES_CM_128_HMAC_SHA1_32 inline: L5+zq2AXJxLk+058lu/XRQWJZiK0c0D0

Tag
A decimal number used to identify the particular crypto attribute.

Crypto Suite
Describes the encryption and authentication transforms to be used in the sRTP media stream.
- AES_CM – AES Counter mode cipher
- 128 – The length of the encryption key (bits)
- HMAC_SHA1 – (Hashed) Message Authentication Code using the SHA1 algorithm

Transport Method
Defines the method of transporting the key. Inline meaning that the key and salt will follow immediately. Other methods (such as a URI) are not defined in rfc4568

Master Key and Master Salt
A 240bit base64 encoded string containing a concatenation of the master key and master salt
## Secure RTP – SDP Security Descriptions (SDES)

### Additional Parameters

<table>
<thead>
<tr>
<th>SRTP Session Parameters: Allows further session flexibility, such as unencrypted/unauthenticated RTCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>a=crypto:1 AES_CM_128_HMAC_SHA1_32 inline: L5+zq2AXJxLk+0581u/XRQWJZiK0c0D0 UNENCRIPTED_SRTCP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key Lifetime and Master Key Index: Not generally used in collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td>a=crypto:1 AES_CM_128_HMAC_SHA1_32 inline: L5+zq2AXJxLk+0581u/XRQWJZiK0c0D0</td>
</tr>
</tbody>
</table>
Secure RTP – SDP Signaling for DTLS

**Secure RTP – SDP Signaling for DTLS**

**DTLS SDP Attributes**

- a=setup:passive
- a=connection:new

**Setup**
Indicates DTLS support. The value is negotiated between both endpoints in the offer answer model.

**Connection**
Used to indicate whether the offer/answer exchange is using an existing connection.

**Fingerprint**
A cryptographic hash of the x509 certificate to be used in the key exchange and the hashing algorithm used.

**Setup Values**

- **active**: The endpoint will initiate an outgoing connection.
- **passive**: The endpoint will accept an incoming connection.
- **actpass**: The endpoint is willing to accept an incoming connection or to initiate an outgoing connection.
- **holdconn**: The endpoint does not want the connection to be established for the time being.

<table>
<thead>
<tr>
<th>Offer</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>passive</td>
</tr>
<tr>
<td>passive</td>
<td>active</td>
</tr>
<tr>
<td>actpass</td>
<td>active</td>
</tr>
<tr>
<td>holdconn</td>
<td>holdconn</td>
</tr>
</tbody>
</table>

**Offer**

- **active**: The endpoint will initiate an outgoing connection.
- **passive**: The endpoint will accept an incoming connection.
- **actpass**: The endpoint is willing to accept an incoming connection or to initiate an outgoing connection.
- **holdconn**: The endpoint does not want the connection to be established for the time being.
Secure RTP

SRTP Packet Format

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X| CC |M| PT | sequence number |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          timestamp                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     synchronization source (SSRC) identifier        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   contributing source (CSRC) identifiers           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                     RTP extension (OPTIONAL) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             payload ...                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          | RTP padding | RTP pad count |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         SRTP MKI (OPTIONAL) |                                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                              authentication tag (RECOMMENDED) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        +- Encrypted Portion*                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Authenticated Portion                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

RTP Headers
RTP Headers are authenticated, but not encrypted

Encrypted Portion
The payload, MKI (if present), and authentication tag are encrypted and authenticated
As per RFC 3711, SRTP is “a profile of the Real-time Transport Protocol (RTP), which can provide confidentiality, message authentication, and replay protection to the RTP traffic”

### Confidentiality
Confidentiality is achieved by encrypting the RTP payload using AES ciphers

### Message Authentication
HMAC provides the authentication mechanism by reducing the packet contents to a 160 bit number using SHA1

### Replay Protection
Replay attacks assume that the attacker has access to the SRTP stream and can send duplicated packets to the receiver in order to waste resources. The SRTP cryptographic context keeps track of the number of packets signed by the master key in a sliding window that is tolerant of loss and out of order packet delivery (up to 2^15 packets)
Secure Media – Expressway Configuration

Configuration Settings

**Force encrypted** – Attempt to encrypt any unencrypted RTP, fail the call if encryption is not available.

**Force unencrypted** – Decrypt any encrypted media. Force unencrypted RTP

**Best effort** – Use encryption if available, otherwise fall back to unencrypted media.

**Auto** – No specific media encryption policy is applied by the VCS. Media encryption is purely dependent on endpoint requests.

Note

Any setting other than **Auto** will force the call media to traverse the VCS and thus consume a traversal call license.
# Security modes explained

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
<th>Expressway Configuration</th>
<th>UCM Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>Media encryption is required. Unencrypted calls should always fail no fallback is allowed.</td>
<td>Force encrypted</td>
<td>Mixed Mode On&lt;br&gt;Phones Security Profile(s)&lt;br&gt;• Device Security Mode: Encrypted&lt;br&gt;• Transport Type: TLS&lt;br&gt;Trunk Settings&lt;br&gt;SIP Trunk Security Profile&lt;br&gt;• Incoming Transport Type: TLS&lt;br&gt;• Outgoing Transport Type: TLS&lt;br&gt;SRTCP Allowed Checked</td>
</tr>
<tr>
<td>Best Effort</td>
<td>Calls that can be encrypted are encrypted. If encryption cannot be established&lt;br&gt;Calls should fall back to unencrypted media if encryption cannot be established</td>
<td>Best Effort</td>
<td>Mixed Mode On&lt;br&gt;Phone and Trunk security settings as required&lt;br&gt;Normalization script applied</td>
</tr>
<tr>
<td>None</td>
<td>No encryption in use</td>
<td>Mixed Mode Off</td>
<td></td>
</tr>
</tbody>
</table>
Secure Media – IOS Configuration

Enabling Secure Media

1. Enable SRTP on Dial-peer

```
dial-peer voice 1 voip
description to CUCM Sub – Secure Signaling preference 1
destination-pattern 418110....
session protocol sipv2
session target ipv4:14.50.248.103
```

- `srtp` voice-class sip srtp-auth sha1-80 sha1-32
- `voice-class sip srtp_pass-thru`

1. Enable SRTP Globally

```
voice service voip

```

- `srtp`
- `srtp_pass-thru` sip
- `srtp-auth sha1-80 sha1-32`

1. Enables SRTP

2. Configure SRTP cipher suite support

In 15.4(1), support for sha1-80 AES_CM_128_HMAC_SHA1_80 was added

3. (Optional) Configure NGE cipher suite support.

