

# Troubleshooting Guide for Cisco Webex Hybrid Call Service Connect

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Call Set-Up Issues](#)

[Mutual TLS Handshake Failures](#)

[Useful Mutual TLS Troubleshooting Tips](#)

[Issue 1. Expressway-E does not Trust Certificate Authority \(CA\) that signed the Cisco Webex Certificate](#)

[Issue 2. Incorrect Name for TLS Subject Verify Name on Expressway-E Cisco Webex Hybrid DNS Zone](#)

[Issue 3. Expressway-E does not Send Full Certificate Chain to Cisco Webex](#)

[Issue 4. Firewall Terminates Mutual TLS Handshake](#)

[Issue 5. Expressway-E is Signed by Public CA but Cisco Webex Control Hub has Alternate Certificates Loaded](#)

[Issue 6. Expressway is not Mapping Inbound Call to Cisco Webex Hybrid DNS Zone](#)

[Issue 7. Expressway-E uses Default Self-Signed Certificate](#)

[Inbound: Cisco Webex to On-Premises](#)

[Issue 1. Cisco Webex is unable to resolve the Expressway-E DNS SRV/hostname](#)

[Issue 2. Socket Failure: Port 5062 is Blocked Inbound to Expressway](#)

[Issue 3. Socket Failure: Expressway-E is not Listening on Port 5062](#)

[Issue 4. Expressway-E or C does not Support Preloaded SIP Route Headers](#)

[Issue 5. Cisco Webex app is receiving two call notifications \(toasts\)](#)

[Outbound: On-Premises to Cisco Webex](#)

[Issue 1. Expressway is unable to resolve the callservice.ciscospark.com address](#)

[Issue 2. Port 5062 is blocked outbound to Cisco Webex](#)

[Issue 3. Expressway Search rule misconfiguration](#)

[Issue 4. Expressway CPL misconfiguration](#)

[Bidirectional: Cisco Webex to On-Premises or On-Premises to Cisco Webex](#)

[Issue 1. IP Phone/Collaboration Endpoint is offering an audio codec other than G.711, G.722, or AAC-LD.](#)

[Issue 2. Unified CM Max Incoming Message Size Exceeded](#)

[Appendix](#)

[Expressway Troubleshooting Tools](#)

[Check Pattern Utility](#)

[Locate Utility](#)

[Diagnostic Logging](#)

[Related Information](#)

# Introduction

This document describes the Cisco Webex Hybrid Call Service Connect solution that allows your existing Cisco call control infrastructure to connect to the Cisco Collaboration Cloud so that they can work together.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

- Knowledge of the Cisco Webex Offer
- Knowledge of the Expressway solution (B2B)
- Knowledge of Cisco Unified Communications Manager (Unified CM) and its integration with Expressway
- Unified CM 10.5(2) SU5 or later.
- Expressway (B2B) version X8.7.1 or later (X8.9.1 is recommended)
- Expressway (Connector Host) -- see [Expressway Connector Host Support for Cisco Webex Hybrid Services](#) for the currently supported versions

### Components Used

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

- Cisco Unified Communications Manager
- Expressways
- Webex for Windows
- Webexfor Mac
- Webexfor iOS
- Webex for Android
- Cisco Collaboration Endpoints
- Collaboration Desk Endpoints
- IP Phones
- Software Clients

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

The solution offers these capabilities:

- Use the Webex app as a mobile soft client for audio and video calls
- Use the app to make and receive calls from anywhere, as if they were in the office
- Use Webex, Cisco Jabber, or their desk phone to call, without the need to worry about which

option they use

- Unlock call history in on-premise phones and integrate that history in Webex

The scope of this guide is to cover issues that are unique to Hybrid Call Service Connect. Since Hybrid Call Service Connect runs over the same Expressway E & C pair as other solutions such as Mobile and Remote Access and Business to Business calls, issues with the other solutions can affect Hybrid Call Service Connect. For customers and partners who deploy an Expressway pair for use with Call Service Connect, the [Cisco VCS Expressway and VCS Control Basic Configuration guide](#) must be referenced before you attempt to deploy Hybrid Call Service Connect. This troubleshooting guide covers Firewall/NAT considerations along with Expressway design in both Appendix 3 & 4. Review this documentation thoroughly. Additionally, this document assumes that the Expressway connector host and Hybrid Call Service activation were completed.

## Call Set-Up Issues

### Mutual TLS Handshake Failures

Hybrid Call Service Connect uses mutual transport layer security (mutual TLS) for authentication between Cisco Webex and the Expressway-E. This means that both the Expressway-E and Cisco Webex check and inspect the certificate that each other present. Since mutual TLS issues are so prevalent during new deployments of the Expressway servers and the enablement of solutions such as Hybrid Call Service Connect, this section provides useful information and tips for troubleshooting certificate-based issues between the Expressways and Cisco Webex.

What does the Expressway-E check?

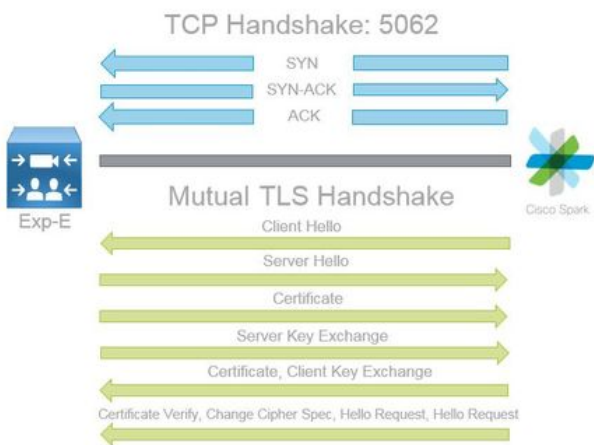
- Was the Cisco Webex certificate signed by a Public CA that is listed in the Expressway-E Trusted CA list?
- Is `callservice.ciscospark.com` present in the Subject Alternate Name field of the Cisco Webex certificate?

What does Cisco Webex check?

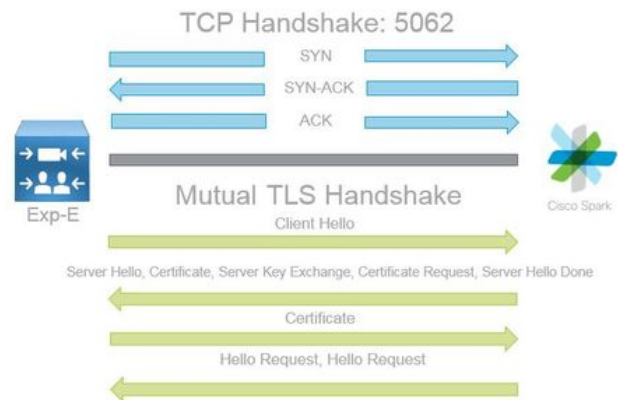
- Has the Expressway-E certificate been signed by one of the Public CAs that Webex trusts? ([Cisco Webex Trusted CA List](#))
- If the Expressway-E does not use a publicly signed certificate, was the Expressway certificate along with any root and intermediate certificates uploaded to the Cisco Webex Control Hub (<https://admin.ciscospark.com>)?

This is explained as shown in the image.

## Spark to On Premise



## On Premise to Spark



## Useful Mutual TLS Troubleshooting Tips

### 1. Decode Mutual TLS Handshake

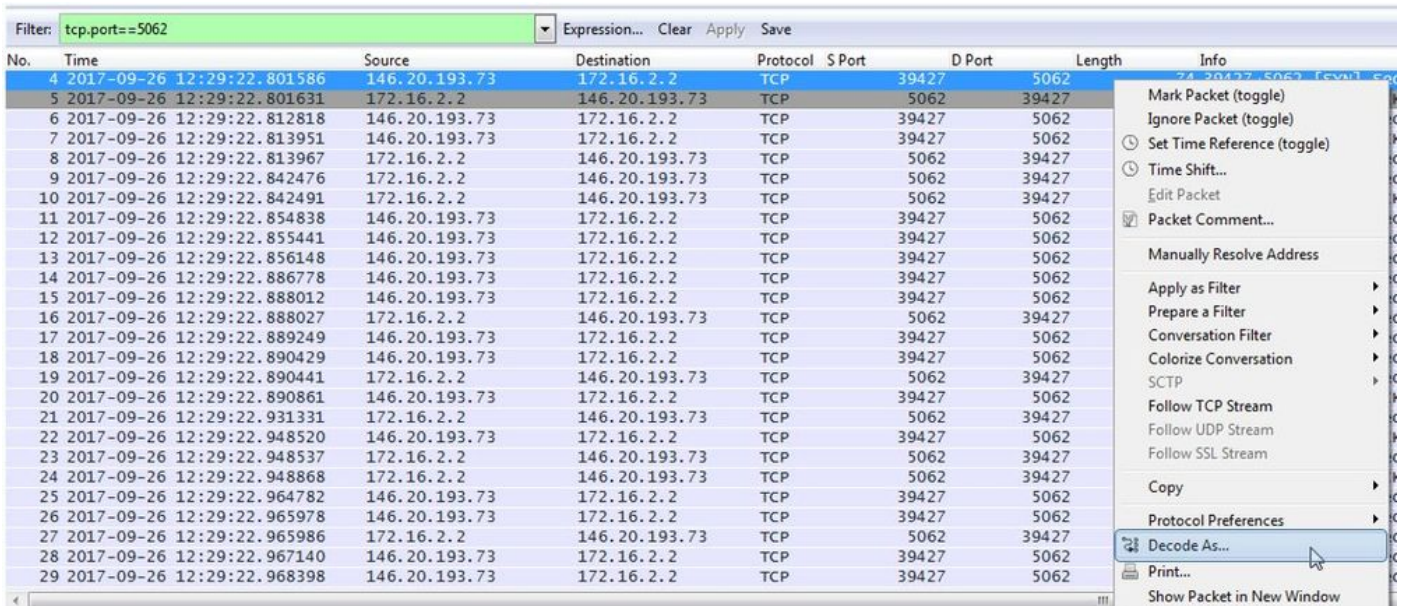
By default, Wireshark marks SIP TLS traffic as port 5061. What this means is that any time you want to analyze a (mutual) TLS handshake that occurs over port 5062, Wireshark will not know how to decode the traffic properly. Here is an example of the Mutual TLS handshake that's occurring over port 5062 as shown in the image.

No.	Time	Source	Destination	Protocol	S Port	D Port	Length	Info
169	2017-09-20 14:22:13.293817	146.20.193.45	172.16.2.2	TCP	48520	5062	74	48520->5062 [SYN] Seq=0 Win=14600 Len=0 MSS=1380 SACK_PERM=1 TSval=3875387337 TSecr=0 WS=128
170	2017-09-20 14:22:13.293846	172.16.2.2	146.20.193.45	TCP	5062	48520	74	5062->48520 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=444315393 TSecr=
171	2017-09-20 14:22:13.304549	146.20.193.45	172.16.2.2	TCP	48520	5062	66	48520->5062 [ACK] Seq=1 Ack=1 Win=14720 Len=0 TSval=3875387348 TSecr=444315393
172	2017-09-20 14:22:13.305898	146.20.193.45	172.16.2.2	TCP	48520	5062	266	48520->5062 [PSH, ACK] Seq=1 Ack=1 Win=14720 Len=200 TSval=3875387349 TSecr=444315393
173	2017-09-20 14:22:13.305911	172.16.2.2	146.20.193.45	TCP	5062	48520	66	5062->48520 [ACK] Seq=1 Ack=201 Win=30080 Len=0 TSval=444315405 TSecr=3875387349
174	2017-09-20 14:22:13.336342	172.16.2.2	146.20.193.45	TCP	5062	48520	2802	5062->48520 [ACK] Seq=1 Ack=201 Win=30080 Len=2736 TSval=444315436 TSecr=3875387349
175	2017-09-20 14:22:13.336358	172.16.2.2	146.20.193.45	TCP	5062	48520	1426	5062->48520 [PSH, ACK] Seq=2737 Ack=201 Win=30080 Len=1360 TSval=444315436 TSecr=3875387349

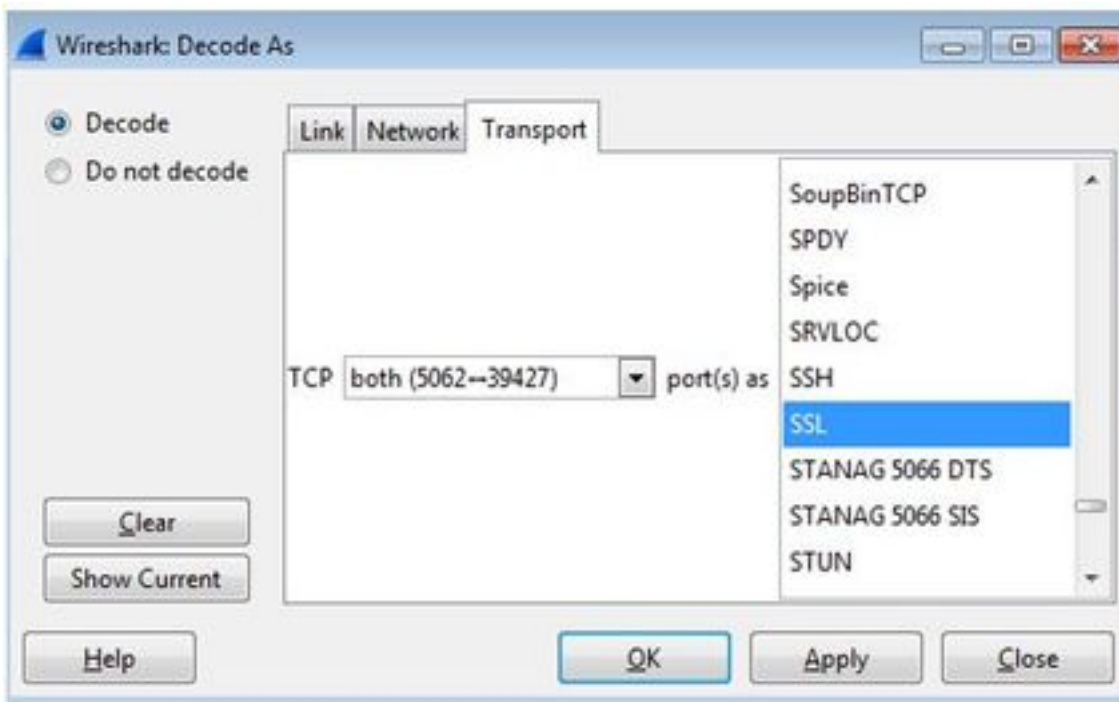
As you can see, this is how the handshake looks with the default settings in Wireshark. Packet number 175 is the certificate the Expressway sends to Cisco Webex. However, you can't determine that without the traffic being decoded. There are two methods that you can use to decode this traffic so that you can more easily see the certificate information and any error messages that are present.

#### 1a. Decod the Stream as SSL

a. When you analyze the Mutual TLS handshake, first filter the capture by **tcp.port==5062**. After this, right-click the first packet in the stream and select **Decode As...** as shown in the image.



b. Once the **Decode As...** option is selected, you see a list where you can select how to Decode the stream you've selected. From the list, select **SSL**, click **Apply** and close the window. At this point, the entire stream shows the certificate and error messages exchanged at the time of the handshake as shown in the image.



1b. Adjust SIP TLS Port

When you adjust the SIP TLS port to 5062 in the Wireshark preferences, you can then see all the details that surround the handshake, which includes the certificates. In order to make this change:

- Open Wireshark
- Navigate to **Edit > Preferences**
- Expand Protocols and select **SIP**
- Set the SIP TLS Port to 5062 and click **Apply**
- Set the value back to 5061 when the analysis is completed as shown in the image.

SIP TCP ports:

SIP TLS Port:

Display raw text for SIP message:

If you analyze the same capture now, you see packets 169 through 175 decoded. Packet 175 shows the Expressway-E certificate and if you drill down on the packet, you can see all the certificate details as shown in the image.

No.	Time	Source	Destination	Protocol	S Port	D Port	Length	Info
169	2017-09-20 14:22:13.293817	146.20.193.45	172.16.2.2	TCP	48520	5062	74	48520->5062 [SYN] Seq=0 Win=14600 Len=0 MSS=1380 SACK_PERM=1 TSval=3875387337 TSecr=0 WS=128
170	2017-09-20 14:22:13.293846	172.16.2.2	146.20.193.45	TCP	5062	48520	74	5062->48520 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=444315393 TSecr=3875387337 WS=128
171	2017-09-20 14:22:13.304549	146.20.193.45	172.16.2.2	TCP	48520	5062	66	48520->5062 [ACK] Seq=1 Ack=1 Win=14720 Len=0 TSval=3875387348 TSecr=444315393
172	2017-09-20 14:22:13.305898	146.20.193.45	172.16.2.2	TLSv1.2	48520	5062	268	Client Hello
173	2017-09-20 14:22:13.305911	172.16.2.2	146.20.193.45	TCP	5062	48520	66	5062->48520 [ACK] Seq=1 Ack=201 Win=30080 Len=0 TSval=444315405 TSecr=3875387349
174	2017-09-20 14:22:13.336342	172.16.2.2	146.20.193.45	TLSv1.2	5062	48520	2802	Server Hello
175	2017-09-20 14:22:13.336358	172.16.2.2	146.20.193.45	TLSv1.2	5062	48520	1426	Certificate

## 2. Wireshark Filtering

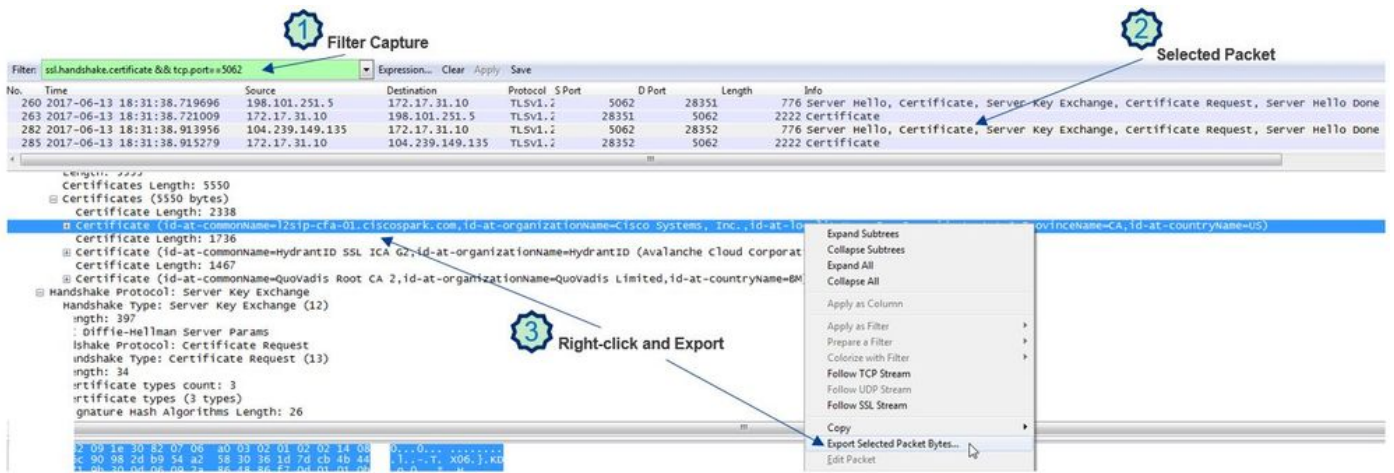
When you analyze packet captures, it's easy to get lost in the sheer amount of packets observed in a given capture. It's important to understand what type of traffic you're most interested in so that you can filter Wireshark to display just that. Here are some common Wireshark filters that can be used to get details about a mutual TLS handshake:

- tcp.port==5062
- ssl && tcp.port==5062
- ssl.handshake.certificate && tcp.port==5062

## 3. Extract Certificate from Pcap

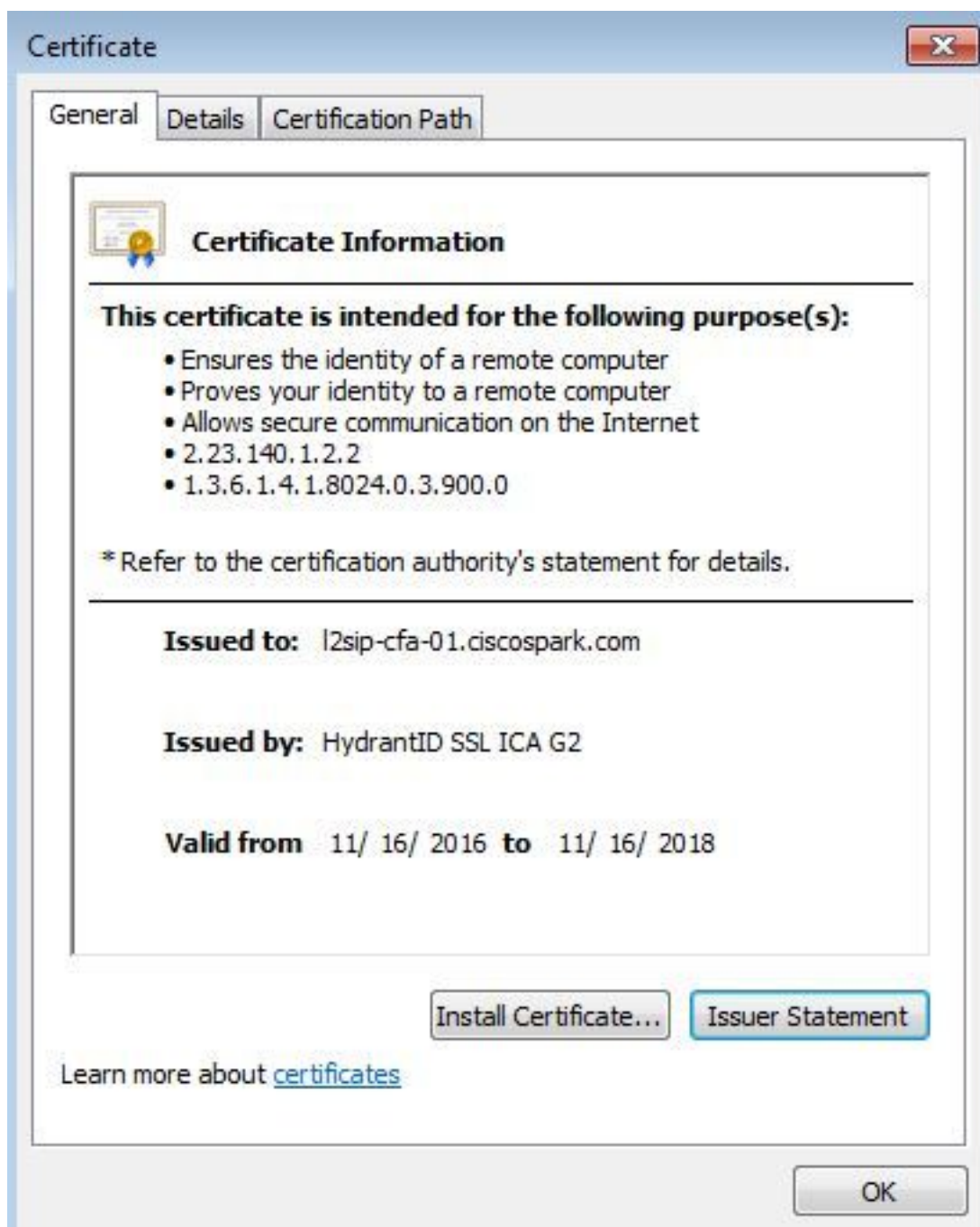
From time to time, you might need to get a copy of a certificate (server, root, or intermediary). If you do not know where to find the certificate you're in search for, you can extract it directly from a packet capture. Here are the steps for how to pull the Cisco Webex certificate that is presented at a mutual TLS handshake.