Introduced in 15.6(1)

Allows for unsupported SRTP cipher suites to be negotiated,
- AEAD_AES_128_GCM
- AEAD_AES_256_GCM
- AEAD_AES_128_CCM
- AEAD_AES_256_CCM

CUBE will pass-thru offered cipher suites and keys from one call-leg to the other call-leg.
Secure Media – IOS-XE Configuration

Enabling Secure Media

1. Create a voice class to define supported SRTP cipher suites.

Define SRTP crypto suite support

- voice class srtp-crypto 1
  - crypto 1 AEAD_AES_256_GCM
  - crypto 2 AEAD_AES_128_GCM
  - crypto 3 AES_CM_128_HMAC_SHA1_80
  - crypto 4 AES_CM_128_HMAC_SHA1_32

2. Apply the defined voice-class either under the dial-peer or globally

2. Enable SRTP and apply voice-class crypto-suite on Dial-peer

dial-peer voice 1 voip
  description to CUCM Sub – Secure Signaling
  preference 1
  destination-pattern 418110....
  session protocol sipv2
  session target ipv4:14.50.248.103

3. Enables SRTP

2. Enable SRTP and apply voice-class crypto-suite Globally

voice service voip
  srtp
  sip
  voice-class sip srtp-crypto 1

Or

srtp
  voice-class sip srtp-crypto 1
CUCM Configuring the IOS SIP Trunk

- SRTP Allowed: When this flag is checked, Encrypted TLS needs to be configured in the network to provide end to end security. Failure to do so will expose keys and other information.
- Consider Traffic on This Trunk Secure: When using both sRTP and TLS
- Route Class Signaling Enabled: Default
- Use Trusted Relay Point: Default
Are Our Calls Completely Secure Now?
Secure to Non-Secure Interoperability
SRTP to RTP
Interworking
CUBE-based SRTP-RTP Interworking

ISR 4000 - 4400/4300-series routers
- Uses built-in crypto-engine
- No additional configuration required

ISR G2 - 2900/3900-series routers
- DSP required
- Leverages DSPfarm configuration
CUBE-based SRTP-RTP Interworking
ISR G2 - DSPfarm Configuration

1. Enable DSPfarm feature on DSP

```plaintext
ISRG2CUBE(config)#voice-card 0
ISRG2CUBE(config-voicecard)#dsp service dspfarm
```

2. Configure DSPfarm profile

```plaintext
dspfarm profile 1 transcode security
codec g729abr8
codec g729ar8
codec g711alaw
codec g711ulaw
maximum sessions 10
associate application CUBE
no shutdown
```
CUBE-based SRTP-RTP Interworking
ISR G2 - DSPfarm Verification

show dspfarm profile

Profile ID = 1, Service = TRANSCODING, Resource ID = 1
Profile Description:

Profile Service Mode : secure
Profile Admin State : UP
Profile Operation State : ACTIVE
Application : CUBE Status : ASSOCIATED
Resource Provider : FLEX_DSPRM Status : UP

Total Number of Resources Configured : 10
Total Number of Resources Available : 10
Total Number of Resources Out of Service : 0
Total Number of Resources Active : 0

Codec Configuration: num_of_codecs:4
Codec : g711ulaw, Maximum Packetization Period : 30
Codec : g711alaw, Maximum Packetization Period : 30
Codec : g729ar8, Maximum Packetization Period : 60
Codec : g729abr8, Maximum Packetization Period : 60
TLS : ENABLED
SRTP Fallback
Secure Call Establishment

```
Invite SDP: RTP/SAVP

200 OK SDP: RTP/SAVP
```

SRTP
Secure Call Establishment Failure

Invite
SDP: RTP/SAVP

488 Not Acceptable Media
SRTP Fallback Call Establishment

Invite
Supported: x-cisco-srtp-fallback
SDP: RTP/SAVP

200 OK
Supported: x-cisco-srtp-fallback
SDP: RTP/AVP
SRTP Fallback - IOS Configuration

Enable on Dial-peer

dial-peer voice 1 voip
description to CUCM - Secure Signaling
preference 1
destination-pattern 418110....
session protocol sipv2
session target ipv4:14.50.248.101
srtp fallback
voice-class sip srtp negotiate cisco
incoming called-number 9.T
dtmf-relay rtp-nge
codec g711ulaw
no vad

Enable Globally

voice service voip
srtp fallback
sip
srtp negotiate cisco
## Secure Media – Expressway and UCM Interop

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
<th>Expressway Behavior</th>
<th>Unified Communications Manager Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Media encryption is not allowed. Calls that require encryption should fail.</td>
<td>m=RTP/AVP media description</td>
<td>m=RTP/AVP media description</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No crypto attributes present in SDP</td>
<td>No crypto attributes present in SDP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>m=RTP/AVP media description</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>a=crypto lines in the SDP</td>
<td>No x-cisco-srtp-fallback header</td>
</tr>
<tr>
<td></td>
<td></td>
<td>x-cisco-srtp-fallback header</td>
<td></td>
</tr>
<tr>
<td>Mandatory</td>
<td>Media encryption is required. Unencrypted calls should always fail. No fallback is allowed</td>
<td>m=RTP/SAVP media description</td>
<td>m=RTP/SAVP media description</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a=crypto lines in the SDP</td>
<td>a=crypto lines in the SDP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>x-cisco-srtp-fallback header</td>
<td>x-cisco-srtp-fallback header</td>
</tr>
<tr>
<td>Best Effort</td>
<td>Calls that can be encrypted are encrypted. If encryption cannot be established, calls attempt to fall back to unencrypted media.</td>
<td>m=RTP/AVP In the Offer’s media description</td>
<td>m=RTP/SAVP media description</td>
</tr>
<tr>
<td></td>
<td></td>
<td>m=RTP/SAVP In the Answer’s media description</td>
<td>a=crypto lines in the SDP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a=crypto lines in the SDP</td>
<td>x-cisco-srtp-fallback header</td>
</tr>
<tr>
<td></td>
<td></td>
<td>x-cisco-srtp-fallback header</td>
<td></td>
</tr>
</tbody>
</table>