1. Filter the packet capture with **ssl.handshake.certificate && tcp.port==5062**
2. Locate the packet that is sourced from the Webex server address and has Certificate printed in the Info section.
3. In the packet details expand **Secure Socket Layer > TLS Certificate > Handshake Protocol > Certificates**. **Note:** The bottom/last certificate in the chain is the root CA.
4. Right-click the certificate of interest and select **Export Selected Packet Bytes...** as shown in the image.



5. Save the file as a **.cer**.

6. Double-click the saved file to open the certificate as shown in the image.



## 4. Adjust Expressway Logging Levels

Two logging modules are available on the Expressway which can help you better understand what logic the Expressway performs when you analyze the certificates:

- developer.ssl
- developer.zone.zonemg

By default, these logging modules are set to an INFO level. When set to a DEBUG level, you can begin to see the information about the certificate inspection that happens, along with what zone traffic gets mapped to. Both of these functions are relevant to Hybrid Call Service.

Example of the Expressway-E that conducts a SAN inspection of Cisco Webex's server certificate.

```
2017-09-22T11:11:19.485-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 15:11:19,485"
Module="developer.ssl" Level="INFO" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1974)"
Method="::ttssl_continueHandshake" Thread="0x7f576cbee700": Detail="Handshake in progress"
Reason="want read/write"
2017-09-22T11:11:19.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 15:11:19,564"
Module="developer.ssl" Level="DEBUG" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1960)"
Method="::ttssl_continueHandshake" Thread="0x7f576cbee700": Detail="Handshake succeeded"
2017-09-22T11:11:19.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 15:11:19,564"
Module="developer.ssl" Level="DEBUG" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1629)"
Method="::TTSSL_retrieveCommonName" Thread="0x7f576cbee700": Detail="Found common name in peer
certificate" CommonName="l2sip-cfa-01.ciscospark.com"
2017-09-22T11:11:19.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 15:11:19,564"
Module="developer.ssl" Level="DEBUG" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1654)"
Method="::TTSSL_retrieveAltNames" Thread="0x7f576cbee700": Detail="Found DNS alt-name in peer
certificate" AltName="l2sip-cfa-01.ciscospark.com"
2017-09-22T11:11:19.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 15:11:19,564"
Module="developer.ssl" Level="DEBUG" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1654)"
Method="::TTSSL_retrieveAltNames" Thread="0x7f576cbee700": Detail="Found DNS alt-name in peer
certificate" AltName="l2sip-cfa-01.wbx2.com"
2017-09-22T11:11:19.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 15:11:19,564"
Module="developer.ssl" Level="DEBUG" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1654)"
Method="::TTSSL_retrieveAltNames" Thread="0x7f576cbee700": Detail="Found DNS alt-name in peer
certificate" AltName="l2sip-cfa-01-web.wbx2.com"
2017-09-22T11:11:19.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 15:11:19,564"
Module="developer.ssl" Level="DEBUG" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1654)"
Method="::TTSSL_retrieveAltNames" Thread="0x7f576cbee700": Detail="Found DNS alt-name in peer
certificate" AltName="l2sip-cfa-web.wbx2.com"
2017-09-22T11:11:19.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 15:11:19,564"
Module="developer.ssl" Level="DEBUG" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1654)"
Method="::TTSSL_retrieveAltNames" Thread="0x7f576cbee700": Detail="Found DNS alt-name in peer
certificate" AltName="callservice.ciscospark.com"
2017-09-22T11:11:19.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 15:11:19,564"
Module="developer.ssl" Level="DEBUG" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1654)"
Method="::TTSSL_retrieveAltNames" Thread="0x7f576cbee700": Detail="Found DNS alt-name in peer
certificate" AltName="callservice.call.ciscospark.com"
```

Example of the Expressway-E mapping the MTLS connection to the Cisco Webex Hybrid DNS Zone:

```
2017-09-22T11:11:19.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 15:11:19,564"
Module="developer.zone.zonemgr" Level="DEBUG"
CodeLocation="ppcmains/oak/zones/ZoneManager.cpp(1226)"
Method="ZoneManager::getDNSZoneByTLSVerifySubjectName" Thread="0x7f577f0a0700":
```



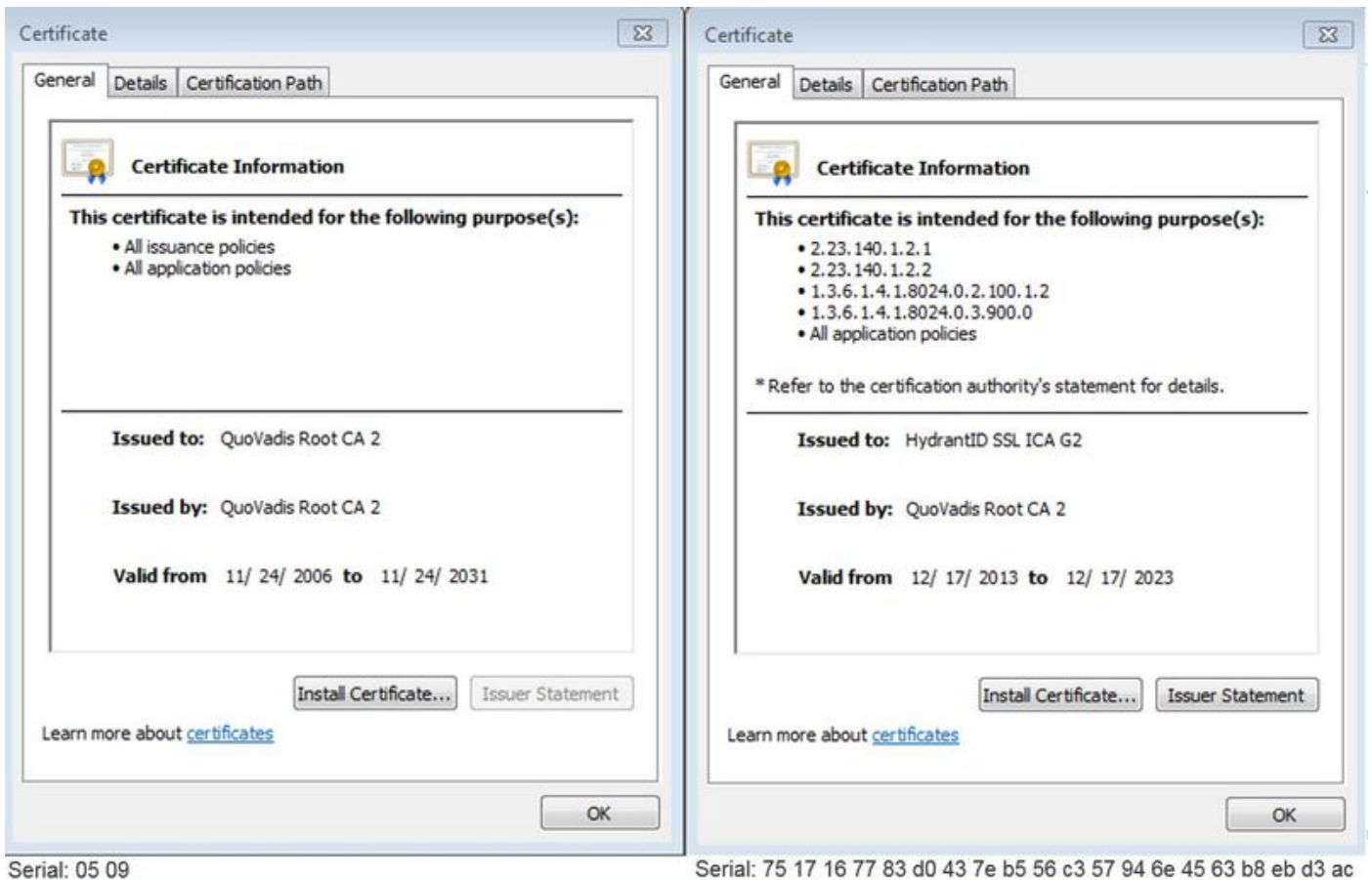
```
this="0x56408ff81220" getDNSZoneByTLSVerifySubjectName classified subject name
callservice.ciscospark.com into DNS zone Hybrid Call Services DNS
2017-09-22T11:11:19.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 15:11:19,564"
Module="developer.zone.zonemgr" Level="DEBUG"
CodeLocation="ppcmains/oak/zones/ZoneManager.cpp(1183)"
Method="ZoneManager::getDNSZoneByTLSVerifySubjectNameList" Thread="0x7f577f0a0700":
this="0x56408ff81220" Detail="Searched for DNS Zones by Subject Name" Found="True"
Candidates="l2sip-cfa-01.ciscospark.coml2sip-cfa-01.ciscospark.coml2sip-cfa-01.wbx2.coml2sip-
cfa-01-web.wbx2.coml2sip-cfa-web.wbx2.comcallservice.ciscospark.com" MatchedZone="Hybrid Call
Services DNS" MatchedIdentity="callservice.ciscospark.com"
2017-09-22T11:11:19.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 15:11:19,564"
Module="developer.zone.zonemgr" Level="DEBUG"
CodeLocation="ppcmains/oak/zones/ZoneManager.cpp(1054)"
Method="ZoneManager::getZoneByIdentities" Thread="0x7f577f0a0700": this="0x56408ff81220"
Detail="getZoneByIdentities, match complete" Identitites="{CN: l2sip-cfa-01.ciscospark.com, Alt-
DNS: l2sip-cfa-01.ciscospark.com, Alt-DNS: l2sip-cfa-01.wbx2.com, Alt-DNS: l2sip-cfa-01-
web.wbx2.com, Alt-DNS: l2sip-cfa-web.wbx2.com, Alt-DNS: callservice.ciscospark.com, Alt-DNS:
callservice.call.ciscospark.com, Alt-DNS: l2sip-a-Webexcall.ciscospark.com, Alt-DNS: l2sip-prod-
11-dfw-public.wbx2.com, Alt-DNS: l2sip-prod-12-dfw-public.wbx2.com, Alt-DNS: l2sip-l2sipproda1-
294-riad-public.wbx2.com, Alt-DNS: l2sip-l2sipproda1-817-riad-public.wbx2.com, Alt-DNS: l2sip-
l2sip-prod-wpsjc-web.ciscospark.com, Alt-DNS: l2sip-l2sip-prod-wpsjc-web.wbx2.com, Alt-DNS:
l2sip-l2sip-prod-wpdfw-web.ciscospark.com, Alt-DNS: l2sip-l2sip-prod-wpdfw-web.wbx2.com, Alt-
DNS: l2sip-cfa-02.wbx2.com, Alt-DNS: Webexcmr-wpa.ciscospark.com, Alt-DNS: Webexcmr-
wpb.ciscospark.com, Alt-DNS: Webexcmr-wpc.ciscospark.com, Alt-DNS: l2sip-wpa-01.wbx2.com, Alt-
DNS: l2sip-wpa-02.wbx2.com, Alt-DNS: l2sip-wpb-01.wbx2.com, Alt-DNS: l2sip-wpb-02.wbx2.com, Alt-
DNS: l2sip-wpc-01.wbx2.com, Alt-DNS: l2sip-wpc-02.wbx2.com}" MatchMechanism="DNSZoneMatch"
MatchedZone="Hybrid Call Services DNS"
```

Here is a list of the most common issues that are related to mutual TLS failures between the Expressway-E and Cisco Webex.

## **Issue 1. Expressway-E does not Trust Certificate Authority (CA) that signed the Cisco Webex Certificate**

The Cisco Webex server that is in direct communication with the Expressway-E is called an L2SIP server. This L2SIP server is to be signed by an intermediary server with a common name of **Hydrant SSL ICA G2**. The intermediary is signed by a root certificate authority that has a common name of **QuoVadis Root CA 2** as shown in the image.

**Note:** This could be subject to change.



The first step to analyze this traffic from the Expressway diagnostic perspective is to search for **TCP Connecting**. After you search **TCP Connecting**, you'll look for the **Dst-port=5062** value. Once you identify the area in the logs where this connection was attempted and established, you can then look for the TLS Handshake which is generally denoted by the log entries that indicates Handshake in progress.

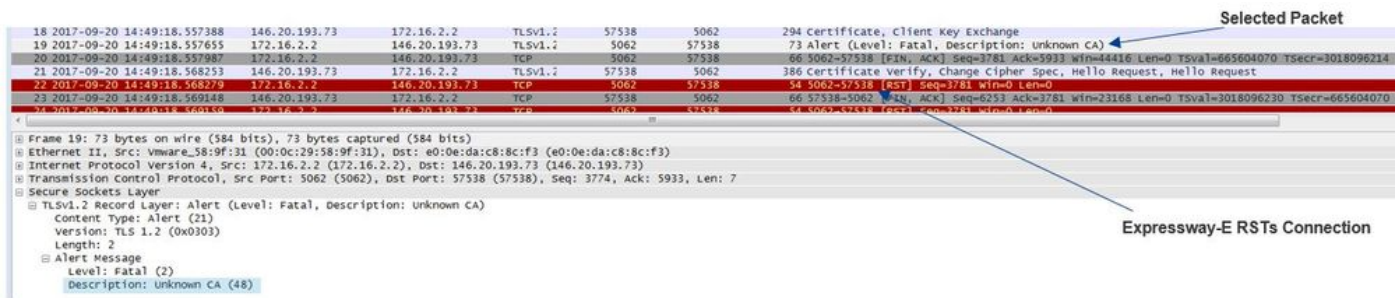
```
2017-09-20T10:49:18.427-04:00 amer-expressway01 tvcs: UTCTime="2017-09-20 14:49:18,426"
Module="developer.ssl" Level="INFO" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1974)"
Method=":ttssl_continueHandshake" Thread="0x7f29ddefa700": Detail="Handshake in progress"
Reason="want read/write"
```

If the Expressway-E does not trust the Cisco Webex signed certificates, you can expect that the Expressway-E can reject the certificate immediately after the handshake completes. This can be spotted in the Expressway-E logging by these log entries:

```
2017-09-20T10:49:18.724-04:00 amer-expressway01 tvcs: Event="Inbound TLS Negotiation Error"
Service="SIP" Src-ip="146.20.193.73" Src-port="58531" Dst-ip="172.16.2.2" Dst-port="5062"
Detail="self signed certificate in certificate chain" Protocol="TLS" Level="1" UTCTime="2017-09-20 14:49:18,724"
2017-09-20T10:49:18.724-04:00 amer-expressway01 tvcs: UTCTime="2017-09-20 14:49:18,724"
Module="developer.ssl" Level="ERROR" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(68)"
Method=":TTSSLErrorOutput" Thread="0x7f29ddefa700": TTSSL_continueHandshake: Failed to
establish SSL connection iResult="-1" error="1" bServer="true"
localAddress="['IPv4','TCP','172.16.2.2:5062']" remoteAddress="['IPv4','TCP','146.20.193.73:58531']"
ssl_error_reason="error:14089086:SSL routines:ssl3_get_client_certificate:certificate verify failed"
2017-09-20T10:49:18.724-04:00 amer-expressway01 tvcs: UTCTime="2017-09-20 14:49:18,724"
Module="network.tcp" Level="DEBUG": Src-ip="146.20.193.73" Src-port="58531" Dst-ip="172.16.2.2"
Dst-port="5062" Detail="TCP Connection Closed" Reason="self signed certificate in certificate chain"
```

The Expressway error message can slightly mislead because it refers to a self-signed certificate in

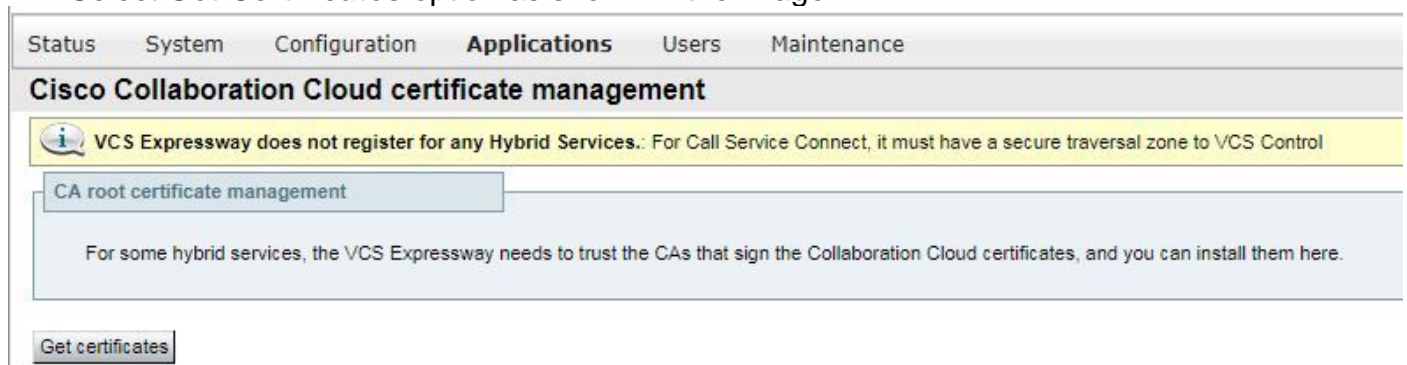
the certificate chain. Wireshark allows you to take a closer look at the exchange. From a Wireshark packet capture analysis perspective, you can clearly see that when the Webex environment presents its certificate then Expressway turns around and rejects with a certificate with an Unknown CA error as shown in the image.



## Solution:

In order to resolve this situation, you must ensure that the Expressway-E trusts the Cisco Webex certificate authorities. While you could simply extract these certificates from a Wireshark trace and upload them to the Trusted CA certificate store on the Expressway, the Expressway offers a simpler method:

- Log into the Expressway-E
- Navigate to **Applications > Cloud Certificate management**
- Select **Get Certificates** option as shown in the image.



At this point, the Cisco Webex certificate authorities are uploaded to the Expressway-E Trusted CA store (**Maintenance > Security > Trusted CA certificate**).

## Issue 2. Incorrect Name for TLS Subject Verify Name on Expressway-E Cisco Webex Hybrid DNS Zone

As part of the mutual TLS handshake, Hybrid Call Service Connect uses TLS Verification. This means that in addition to trusting the Cisco Webex CA certificates, the Expressway verifies the certificate by checking the Subject Alternate Name (SAN) field of the certificate that is presented to ensure it has a value such as **callservice.ciscopark.com** present. If this value is not present, the inbound call fails.

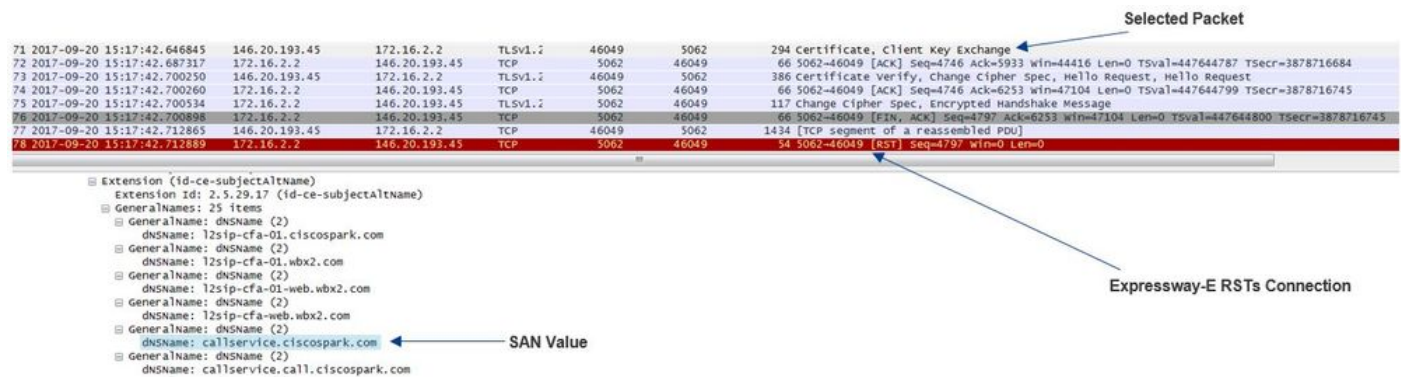
In this particular scenario, the Cisco Webex server presents its certificate to the Expressway-E. The certificate actually has 25 different SANs. Consider the case where the Expressway-E checks the certificate for the callservice.ciscopark.com SAN but doesn't find that. When this condition is met, you can see an error similar to this within the diagnostic logging:

```

2017-09-20T11:17:42.701-04:00 amer-expressway01 tvcs: Event="Inbound TLS Negotiation Error"
Service="SIP" Src-ip="146.20.193.45" Src-port="46049" Dst-ip="172.16.2.2" Dst-port="5062"
Detail="Peer's TLS certificate identity was unacceptable" Protocol="TLS" Level="1"
UTCTime="2017-09-20 15:17:42,700"
2017-09-20T11:17:42.701-04:00 amer-expressway01 tvcs: UTCTime="2017-09-20 15:17:42,700"
Module="network.tcp" Level="DEBUG": Src-ip="146.20.193.45" Src-port="46049" Dst-ip="172.16.2.2"
Dst-port="5062" Detail="TCP Connection Closed" Reason="Peer's TLS certificate identity was
unacceptable"

```

If you use Wireshark to analyze this certificate handshake, you can find that after Cisco Webex presents its certificate, the Expressway RSTs the connection shortly after as shown in the image.