### Warning

Expressway will not send crypto keys without a secure signaling transport (TLS)
UCM Does not have this limitation and will send crypto keys in clear text over non secure (TCP, UDP) transport types
## Secure Media – vcs-interop Lua Script

<table>
<thead>
<tr>
<th>Direction</th>
<th>Conditions</th>
<th>Behavior</th>
<th>Applies to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound to UCM</td>
<td>m=RTP/AVP media description lines in the SDP</td>
<td>Convert media descriptions to RTP/SAVP</td>
<td>All requests containing SDP</td>
</tr>
<tr>
<td></td>
<td>a=crypto</td>
<td>Add ( x)-cisco-srtp-fallback header</td>
<td></td>
</tr>
<tr>
<td>Outbound to Expressway</td>
<td>m=RTP/SAVP media description lines in the SDP, or both of the a=setup and a=fingerprint attributes</td>
<td>Convert media descriptions to RTP/AVP</td>
<td>INVITEs only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outbound to Expressway</td>
<td>All Requests and Responses</td>
<td>Modify the RHS of the SIP URI to the Top Level Domain on the any of the following headers (if present): From, Remote-Party-Id, P-Asserted-Identity</td>
<td>All requests, including INVITEs with modified media descriptions</td>
</tr>
</tbody>
</table>
Troubleshooting
Troubleshooting Methodology
Identify and Quantify the Problem

- When was it first reported?
- How often does it occur?
- What’s the impact?
  - Individual user
  - Site
  - Entire deployment

Gather Information

- What’s changed?
- Software versions?
- Call flow?
- Network topology?
- Debugs/Traces/Packet Captures

Analyze Gathered Information and Narrow the Scope
Narrowing the Scope
Is it a Signaling or Media problem?

**Signaling Path**
- SIP
- TLS
- Media Negotiation

**Media Path**
- Media Encryption
- Media Decryption
- Voice Quality
Identifying the Problem

Start

Does an unsecure call work in the same call flow? No

• Troubleshoot and resolve before implementing secure configuration.
Identifying the Problem

Start

Does an unsecure call work in the same call flow?

Yes

Does the calling party hear ringback?

No

• Indicates the TCP or TLS connection failed to establish between CUCM, Expressway and/or CUBE.

• Focus on certificate exchange between client and server.
Identifying the Problem

Start

Does an unsecure call work in the same call flow?

Yes

Does the calling party hear ringback?

Yes

Does the call connect?

No

- Possible issue with media negotiation.
- Check SDP for codec/dtmf advertisement
Identifying the Problem

Start

Does an unsecure call work in the same call flow?

Yes

Does the calling party hear ringback?

Yes

Does the call connect?

Yes

Does the lock icon appear on the phone?

No

- Problem with establishing secure media
- Check SDP answer and verify secure media is being advertised.
Identifying the Problem

Start
Does an unsecure call work in the same call flow? Yes

Does the calling party hear ringback? Yes

Does the call connect? Yes

Does the lock icon appear on the phone? Yes

Secure call successfully negotiated!

End
Troubleshooting Tools
IOS Command Output

show call [active|history] voice brief

ISR4ICUBE#show call active voice brief

2: 11 572330360ms.1 +1380 pid:31 Answer 1052061 active
dur 00:00:10 tx:493/98600 rx:492/98400 dscp:0 media:0 audio tos:0xB8 video tos:0x88
IP 14.50.212.59:25482 SRTP: off rtt:0ms pl:0/0ms lost:0/0/0 delay:0/0/0ms g711ulaw
long duration call detected:n long duration call duration:n/a timestamp:n/a
LostPacketRate:0.00 OutOfOrderRate:0.00

2: 13 572330370ms.1 +1360 pid:21 Originate 4181101002 active
dur 00:00:10 tx:492/100368 rx:493/100572 dscp:0 media:0 audio tos:0xB8 video tos:0x88
IP 14.50.248.150:17102 SRTP: on rtt:0ms pl:0/0ms lost:0/0/0 delay:0/0/0ms g711ulaw
long duration call detected:n long duration call duration:n/a timestamp:n/a
LostPacketRate:0.00 OutOfOrderRate:0.00
**IOS Command Output**

```
show call [active|history] voice brief
```

```
ISR4KCUBE#show call active voice brief

2 : 11 572330360ms.1 +1380 pid:31 Answer 1052061 active
dur 00:00:10 tx:493/98600 rx:492/98400 dscp:0 media:0 audio tos:0xB8 video tos:0x88
IP 14.50.212.59:25482 SRTP: off rtt:0ms pl:0/0ms lost:0/0/0 delay:0/0/0ms g711ulaw
long duration call detected:n long duration call duration:n/a timestamp:n/a
LostPacketRate:0.00 OutOfOrderRate:0.00

2 : 13 572330370ms.1 +1360 pid:21 Originate 4181101002 active
dur 00:00:10 tx:492/100368 rx:493/100572 dscp:0 media:0 audio tos:0xB8 video tos:0x88
IP 14.50.248.150:17102 SRTP: on rtt:0ms pl:0/0ms lost:0/0/0 delay:0/0/0ms
g711ulaw long duration call detected:n long duration call duration:n/a timestamp:n/a
LostPacketRate:0.00 OutOfOrderRate:0.00
```
### IOS Command Output

**show sip-ua calls**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SIP Call ID</strong></td>
<td>4A540093-1F8511E6-8025BF77@14.50.248.31</td>
</tr>
<tr>
<td><strong>Calling Number</strong></td>
<td>1052061</td>
</tr>
<tr>
<td><strong>Called Number</strong></td>
<td>4181101002</td>
</tr>
<tr>
<td><strong>Called URI</strong></td>
<td>sip:4181101002@14.50.248.101:5061</td>
</tr>
<tr>
<td><strong>Source IP Address (Sig)</strong></td>
<td>14.50.248.31</td>
</tr>
<tr>
<td><strong>Destn SIP Req Addr:Port</strong></td>
<td>[14.50.248.101]:5061</td>
</tr>
<tr>
<td><strong>Destn SIP Resp Addr:Port</strong></td>
<td>[14.50.248.101]:5061</td>
</tr>
<tr>
<td><strong>Destination Name</strong></td>
<td>14.50.248.101</td>
</tr>
<tr>
<td><strong>State of the stream</strong></td>
<td>STREAM_ACTIVE</td>
</tr>
<tr>
<td><strong>Stream Call ID</strong></td>
<td>71</td>
</tr>
<tr>
<td><strong>Stream Type</strong></td>
<td>voice+dtmf (1)</td>
</tr>
<tr>
<td><strong>Negotiated Codec</strong></td>
<td>g711ulaw (160 bytes)</td>
</tr>
<tr>
<td><strong>Negotiated Dtmf-relay</strong></td>
<td>rtp-nte</td>
</tr>
<tr>
<td><strong>Media Source IP Addr:Port</strong></td>
<td>[14.50.248.31]:8128</td>
</tr>
<tr>
<td><strong>Media Dest IP Addr:Port</strong></td>
<td>[14.50.248.150]:24696</td>
</tr>
<tr>
<td><strong>Local Crypto Suite</strong></td>
<td>AES_CM_128_HMAC_SHA1_32</td>
</tr>
<tr>
<td><strong>Remote Crypto Suite</strong></td>
<td>AES_CM_128_HMAC_SHA1_32</td>
</tr>
</tbody>
</table>

**Unique SIP signaling identifier**

**Call leg identifier**

**IP address of media peer**

**Negotiated SRTP media crypto suites**
Debugging in IOS
Best Practices

- `service sequence-numbers`
- `service timestamps debug datatime localtime msec`
- `logging buffered 1000000 debug`
- `no logging console`
- `no logging monitor`

015839: May 22 19:04:40.642: CRYPTO_PKI(Cert Lookup) issuer="hostname=ISR4KCUBE,cn=ISR4KCUBE"

Provides more granular time stamps and correct timezone. Makes comparing debugs easier between platforms.

Identifies whether messages were dropped due to rate or queue limiting.
Debugging in IOS
Logging to a Memory Buffer (Short-term debugging)

```
service sequence-numbers
service timestamps debug datatime localtime msec
logging buffered 1000000 debug
no logging console
no logging monitor

Prevents debug output from displaying to telnet/ssh sessions.
Prevents debug output from displaying to console sessions.
Creates a buffer in memory to hold debug-level messages.
```
Debugging in IOS

Logging to syslog server

- service sequence-numbers
- service timestamps debug datetime localtime msec
- no logging console
- no logging monitor
- logging host <syslog_server> transport <type> port <port>
- logging trap debug

Routes debug-level output to configured syslog server

Defines the IP address or hostname of the syslog server

TCP is the recommended and more reliable transport

TCP: 601
UDP: 514
Diagnostic Logs

Anatomy

• Available through the Expressway Web UI (Maintenance > Diagnostics > Diagnostic Logging)

If packet capture is needed, be sure to select

Click to Download the logs

Click to Start

Click to Stop after the issue is reproduced
Diagnostic Logs

What do they capture?

- SIP/H.323 traffic
- MRA Reverse Proxy Traffic
- TCP, SSL, and DNS traffic
- Application logic, and much more

```
2017-09-19T14:01:46.462-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 18:01:46,462" Module="network.tcp" Level="DEBUG": Src-ip="146.20.193.73" Src-port="40342" Dst-ip="172.16.2.2" Dst-port="5062" Detail="TCP Connecting"
```

```
2017-09-19T14:01:46.462-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 18:01:46,462" Module="network.tcp" Level="DEBUG": Src-ip="146.20.193.73" Src-port="40342" Dst-ip="172.16.2.2" Dst-port="5062" Detail="TCP Connection Established"
```

```
2017-09-19T14:01:46.491-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 18:01:46,491" Module="developer.ssl" Level="INFO" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1974)" Method="::ttssl_continueHandshake" Thread="0x7f420863b700": Detail="Handshake in progress" Reason="want read/write"
```
Cisco TAC Tool: Collaboration Solution Analyzer

Upload log files

Click or drop files here

About the tool

This tool analyzes the log files from multiple products in collaboration space and displays details about great amount of communications flows (calls, MRA login, RTP/TCP/UDP streams, XMPP, STUN, etc.), configuration overview and diagnostic signatures highlighting known issues found and next action plan to resolve them. More info

When to use

Use this tool when troubleshooting any issue on your collaboration servers or endpoints. Diagnostic signatures will suggest next action plan in case any known issues were found. Alternatively, use the tool output to visualize and better understand the communication flows and configuration to troubleshooting the issue further. More info

Available files

Select Filename Size Product type

- diagnostic_log_vcse_2016-07-02_14_43_12.tar.gz

49.891 MB VCS

Select all

Run Analysis

Delete all
OpenSSL is an open-source implementation of the SSL and TLS protocols. It is widely used in Internet web servers, serving a majority of all web sites. The core library, written in the C programming language, implements basic cryptographic functions and provides various utility functions.