In order to confirm the configuration of this value, you can go to the Webex Hybrid DNS Zone that was configured for the solution. If you have the Expressway-E xConfiguration, you can look for the Zone configuration section to determine how the TLS verify subject name was configured. For xConfiguration, note that the zones are ordered with Zone 1 being the first. Here is an xConfiguration from the problematic environment analyzed above.

```

*c xConfiguration Zones Zone 6 DNS SIP TLS Verify Mode: "On"
*c xConfiguration Zones Zone 6 DNS SIP TLS Verify Subject Name: "call1service.ciscospark.com"

```

As you can see in the example the TLS Verify Subject Name is set to **call1service.ciscospark.com** instead of **callservice.ciscospark.com**. (note the extra "1").

Solution:

In order to resolve this issue, the TLS Verify Subject Name must be modified:

- Log into the Expressway-E
- Navigate to **Configuration > Zones > Zones**
- Select **Webex Hybrid Services DNS Zone**
- Set the **TLS verify subject name** to **callservice.ciscospark.com**
- Select **Save**

**Note:** See thefor baseline logging behavior. This section shows the Expressway performing certificate verification and the mapping to the Webex Hybrid DNS Zone.

**Note:** As of Expressway code x12.5 and later a new "Webex" zone has been released. This Webex zone prepopulates the configuration of the zone required for communication out to Webex. This means you no longer have to set the TLS Subject Verify Mode and TLS Verify Subject Name. For configuration simplification it's recommended to leverage the Webex zone if you are running x12.5 or later of Expressway code.

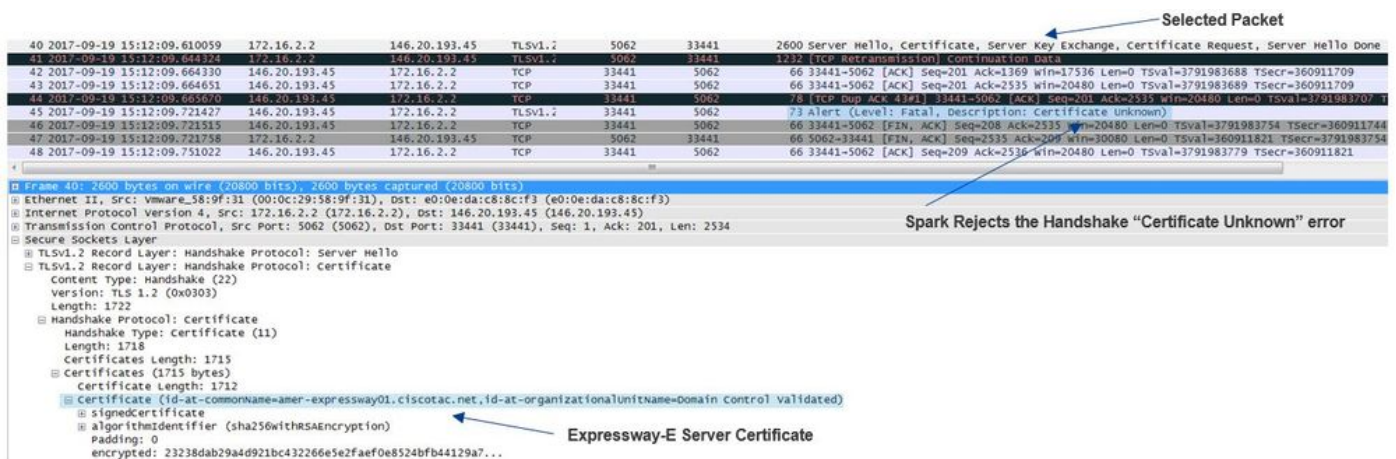
### Issue 3. Expressway-E does not Send Full Certificate Chain to Cisco Webex

As part of the mutual TLS handshake, Cisco Webex must trust the Expressway-E certificate. Cisco Webex has a full list of public CAs that it trusts. Typically, a TLS handshake is successful when your Expressway-E certificate is signed by a public CA that Cisco Webex supports. By design, the Expressway-E only sends its certificate during a TLS handshake despite being signed by a public CA. In order to send the full chain of certificates (root and intermediate), those certificates must be added to the Trusted CA certificate store on the Expressway-E itself.

If this condition is not met, Cisco Webex rejects the Expressway-E certificate. When you troubleshoot a condition that matches this problem, you can use the diagnostic logs and tcpdump from the Expressway-E. When you analyze the Expressway-E diagnostic logs, you'll see an error similar to that here:

```
2017-09-19T11:12:09.721-04:00 amer-expressway01 tvcs: Event="Inbound TLS Negotiation Error"
Service="SIP" Src-ip="146.20.193.45" Src-port="33441" Dst-ip="172.16.2.2" Dst-port="5062"
Detail="sslv3 alert certificate unknown" Protocol="TLS" Level="1" UTCTime="2017-09-19
15:12:09,721"
2017-09-19T11:12:09.721-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 15:12:09,721"
Module="developer.ssl" Level="ERROR" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(68)"
Method="::TTSSLErrorOutput" Thread="0x7fc67c6ec700": TTSSL_continueHandshake: Failed to
establish SSL connection iResult="0" error="1" bServer="true"
localAddress="[ 'IPv4' 'TCP' '172.16.2.2:5062' ]" remoteAddress="[ 'IPv4' 'TCP' '146.20.193.45:33441' ]"
ssl_error_reason="error:14094416:SSL routines:ssl3_read_bytes:sslv3 alert certificate unknown"
2017-09-19T11:12:09.721-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 15:12:09,721"
Module="network.tcp" Level="DEBUG": Src-ip="146.20.193.45" Src-port="33441" Dst-ip="172.16.2.2"
Dst-port="5062" Detail="TCP Connection Closed" Reason="Got EOF on socket"
```

If you analyze this from a Wireshark perspective, you see that the Expressway-E presents its certificate. If you expand the packet, you can see that only the server certificate is sent. Cisco Webex then rejects this TLS handshake with an Unknown CA error message as shown in the image.



#### Solution:

In order to address the issue in this scenario, you must upload the intermediate and root CAs that are involved in the signing of the Expressway-E certificate to the Trusted CA certificate store:

- Step 1. Log into the Expressway-E.
- Step 2. Navigate to **Maintenance > Security > Trusted CA certificate**.
- Step 3. Select **Choose File** under the Upload menu near the bottom of the UI.
- Step 4. Choose the CA certificate that was involved in the signing of the Expressway-E.

Step 5. Select **Append CA Certificate**.

Step 6. Repeat steps for all CA certificates involved in the signing of the Expressway-E certificate (Intermediate, Root).

Step 7. Select **Append CA Certificate**.

Once this process is completed, you see that the full chain of certificates involved in signing the Expressway-E server certificate included in the key exchange. Here is a sample of what you would see if you analyzing a packet capture with Wireshark.

The image shows a Wireshark packet capture of a TLS handshake. The selected packet (175) is a Certificate. The details pane shows the certificate chain: Server Certificate, Intermediate Certificate, and Root Certificate.

```
175 2017-09-20 14:22:13.336358 172.16.2.2 146.20.193.45 TLSv1.2 5062 48520 1426 Certificate
176 2017-09-20 14:22:13.351899 146.20.193.45 172.16.2.2 TCP 48520 5062 66 48520-5062 [ACK] Seq=201 Ack=1369 win=17536 Len=0 TSval=3875387398 TSecr=444315436
177 2017-09-20 14:22:13.354815 146.20.193.45 172.16.2.2 TCP 48520 5062 66 48520-5062 [ACK] Seq=201 Ack=2737 win=20480 Len=0 TSval=3875387399 TSecr=444315436
178 2017-09-20 14:22:13.355985 146.20.193.45 172.16.2.2 TCP 48520 5062 66 48520-5062 [ACK] Seq=201 Ack=4097 win=23296 Len=0 TSval=3875387400 TSecr=444315436
179 2017-09-20 14:22:13.355999 172.16.2.2 146.20.193.45 TLSv1.2 5062 48520 715 Server Key Exchange
180 2017-09-20 14:22:13.366930 146.20.193.45 172.16.2.2 TCP 48520 5062 66 48520-5062 [ACK] Seq=201 Ack=4746 win=26112 Len=0 TSval=3875387411 TSecr=444315455
197 2017-09-20 14:22:13.668592 146.20.193.45 172.16.2.2 TLSv1.2 48520 5062 73 Alert (Level: Fatal, Description: Certificate unknown)
198 2017-09-20 14:22:13.668644 146.20.193.45 172.16.2.2 TCP 48520 5062 66 48520-5062 [FIN, ACK] Seq=209 Ack=4746 win=0 Len=0 TSval=3875387711 TSecr=444315455
199 2017-09-20 14:22:13.668871 172.16.2.2 146.20.193.45 TCP 5062 48520 66 5062-48520 [FIN, ACK] Seq=4746 Ack=209 win=30080 Len=0 TSval=444315768 TSecr=3875387711
200 2017-09-20 14:22:13.681586 146.20.193.45 172.16.2.2 TCP 48520 5062 66 48520-5062 [ACK] Seq=209 Ack=4747 win=26112 Len=0 TSval=3875387725 TSecr=444315768
```

Frame 175: 1426 bytes on wire (11408 bits), 1426 bytes captured (11408 bits) on interface 0  
Ethernet II, Src: Vmware\_58:9f:31 (00:0c:29:58:9f:31), Dst: e0:0e:da:c8:8c:f3 (e0:0e:da:c8:8c:f3)  
Internet Protocol Version 4, Src: 172.16.2.2 (172.16.2.2), Dst: 146.20.193.45 (146.20.193.45)  
Transmission Control Protocol, Src Port: 5062 (5062), Dst Port: 48520 (48520), Seq: 2737, Ack: 201, Len: 1360  
[2 Reassembled TCP Segments (3938 bytes): #174(2642), #175(1296)]  
Secure Sockets Layer  
 TLSv1.2 Record Layer: Handshake Protocol: Certificate  
 Content Type: Handshake (22)  
 Version: TLS 1.2 (0x0303)  
 Length: 3933  
 Handshake Protocol: Certificate  
 Handshake Type: certificate (11)  
 Length: 3929  
 Certificates Length: 3926  
 Certificates (3926 bytes)  
 Certificate Length: 1712  
 Certificate (id-at-commonName=amer-expressway01.ciscotac.net,id-at-organizationalunitName=Domain control validated)  
 Certificate Length: 1236  
 Certificate (id-at-commonName=Go Daddy Secure Certificate Authority - G2,id-at-organizationalunitName=http://certs.godaddy.com/repository,id-at-organizationName=GoDaddy.com, Inc.,id-at-localityName=Scottsdale,Arizona,US)  
 Certificate Length: 969  
 Certificate (id-at-commonName=Go Daddy Root Certificate Authority - G2,id-at-organizationName=GoDaddy.com, Inc.,id-at-localityName=Scottsdale,Arizona,US)

## Issue 4. Firewall Terminates Mutual TLS Handshake

The Expressway solution typically interfaces with a firewall. Many times, the inline firewall for the solution is runs some type of application layer inspection. Often with the Expressway solution, when the firewall runs application layer inspection, administrators see undesirable results. This particular issue helps you identify when a firewall's application layer inspection abruptly tore down the connection.

With the use of the diagnostic logs from the Expressway, you can look for the attempted Mutual TLS handshake. This handshake, as mentioned earlier, should come shortly after the TCP Connection is established over port 5062. In this scenario, when the firewall tears down the connection, you see these errors within the diagnostic logging.

```
Thread="0x7f6496669700": TTSSL_continueHandshake: Failed to establish SSL connection iResult="-1" error="5" bServer="false" localAddress="['IPv4 'TCP' '172.17.31.10:28351']"  
2017-06-13T13:31:38.760-05:00 vcse tvcs: Event="Outbound TLS Negotiation Error" Service="SIP"  
Src-ip="172.17.31.10" Src-port="28351" Dst-ip="198.101.251.5" Dst-port="5062" Detail="No SSL error available, probably remote disconnect" Protocol="TLS" Common-name="callservice.ciscopark.com" Level="1" UTCTime="2017-06-13 18:31:38,758"  
2017-06-13T13:31:38.760-05:00 vcse tvcs: UTCTime="2017-06-13 18:31:38,758" Module="network.tcp" Level="DEBUG": Src-ip="172.17.31.10" Src-port="28351" Dst-ip="198.101.251.5" Dst-port="5062" Detail="TCP Connection Closed" Reason="Got EOF on socket"
```

From a packet capture perspective, you'll see that the Expressway-E presents its certificate to Cisco Webex. You see a TCP RST come in from the direction of Cisco Webex as shown in the image.

The image shows a Wireshark capture of a TLS handshake. Packet 266 is selected, showing a Certificate packet. The certificate chain includes Expressway-E, Go Daddy, and a Public CA. An arrow points to the selected packet with the label "Selected Packet". Another arrow points to the packet details with the label "Unexpected RST with no error code".

At first glance, you may think something is wrong with the Expressway-E certificate. In order to troubleshoot this issue, you first have to determine answers to these questions:

- Is the Expressway-E signed by a Public CA that Cisco Webex trusts?
- Is the Expressway-E certificate and any certificates involved in the signing of the Expressway-E certificate manually uploaded to the Cisco Webex Control Hub (<https://admin.ciscospark.com>)?

In this particular condition, the solution was not to use the Cisco Webex Control Hub to manage the Expressway-E certificates. This means the Expressway-E certificate must be signed by a public CA that Cisco Webex trusts. By selecting on the Certificate packet in the Wireshark capture (as illustrated above), you can see that the certificate was signed by a Public CA and that the full chain was sent to Cisco Webex. Therefore, the issue should not be related to the Expressway-E certificate.

At this point, if further isolation is required, you could take a packet capture off the outside interface of the firewall. However, the lack of SSL error in the diagnostic log is an important data point. If you recall above (Issue 3.), if Cisco Webex doesn't trust the Expressway-E certificate, you must see some type of SSL disconnect reason. In this condition, there was no SSL error available.

**Note:** If you were to get a packet capture off the firewall outside interface you would not see a TCP RST coming in from the Cisco Webex environment.

## Solution

For this particular solution, you as a partner or customer must rely on your security team. The team must investigate if they use any sort of application layer inspection for the Expressway solution and if they are, this should be disabled. [Appendix 4](#) of the **VCS Control and Expressway Deployment Guide** explains why it is recommended customers turn off this functionality.

## Issue 5. Expressway-E is Signed by Public CA but Cisco Webex Control Hub has Alternate Certificates Loaded

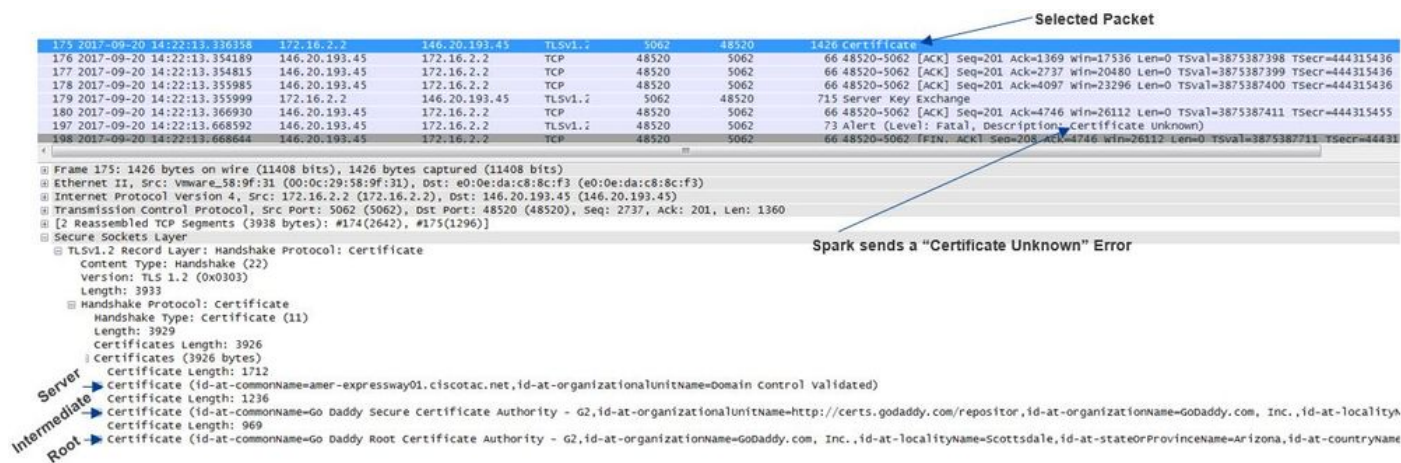
This particular condition can often occur when you deployed the Expressway solution from scratch and you do not have the Expressway-E certificate signed by a public CA initially. What happens in this scenario is that you upload the Expressway-E server certificate (which has been signed internally) to the Cisco Webex Control Hub so that you can complete the mutual TLS negotiation successfully. Afterwards, you end up getting the Expressway-E certificate signed by a Public CA, however you forget to remove the server certificate from the Cisco Webex Control Hub. It's

important to know that when a certificate is uploaded to the Cisco Webex Control Hub, that certificate takes priority over what certificate and chain the Expressway presents during the TLS handshake.

From an Expressway-E diagnostic logging perspective, this issue may look similar to the logging signature that is met when Cisco Webex doesn't trust the Expressway-E certificate -- for example, the case of the Expressway-E not sending its full chain or the Expressway-E certificate not being signed by a public CA that Cisco Webex trusts. Below is a sample of what you can expect in the Expressway-E logging during the TLS handshake:

```
2017-09-20T10:22:13.669-04:00 amer-expressway01 tvcs: Event="Inbound TLS Negotiation Error"
Service="SIP" Src-ip="146.20.193.45" Src-port="48520" Dst-ip="172.16.2.2" Dst-port="5062"
Detail="sslv3 alert certificate unknown" Protocol="TLS" Level="1" UTCTime="2017-09-20
14:22:13,668"
2017-09-20T10:22:13.669-04:00 amer-expressway01 tvcs: UTCTime="2017-09-20 14:22:13,668"
Module="developer.ssl" Level="ERROR" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(68)"
Method="::TTSSLErrorOutput" Thread="0x7f4a2c16f700": TTSSL_continueHandshake: Failed to
establish SSL connection iResult="0" error="1" bServer="true"
localAddress="['IPv4' 'TCP' '172.16.2.2:5062']" remoteAddress="['IPv4' 'TCP' '146.20.193.45:48520']"
ssl_error_reason="error:14094416:SSL routines:ssl3_read_bytes:sslv3 alert certificate unknown"
2017-09-20T10:22:13.669-04:00 amer-expressway01 tvcs: UTCTime="2017-09-20 14:22:13,668"
Module="network.tcp" Level="DEBUG": Src-ip="146.20.193.45" Src-port="48520" Dst-ip="172.16.2.2"
Dst-port="5062" Detail="TCP Connection Closed" Reason="Got EOF on socket"
```

Take a look at this from the Wireshark perspective you can see here that the Expressway-E presents its certificate in line item 175. A few line items later, the Cisco Webex environment rejects the certificate with a Certificate Unknown error as shown in the image.



If you select the Certificate packet that the Expressway-E sends, you can expand the certificate information to determine if the Expressway-E

1. is signed by a [Public CA that Cisco Webex trusts](#), and
2. is including its full chain involved in the signing.

In this situation, both of these conditions are met. This suggests there is nothing wrong with the Expressway-E certificate.

### Solution

Step 1. Log into the [Cisco Webex Control Hub](#).



Step 2. Select **Services** from the left pane.

Step 3. Choose **Settings** under the Hybrid Call card.

Step 4. Scroll to the Call Service Connect section and look under the Certificates for Encrypted SIP Calls to see if undesired certificates are listed. If so, click the trash can icon next to the certificate.

step 5. Select **Remove**.

**Note:** It is important that the analysis is conducted and it is determined that the customer is not using the certificates uploaded to the Webex Control Hub prior to removing them.

For more information about uploading your Expressway-E certificate in the Cisco Webex Control Hub, check [this section of the Hybrid Call Deployment Guide](#).