- **s_client** - A generic SSL/TLS client that can complete a TLS connection to any remote server:port combination
- **x509** - A collection of utilities for reading, creating, and verifying x509 certificates
- **errstr** - Error Number to Error String Conversion.
- **ciphers** - Cipher Suite Description Determination.
openssl s_client

```
panholt@whiskeyjack:~$ openssl s_client -connect amer-expressway01.ciscotac.net:5061
CONNECTED(00000003)
depth=2 C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", CN=Go Daddy Root Certificate Authority - G2
verify return:1
depth=1 C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2
verify return:1
depth=0 OU=Domain Control Validated, CN=amer-expressway01.ciscotac.net
verify return:1
---
Certificate chain
0 s:/OU=Domain Control Validated/CN=amer-expressway01.ciscotac.net
   i:/C=US/ST=Arizona/L=Scottsdale/O=GoDaddy.com, Inc./CN=Go Daddy Root Certificate Authority - G2
2 s:/C=US/ST=Arizona/L=Scottsdale/O=GoDaddy.com, Inc./CN=Go Daddy Root Certificate Authority - G2
   i:/C=US/ST=Arizona/L=Scottsdale/O=GoDaddy.com, Inc./CN=Go Daddy Root Certificate Authority - G2
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIGrDCCBZSg...
-----END CERTIFICATE-----
subject=/OU=Domain Control Validated/CN=amer-expressway01.ciscotac.net
---
No client certificate CA names sent
Peer signing digest: SHA512
Server Temp Key: ECDH, P-521, 521 bits
---
SSL handshake has read 4901 bytes and written 499 bytes
```

Socket is connected and the peers server cert is verified against local CA trust store

Server cert’s trust chain is displayed

PEM encoded server cert is displayed (use `showcerts` to display all the certs in the trust chain)

SSL/TLS Handshake is complete
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 4096 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>TLSv1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cipher</td>
<td>ECDHE-RSA-AES256-GCM-SHA384</td>
</tr>
<tr>
<td>Session-ID</td>
<td>A6A286C01CAE78D6F4A2F0A10E413AD121578DD01CFD1160B776E73F2E69A130</td>
</tr>
<tr>
<td>Session-ID-ctx</td>
<td>Master-Key: 3EF487C8C18BCA4D10A16D4E25DB31F68238425B80E11CC2869780E45047FED5CD8EC3F2D8858F6D68B5602808953F6</td>
</tr>
<tr>
<td>Key-Arg</td>
<td>None</td>
</tr>
<tr>
<td>PSK identity</td>
<td>None</td>
</tr>
<tr>
<td>PSK identity hint</td>
<td>None</td>
</tr>
<tr>
<td>SRP username</td>
<td>None</td>
</tr>
<tr>
<td>TLS session ticket lifetime hint: 300 (seconds)</td>
<td></td>
</tr>
</tbody>
</table>

Start Time: 1525379195
Timeout   : 300 (sec)
Verify return code: 0 (ok)
---
^C

Session statistics and verification results
openssl s_client (other usage)

Change the key/cert and/or CA trust file

- CApath arg: PEM format directory of CA's certificate file
- CAfile arg: PEM format file of CA's certificate file
- cert arg: certificate file to use, PEM format assumed
- certfile arg: certificate format (PEM or DER) PEM default
- key arg: Private key file to use, in cert file if not specified but cert file is.

Test different protocols or cipher suites

- starttls prot: use the STARTTLS command before starting TLS for those protocols that support it, where 'prot' defines which one to assume. Currently, only "smtp", "pop3", "imap", "ftp" and "xmpp" are supported.
- ssl2: just use SSLv2
- ssl3: just use SSLv3
- tls1_2: just use TLSv1.2
- tls1_1: just use TLSv1.1
- tls1: just use TLSv1
- no_tls1_2/-no_tls1_1/-no_tls1/-no_ssl3/-no_ssl2: turn off that protocol

Various debugging and extra output

- mtu: set the link layer MTU
- showcerts: show all certificates in the chain
- debug: extra output
- msg: Show protocol messages
- tlsextdebug: hex dump of all TLS extensions received
- servername host: Set TLS extension servername in ClientHello
- alpn arg: enable ALPN extension, named protocols supported (comma-separated list)

TLS Extension testing and debugging (SNI, ALPN)
openssl x509

Certificate:
Data:
Version: 3 (0x2)
Serial Number: 8041775998364328096 (0x6f9a20a9d1c4a0a0)
Signature Algorithm: sha256WithRSAEncryption
Validity
Not Before: May 31 14:48:01 2017 GMT
Not After : May 31 14:48:01 2020 GMT
Subject: OU=Domain Control Validated, CN=amer-expressway01.ciscotac.net
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (4096 bit)
Modulus:
00:8e:26:ce:19:f6:3e:a4:33:
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints: critical
 CA:FALSE
X509v3 Extended Key Usage:
 TLS Web Server Authentication, TLS Web Client Authentication
X509v3 Key Usage: critical
 Digital Signature, Key Encipherment
X509v3 CRL Distribution Points:

{snip}

X509v3 Subject Alternative Name:
 DNS:amer-expressway01.ciscotac.net, DNS:www.amer-expressway01.ciscotac.net, DNS:rtp.ciscotac.net, DNS:ciscotac.net, DNS:amer-expressway01.rtp.ciscotac.net
**openssl x509 (other uses)**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inform arg</td>
<td>- input format - default PEM (one of DER, NET or PEM)</td>
</tr>
<tr>
<td>outform arg</td>
<td>- output format - default PEM (one of DER, NET or PEM)</td>
</tr>
<tr>
<td>keyform arg</td>
<td>- private key format - default PEM</td>
</tr>
<tr>
<td>in arg</td>
<td>- input file - default stdin</td>
</tr>
<tr>
<td>out arg</td>
<td>- output file - default stdout</td>
</tr>
<tr>
<td>subject</td>
<td>- print subject DN</td>
</tr>
<tr>
<td>issuer</td>
<td>- print issuer DN</td>
</tr>
<tr>
<td>email</td>
<td>- print email address(es)</td>
</tr>
<tr>
<td>startdate</td>
<td>- notBefore field</td>
</tr>
<tr>
<td>enddate</td>
<td>- notAfter field</td>
</tr>
<tr>
<td>dates</td>
<td>- both Before and After dates</td>
</tr>
<tr>
<td>text</td>
<td>- print the certificate in text form</td>
</tr>
<tr>
<td>x509toreq</td>
<td>- output a certification request object</td>
</tr>
</tbody>
</table>
## Data to Collect

### Media Establishment

<table>
<thead>
<tr>
<th>Type of Problem</th>
<th>IOS Debugs</th>
<th>IOS Command Output</th>
<th>CUCM</th>
<th>Expressway</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRTP-RTP Interworking ISR-G2 only</td>
<td>debug voip ipipgw</td>
<td>show dspfarm profile active</td>
<td></td>
<td></td>
<td>Start Log Collection</td>
</tr>
<tr>
<td></td>
<td>debug voip hpi</td>
<td>show voip rtp connection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>debug ccsip</td>
<td>error</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>info</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Media</td>
<td>debug ccsip</td>
<td>media</td>
<td>show call active</td>
<td>history voice brief</td>
<td>CCM traces</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>show sip-ua call</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Phone Registration

<table>
<thead>
<tr>
<th>Type of Problem</th>
<th>CUCM</th>
<th>Expressway</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSC Installation</td>
<td>CAPF traces</td>
<td>Start Log Collection</td>
<td>Phone console logs</td>
</tr>
<tr>
<td>Secure Phone Registration</td>
<td>CCM traces</td>
<td></td>
<td>Packet capture</td>
</tr>
<tr>
<td>CTL Installation</td>
<td>TFTP traces</td>
<td>Start Log Collection</td>
<td></td>
</tr>
<tr>
<td>Type of Problem</td>
<td>IOS Debugs</td>
<td>IOS command output</td>
<td>CUCM</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------</td>
<td>------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>TCP connection failure</td>
<td>debug ip tcp</td>
<td>transaction packet show tcp brief</td>
<td></td>
</tr>
<tr>
<td>TLS connection failure</td>
<td>debug crypto pki</td>
<td>messages transactions validation api</td>
<td></td>
</tr>
<tr>
<td></td>
<td>debug ssl openssl</td>
<td>errors msg states</td>
<td></td>
</tr>
<tr>
<td>SIP call establishment</td>
<td>debug ccsip</td>
<td>message error transport</td>
<td></td>
</tr>
<tr>
<td>Call Routing</td>
<td>debug voip ccapi inout</td>
<td>show dial-peer voice summary</td>
<td></td>
</tr>
</tbody>
</table>
Troubleshooting Scenarios
Scenario 1: Call Setup Failure
Problem Description

Initial Problem Description

Outbound calls work but inbound fail.

Secure voice between CUCM and SIP-SIP CUBE on an ISR4451-X.
Unsecure between CUBE and PSTN.
IOS Command Output after Outbound Call

### show sip-ua connection tcp tls detail

---

**ISR4KCUBE#show sip-ua connection tcp tls detail**

<table>
<thead>
<tr>
<th>Total active connections : 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of send failures          : 0</td>
</tr>
<tr>
<td>No. of remote closures        : 0</td>
</tr>
<tr>
<td>No. of conn. failures         : 0</td>
</tr>
<tr>
<td>No. of inactive conn. ageouts : 0</td>
</tr>
<tr>
<td>TLS client handshake failures : 0</td>
</tr>
<tr>
<td>TLS server handshake failures : 0</td>
</tr>
</tbody>
</table>

---

**Printing Detailed Connection Report**

- **Remote-Agent:** 14.50.248.103, **Connections-Count:** 2
- **Connection failure counters**
- **Active connections counter**
- **IP address of signaling peer**

```
                  Remote-Port Conn-Id Conn-State      WriteQ-Size Local-Address TLS-Version
--------- ------- ----------- ------------------ --------------- ------------------ ---------------
54715 122     Established 0                     0               -                  TLSv1.2
```
**IOS Command Output after Inbound Call Attempt**

*show sip-ua connection tcp tls detail*

---

ISR4KCUBE#show sip-ua connection tcp tls detail
Total active connections : 1
No. of send failures     : 0
No. of remote closures   : 0
No. of conn. failures    : 1
No. of inactive conn. ageouts : 0
TLS client handshake failures : 1
TLS server handshake failures : 0

---------Printing Detailed Connection Report---------

Remote-Agent:14.50.248.101, Connections-Count:2
Remote-Port Conn-Id Conn-State WriteQ-Size Local-Address TLS-Version
================= ======= =========== =========== =============== ===========
54715            122 Established          0             -             TLSv1.2

---

Increase in failure counters
Troubleshooting

Inbound call

14.50.211.3  14.50.211.41

INVITE

Jun 22 00:16:55.003: //1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:

Received:
INVITE sip:1001@14.50.211.41:5060 SIP/2.0
Via: SIP/2.0/TCP 14.50.211.31:5060;branch=z9hG4bx6e7c6463
From: <sip:1052062@14.50.211.31>;tag=44900~bb0e39ab-e0bf-401d
To: <sip:1001@14.50.211.41>
Date: Thu, 22 Jun 2017 00:16:55 GMT
Call-ID: 17c53b00-94b10c77-52-1fd3320e@14.50.211.31

Unique call identifier

Remote media IP address and RTP port

Audio codec
G711ulaw

DTMF-relay RFC2833

debug ccsip message
Troubleshooting

Inbound call

14.50.211.31  14.50.211.182

INVITE

Jun 22 00:16:55.021: //1/17C53B000000/CCAPI/cc_api_call_setup_ind_common
Interface=0x7F12BB8C3E80, Call Info
Calling Number=1052062 (Calling Name=)(TON=Unknown, NPI=Unknown),
Called Number=1001 (TON=Unknown, NPI=Unknown),
Incoming Dial-peer=3, Progress Indication=NULL(0), Calling IE Present=TRUE,
Source Trkgrp Route Label=, Target Trkgrp Route Label=, CLID Transparent=FALSE), Call