## Issue 6. Expressway is not Mapping Inbound Call to Cisco Webex Hybrid DNS Zone

The Inbound TLS mapping feature works in conjunction with the TLS Verify Subject Name, both of which are configured on the Hybrid Call DNS Zone. This scenario articulates issues and challenges observed with the Expressway prior to x12.5. In x12 and later a new zone type was implemented called the "Webex" zone. This zone pre-populates all the required configuration for the integration with Webex. If you're running x12.5 and deploying Webex Hybrid Call it's recommend to use the **Webex** Zone type so that the Hybrid Call Services Domain (callservice.webex.com) is auto configured for you. This value matches the Subject Alternate Name of the Webex certificate that is presented during the Mutual TLS handshake and allows the connection and inbound mapping to the Expressway to succeed.

If you're using any code version below x12.5 or are not using the Webex zone you'll want to proceed with the explanation below that demonstrates how to identify and correct issues where the Expressway is not mapping the inbound call to the Webex Hybrid DNS Zone.

The feature breaks down into a three step process:

1. Expressway-E accepts the Cisco Webex certificate.
2. Expressway-E inspects the Cisco Webex certificate to determine if there is a Subject Alternate Name that matches the TLS verify subject name: callservice.ciscopark.com.
3. Expressway-E maps the inbound connection through the Cisco Webex Hybrid DNS Zone.

If authentication is not successful, this means that the certificate validation failed. The call enters into the Default Zone and is routed according to the search rules provided for business-to-business scenarios, if business-to-business is configured on Expressway-E.

Like the other scenarios, you must use both the diagnostic logging and packet captures to determine what this failure looks like, then use the packet capture to see which side is sending the RST. Here is a sample of the TCP Connection being attempted, then establishing.

```
2017-09-22T10:09:56.471-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 14:09:56,471"  
Module="network.tcp" Level="DEBUG": Src-ip="148.62.40.52" Src-port="44205" Dst-ip="172.16.2.2"  
Dst-port="5062" Detail="TCP Connecting"  
2017-09-22T10:09:56.471-04:00 amer-expressway01 tvcs: UTCTime="2017-09-22 14:09:56,471"  
Module="network.tcp" Level="DEBUG": Src-ip="148.62.40.52" Src-port="44205" Dst-ip="172.16.2.2"
```

Dst-port="5062" Detail="TCP Connection Established"

Now that the TCP connection has established, the TLS Handshake can ensue. You can see shortly after the handshake starts, it quickly errors out.

```
2017-09-22T10:09:57.044-04:00 amer-expressway01 tvcs: UTCtime="2017-09-22 14:09:57,044"
Module="developer.ssl" Level="INFO" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1974)"
Method=":ttssl_continueHandshake" Thread="0x7f044e7cc700": Detail="Handshake in progress"
Reason="want read/write"
2017-09-22T10:09:57.123-04:00 amer-expressway01 tvcs: Event="Inbound TLS Negotiation Error"
Service="SIP" Src-ip="148.62.40.52" Src-port="44205" Dst-ip="172.16.2.2" Dst-port="5062"
Detail="Peer's TLS certificate identity was unacceptable" Protocol="TLS" Level="1"
UTCtime="2017-09-22 14:09:57,123"
2017-09-22T10:09:57.123-04:00 amer-expressway01 tvcs: UTCtime="2017-09-22 14:09:57,123"
Module="network.tcp" Level="DEBUG": Src-ip="148.62.40.52" Src-port="44205" Dst-ip="172.16.2.2"
Dst-port="5062" Detail="TCP Connection Closed" Reason="Peer's TLS certificate identity was unacceptable"
```

Look at this situation from a pcap perspective, you can get a better sense of

- who is sending the RST, and
- what certificates are being passed to determine if they are correct.

When you analyze this particular capture, you can see that the Expressway-E sends the RST. When you look at the Cisco Webex certificate that is passed, you can see that it sends the full chain. Additionally, you can conclude that based off the error message in the diagnostic log, you can rule out the scenario where the Expressway-E doesn't trust the Cisco Webex Public CAs. Otherwise, you would see an error like "self signed certificate in certificate chain". You can dig into the packet details as shown in the image.

The image shows a Wireshark capture of a network session. The packet list pane at the top shows several packets, with packet 70 highlighted in red, indicating a Reset (RST). The details pane for packet 70 shows the following structure:

- Frame 62: 294 bytes on wire (2352 bits), 294 bytes captured (2352 bits)
- Ethernet II, Src: e0:0e:da:c8:8c:f3 (e0:0e:da:c8:8c:f3), Dst: vmware\_58:9f:31 (00:0c:29:58:9f:31)
- Internet Protocol Version 4, Src: 148.62.40.52 (148.62.40.52), Dst: 172.16.2.2 (172.16.2.2)
- Transmission Control Protocol, Src Port: 44205 (44205), Dst Port: 5062 (5062), Seq: 5673, Ack: 4746, Len: 228
- [3] Reassembled TCP segments (5700 bytes): #54(1368), #56(1368), #59(1368), #60(1368), #62(228)
- Secure Sockets Layer
  - TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    - Content Type: Handshake (22)
    - Version: TLS 1.2 (0x0303)
    - Length: 5695
    - Handshake Protocol: Certificate
      - Handshake Type: Certificate (11)
      - Length: 5553
      - Certificates Length: 5550
      - Certificates (5550 bytes)
        - Certificate Length: 2338
          - Certificate (id-at-commonName=l2sip-cfa-01.ciscospark.com,id-at-organizationName=Cisco Systems, Inc.,id-at-localityName=San Jose,id-at-stateOrProvinceName=CA,id-at-countryName=US)
        - Certificate Length: 1736
          - Certificate (id-at-commonName=HydrantID SSL ICA G2,id-at-organizationName=HydrantID (Avalanche Cloud Corporation),id-at-countryName=US)
        - Certificate Length: 1467
          - Certificate (id-at-commonName=Quovadis Root CA 2,id-at-organizationName=Quovadis Limited,id-at-countryName=BM)

By click the Webex server certificate and expanding it to see the Subject Alternate Names (dnsName) you can verify to ensure it has **callservice.ciscospark.com** listed.

Navigate to **Wireshark: Certificate > Extension > General Names > General Name > dnsName: callservice.ciscospark.com**

This fully confirms that the Webex certificate looks just fine.

You can now confirm that the TLS Verify Subject Name is correct. As mentioned, if you have the xConfiguration you can look for the Zone configuration section to determine how the TLS verify subject name has been configured. One thing to note about the xConfiguration is that the zones are ordered with Zone 1 is the first created. Here is an xConfiguration from the problematic

environment analyzed above. It's clear that nothing is wrong with the TLS Verify Subject Name.

```
*c xConfiguration Zones Zone 6 DNS SIP TLS Verify Mode: "On"
```

```
*c xConfiguration Zones Zone 6 DNS SIP TLS Verify Subject Name: "callservice.ciscospark.com"
```

The next thing that must be investigated is the **TLS verify inbound mapping**. This confirms if you are correctly mapping the TLS connection to the Webex Hybrid DNS Zone. The xConfiguration can be leveraged to analyze this as well. In the xConfiguration the **TLS verify inbound mapping** is called **DNS ZIP TLS Verify InboundClassification**. As you can see in this example the value is set to Off.

```
*c xConfiguration Zones Zone 6 DNS SIP TLS Verify InboundClassification: "Off"
```

```
*c xConfiguration Zones Zone 6 Name: "Hybrid Call Services DNS"
```

Given that this value is set to Off, what this means is that the VCS is prevented from attempting to map inbound TLS connections to this zone. The call thus enters into the Default Zone and is checked and routed according to the search rules provided for business-to-business scenarios, if business-to-business is configured on Expressway-E..

## Solution

In order to address this you need to set the TLS verify inbound mapping on the Hybrid Call DNS Zone to On. Here are the steps to complete that.

1. Log into the Expressway-E
2. Navigate to **Configuration > Zones > Zones**
3. Select **Hybrid Call DNS Zone**
4. For the **TLS verify inbound mapping**, choose **On**
5. Select **Save**

**Note:** See the for baseline logging behavior. This section shows the Expressway performing certificate verification and the mapping to the Webex Hybrid DNS Zone.

## Issue 7. Expressway-E uses Default Self-Signed Certificate

In some new deployments of Hybrid Call Service Connect, the signing of the Expressway-E certificate is overlooked or it's believed that the default server certificate can be used. Some people think that this is possible because the Cisco Webex Control Hub lets you load a custom certificate into the portal. (**Services > Settings (Under Hybrid Call card) > Upload (Under Certificates for Encrypted Calls)**)

If you pay close attention to the wording about the **Certificates for Encrypted SIP Calls**, you see this: 'Use certificates provided from the Cisco Collaboration default trust list or upload your own. If you use your own, ensure the hostnames are on a verified domain.' The key piece to that statement is "**make sure hostnames are on a verified domain.**"

When you troubleshoot an issue that matches this condition, keep in mind that the symptom is going to be dependent on the direction of the call. If the call originated by an on-premises phone, you can expect that the Cisco Webex app would not ring. Also, if you tried to trace the call from the Expressways Search History, you'd find that the call would make it to the Expressway-E and stop there. If the call originated from a Cisco Webex app and was destined for the premises, the on-premises phone does not ring. In that instance, the Expressway-E and Expressway-C Search

History would not show anything.

In this particular scenario, the call originated from an on-premises phone. Using the Expressway-E Search History, you can determine that the call made it to the server. At this point, you can dive into the diagnostic logging to determine what happened. To start this analysis, first look to see if a TCP Connection was attempted and established over port 5062. By searching the Expressway-E diagnostic logs for "TCP Connecting" and searching the line item with the tag "Dst-port=5062", you can determine if the connection establishes.

```
2017-09-26T08:18:08.428-04:00 amer-expressway01 tvcs: UTCTime="2017-09-26 12:18:08,426"
Module="network.tcp" Level="DEBUG": Src-ip="146.20.193.45" Src-port="59720" Dst-ip="172.16.2.2"
Dst-port="5062" Detail="TCP Connecting"
```

```
2017-09-26T08:18:08.428-04:00 amer-expressway01 tvcs: UTCTime="2017-09-26 12:18:08,426"
Module="network.tcp" Level="DEBUG": Src-ip="146.20.193.45" Src-port="59720" Dst-ip="172.16.2.2"
Dst-port="5062" Detail="TCP Connection Established"
```

Now that you confirmed the TCP Connection established, you can analyze the mutual TLS handshake that happens immediately after. As you can see in the snippet here, the handshake fails and the certificate is unknown (**Detail="sslv3 alert certificate unknown"**)

```
2017-09-26T08:18:08.441-04:00 amer-expressway01 tvcs: UTCTime="2017-09-26 12:18:08,441"
Module="developer.ssl" Level="INFO" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1974)"
Method="::ttssl_continueHandshake" Thread="0x7f930adab700": Detail="Handshake in progress"
Reason="want read/write"
```

```
2017-09-26T08:18:08.455-04:00 amer-expressway01 tvcs: Event="Inbound TLS Negotiation Error"
Service="SIP" Src-ip="146.20.193.45" Src-port="59720" Dst-ip="172.16.2.2" Dst-port="5062"
Detail="sslv3 alert certificate unknown" Protocol="TLS" Level="1" UTCTime="2017-09-26
12:18:08,455"
```

```
2017-09-26T08:18:08.455-04:00 amer-expressway01 tvcs: UTCTime="2017-09-26 12:18:08,455"
Module="developer.ssl" Level="DEBUG" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(1997)"
Method="::ttssl_continueHandshake" Thread="0x7f930adab700": Detail="Handshake Failed"
Reason="want error ssl"
```

```
2017-09-26T08:18:08.455-04:00 amer-expressway01 tvcs: UTCTime="2017-09-26 12:18:08,455"
Module="developer.ssl" Level="ERROR" CodeLocation="ppcmains/ssl/ttssl/ttssl_openssl.cpp(68)"
Method="::TTSSL_ErrorOutput" Thread="0x7f930adab700": TTSSL_continueHandshake: Failed to
establish SSL connection iResult="0" error="1" bServer="true"
localAddress="[ 'IPv4' 'TCP' '172.16.2.2:5062' ]" remoteAddress="[ 'IPv4' 'TCP' '146.20.193.45:59720' ]"
ssl_error_reason="error:14094416:SSL routines:ssl3_read_bytes:sslv3 alert certificate unknown"
```

```
2017-09-26T08:18:08.455-04:00 amer-expressway01 tvcs: UTCTime="2017-09-26 12:18:08,455"
Module="network.tcp" Level="DEBUG": Src-ip="146.20.193.45" Src-port="59720" Dst-ip="172.16.2.2"
Dst-port="5062" Detail="TCP Connection Closed" Reason="Got EOF on socket"
```

Take a closer look at the packet capture provided with the Expressway-E diagnostic logging, you can see that the Certificate Unknown error is getting sourced from the direction of Cisco Webex as shown in the image.

No.	Time	Source	Destination	Protocol	S Port	D Port	Length	Info
3	2017-09-26 12:18:08.415918	146.20.193.45	172.16.2.2	TCP	59720	5062	74	59720->5062 [SYN] Seq=0 win=14600 Len=0 MSS=1380 SACK_PERM=1 TSval=91375166 TSecr=0 W
4	2017-09-26 12:18:08.415941	172.16.2.2	146.20.193.45	TCP	5062	59720	74	5062->59720 [SYN, ACK] Seq=0 Ack=1 win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=9552705
5	2017-09-26 12:18:08.426317	146.20.193.45	172.16.2.2	TCP	59720	5062	66	59720->5062 [ACK] Seq=1 Ack=1 win=14720 Len=0 TSval=91375177 TSecr=955270515
6	2017-09-26 12:18:08.427715	146.20.193.45	172.16.2.2	TLSv1.2	59720	5062	266	client Hello
7	2017-09-26 12:18:08.427728	172.16.2.2	146.20.193.45	TCP	5062	59720	66	5062->59720 [ACK] Seq=1 Ack=201 win=30080 Len=0 TSval=955270527 TSecr=91375178
8	2017-09-26 12:18:08.440978	172.16.2.2	146.20.193.45	TLSv1.2	5062	59720	1780	Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Do
9	2017-09-26 12:18:08.453269	146.20.193.45	172.16.2.2	TCP	59720	5062	66	59720->5062 [ACK] Seq=201 Ack=1369 win=17536 Len=0 TSval=91375204 TSecr=955270540
10	2017-09-26 12:18:08.453308	146.20.193.45	172.16.2.2	TCP	59720	5062	66	59720->5062 [ACK] Seq=201 Ack=1715 win=20352 Len=0 TSval=91375204 TSecr=955270540
11	2017-09-26 12:18:08.455698	146.20.193.45	172.16.2.2	TLSv1.2	59720	5062	73	Alert (Level: Fatal, Description: Certificate Unknown)

Certificate Unknown Sourced from Spark

If you inspect the Default Server certificate from the Expressway-E, you can see that the 'Common Name' and 'Subject Alternate Names' do not contain the 'Verified Domain' (**rtp.ciscotac.net**). You then have evidence about what causes this issue as shown in the image.

The image displays a network packet capture (Wireshark) and a Windows Certificate dialog box. The packet capture shows a TLS handshake where the server sends a certificate. The certificate's 'Common Name' is 'amer-expressway01'. The Windows dialog box shows the certificate information, including the issuer 'Temporary CA 01162d22-e216-470f-991b-882c49981ae7' and the validity period from 9/26/2017 to 9/26/2018. A warning message states: 'Windows does not have enough information to verify this certificate.' A red arrow points from the 'Common Name' field in the packet capture to the 'Issued to' field in the Windows dialog box.

At this point, you determined that the Expressway-E server certificate needs to be signed by either a Public CA or an Internal CA.

## Solution

In order to resolve this issue, you have two options:

1. Have the Expressway-E certificate be signed by a [Public CA that Cisco Webex trusts](#).  
 Log into the Expressway. Navigate to **Maintenance > Security > Server certificate**. Select **Generate CSR**. Enter the required certificate information and ensure that the **Additional alternative names** field contains the **Verified Domain** listed in the Webex Control Hub. Click **Generate CSR**. Provide the CSR to a 3rd party Public CA for signing. Upon return of the certificate, navigate to **Maintenance > Security > Server certificates**. In the **Upload New Certificate** section next to **Select the server certificate file**, select **Choose File** and select the **signed certificate**. Select **Upload server certificate data**. Navigate to **Maintenance > Security > Trusted CA certificate**. In the **Upload** section next to **Select the file containing trusted CA certificates** select **Choose File**. Select any root and intermediate CA certificates provided by the Public CA. Select **Append CA certificate**. Restart the Expressway-E.
2. Have the Expressway-E certificate be signed by an Internal CA and then upload the Internal CA and Expressway-E to the Cisco Webex Control Hub.  
 Log into the Expressway. Navigate to **Maintenance > Security > Server certificate**. Select **Generate CSR**. Enter the required certificate information ensuring that the *Additional alternative names field* contains the **Verified Domain** listed in the Webex Control Hub. Click **Generate CSR**. Provide the CSR to a 3rd party Public CA for signing. Upon return of the certificate, navigate to **Maintenance > Security > Server certificates**. In the **Upload New Certificate** section next to **Select the server certificate file**, select **Choose File** and select the **signed certificate**. Select **Upload server certificate data**. Navigate to **Maintenance > Security > Trusted CA certificate**. In the **Upload** section next to **Select the file containing trusted CA certificates** select **Choose File**. Select any root and intermediate CA certificates provided by the Public CA. Select **Append CA certificate**. Restart the Expressway-E.

2a. Upload the Internal CA and Expressway-E certificate to the Cisco Webex Control Hub

1. Log into the [Cisco Webex Control Hub](#) as an Administrator.
2. Select **Services**.
3. Select **Settings** under the Hybrid Call Service card.
4. In the **Certificates for Encrypted SIP Calls** section select **Upload**.
5. Choose the Internal CA and Expressway-E certificates.

## Inbound: Cisco Webex to On-Premises

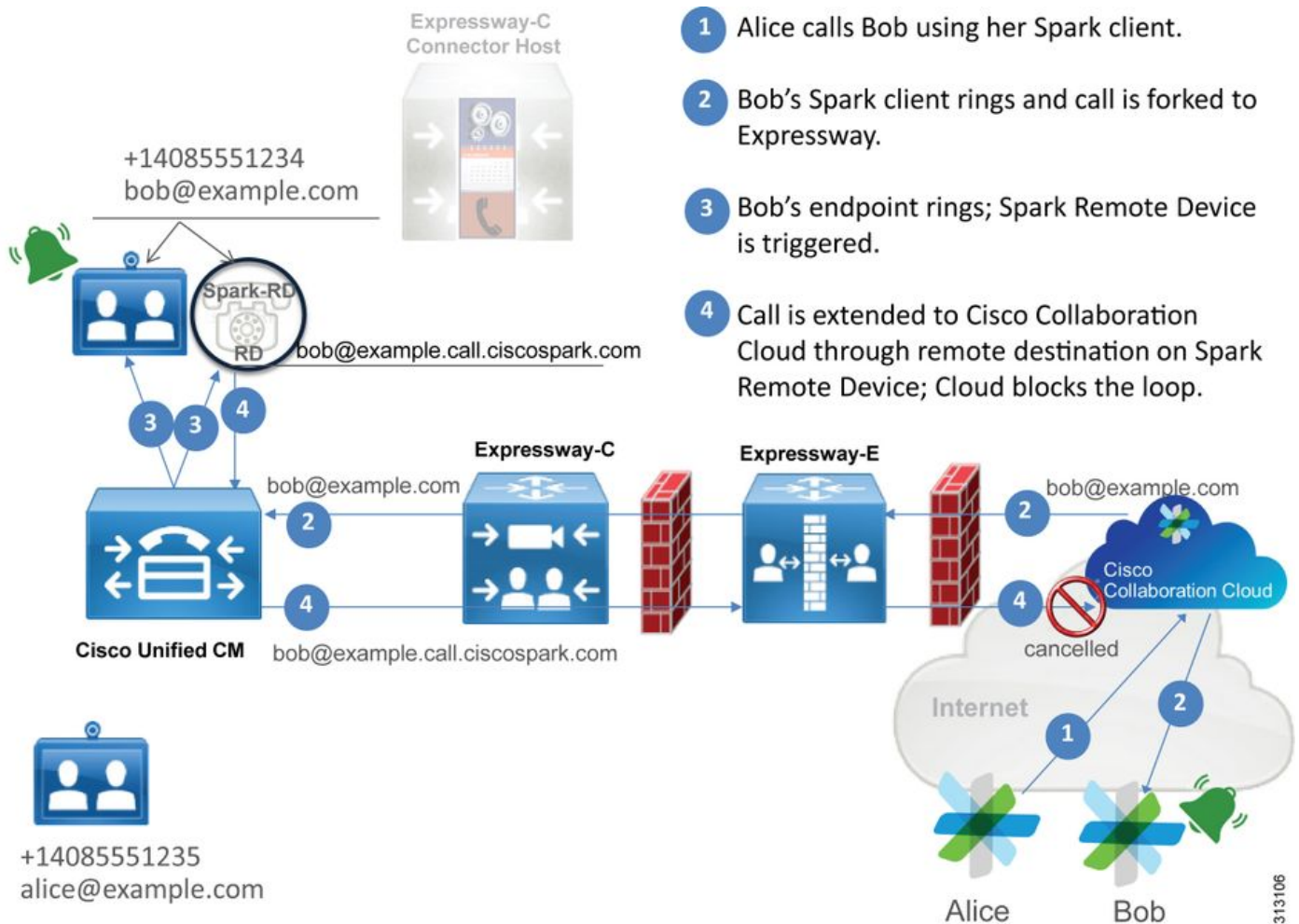
Almost every inbound Cisco Webex to on-premises failure results in the same reported symptom: "When I call from my Cisco Webex app to another colleague's app, the colleague's app rings but the on-premises phone does not." In order to troubleshoot this scenario, you'll find it helpful to understand both the call flow and logic that occurs when this type of call is being placed.