Jun 22 00:16:55.025: //33288/17C53B000000/CCAPI/ccCallSetupRequest
Calling Number=1052062 (TON=Unknown, NPI=Unknown, Screening=User, Passed),
Called Number=1001 (TON=Unknown, NPI=Unknown),
Account Number=1052062, Final Destination Flag=TRUE,
Guid=17C53B00-0001-0000-003A1FD3320E, Outgoing Dial-peer=2

debug voip ccapi inout

dial-peer voice 3 voip
description inbound PSTN unsecure translation-profile incoming switch21
session transport tcp
incoming called-number 1...
dtmf-relay rtp-n-te
codec g711ulaw
no vad

dial-peer voice 2 voip
description to CUCM Pub – Secure
preference 1
destination-pattern 1...
session protocol sipv2
session target ipv4:14.50.248.100
session transport tcp tls
incoming called-number 8.T
dtmf-relay rtp-n-te
srtp
codec g711ulaw
no vad
Troubleshooting

14.50.211.31

TCP Established

14.50.248.100

Jun 22 00:16:55.053: //-/xxxxxxxxxxxxx/SIP/Transport/sipTransportPostRequestConnection:
Posting TLS conn create request for addr=14.50.248.100, port=5061, context=0x7F12BB4CB6A8

Jun 22 00:16:55.055: TCP: sending SYN, seq 1205252257, ack 0
Jun 22 00:16:55.056: TCP0: Connection to 14.50.248.100:5061, advertising MSS 1460
Jun 22 00:16:55.056: TCP0: state was CLOSED -> SYNSENT [32347 -> 14.50.248.100(5061)]
Jun 22 00:16:55.057: TCP0: state was SYNSENT -> ESTAB [32347 -> 14.50.248.100(5061)]

Jun 22 00:16:55.059: openssl_SetPKIInfo entry
Jun 22 00:16:55.059: CRYPTO_PKI: (A0050) Session started - identity selected (caServer)
Jun 22 00:16:55.059: CRYPTO_PKI(Cert Lookup) issuer="cn=BRKUCC_CA" serial number=
10 FC B1 6C 00 00 00 00 00 04
Troubleshooting

Jun 22 00:16:55.062: Handshake start: before/connect initialization
Jun 22 00:16:55.062: SSL_connect:before/connect initialization
Jun 22 00:16:55.063: >>> TLS 1.2 Handshake [length 0072], ClientHello
Jun 22 00:16:55.063:     01 00 00 6E 03 03 9D 9A D4 49 99 E2 98 90 A0 2A

Jun 22 00:16:55.073: <<< TLS 1.2 Handshake [length 003E], ServerHello
Jun 22 00:16:55.073:     02 00 00 3A 03 03 BF 47 15 B3 D1 DD 5F 01 A4 99

Jun 22 00:16:55.075: <<< TLS 1.2 Handshake [length 03C3], Certificate
Jun 22 00:16:55.075:     0B 00 03 BF 00 03 BC 00 03 B9 30 82 03 B5 30 82
Troubleshooting

Server Certificate analysis

Jun 22 00:16:55.101: CRYPTO_PKI: Added x509 peer certificate - (953) bytes

Jun 22 00:16:55.102: CRYPTO_PKI(Cert Lookup) issuer="l=RTP,st=NC,cn=cucmpub.cisco-uc.com,ou=TAC,o=Cisco,c=US" serial number= 6C 47 C6 EC 38 9B D1 23 75 7F 04 65 63 AE AB 09

Jun 22 00:16:55.102: CRYPTO_PKI: Cert record not found, returning E_NOT_FOUND

Jun 22 00:16:55.102: CRYPTO_PKI: (A0051) No suitable trustpoints found
Troubleshooting

TLS Alert
Jun 22 00:16:55.103: >>> TLS 1.2 Alert [length 0002], fatal bad_certificate
Jun 22 00:16:55.103: 02 2A

Jun 22 00:16:55.112: //33288/17C53B000000/CCAPI/ccCallDisconnect:
Cause Value=38, Tag=0x0, Call Entry(Previous Disconnect Cause=0, Disconnect Cause=0)
Jun 22 00:16:55.112: //33288/17C53B000000/CCAPI/ccCallDisconnect:
Cause Value=38, Call Entry(Responded=TRUE, Cause Value=38)
Verifying Certificates

ISR4KCUBE#sh crypto pki certificate
CA Certificate
  Status: Available
  Certificate Serial Number (hex): 3A0E15FE9124C0B040801DC535DBA5C3
  Certificate Usage: Signature
  Issuer:
    cn=BRKUCC_CA
  Subject:
    cn=BRKUCC_CA
  Validity Date:
    start date: 17:44:42 EDT May 29 2017
    end date: 17:54:41 EDT May 29 2022
  Associated Trustpoints: caServer
Verifying Certificates

Jun 22 00:16:55.101: CRYPTO_PKI: Added x509 peer certificate - (953) bytes
Jun 22 00:16:55.102: CRYPTO_PKI(Cert Lookup) issuer="l=RTP,st=NC,cn=cucmpub.cisco-uc.com,ou=TAC,o=Cisco,c=US" serial number= 6C 47 C6 EC 38 9B D1 23 75 7F 04 65 63 AE AB 09

<table>
<thead>
<tr>
<th>Certificate</th>
<th>Common Name</th>
<th>Type</th>
<th>Key Type</th>
<th>Distribution</th>
<th>Issued By</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallManager</td>
<td>cucmpub.cisco-uc.com</td>
<td>Self-signed</td>
<td>RSA</td>
<td>cucmpub.cisco-uc.com</td>
<td>cucmpub.cisco-uc.com</td>
</tr>
<tr>
<td>CallManager-ECDSA</td>
<td>cucmpub-EC.cisco-uc.com</td>
<td>Self-signed</td>
<td>EC</td>
<td>cucmpub.cisco-uc.com</td>
<td>cucmpub-EC.cisco-uc.com</td>
</tr>
<tr>
<td>CallManager-trust</td>
<td>CAPF-26688e56</td>
<td>CA-signed</td>
<td>RSA</td>
<td>CAPF-26688e56</td>
<td>BRKUCC_CA</td>
</tr>
<tr>
<td>CallManager-trust</td>
<td>CAP-RTP-001</td>
<td>Self-signed</td>
<td>RSA</td>
<td>CAP-RTP-001</td>
<td>CAP-RTP-001</td>
</tr>
<tr>
<td>CallManager-trust</td>
<td>ACT2_SUDI_CA</td>
<td>CA-signed</td>
<td>RSA</td>
<td>ACT2_SUDI_CA</td>
<td>Cisco_Root_CA_2014</td>
</tr>
<tr>
<td>CallManager-trust</td>
<td>Cisco_Manufacturing_CA_SHA2</td>
<td>CA-signed</td>
<td>RSA</td>
<td>Cisco_Manufacturing_CA_SHA2</td>
<td>Cisco_Root_CA_2014</td>
</tr>
<tr>
<td>CallManager-trust</td>
<td>Cisco_Root_CA_2014</td>
<td>CA-signed</td>
<td>RSA</td>
<td>Cisco_Root_CA_2014</td>
<td>Cisco_Root_CA_2014</td>
</tr>
<tr>
<td>CallManager-trust</td>
<td>Cisco_Root_CA_M2</td>
<td>Self-signed</td>
<td>RSA</td>
<td>Cisco_Root_CA_M2</td>
<td>Cisco_Root_CA_M2</td>
</tr>
<tr>
<td>CallManager-trust</td>
<td>BRKUCC_CA</td>
<td>Self-signed</td>
<td>RSA</td>
<td>BRKUCC_CA</td>
<td>BRKUCC_CA</td>
</tr>
<tr>
<td>CallManager-trust</td>
<td>CAPP-0a5e9493</td>
<td>Self-signed</td>
<td>RSA</td>
<td>CAPP-0a5e9493</td>
<td>CAPP-0a5e9493</td>
</tr>
<tr>
<td>CallManager-trust</td>
<td>cucmsub.cisco-uc.com</td>
<td>CA-signed</td>
<td>RSA</td>
<td>cucmsub.cisco-uc.com</td>
<td>BRKUCC_CA</td>
</tr>
<tr>
<td>CallManager-trust</td>
<td>Cisco_Manufacturing_CA</td>
<td>CA-signed</td>
<td>RSA</td>
<td>Cisco_Manufacturing_CA</td>
<td>Cisco_Root_CA_2014</td>
</tr>
<tr>
<td>CallManager-trust</td>
<td>CAP-RTP-002</td>
<td>Self-signed</td>
<td>RSA</td>
<td>CAP-RTP-002</td>
<td>CAP-RTP-002</td>
</tr>
</tbody>
</table>
Summary

Root Cause

• CUBE only has a dial-peer pointing to CUCM Pub for inbound PSTN calls.
• CA-signed identity certificate was not imported into CUCM Pub
• Working outbound calls were sent by the CUCM Subscriber.

Solution

• Import CA-signed identity certificate to CUCM Pub and restart CallManager service.
• Create dial-peer pointing to CUCM Subscriber for redundancy.
Additional UC Security Sessions

**BRKCOL-3501**: Implementing and Troubleshooting Secure IP Phones and Endpoints
- Wednesday 13th at 1:30pm

**BRKCOL-2014**: Introduction to Cisco UC Security
- Watch the video online at [www.ciscolive.com](http://www.ciscolive.com) when it’s posted in 2–3 weeks!
Complete your online session evaluation

Give us your feedback to be entered into a Daily Survey Drawing.

Complete your session surveys through the Cisco Live mobile app or on www.CiscoLive.com/us.

Don’t forget: Cisco Live sessions will be available for viewing on demand after the event at www.CiscoLive.com/Online.
Continue your education

- Demos in the Cisco campus
- Walk-in self-paced labs
- Meet the engineer 1:1 meetings
- Related sessions
Thank you
## Collaboration Cisco education offerings

<table>
<thead>
<tr>
<th>Course</th>
<th>Description</th>
<th>Cisco Certification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCIE Collaboration Advanced Workshop (CIEC)</td>
<td>Gain expert-level skills to integrate, configure, and troubleshoot complex collaboration networks</td>
<td>CCIE® Collaboration</td>
</tr>
<tr>
<td>Implementing Cisco Collaboration Applications (CAPPS)</td>
<td>Understand how to implement the full suite of Cisco collaboration applications including Jabber,</td>
<td>CCNP® Collaboration</td>
</tr>
<tr>
<td></td>
<td>Cisco Unified IM and Presence, and Cisco Unity Connection.</td>
<td></td>
</tr>
<tr>
<td>Implementing Cisco IP Telephony and Video Part 1 (CIPTV1)</td>
<td>Learn how to implement Cisco Unified Communications Manager, CUBE, and audio and videoconferences in a single-site voice and video network.</td>
<td>CCNP® Collaboration</td>
</tr>
<tr>
<td>Implementing Cisco IP Telephony and Video Part 2 (CIPTV2)</td>
<td>Obtain the skills to implement Cisco Unified Communications Manager in a modern, multisite collaboration environment.</td>
<td></td>
</tr>
<tr>
<td>Troubleshooting Cisco IP Telephony and Video (CTCOLLAB)</td>
<td>Troubleshoot complex integrated voice and video infrastructures.</td>
<td></td>
</tr>
<tr>
<td>Implementing Cisco Collaboration Devices (CICD)</td>
<td>Acquire a basic understanding of collaboration technologies like Cisco Call Manager and Cisco Unified Communications Manager.</td>
<td>CCNA® Collaboration</td>
</tr>
<tr>
<td>Implementing Cisco Video Network Devices (CIVND)</td>
<td>Learn how to evaluate requirements for video deployments, and implement Cisco Collaboration endpoints in converged Cisco infrastructures.</td>
<td></td>
</tr>
</tbody>
</table>

For more details, please visit: [http://learningnetwork.cisco.com](http://learningnetwork.cisco.com)

Questions? Visit the Learning@Cisco Booth