### High Level Logic Flow

1. Cisco Webex app calling party initiates the call
2. Called party's app rings
3. The call is forked to the Cisco Webex environment
4. The Cisco Webex environment must perform a DNS Lookup based on the customer's configured SIP Destination in the Cisco Webex Control Hub
5. The Cisco Webex environment attempts to connect to the Expressway over port 5062
6. The Cisco Webex environment attempts to perform a mutual TLS handshake
7. The Cisco Webex environment sends a SIP INVITE to the Expressway which is passed down to the on-premises collaboration endpoint/IP phone
8. Cisco Webex and the enterprise complete the SIP negotiation
9. Cisco Webex and the enterprise begin sending and receiving media.

### Call Flow

Navigate to **Cisco Webex app > Cisco Webex environment > Expressway-E > Expressway-C > On-Premises Collaboration Endpoint/IP Phone** as shown in the image.



- 1 Alice calls Bob using her Spark client.
- 2 Bob's Spark client rings and call is forked to Expressway.
- 3 Bob's endpoint rings; Spark Remote Device is triggered.
- 4 Call is extended to Cisco Collaboration Cloud through remote destination on Spark Remote Device; Cloud blocks the loop.

Here are some of the common issues observed with Inbound calls from Webex to the on-premises infrastructure.

### Issue 1. Cisco Webex is unable to resolve the Expressway-E DNS SRV/hostname

When thinking about the Cisco Webex to on-premises call flow, Cisco Webex's first logical step is how to contact the on-premises Expressway. As noted above, Cisco Webex will attempt to connect to the on-premises Expressway by performing an SRV lookup based on the configured **SIP Destination** that is listed in the **Hybrid Call Service Settings** page in the [Cisco Webex Control Hub](#).

If you attempt to troubleshoot this situation from an Expressway-E diagnostic log perspective, you do not see any traffic from Cisco Webex. If you try to search for TCP Connecting, you would not see the Dst-port=5062, nor would you see any subsequent MTLS handshake or SIP Invite from Cisco Webex.

If this is the situation, you must check how the **SIP Destination** was configured in the Cisco Webex Control Hub. You can also use the **Hybrid Connectivity Test Tool** to aid in troubleshooting. The Hybrid Connectivity Test Tool checks if there is a valid DNS address, if Cisco Webex can connect to the port returned in the SRV lookup, and if the on-premises Expressway has a valid certificate that Cisco Webex trusts.

1. Log into the [Cisco Webex Control Hub](#)
2. Select **Services**
3. Select the **Settings** link in the **Hybrid Call** card.

4. In the Call Service Connect section verify the domain used for the public SIP SRV address in the **SIP Destination** field.
5. If the record has been entered correct, click **Test** to see if the record is valid.
6. As pictured below, you can clearly see that the public domain does not have a corresponding SIP SRV record associated to it as shown in the image.



select **View test results** and you can see more detail about what failed as shown in the image.



As another approach, you can also look up the SRV record by using nslookup. Here are the commands you can run to verify if the SIP Destination exists.

```
C:\Users\pstoiano>nslookup
> server 8.8.8.8
Default Server: google-public-dns-a.google.com
Address: 8.8.8.8
> set type=SRV
> _sips._tcp.mtls.rtp.ciscotac.net
Server: google-public-dns-a.google.com
Address: 8.8.8.8
DNS request timed out.
timeout was 2 seconds.
DNS request timed out.
timeout was 2 seconds.
*** Request to google-public-dns-a.google.com timed-out
```

As you can see in the code block above, the nslookup command was initiated then the server is set to 8.8.8.8 which is a public Google DNS server. Lastly, you are setting the record types to lookup to SRV records. At that point, you can then issue the full SRV record you want to look up. The net result is that the requests ultimately times out.

### Solution

1. Configure a public SIP SRV address for the Expressway-E on the site they use to host public domain names.
2. Configure a hostname that will resolve to the public IP address of the Expressway-E
3. Configure the SIP Destination to list the domain used for the SIP SRV address created in Step 1. Log into the [Cisco Webex Control Hub](#) Select **Services** Select the **Settings** link in the *Hybrid Call card* In the Call Service Connect section enter the domain used for the public SIP SRV address in the **SIP Destination** field. Select Save



**Note:** If the SIP SRV record you would like to use is already being leveraged for business-to-business communications, we recommend specifying a subdomain of the corporate domain as the SIP discovery address in Cisco Webex Control Hub, and consequently a public DNS SRV record, as follows:

Service and protocol: `_sips._tcp.mtls.example.com`  
Priority: 1  
Weight: 10  
Port number: 5062  
Target: `us-expe1.example.com`

The above recommendation was pulled directly from the [Cisco Webex Hybrid Design Guide](#).

### Alternate Solution

If the customer does not have a SIP SRV record present (and does not plan to create one), they can alternatively list the Expressway Public IP address suffixed by `":5062"`. By doing this, the Webex environment will not attempt an SRV lookup but rather connect directly to the `%Expressway_Pub_IP%:5062`. (Example: `64.102.241.236:5062`)

1. Configure the SIP Destination to be formatted as `%Expressway_Pub_IP%:5062`. (Example: `64.102.241.236:5062`) Log into the [Cisco Webex Control Hub](#) Select **Services** Select the **Settings** link in the *Hybrid Call card* In the Call Service Connect section enter the `%Expressway_Pub_IP%:5062` in the **SIP Destination** field. Select Save

For more information about the SIP Destination address and/or SRV record that must be setup. Refer the [Enable Hybrid Call Service Connect for Your Organization](#) section of the Cisco Webex Hybrid Call Service Deployment Guide or the [Cisco Webex Hybrid Design Guide](#).

### Issue 2. Socket Failure: Port 5062 is Blocked Inbound to Expressway

After the DNS resolution completes, the Cisco Webex environment to attempt to establish a TCP connection over port 5062 to the IP address that was returned during the DNS lookup. This IP address is going to be the public IP address of the on-premises Expressway-E. If the Cisco Webex environment is unable to establish this TCP connection, the call inbound to the premises is subsequently fail. The symptom for this particular condition is the same as almost every other Cisco Webex inbound call failure: the on-premises phone does not ring.

If you're troubleshooting this issue using the Expressway diagnostic logs, you will not see any traffic from Cisco Webex. If you try to search for TCP Connecting, you would not see any connection attempts for the `Dst-port=5062`, nor would you see any subsequent MTLN handshake or SIP Invite from Cisco Webex. Since the Expressway-E diagnostic logging is of no use in this situation, you have a few possible methods for verification:

1. Get a packet capture off the outside interface of the firewall
2. Leverage a port checking utility
3. Use the Hybrid Connectivity Test tool

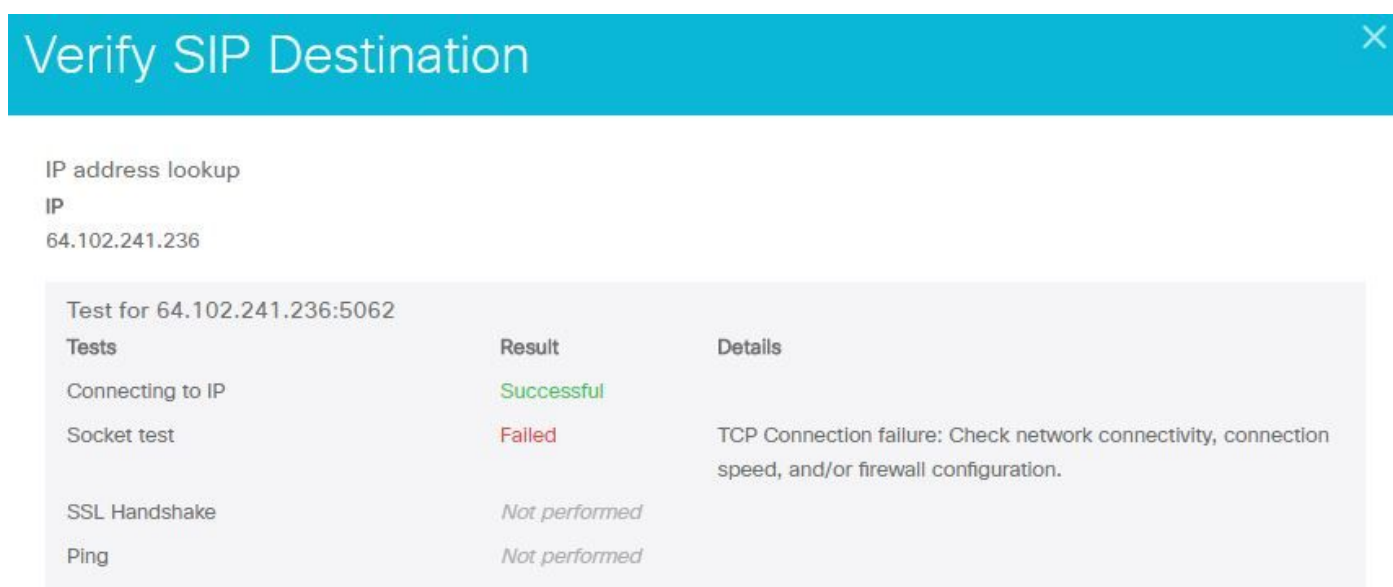
Since the Hybrid Connectivity Test tool is built right into the Cisco Webex Control Hub and simulates the Cisco Webex environment trying to connect to the on-premises Expressway, it is the most ideal verification method available. To test the TCP Connectivity into the organization:

1. Log into the [Cisco Webex Control Hub](#)
2. Select **Services**
3. Select the **Settings** link in the **Hybrid Call card**
4. In the Call Service Connect section ensure the value entered in the SIP Destination is correct
5. Click Test as shown in the image.

SIP Destination ⓘ



6. Since the test has failed you can click the **View test results** link to check the details as shown in the image.



As observed in the image above, you can see that the Socket test has failed when trying to connect to 64.102.241.236:5062. Having this data in addition to the Expressway diagnostic logs/pcaps not show any connection attempts, you now have enough evidence to investigate the firewall ACL/NAT/Routing configuration.

### Solution

Since this particular issue isn't caused by the Cisco Webex environment or the on-premises collaboration equipment, you need to focus on the firewall configuration. Since you cannot necessarily predict the type of firewall you will be interfacing with, you need to rely on someone with familiarity with the device. It's possible that the issue could be related to a firewall ACL, NAT, or routing misconfiguration.

### Issue 3. Socket Failure: Expressway-E is not Listening on Port 5062

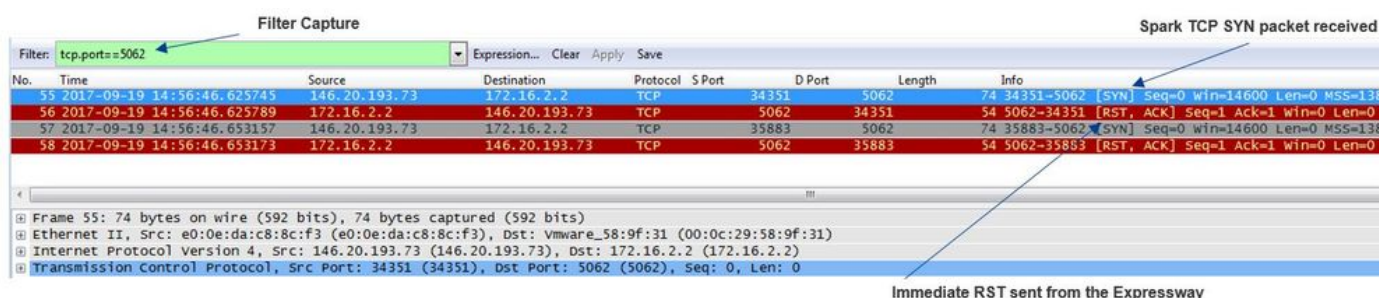
This particular condition is often diagnosed incorrectly. Many times, it is assumed that the firewall is the cause for why the traffic over port 5062 is getting blocked. To troubleshoot this particular condition, you can use the techniques in the "Port 5062 is blocked inbound to the Expressway" scenario above. You will find that the Hybrid Connectivity Test tool and any other tool used to

check port connectivity will fail. The first assumption is that the firewall is blocking the traffic. Most people will then double check the diagnostic logging from the Expressway-E to determine if they can see the TCP connection trying to establish. They will general look for a log line item such as this as shown in the image.

```
2017-09-19T14:01:46.462-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 18:01:46,461"
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"
```

In this condition, the particular log entry above will not exist. Therefore, many people will misdiagnose the condition and assume it is the firewall.

If a packet capture is included with the diagnostic logging, you can verify that the firewall is not the cause. Below is a packet capture sample from thescenariio where the Expressway-E was not listening over port 5062. This capture filtered by using **tcp.port==5062** as the applied filter as shown in the image.



As you can see in the packet capture that was obtained from the Expressway-E, the traffic over tcp port 5062 is not being blocked by the firewall but is in fact arriving. In packet number 56, you can see that the Expressway-E is sending the RST immediately after the initial TCP SYN packet arrived. With this information, you can conclude that the issue is isolated to the Expressway-E receiving the packet; you must troubleshoot the issue from the Expressway-E perspective. Given the evidence, consider possible reasons for why the Expressway-E would RST the packet. Two possibilities that could attribute to this behavior are:

1. The Expressway-E has some type of firewall rules set up that could be blocking the traffic
2. The Expressway-E is not listening for Mutual TLS traffic and/or not listening for traffic over port 5062.

The Expressway-E's firewall functionality exists under *System > Protection > Firewall rules > Configuration*. When this was checked in this environment, there was no firewall configuration present.

There are several ways to verify if the Expressway-E is listening for Mutual TLS traffic over port 5062. You can do this either through the Web Interface or the CLI as a root user.

From the root of the Expressway, if you issue **netstat -an | grep ':5062'**, you should get some output similar to what you see below.

```
~ # netstat -an | grep ':5062'
tcp        0      0 172.16.2.2:5062      0.0.0.0:*           LISTEN    <-- Outside
Interface
tcp        0      0 192.168.1.6:5062     0.0.0.0:*           LISTEN    <-- Inside Interface
tcp        0      0 127.0.0.1:5062       0.0.0.0:*           LISTEN
```

```
tcp          0          0 ::::5062          :::*          LISTEN
```

This information can also be captured through the web interface of the Expressway-E. See the steps below to gather this information

1. Log into the Expressway-E
2. Navigate to **Maintenance Tools > Port usage > Local inbound ports**
3. Search for Type SIP and IP port 5062. (highlighted in red as shown in the image)

Type	Description	Protocol	IP address	IP port	Transport	Actions
H.323	Registration UDP port	H.323	192.168.1.6	1719	UDP	<a href="#">View/Edit</a>
H.323	Registration UDP port	H.323	172.16.2.2	1719	UDP	<a href="#">View/Edit</a>
SIP	TCP port	SIP	192.168.1.6	5060	TCP	<a href="#">View/Edit</a>
SIP	TCP port	SIP	172.16.2.2	5060	TCP	<a href="#">View/Edit</a>
SIP	TLS port	SIP	192.168.1.6	5061	TCP	<a href="#">View/Edit</a>
SIP	TLS port	SIP	172.16.2.2	5061	TCP	<a href="#">View/Edit</a>
SIP	Mutual TLS port	SIP	192.168.1.6	5062	TCP	<a href="#">View/Edit</a>
SIP	Mutual TLS port	SIP	172.16.2.2	5062	TCP	<a href="#">View/Edit</a>

Now that you know what you should see, you can compare that to the current environment. From the CLI perspective, when you run **netstat -an | grep ':5062'**, the output looks like this:

```
~ # netstat -an | grep ':5062'
tcp          0          0 127.0.0.1:5062          0.0.0.0:*          LISTEN
tcp          0          0 ::::5062          :::*          LISTEN
~ #
```

Additionally, the web UI does not show the Mutual TLS port listed under Local inbound ports

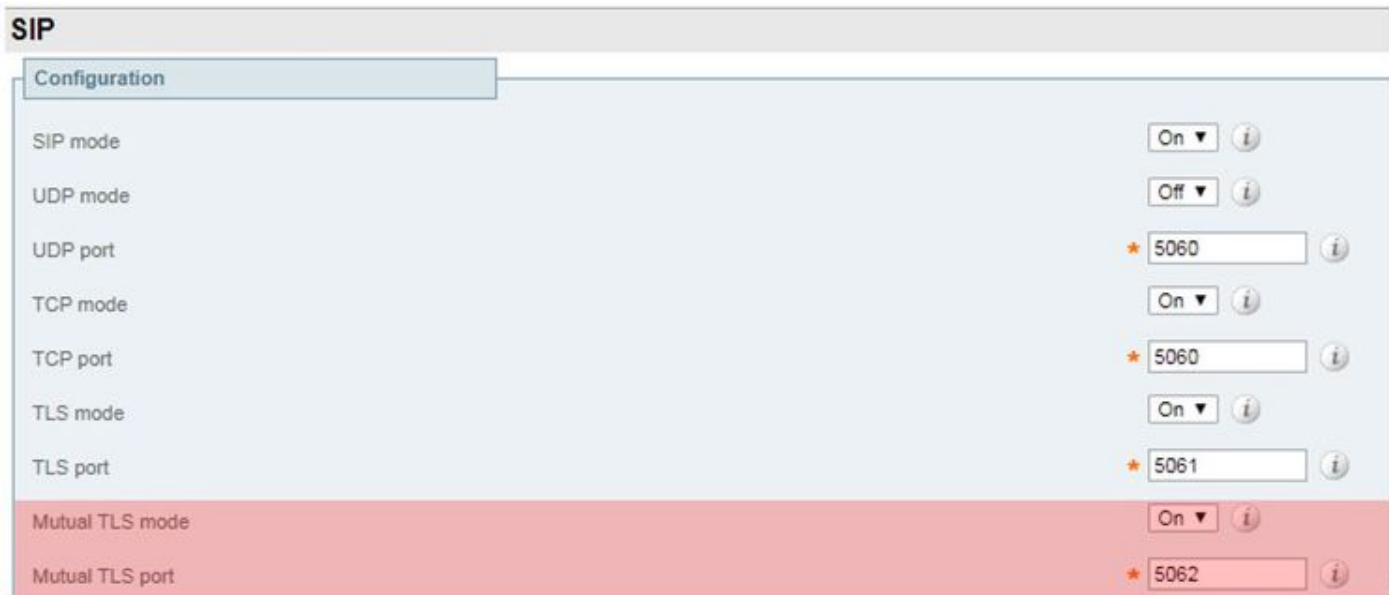
Type	Description	Protocol	IP address	IP port	Transport
H.323	Call signaling port range	H.323	192.168.1.6	15000-19999	TCP
H.323	Call signaling port range	H.323	172.16.2.2	15000-19999	TCP
H.323	Registration UDP port	H.323	192.168.1.6	1719	UDP
H.323	Registration UDP port	H.323	172.16.2.2	1719	UDP
SIP	TCP port	SIP	192.168.1.6	5060	TCP
SIP	TCP port	SIP	172.16.2.2	5060	TCP
SIP	TLS port	SIP	192.168.1.6	5061	TCP
SIP	TLS port	SIP	172.16.2.2	5061	TCP

With this data, you can conclude that the Expressway-E is not listening for Mutual TLS traffic.

## Solution

In order to solve this problem, you must ensure that the Mutual TLS mode is enabled and that the Mutual TLS port is set to 5062 on the Expressway-E:

1. Log into the Expressway-E
2. Navigate to **Configuration > Protocols > SIP**
3. Ensure the Mutual TLS mode is set to **On**
4. Ensure the Mutual TLS port is set to **5062**
5. Click **Save** as shown in the image.



#### Issue 4. Expressway-E or C does not Support Preloaded SIP Route Headers

With Hybrid Call Service Connect, the call routing is done based on **route header**. The route header is populated based on the information that the Call Service Aware (Expressway Connector) portion of the solution delivers to Cisco Webex. The Expressway connector host queries the Unified CM for users who are enabled for the Call Service and pull both their **Directory URI** and the **Cluster FQDN of their Unified CM home cluster**. See these example, using Alice and Bob:

<b>Directory URI</b>	<b>Destination Route Header</b>
bob@example.com	emea-cucm.example.com
alice@example.com	us-cucm.example.com

If Alice or Bob make a call, the call is routed to their on-premises Unified CM so that it can be anchored to their Cisco WebexRD before routing out to the called user.

If Alice were to call Bob, the call would route to *Alice's Unified CM Home Cluster FQDN (us-cucm.example.com)*. If you analyze the SIP INVITE that Cisco Webex sends inbound to the Expressway-E, you'd find the following information within the SIP header

**Request URI** sip: bob@example.com  
**Route Header** sip:us-cucm.example.com;lr

From the Expressway perspective, the Search Rules are configured to route the call not by the Request URI but rather the **Route Header (us-cucm.example.com)** -- in this case Alice's Unified CM home cluster.

With this foundation set, you can understand troubleshoot situations where the Expressways are misconfigured, which causes the above logic not to work. As nearly every other inbound Hybrid Call Service Connect call setup failure, the symptom is that *the on-premises phone does not ring*.

Before you analyze the diagnostic logs on the Expressway, consider how to identify this call:

1. The SIP Request URI will be the **Directory URI of the Called Party**.
2. The SIP FROM field will be formatted with the **Calling Party** listed as "**First Name Last Name**" <sip:WebexDisplayName@subdomain.call.ciscopark.com>

With this information, you can search the diagnostic logs by **Directory URI of Called Party, First and Last Name of Calling Party, or Cisco Webex SIP Address of the Calling Party**. If you don't have any of this information, you can search on "**INVITE SIP:**" which locates all SIP calls running over the Expressway. Once you have identified the SIP INVITE for the Inbound call, you can then locate and copy the SIP Call ID. After you have this value, you can simply search the diagnostic logs based on the Call-ID to see all messages that correlate to this call leg.

Another thing to help isolate the routing problem is to determine how far the call goes into the enterprise. You can try to search for the information noted above on the Expressway-C to see if the call was routed that far. If so, you will likely want to start your investigation there.

In this scenario, you can see that the Expressway-C has received the INVITE from the Expressway-E.

```
2017-09-19T14:16:15.836-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-19 18:16:15,830"
Module="network.sip" Level="DEBUG": Action="Received" Local-ip="192.168.1.5" Local-port="26847"
Src-ip="192.168.1.6" Src-port="7003" Msg-Hash="11449260850208794722"
SIPMSG:
|INVITE sip:jorobb@rtp.ciscotac.net SIP/2.0
Via: SIP/2.0/TLS 192.168.1.6:7003;egress-
zone=HybridCallServiceTraversal;branch=z9hG4bKc81c6c4dddef7ed6be5bdce9868fb019913;proxy-call-
id=a82052ef-6fd7-4506-8173-e73af6655b5d;rport
Via: SIP/2.0/TLS 192.168.1.6:5073;branch=z9hG4bKb0eba6d700dfdf761a8ad97fff3c240124;x-cisco-
local-service=nettle;received=192.168.1.6;rport=43119;ingress-zone=DefaultZone
Via: SIP/2.0/TLS 64.102.241.236:5061;egress-
zone=DefaultZone;branch=z9hG4bK6fe399bae58fb0d70c9d69b8e37e13e5912.4248943487bff4af6f649b586c769
6bb;proxy-call-id=f2d15853-c81f-462f-b3e5-c08124f344a3;received=172.16.2.2;rport=25016
Via: SIP/2.0/TLS
192.168.5.66:5062;branch=z9hG4bK0f455ca79cf1b0af5637333aa5286436;received=146.20.193.45;rport=35
464;ingress-zone=HybridCallServicesDNS
Via: SIP/2.0/TLS 127.0.0.1:5070;branch=z9hG4bK-383039-
8f0d64025c04d23b6d5e1d5142db46ec;rport=52706
Call-ID: 9062bca7eca2afe71b4a225048ed5101@127.0.0.1
CSeq: 1 INVITE
Contact: <sip:192.168.1.6:5073;transport=tls>;call-type=squared
From: "pstojoano test" <sip:pstojoano-test@dmzlab.call.ciscospark.com>;tag=872524918
To: <sip:jorobb@rtp.ciscotac.net>
Max-Forwards: 15
Route: <sip:cucm.rtp.ciscotac.net;lr>
Record-Route: <sip:proxy-call-id=a82052ef-6fd7-4506-8173-
e73af6655b5d@192.168.1.6:7003;transport=tls;lr>
Record-Route: <sip:proxy-call-id=a82052ef-6fd7-4506-8173-
e73af6655b5d@192.168.1.6:5061;transport=tls;lr>
```

The important thing is that the **route header (Cluster FQDN)** is still intact. However, there's no search logic being performed based on the route header (Cluster FQDN) **cucm.rtp.ciscotac.net**. Rather, you see the message getting rejected immediately with a **404 Not Found**.

```
2017-09-19T14:16:15.836-04:00 rtp12-tpdmz-118-VCSC tvcs: Event="Call Attempted" Service="SIP"
Src-ip="192.168.1.6" Src-port="7003" Src-alias-type="SIP" Src-alias="sip:pstojoano-
test@dmzlab.call.ciscospark.com" Dst-alias-type="SIP" Dst-alias="sip:jorobb@rtp.ciscotac.net"
Call-serial-number="a3e44231-f62a-4e95-a70e-253701a89515" Tag="73c276e2-3917-4a0c-9fc5-
ddde83b49fd0" Protocol="TLS" Auth="NO" Level="1" UTCTime="2017-09-19 18:16:15,832"
2017-09-19T14:16:15.836-04:00 rtp12-tpdmz-118-VCSC tvcs: Event="Search Attempted" Service="SIP"
Src-alias-type="SIP" Src-alias="pstojoano-test@dmzlab.call.ciscospark.com" Dst-alias-type="SIP"
Dst-alias="sip:jorobb@rtp.ciscotac.net" Call-serial-number="a3e44231-f62a-4e95-a70e-
253701a89515" Tag="73c276e2-3917-4a0c-9fc5-ddde83b49fd0" Detail="searchtype:INVITE" Level="1"
UTCTime="2017-09-19 18:16:15,834"
2017-09-19T14:16:15.836-04:00 rtp12-tpdmz-118-VCSC tvcs: Event="Search Completed" Reason="Not
```

**Found** Service="SIP" Src-alias-type="SIP" **Src-alias="pstojoano-test@dmzlab.call.ciscospark.com"** Dst-alias-type="SIP" **Dst-alias="sip:jorobb@rtp.ciscotac.net"** Call-serial-number="a3e44231-f62a-4e95-a70e-253701a89515" Tag="73c276e2-3917-4a0c-9fc5-ddde83b49fd0" **Detail="found:false**, searchtype:INVITE, Info:Policy Response" Level="1" UTCTime="2017-09-19 18:16:15,835" 2017-09-19T14:16:15.836-04:00 rtp12-tpdmz-118-VCSC tvcs: Event="Call Rejected" Service="SIP" Src-ip="192.168.1.6" Src-port="7003" Src-alias-type="SIP" **Src-alias="sip:pstojoano-test@dmzlab.call.ciscospark.com"** **Dst-alias-type="SIP" Dst-alias="sip:jorobb@rtp.ciscotac.net"** Call-serial-number="a3e44231-f62a-4e95-a70e-253701a89515" Tag="73c276e2-3917-4a0c-9fc5-ddde83b49fd0" **Detail="Not Found"** Protocol="TLS" **Response-code="404"** Level="1" UTCTime="2017-09-19 18:16:15,835" 2017-09-19T14:16:15.836-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-19 18:16:15,830" Module="network.sip" Level="INFO": Action="Received" Local-ip="192.168.1.5" Local-port="26847" Src-ip="192.168.1.6" Src-port="7003" Detail="Receive Request Method=INVITE, CSeq=1, **Request-URI=sip:jorobb@rtp.ciscotac.net**, Call-ID=9062bca7eca2afe71b4a225048ed5101@127.0.0.1, From-Tag=872524918, To-Tag=, Msg-Hash=11449260850208794722, Local-SessionID=daf7c278732bb5a557fb57925dffcbf7, Remote-SessionID=00000000000000000000000000000000" 2017-09-19T14:16:15.836-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-19 18:16:15,836" Module="network.sip" Level="INFO": Action="Sent" Local-ip="192.168.1.5" Local-port="26847" Dst-ip="192.168.1.6" Dst-port="7003" Detail="**Sending Response Code=404**, Method=INVITE, CSeq=1, **To=sip:jorobb@rtp.ciscotac.net**, Call-ID=9062bca7eca2afe71b4a225048ed5101@127.0.0.1, From-Tag=872524918, To-Tag=96b9a0eaf669a590, Msg-Hash=254718822158415175, Local-SessionID=00000000000000000000000000000000, Remote-SessionID=daf7c278732bb5a557fb57925dffcbf7" 2017-09-19T14:16:15.836-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-19 18:16:15,836" Module="network.sip" Level="DEBUG": Action="Sent" Local-ip="192.168.1.5" Local-port="26847" Dst-ip="192.168.1.6" Dst-port="7003" Msg-Hash="254718822158415175" SIPMSG: |**SIP/2.0 404 Not Found** Via: SIP/2.0/TLS 192.168.1.6:7003;egress-zone=HybridCallServiceTraversal;branch=z9hG4bKc81c6c4dddef7ed6be5bdce9868fb019913;proxy-call-id=a82052ef-6fd7-4506-8173-e73af6655b5d;received=192.168.1.6;rport=7003;ingress-zone=HybridCallServiceTraversal Via: SIP/2.0/TLS 192.168.1.6:5073;branch=z9hG4bKb0eba6d700dfdf761a8ad97fff3c240124;x-cisco-local-service=nettle;received=192.168.1.6;rport=43119;ingress-zone=DefaultZone Via: SIP/2.0/TLS 64.102.241.236:5061;egress-zone=DefaultZone;branch=z9hG4bK6fe399bae58fb0d70c9d69b8e37e13e5912.4248943487bff4af6f649b586c7696bb;proxy-call-id=f2d15853-c81f-462f-b3e5-c08124f344a3;received=172.16.2.2;rport=25016 Via: SIP/2.0/TLS 192.168.5.66:5062;branch=z9hG4bK0f455ca79cf1b0af5637333aa5286436;received=146.20.193.45;rport=35464;ingress-zone=HybridCallServicesDNS Via: SIP/2.0/TLS 127.0.0.1:5070;branch=z9hG4bK-383039-8f0d64025c04d23b6d5eld5142db46ec;rport=52706 Call-ID: **9062bca7eca2afe71b4a225048ed5101@127.0.0.1** CSeq: 1 INVITE From: "**pstojoano test**" <sip:pstojoano-test@dmzlab.call.ciscospark.com>;tag=872524918 To: <sip:jorobb@rtp.ciscotac.net>;tag=96b9a0eaf669a590 Server: TANDBERG/4135 (X8.10.2) Warning: 399 192.168.1.5:5061 "Policy Response" Session-ID: 00000000000000000000000000000000;remote=daf7c278732bb5a557fb57925dffcbf7 Content-Length: 0

Compared to a working scenario, you would see that in the working scenario the the search logic is being performed based on the Router Header (Cluster FQDN)

2017-09-22T13:56:02.215-04:00 rtp12-tpdmz-118-VCSC tvcs: Event="Search Attempted" Service="SIP" Src-alias-type="SIP" **Src-alias="pstojoano-test@dmzlab.call.ciscospark.com"** Dst-alias-type="SIP" **Dst-alias="sip:jorobb@rtp.ciscotac.net"** Call-serial-number="17aa8dc7-422c-42ef-bdd9-b9750fbd0edf" Tag="8bd936da-f2ab-4412-96df-d64558f7597b" Detail="searchtype:INVITE" Level="1" UTCTime="2017-09-22 17:56:02,215" 2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,217"

```
Module="network.cpl" Level="DEBUG": Remote-ip="192.168.1.6" Remote-port="7003" Detail="CPL:
<routed> "
2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,218"
Module="network.cpl" Level="DEBUG": Remote-ip="192.168.1.6" Remote-port="7003" Detail="CPL:
<location clear="yes" url="sip:cucm.rtp.ciscotac.net;lr" diversion="" dest-url-for-
message="sip:jorobb@rtp.ciscotac.net" sip-route-set="" dest-service=""> added
sip:cucm.rtp.ciscotac.net;lr to location set "
2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,218"
Module="network.cpl" Level="DEBUG": Remote-ip="192.168.1.6" Remote-port="7003" Detail="CPL:
<proxy stop-on-busy="no" timeout="0"/> "
2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,218"
Module="network.search" Level="DEBUG": Detail="Search rule 'Inbound MS to CMS' did not match
destination alias 'cucm.rtp.ciscotac.net;lr'"
2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,218"
Module="network.search" Level="DEBUG": Detail="Search rule 'multiway' did not match destination
alias 'cucm.rtp.ciscotac.net;lr'"
2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,218"
Module="network.search" Level="DEBUG": Detail="Search rule 'WebEx Search Rule' did not match
destination alias 'cucm.rtp.ciscotac.net;lr'"
2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,218"
Module="network.search" Level="DEBUG": Detail="Search rule 'ISDN Inbound' ignored due to source
filtering"
2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,218"
Module="network.search" Level="DEBUG": Detail="Search rule 'recalls into CMS' did not match
destination alias 'cucm.rtp.ciscotac.net;lr'"
2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,218"
Module="network.search" Level="DEBUG": Detail="Search rule 'CEtcp-rtp12-tpdmz-118-ucmpub' did
not match destination alias 'cucm.rtp.ciscotac.net;lr'"
2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,218"
Module="network.search" Level="DEBUG": Detail="Search rule 'Conference Factory' did not match
destination alias 'cucm.rtp.ciscotac.net;lr'"
2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,219"
Module="network.search" Level="DEBUG": Detail="Search rule 'Inbound B2B Calling' did not match
destination alias 'cucm.rtp.ciscotac.net;lr'"
2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,219"
Module="network.search" Level="DEBUG": Detail="Search rule 'Calls to Cisco Webex' did not match
destination alias 'cucm.rtp.ciscotac.net;lr'"
2017-09-22T13:56:02.218-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,219"
Module="network.search" Level="DEBUG": Detail="Considering search rule 'as is local' towards
target 'LocalZone' at priority '1' with alias 'cucm.rtp.ciscotac.net;lr'"
2017-09-22T13:56:02.219-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,219"
Module="network.search" Level="DEBUG": Detail="Considering search rule 'Hybrid Call Service
Inbound Routing' towards target 'CUCM11' at priority '2' with alias 'cucm.rtp.ciscotac.net;lr'"
```

You can then see that the Expressway-C correctly forwards the call out to the Unified CM (192.168.1.21).

```
2017-09-22T13:56:02.232-04:00 rtp12-tpdmz-118-VCSC tvcs: UTCTime="2017-09-22 17:56:02,232"
Module="network.sip" Level="DEBUG": Action="Sent" Local-ip="192.168.1.5" Local-port="25606" Dst-
ip="192.168.1.21" Dst-port="5065" Msg-Hash="866788495063340574"
SIPMSG:
|INVITE sip:jorobb@rtp.ciscotac.net SIP/2.0
Via: SIP/2.0/TCP 192.168.1.5:5060;egress-
zone=CUCM11;branch=z9hG4bK251d6daf044e635607cc13d244b9ea45138220.69ccb8de20a0e853c1313782077f77b
5;proxy-call-id=17aa8dc7-422c-42ef-bdd9-b9750fbd0edf;rport
Via: SIP/2.0/TLS 192.168.1.6:7003;egress-
zone=HybridCallServiceTraversal;branch=z9hG4bKba323da436b2bc288200d56d11f02d4d272;proxy-call-
id=32c76cef-e73c-4911-98d0-e2d2bb6fec77;received=192.168.1.6;rport=7003;ingress-
zone=HybridCallServiceTraversal
Via: SIP/2.0/TLS 192.168.1.6:5073;branch=z9hG4bK06cde3f662d53a210b5b4b11b85500c19;x-cisco-local-
service=nettle;received=192.168.1.6;rport=42533;ingress-zone=DefaultZone
Via: SIP/2.0/TLS 64.102.241.236:5061;egress-
zone=DefaultZone;branch=z9hG4bK297799f31d0785ff7449e1d7dbe3595b271.2ed90cbcd5b79c6cffad9ecd84cc8
```



```
337;proxy-call-id=3be87d96-d2e6-4489-b936-8f9cb5ccaa5f;received=172.16.2.2;rport=25005
Via: SIP/2.0/TLS
192.168.4.146:5062;branch=z9hG4bK043ca6360f253c6abed9b23fbfeff9819;received=148.62.40.64;rport=36
149;ingress-zone=HybridCallServicesDNS
Via: SIP/2.0/TLS 127.0.0.1:5070;branch=z9hG4bK-353038-
8c648a16c2c5d7b85fa5c759d59aa190;rport=47732
Call-ID: daala6fa546ce76591fc464f0a50ee32@127.0.0.1
CSeq: 1 INVITE
Contact: <sip:192.168.1.6:5073;transport=tls>;call-type=squared
From: "pstoiano test" <sip:pstoiano-test@dmzlab.call.ciscospark.com>;tag=567490631
To: <sip:jorobb@rtp.ciscotac.net>
Max-Forwards: 14
Route: <sip:cucm.rtp.ciscotac.net;lr>
Record-Route: <sip:proxy-call-id=17aa8dc7-422c-42ef-bdd9-
b9750fbd0edf@192.168.1.5:5060;transport=tcp;lr>
Record-Route: <sip:proxy-call-id=17aa8dc7-422c-42ef-bdd9-
b9750fbd0edf@192.168.1.5:5061;transport=tls;lr>
Record-Route: <sip:proxy-call-id=32c76cef-e73c-4911-98d0-
e2d2bb6fec77@192.168.1.6:7003;transport=tls;lr>
Record-Route: <sip:proxy-call-id=32c76cef-e73c-4911-98d0-
e2d2bb6fec77@192.168.1.6:5061;transport=tls;lr>
Allow: INVITE,ACK,BYE,CANCEL,INFO,OPTIONS,REFER,SUBSCRIBE,NOTIFY
User-Agent: TANDBERG/4352 (X8.10.2-b2bua-1.0)
```

Having analyzed the diagnostic logging which isolated the problem to the Expressway-C and a specific error (404 Not Found), you can focus on what would cause this type of behavior. Some things to consider are the following:

1. Calls are moved in and out of Zones on the Expressway by way of Search Rules.
2. The Expressways use logic called Preloaded SIP routes support which processes SIP INVITE requests that contain Router header. This value can be turned On or Off in the Zones (Traversal server, Traversal client, Neighbor) on both the Expressway-C and Expressway-E.

You can now use the xConfiguration to view the configuration on both the Expressway-E Traversal server and Expressway-C client zones, specifically those that are set up for Hybrid Call Service Connect. In addition to the Zone configuration, you can analyze the Search Rules that are configured to pass this call through from one Zone to another. You also know that the Expressway-E did pass the call to the Expressway-C so the Traversal server zone configuration there is most likely set up correctly.

To break this down, the xConfig below tells us that the name of this zone is called **Hybrid Call Service Traversal**. It's of the **TraversalServer** zone type. It communicates to the Expressway-C over SIP TCP port **7003**.

The key piece for Hybrid Call Service is that it must have Preloaded SIP routes support On. The Expressway Web interface calls this value **Preloaded SIP routes support** whereas the xConfiguration will display it as **SIP PreloadedSipRoutes Accept**

#### Expressway-E

```
*c xConfiguration Zones Zone 7 Name: "Hybrid Call Service Traversal"
*c xConfiguration Zones Zone 7 TraversalServer Authentication Mode: "DoNotCheckCredentials"
*c xConfiguration Zones Zone 7 TraversalServer Authentication UserName: "hybridauth"
*c xConfiguration Zones Zone 7 TraversalServer Collaboration Edge: "Off"
*c xConfiguration Zones Zone 7 TraversalServer H323 H46019 Demultiplexing Mode: "Off"
*c xConfiguration Zones Zone 7 TraversalServer H323 Port: "6007"
*c xConfiguration Zones Zone 7 TraversalServer H323 Protocol: "Assent"
*c xConfiguration Zones Zone 7 TraversalServer Registrations: "Allow"
*c xConfiguration Zones Zone 7 TraversalServer SIP Media AesGcm Support: "Off"
*c xConfiguration Zones Zone 7 TraversalServer SIP Media Encryption Mode: "Auto"
```

```

*c xConfiguration Zones Zone 7 TraversalServer SIP Media ICE Support: "Off"
*c xConfiguration Zones Zone 7 TraversalServer SIP Multistream Mode: "On"
*c xConfiguration Zones Zone 7 TraversalServer SIP ParameterPreservation Mode: "On"
*c xConfiguration Zones Zone 7 TraversalServer SIP Poison Mode: "Off"
*c xConfiguration Zones Zone 7 TraversalServer SIP Port: "7003"
*c xConfiguration Zones Zone 7 TraversalServer SIP PreloadedSipRoutes Accept: "On"
*c xConfiguration Zones Zone 7 TraversalServer SIP Protocol: "Assent"
*c xConfiguration Zones Zone 7 TraversalServer SIP TLS Verify Mode: "On"
*c xConfiguration Zones Zone 7 TraversalServer SIP TLS Verify Subject Name: "rtp12-tpdmz-118-
VCSC.rtp.ciscotac.net"
*c xConfiguration Zones Zone 7 TraversalServer SIP Transport: "TLS"
*c xConfiguration Zones Zone 7 TraversalServer TCPProbe KeepAliveInterval: "20"
*c xConfiguration Zones Zone 7 TraversalServer TCPProbe RetryCount: "5"
*c xConfiguration Zones Zone 7 TraversalServer TCPProbe RetryInterval: "2"
*c xConfiguration Zones Zone 7 TraversalServer UDPPProbe KeepAliveInterval: "20"
*c xConfiguration Zones Zone 7 TraversalServer UDPPProbe RetryCount: "5"
*c xConfiguration Zones Zone 7 TraversalServer UDPPProbe RetryInterval: "2"
*c xConfiguration Zones Zone 7 Type: "TraversalServer"

```

You can also determine that this Zone has Search Rule 3 (Webex Hybrid) tied to it. Essentially the Search Rule is sending an "Any" alias that comes in through the Hybrid Call Services' DNS zone and passing it to the zone above, Hybrid Call Service Traversal. As expected, both the Search Rule and Traversal Server zone on the Expressway-E are configured correctly.

```

*c xConfiguration Zones Policy SearchRules Rule 3 Authentication: "No"
*c xConfiguration Zones Policy SearchRules Rule 3 Description: "Calls to VCS-C"
*c xConfiguration Zones Policy SearchRules Rule 3 Mode: "AnyAlias"
*c xConfiguration Zones Policy SearchRules Rule 3 Name: "Webex Hybrid"
*c xConfiguration Zones Policy SearchRules Rule 3 Pattern Behavior: "Strip"
*c xConfiguration Zones Policy SearchRules Rule 3 Pattern Replace:
*c xConfiguration Zones Policy SearchRules Rule 3 Pattern String:
*c xConfiguration Zones Policy SearchRules Rule 3 Pattern Type: "Prefix"
*c xConfiguration Zones Policy SearchRules Rule 3 Priority: "15"
*c xConfiguration Zones Policy SearchRules Rule 3 Progress: "Stop"
*c xConfiguration Zones Policy SearchRules Rule 3 Protocol: "SIP"
*c xConfiguration Zones Policy SearchRules Rule 3 SIPTrafficType: "Any"
*c xConfiguration Zones Policy SearchRules Rule 3 Source Mode: "Named"
*c xConfiguration Zones Policy SearchRules Rule 3 Source Name: "Hybrid Call Services DNS"
*c xConfiguration Zones Policy SearchRules Rule 3 State: "Enabled"
*c xConfiguration Zones Policy SearchRules Rule 3 SystemGenerated: "No"
*c xConfiguration Zones Policy SearchRules Rule 3 Target Name: "Hybrid Call Service Traversal"
*c xConfiguration Zones Policy SearchRules Rule 3 Target SIPVariant: "Any"
*c xConfiguration Zones Policy SearchRules Rule 3 Target Type: "Zone"

```

If you focus on the xConfiguration of the Expressway-C, you can start by looking for the Traversal Client zone for Webex Hybrid. One easy way to find it is to search on the port number you learned from the Expressway-E xConfiguration (**SIP Port: "7003"**). This helps you quickly identify the correct Zone in the xConfiguration.

As before, you can learn the Zone Name (Hybrid Call Service Traversal), the Type (Traversal Client), and what has been configured for the SIP PreloadedSipRoutes Accept (Preloaded SIP routes support). As you can see from this xConfiguration, this value is set to Off. Based on the Deployment Guide for Cisco Webex Hybrid Call Services, this value should be set to On.

Additionally, if we check the definition of the Preloaded SIP routes support we can see clearly that the Expressway-C should REJECT a message if this value is set to Off AND the INVITE contains a route header: **"Switch Preloaded SIP routes support Off if you want the zone to reject SIP INVITE requests containing this header."**

## Expressway-C

```
*c xConfiguration Zones Zone 6 Name: "Hybrid Call Service Traversal"
*c xConfiguration Zones Zone 6 TraversalClient Accept Delegated Credential Checks: "Off"
*c xConfiguration Zones Zone 6 TraversalClient Authentication Mode: "DoNotCheckCredentials"
*c xConfiguration Zones Zone 6 TraversalClient Authentication Password:
"{cipher}qeh8eq+fuVY1GHGgRLder/1lYDd76O/6KrHGA7g8bJs="
*c xConfiguration Zones Zone 6 TraversalClient Authentication UserName: "hybridauth"
*c xConfiguration Zones Zone 6 TraversalClient Collaboration Edge: "Off"
*c xConfiguration Zones Zone 6 TraversalClient H323 Port: "1719"
*c xConfiguration Zones Zone 6 TraversalClient H323 Protocol: "Assent"
*c xConfiguration Zones Zone 6 TraversalClient Peer 1 Address: "amer-expressway01.ciscotac.net"
*c xConfiguration Zones Zone 6 TraversalClient Peer 2 Address:
*c xConfiguration Zones Zone 6 TraversalClient Peer 3 Address:
*c xConfiguration Zones Zone 6 TraversalClient Peer 4 Address:
*c xConfiguration Zones Zone 6 TraversalClient Peer 5 Address:
*c xConfiguration Zones Zone 6 TraversalClient Peer 6 Address:
*c xConfiguration Zones Zone 6 TraversalClient Registrations: "Allow"
*c xConfiguration Zones Zone 6 TraversalClient RetryInterval: "120"
*c xConfiguration Zones Zone 6 TraversalClient SIP Media AesGcm Support: "Off"
*c xConfiguration Zones Zone 6 TraversalClient SIP Media Encryption Mode: "Auto"
*c xConfiguration Zones Zone 6 TraversalClient SIP Media ICE Support: "Off"
*c xConfiguration Zones Zone 6 TraversalClient SIP Multistream Mode: "On"
*c xConfiguration Zones Zone 6 TraversalClient SIP ParameterPreservation Mode: "On"
*c xConfiguration Zones Zone 6 TraversalClient SIP Poison Mode: "Off"
*c xConfiguration Zones Zone 6 TraversalClient SIP Port: "7003"
*c xConfiguration Zones Zone 6 TraversalClient SIP PreloadedSipRoutes Accept: "Off"
*c xConfiguration Zones Zone 6 TraversalClient SIP Protocol: "Assent"
*c xConfiguration Zones Zone 6 TraversalClient SIP TLS Verify Mode: "On"
*c xConfiguration Zones Zone 6 TraversalClient SIP TURN Server Address:
*c xConfiguration Zones Zone 6 TraversalClient SIP TURN Server Port:
*c xConfiguration Zones Zone 6 TraversalClient SIP Transport: "TLS"
*c xConfiguration Zones Zone 6 Type: "TraversalClient"
```

At this point, you've isolated the problem to a misconfiguration of the Expressway-C Traversal client zone configuration. You must switch the Preloaded SIP routes support to On.

## Solution

To properly set the Preloaded SIP routes support:

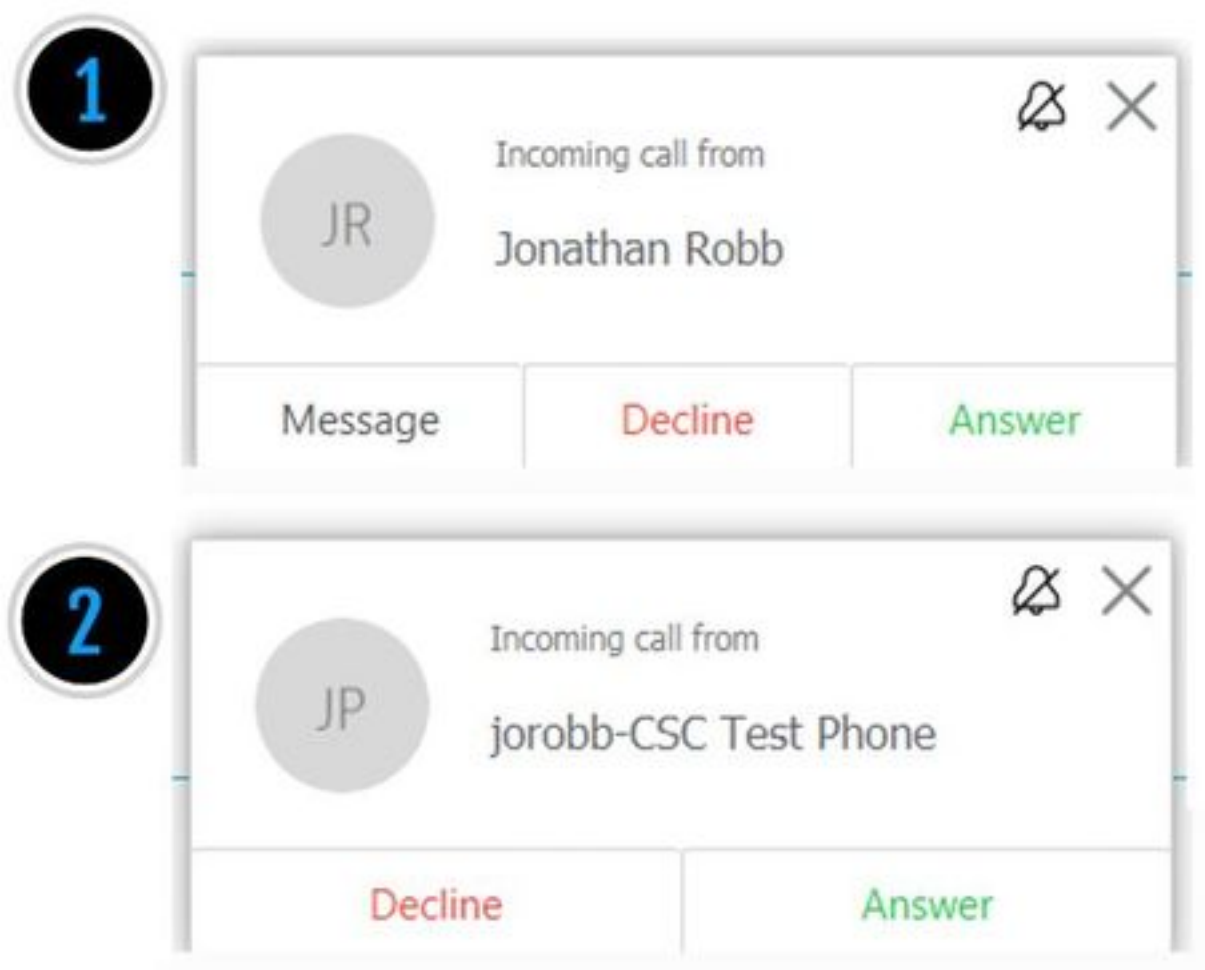
1. Log into the Expressway-C
2. Navigate to **Configuration > Zones > Zones**
3. Select the Hybrid Call Service Traversal client zone (naming will vary customer to customer)
4. Set the **Preloaded SIP routes support** to **On**
5. Select **Save**

**Note:** While this scenario demonstrated the failure on the Expressway-C, the same diagnostic logging errors could be observed on the Expressway-E if the **Preloaded SIP routes support** was Off on the Webex Hybrid Call Traversal Server zone. In that event you would have never seen the call reach the Expressway-C and the Expressway-E would have been responsible for Rejecting the call and sending the 404 Not Found.

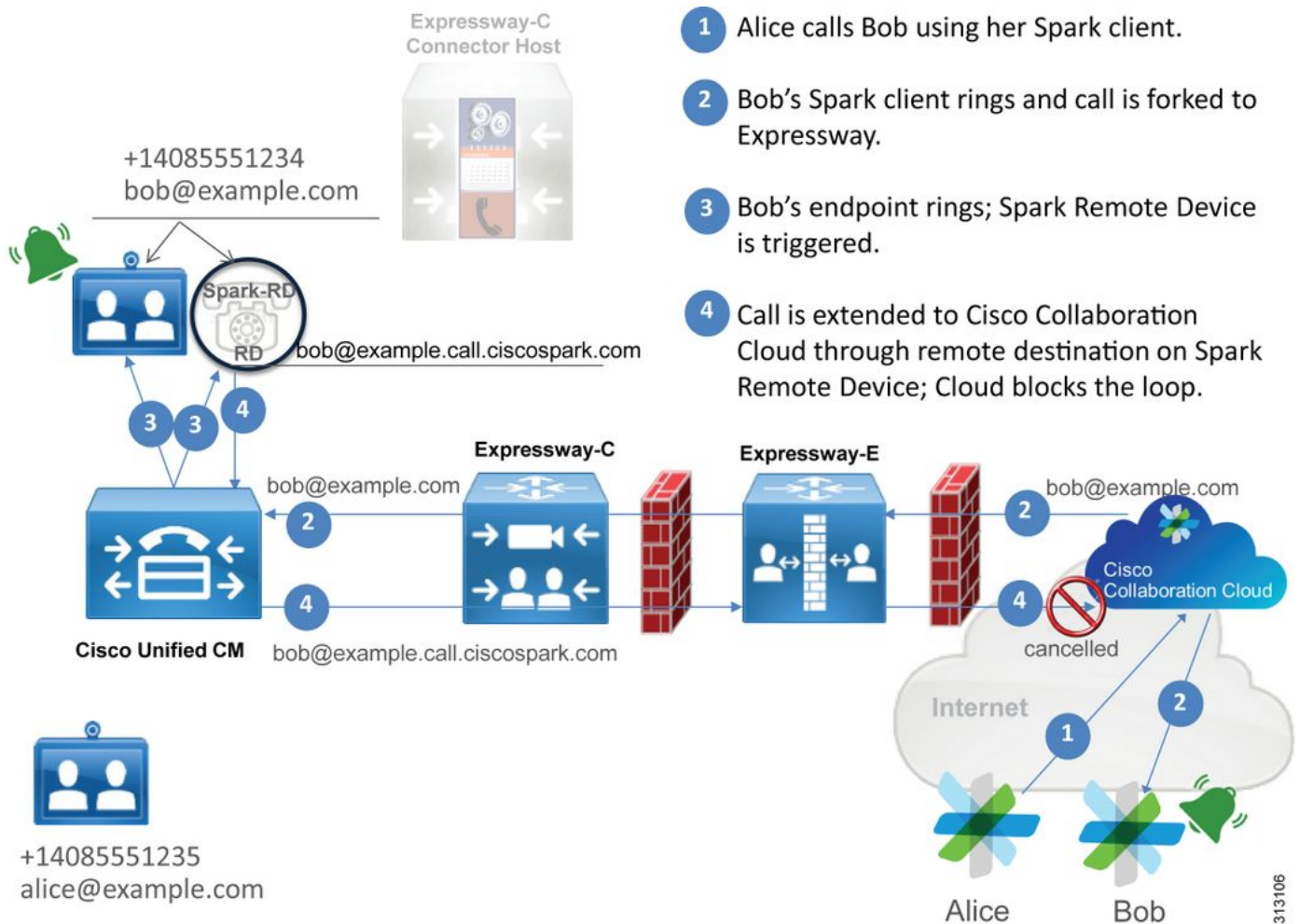
## Issue 5. Cisco Webex app is receiving two call notifications (toasts)

This particular issue happens to be the only inbound calling scenario that doesn't result in the call dropping. For this issue, the person receiving the call (called party) is receiving two notifications (toasts) in the Cisco Webex app from the person who had placed the call (calling party). The first notification is generated from Cisco Webex and the second notification comes from the on-

premises infrastructure. Below are samples of the two notifications that are received as shown in the image.



The first notification (toast) is from the person who is initiating the call (calling party) from the Cisco Webex side. The calling ID in this instance is the Display Name of the user initiating that call. The second notification (toast) is coming from the on premises CTI or Cisco Webex RD that is assigned to the user who is making the call. At first, this behavior seems peculiar. However, if you review the inbound calling diagram (from the Cisco Webex Hybrid Call Design Guide), the behavior makes more sense as shown in the image.



From the illustration, you can see the Alice is calling Bob from her Cisco Webex app and that the call is being forked down to the premises. This call should match the Directory URI that is assigned to Bob's phone. The problem is that with this design, the Directory URI is also assigned to his CTI-RD or Cisco Webex RD. Therefore, when the call is offered to the CTI-RD or Cisco Webex RD, the call is sent back out to Cisco Webex because the device has a Remote Destination configured for bob@example.call.ciscospark.com. The way Cisco Webex handles this situation is that it cancels the particular call leg.

For Cisco Webex to properly cancel the call leg, Cisco Webex initially needed to put a parameter in the SIP header which it would look for to cancel that given leg. The parameter Cisco Webex inserts into the SIP INVITE is called "**call-type=squared**" and this value is entered into the Contact header. If this value is stripped from the message, Cisco Webex does not understand how to cancel the call.

With this information, you can revisit the scenario presented earlier where the user's Cisco Webex app was receiving two notifications (toasts) when Cisco Webex user Jonathan Robb was making a call. To troubleshoot this type of problem, you're always going to need to collect diagnostic logging off the Expressway-C and Expressway-E. As a starting point, you can review the Expressway-E logs to determine that the SIP INVITE does in fact have the **call-type=squared** value present in the Contact header of the initial Cisco Webex INVITE sent inbound. This will ensure that the firewall is not manipulating the message in any way. Below is a sample snippet of the INVITE coming inbound to the Expressway-E from this scenario.

```
2017-09-19T14:01:48.140-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 18:01:48,140"
Module="network.sip" Level="DEBUG": Action="Received" Local-ip="172.16.2.2" Local-port="5062"
Src-ip="146.20.193.73" Src-port="40342" Msg-Hash="11658696457333185909"
```

```
SIPMSG:
|INVITE sip:pstojano-test@rtp.ciscotac.net SIP/2.0
Via: SIP/2.0/TLS 192.168.5.164:5062;branch=z9hG4bK564cd36d87f3417513c9b559dc666f71,SIP/2.0/TLS
127.0.0.1:5070;branch=z9hG4bK-3237-5c5060d07ecc546a0bb861ef52a5f507;rport=43306
Call-ID: 6bc0ca8210c0b48df69f38057ec1e48b@127.0.0.1
CSeq: 1 INVITE
Contact: "l2sip-UA" <sip:l2sip-UA@l2sip-cfa-01.wbx2.com:5062;transport=tls>;call-type=squared
<-- Webex inserted value
From: "Jonathan Robb" <sip:jorobb@dmzlab.call.ciscospark.com>;tag=540300020
To: <sip:pstojano-test@rtp.ciscotac.net>
Max-Forwards: 70
Route: <sip:l2sip@64.102.241.236:5062;transport=tls;lr>, <sip:cucm.rtp.ciscotac.net;lr>
```

The Contact header has the **call-type=squared** value present. At this point, the call must route through the Expressway and be sent out of the Webex Hybrid Traversal Server zone. We can search the Expressway-E logs to determine how the call was sent out of the Expressway-E. This will give us an idea if the Expressway-E is manipulating the INVITE in any way.

```
2017-09-19T14:01:48.468-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 18:01:48,468"
Module="network.sip" Level="DEBUG": Action="Sent" Local-ip="192.168.1.6" Local-port="7003" Dst-
ip="192.168.1.5" Dst-port="26686" Msg-Hash="1847271284712495612"
```

```
SIPMSG:
INVITE sip:pstojano-test@rtp.ciscotac.net SIP/2.0
Via: SIP/2.0/TLS 192.168.1.6:7003;egress-
zone=HybridCallServiceTraversal;branch=z9hG4bKec916b02b6d469abad0a30b93753f4b0859;proxy-call-
id=d7372034-85d1-41f8-af84-dffed6d1a9a9;rport
Via: SIP/2.0/TLS 192.168.1.6:5073;branch=z9hG4bKd91699370129b4c10d09e269525de00c2;x-cisco-local-
service=nettle;received=192.168.1.6;rport=43119;ingress-zone=DefaultZone
Via: SIP/2.0/TLS 64.102.241.236:5061;egress-
zone=DefaultZone;branch=z9hG4bK52aac9a181192566e01b98ae0280bdf858.0e65cdfe078cabb269eecb6bce132
8be;proxy-call-id=ec51e8da-e1a3-4210-95c9-494d12debc8;received=172.16.2.2;rport=25016
Via: SIP/2.0/TLS
192.168.5.164:5062;branch=z9hG4bK564cd36d87f3417513c9b559dc666f71;received=146.20.193.73;rport=4
0342;ingress-zone=HybridCallServicesDNS
Via: SIP/2.0/TLS 127.0.0.1:5070;branch=z9hG4bK-3237-5c5060d07ecc546a0bb861ef52a5f507;rport=43306
Call-ID: 6bc0ca8210c0b48df69f38057ec1e48b@127.0.0.1
CSeq: 1 INVITE
Contact: <sip:192.168.1.6:5073;transport=tls> <-- Webex inserted value is now missing
From: "Jonathan Robb" <sip:jorobb@dmzlab.call.ciscospark.com>;tag=540300020
To: <sip:pstojano-test@rtp.ciscotac.net>
Max-Forwards: 15
Route: <sip:cucm.rtp.ciscotac.net;lr>
```

When reviewing this SIP INVITE that is being sent from the Expressway-E to the Expressway-C, note that the Contact header is missing the **call-type=squared**. One other thing to point out is that in line item 4, you can see that the egress-zone is equal to **HybridCallServiceTraversal**. You can now conclude that the reason the Cisco Webex app is getting a second notification (toast) when dialed is because of the Expressway-E stripping the **call-type=squared** tag from the SIP INVITE Contact header. The question to answer is what could be causing this stripped header.

The call must route through the Hybrid Call Service Traversal you set up on the Expressway, so that is a good place to start the investigation. If you have the xConfiguration, you can see how this zone has been configured. To identify the Zone in the xConfiguration, you can simply use the name recorded in the Via line that gets printed in the logs. You can see above it was called egress-zone=HybridCallServiceTraversal. When this name is printed into the Via line of the SIP Header, the spaces are removed. The real zone name from the xConfiguration perspective would have spaces and is formatted at Hybrid Call Service Traversal.

```
*c xConfiguration Zones Zone 7 TraversalServer Authentication Mode: "DoNotCheckCredentials"
```

```

*c xConfiguration Zones Zone 7 TraversalServer Authentication UserName: "hybridauth"
*c xConfiguration Zones Zone 7 TraversalServer Collaboration Edge: "Off"
*c xConfiguration Zones Zone 7 TraversalServer H323 H46019 Demultiplexing Mode: "Off"
*c xConfiguration Zones Zone 7 TraversalServer H323 Port: "6007"
*c xConfiguration Zones Zone 7 TraversalServer H323 Protocol: "Assent"
*c xConfiguration Zones Zone 7 TraversalServer Registrations: "Allow"
*c xConfiguration Zones Zone 7 TraversalServer SIP Media AesGcm Support: "Off"
*c xConfiguration Zones Zone 7 TraversalServer SIP Media Encryption Mode: "Auto"
*c xConfiguration Zones Zone 7 TraversalServer SIP Media ICE Support: "Off"
*c xConfiguration Zones Zone 7 TraversalServer SIP Multistream Mode: "On"
*c xConfiguration Zones Zone 7 TraversalServer SIP ParameterPreservation Mode: "Off" <--
Possible Suspect Value
*c xConfiguration Zones Zone 7 TraversalServer SIP Poison Mode: "Off"
*c xConfiguration Zones Zone 7 TraversalServer SIP Port: "7003"
*c xConfiguration Zones Zone 7 TraversalServer SIP PreloadedSipRoutes Accept: "On" <--
Possible Suspect Value
*c xConfiguration Zones Zone 7 TraversalServer SIP Protocol: "Assent"
*c xConfiguration Zones Zone 7 TraversalServer SIP TLS Verify Mode: "On"
*c xConfiguration Zones Zone 7 TraversalServer SIP TLS Verify Subject Name: "rtp12-tpdmz-118-
VCSC.rtp.ciscotac.net"
*c xConfiguration Zones Zone 7 TraversalServer SIP Transport: "TLS"
*c xConfiguration Zones Zone 7 TraversalServer TCPProbe KeepAliveInterval: "20"
*c xConfiguration Zones Zone 7 TraversalServer TCPProbe RetryCount: "5"
*c xConfiguration Zones Zone 7 TraversalServer TCPProbe RetryInterval: "2"
*c xConfiguration Zones Zone 7 TraversalServer UDPProbe KeepAliveInterval: "20"
*c xConfiguration Zones Zone 7 TraversalServer UDPProbe RetryCount: "5"
*c xConfiguration Zones Zone 7 TraversalServer UDPProbe RetryInterval: "2"
*c xConfiguration Zones Zone 7 Name: "Hybrid Call Service Traversal"

```

With the settings identified for the Hybrid Call Service Traversal, you can look for potential settings that stand out, such as:

- SIP PreloadedSIPRoutes Accept: On
- SIP ParameterPreservatoin Mode: Off

Using the web interface of any Expressway, you can see what the definition of these values are and what they do.

## Preloaded SIP Routes support

Switch Preloaded SIP routes support On to enable this zone to process SIP INVITE requests that contain the Route header.

Switch Preloaded SIP routes support Off if you want the zone to reject SIP INVITE requests containing this header.

## SIP parameter preservation

Determines whether the Expressway's B2BUA preserves or rewrites the parameters in SIP requests routed via this zone.

**On**preserves the SIP Request URI and Contact parameters of requests routing between this zone and the B2BUA.

**Off**allows the B2BUA to rewrite the SIP Request URI and Contact parameters of requests routing between this zone and the B2BUA, if necessary.

Based off these definitions, the xConfiguration, and that the **call-type=squared** value is placed in the "Contact" header of the SIP INVITE, you can conclude that having the SIP parameter preservation value Off on the Hybrid Call Service Traversal zone is the reason that tag is getting stripped and the Cisco Webex app is getting double ring notifications.

## Solution

To preserve the `call-type=squared` value in the Contact header of the SIP INVITE, you must ensure that the Expressways support SIP parameter preservation for all Zones involved in handling the call:

1. Log into the Expressway-E
2. Navigate to **Configuration > Zones > Zones**
3. Select the Zone that's being used for the Hybrid Traversal Server
4. Set the SIP parameter preservation value to **On**
5. Save the settings.

#####

Note: In this example scenario it was the Webex Hybrid Traversal Server zone on the Expressway-E that was misconfigured. Keep in mind that it is entirely possible for the SIP parameter preservation value to be set to Off on the Webex Hybrid Traversal client or CUCM neighbor zones. Both of these configurations would be done on the Expressway-C. If that were the case you could expect that the Expressway-E would have sent the **call-type=squared** value to the Expressway-C and it would have been the Expressway-C stripping it off.

## Outbound: On-Premises to Cisco Webex

Almost every call failure involving outbound on-premises to Cisco Webex results in the same reported symptom: "When I call from my Unified CM-registered phone to another user who is enabled for Call Service Connect, their on-premises phone rings but their Cisco Webex app does not." To troubleshoot this scenario, it's important to understand both the call flow and logic that are occurring when this type of call is being placed.

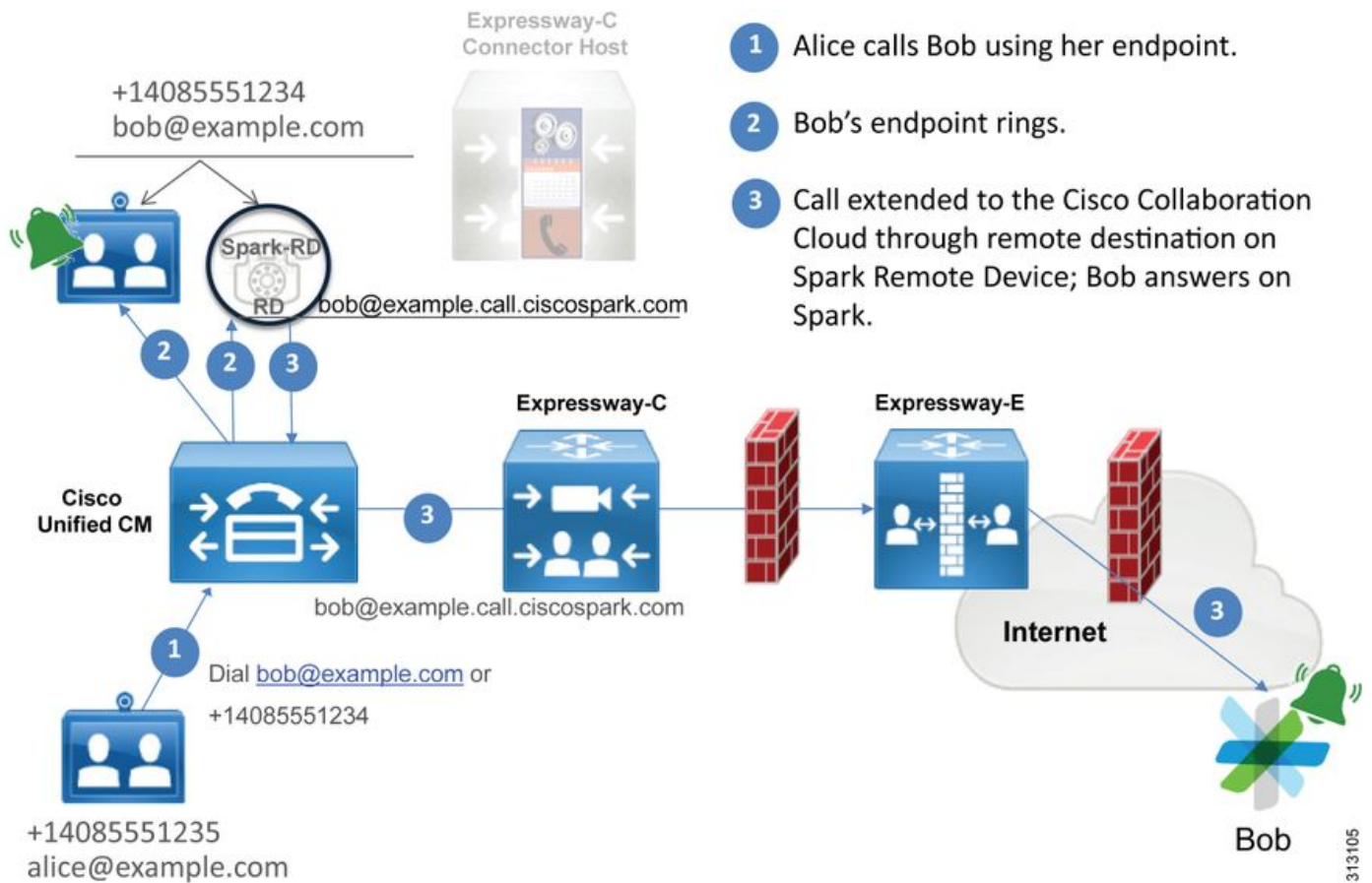
### High Level Logic Flow

1. User A makes a call from their on-premises phone to the Directory URI of User B
2. User B's on-premises phone and CTI-RD/Webex-RD accept the call
3. User B's on-premises phone begins to ring
4. User B's CTI-RD/Webex-RD forks this call out to the destination of `UserB@example.call.ciscopark.com`
5. Unified CM passes this call to the Expressway-C
6. Expressway-C sends the call to the Expressway-E
7. Expressway-E performs a DNS lookup on the `callservice.ciscopark.com` domain
8. Expressway-E attempts to connect to the Cisco Webex environment over port 5062.
9. Expressway-E and the Cisco Webex environment begin a Mutual handshake
10. The Cisco Webex environment passes the call onto User B's available Cisco Webex app
11. User B's available Cisco Webex app begins to ring.

### Call Flow

Navigate to **User B on-prem phone > Unified CM > CTI-RD/Webex-RD > Expressway-C > Expressway-E > Cisco Webex environment > Cisco Webex app** as shown in the image.





Note: Image has been pulled from the [Cisco Webex Hybrid Design Guide](#).

## Log Analysis Tips

If you were troubleshooting a situation where the outbound forked calls to Cisco Webex were failing, you'd want to collect the Unified CM, Expressway-C, and Expressway-E logs. By having these sets of logs, you can see how the call is passing through the environment. Another quick way to understand how far the call is getting within your on-premises environment is to use the Expressway "Search History". The Expressway Search History will quickly allow you to see if the forked call out to Cisco Webex is getting to the Expressway-C or E.

### To use the Search History you can perform these:

1. Log into the Expressway-E

Place a test call

Navigate to **Status > Search History**

Verify if you see a call that has a destination address of the Webex SIP URI that should be called (user@example.call.ciscospark.com)

If the Search History does not show the call hitting the Expressway-E Search History repeat this process on the Expressway-C

Before you analyze the diagnostic logs on the Expressway, consider how to identify this call:

1. The SIP Request URI will be the Cisco Webex User's SIP Address
2. The SIP FROM field will be formatted to have the Calling Party listed as "First Name Last Name" <sip:Alias@Domain>

With this information you can search the diagnostic logs by Directory URI of Calling Party, First and Last Name of Calling Party, or Cisco Webex SIP Address of the Called Party. If you don't

have any of this information, you can do a search on "**INVITE SIP:**" which will locate all SIP calls running over the Expressway. Once you have identified the SIP INVITE for the Outbound call, you can then locate and copy the SIP **Call-ID**. After you have this you can simply search the diagnostic logs based on the **Call-ID** to see all messages that correlate to this call leg.

Here are some of the most common issues observed with outbound calls from the Unified CM-registered phone to the Cisco Webex environment when the call is made to a user who is enabled for Call Service Connect.

## Issue 1. Expressway is unable to resolve the **callservice.ciscopark.com** address

The standard operating procedure for an Expressway DNS zone is to perform a DNS lookup based on the domain that shows up on the righthand side of a Request URI. To explain this, consider an example. If the DNS Zone were to receive a call that had a Request URI of **pstojano-test@dmzlab.call.ciscopark.com**, a typical Expressway DNS Zone would perform the DNS SRV Lookup logic on **dmzlab.call.ciscopark.com** which is the right hand side of the Request URI. If the Expressway were to do this you could expect that the following lookup and response would occur.

```
_sips._tcp.dmzlab.call.ciscopark.com.  
Response: 5 10 5061 l2sip-cfa-01.wbx2.com.  
l2sip-cfa-01.wbx2.com  
Response: 146.20.193.64
```

If you look closely, you see that the SRV record response is providing a server address and port 5061, not 5062.

This means that the Mutual TLS handshake that occurs over port 5062 will not happen and a separate port is used for signaling between the Expressway and Cisco Webex. The challenge with this is that the *Deployment Guide for Cisco Webex Hybrid Call Services* doesn't explicitly call out the use of port 5061 because some environments do not allow business to business calling.

The way to work past this standard DNS Zone SRV lookup logic on the Expressway is to configure the Expressway so that it does explicit searches based on a value that you provide.

Now when analyzing this particular call, you can focus on the Expressway-E because you determined (using Search History) that the call has made it this far. Start with the first SIP INVITE that comes into the Expressway-E to see what zone it came in over, which Search Rules are being used, which Zone the call goes out, and if sent correctly to the DNS zone, what DNS lookup logic occurs.

```
2017-09-19T13:18:50.562-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 17:18:50,556"  
Module="network.sip" Level="DEBUG": Action="Received" Local-ip="192.168.1.6" Local-port="7003"  
Src-ip="192.168.1.5" Src-port="26686" Msg-Hash="4341754241544006348"  
SIPMSG:  
|INVITE sip:pstojano-test@dmzlab.call.ciscopark.com SIP/2.0  
Via: SIP/2.0/TLS 192.168.1.5:5061;egress-  
zone=HybridCallServiceTraversal;branch=z9hG4bK6d734eaf7a6d733bd1e79705b7445ebb46175.1d33be65c99c  
56898f85df813f1db3a7;proxy-call-id=47454c92-2b30-414a-b7fe-aff531296bcf;rport  
Via: SIP/2.0/TCP 192.168.1.21:5065;branch=z9hG4bK13187594dd412;received=192.168.1.21;ingress-  
zone=CUCM11  
Call-ID: 991f7e80-9c11517a-130ac-1501a8c0@192.168.1.21
```

CSeq: 101 INVITE  
Call-Info: <urn:x-cisco-remotecallinfo>;x-cisco-video-traffic-class=DESKTOP  
Remote-Party-ID: "Jonathan Robb"  
<sip:5010@rtp.ciscotac.net>;party=calling;screen=yes;privacy=off  
Contact: <sip:5010@192.168.1.21:5065;transport=tcp>;video;audio  
**From: "Jonathan Robb" <sip:5010@rtp.ciscotac.net>;tag=332677~c9cc7ddc-9592-49e8-a13c-79e26f48eebc-30106860**  
**To: <sip:pstojano-test@dmzlab.call.ciscospark.com>**  
Max-Forwards: 15  
Record-Route: <sip:proxy-call-id=47454c92-2b30-414a-b7fe-aff531296bcf@192.168.1.5:5061;transport=tls;lr>  
Record-Route: <sip:proxy-call-id=47454c92-2b30-414a-b7fe-aff531296bcf@192.168.1.5:5060;transport=tcp;lr>  
Allow: INVITE, OPTIONS, INFO, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY  
**User-Agent: Cisco-CUCM11.5**  
Expires: 180  
Date: Tue, 19 Sep 2017 17:18:50 GMT  
Supported: timer, resource-priority, replaces, X-cisco-srtp-fallback, X-cisco-original-called  
Session-Expires: 1800  
Min-SE: 1800  
Allow-Events: presence  
X-TAATag: 2272025a-ce36-49d0-8d93-cb6a5e90ffe0  
Session-ID: 75957d4fb66a13e835c10737aa332675;remote=00000000000000000000000000000000  
Cisco-Guid: 2568978048-0000065536-0000000148-0352430272  
Content-Type: application/sdp  
Content-Length: 714

<SDP Omitted>

In this SIP INVITE, you can gather up the **Request URI** (pstojano-test@dmzlab.call.ciscospark.com), the **Call-ID** (991f7e80-9c11517a-130ac-1501a8c0), **From** ("Jonathan Robb" <sip:5010@rtp.ciscotac.net>), **To** (sip:pstojano-test@dmzlab.call.ciscospark.com), and **User-Agent** (Cisco-CUCM11.5). After this INVITE is received, the Expressway must now make logic decisions to determine if it can route the call to another Zone. The Expressway will do this based on Search Rules.

```
2017-09-19T13:18:50.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 17:18:50,564"  
Module="network.search" Level="DEBUG": Detail="Search rule 'B2B calls to VCS-C' did not match  
destination alias 'pstojano-test@dmzlab.call.ciscospark.com'"  
2017-09-19T13:18:50.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 17:18:50,564"  
Module="network.search" Level="DEBUG": Detail="Search rule 'Webex Hybrid' ignored due to source  
filtering"  
2017-09-19T13:18:50.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 17:18:50,564"  
Module="network.search" Level="DEBUG": Detail="Search rule 'Calls to Webex' did not match  
destination alias 'pstojano-test@dmzlab.call.ciscospark.com'"  
2017-09-19T13:18:50.564-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 17:18:50,564"  
Module="network.search" Level="DEBUG": Detail="Considering search rule 'Webex Hybrid - to Webex  
Cloud' towards target 'Hybrid Call Services DNS' at priority '90' with alias 'pstojano-  
test@dmzlab.call.ciscospark.com'"
```

Based on the log snippet above, you can see that the Expressway-E parsed through four Search Rules, however only one (Webex Hybrid - to Webex Cloud) was considered. The Search Rule had a priority of 90 and was targeted to go to the Hybrid Call Services DNS Zone. Now that the call is being sent to a DNS Zone, you can review the DNS SRV Lookups that are occurring on the Expressway-E

```
2017-09-19T13:18:50.565-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 17:18:50,565"  
Module="network.dns" Level="DEBUG": Detail="Sending DNS query"
```

```
Name="dmzlab.call.ciscospark.com" Type="NAPTR (IPv4 and IPv6)"
2017-09-19T13:18:50.718-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 17:18:50,718"
Module="network.dns" Level="DEBUG": Detail="Sending DNS query"
Name="_sips._tcp.dmzlab.call.ciscospark.com" Type="SRV (IPv4 and IPv6)"
2017-09-19T13:18:50.795-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 17:18:50,795"
Module="network.dns" Level="DEBUG": Detail="Resolved hostname to:
['IPv4','TCP','146.20.193.64:5061'] (A/AAAA) Hostname:'l2sip-cfa-01.wbx2.com' Port:'5061'
Priority:'5' TTL:'300' Weight:'10' (SRV) Number of relevant records retrieved: 2"
```

In the snippet above, you can see that the Expressway-E performed the SRV lookup based on the right hand side on the Request URI (\_sips.\_tcp.dmzlab.call.ciscospark.com) and it has resolved to a hostname of l2sip-cfa-01.wbx2.com and port 5061. The hostname l2sip-cfa-01.wbx2.com resolves to 146.20.193.64. With this information, the next logical step the Expressway will take is to send a TCP SYN packet to 146.20.193.64 so it can try to setup the call. From the Expressway-E logging, you can review to see if this is happening.

```
2017-09-19T13:18:51.145-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 17:18:51,145"
Module="network.tcp" Level="DEBUG": Src-ip="172.16.2.2" Src-port="25010" Dst-ip="146.20.193.64"
Dst-port="5061" Detail="TCP Connecting"
2017-09-19T13:19:01.295-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 17:19:01,289"
Module="network.tcp" Level="ERROR": Src-ip="172.16.2.2" Src-port="25010" Dst-ip="146.20.193.64"
Dst-port="5061" Detail="TCP Connection Failed"
```

In the above Expressway-E diagnostic logging snippet, you can see that the Expressway-E is trying to connect to the IP 146.20.193.64 which was previously resolved over TCP port 5061 however this connection is outright failing. The same can be seen from the packet capture that was collected.

Expressway-E attempts TCP Connection

No.	Time	Source	Destination	Protocol	S Port	D Port	Length	Info
3878	2017-09-19 17:18:08.801765	68.67.59.22	172.16.2.2	TCP	25876	5061	66	25876->5061 [FIN, ACK] Seq=1 Ack=1 Win=0 Len=0 Tsva=231154828 TSecr=4109470239
3879	2017-09-19 17:18:08.801923	172.16.2.2	68.67.59.22	TCP	5061	25876	66	5061->25876 [FIN, ACK] Seq=1 Ack=1 Win=0 Len=0 Tsva=4111465862 TSecr=231154828
3882	2017-09-19 17:18:08.822153	68.67.59.22	172.16.2.2	TCP	25876	5061	66	25876->5061 [ACK] Seq=2 Ack=2 Win=0 Len=0 Tsva=231154849 TSecr=4111465862
8109	2017-09-19 17:18:25.110830	192.33.146.113	172.16.2.2	TCP	50714	5061	60	50714->5061 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
14878	2017-09-19 17:18:51.145472	172.16.2.2	146.20.193.64	TCP	25010	5061	74	25010->5061 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 Tsva=314491012 TSecr=0 Ws=128
15158	2017-09-19 17:18:52.203326	172.16.2.2	146.20.193.64	TCP	25010	5061	74	[TCP Retransmission] 25010->5061 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 Tsva=314491012 TSecr=0 Ws=128
15702	2017-09-19 17:18:54.251324	172.16.2.2	146.20.193.64	TCP	25010	5061	74	[TCP Retransmission] 25010->5061 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 Tsva=314491012 TSecr=0 Ws=128
16770	2017-09-19 17:18:55.203326	172.16.2.2	146.20.193.64	TCP	25010	5061	74	[TCP Retransmission] 25010->5061 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 Tsva=314491012 TSecr=0 Ws=128
17377	2017-09-19 17:19:01.338601	172.16.2.2	146.20.193.64	TCP	25011	5061	74	25011->5061 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 Tsva=314501195 TSecr=0 Ws=128
17846	2017-09-19 17:19:02.379327	172.16.2.2	146.20.193.64	TCP	25011	5061	74	[TCP Retransmission] 25011->5061 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 Tsva=314501195 TSecr=0 Ws=128
18425	2017-09-19 17:19:04.427323	172.16.2.2	146.20.193.64	TCP	25011	5061	74	[TCP Retransmission] 25011->5061 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 Tsva=314501195 TSecr=0 Ws=128
19459	2017-09-19 17:19:08.459332	172.16.2.2	146.20.193.64	TCP	25011	5061	74	[TCP Retransmission] 25011->5061 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 Tsva=314501195 TSecr=0 Ws=128

The Expressway-E doesn't receive a SYN-ACK so it retries the SYN packet again 3 times

Based on these results, it's clear that traffic over port 5061 is not succeeding. However, Hybrid Call Service Connect intended to use TCP port 5062, not 5061. Therefore, you need to think about why isn't the Expressway-E resolving an SRV record that would return port 5062. To attempt to answer that question, you can look for possible configuration issues on the Expressway-E Webex Hybrid DNS Zone.

```
*c xConfiguration Zones Zone 6 Name: "Hybrid Call Services DNS"
*c xConfiguration Zones Zone 6 DNS SIP Authentication Trust Mode: "Off"
*c xConfiguration Zones Zone 6 DNS SIP Default Transport: "TLS"
*c xConfiguration Zones Zone 6 DNS SIP DnsOverride Name: "ciscospark.com"
*c xConfiguration Zones Zone 6 DNS SIP DnsOverride Override: "Off"
*c xConfiguration Zones Zone 6 DNS SIP Media AesGcm Support: "Off"
*c xConfiguration Zones Zone 6 DNS SIP Media Encryption Mode: "On"
*c xConfiguration Zones Zone 6 DNS SIP Media ICE Support: "Off"
*c xConfiguration Zones Zone 6 DNS SIP ParameterPreservation Mode: "Off"
*c xConfiguration Zones Zone 6 DNS SIP Poison Mode: "Off"
*c xConfiguration Zones Zone 6 DNS SIP PreloadedSipRoutes Accept: "On"
*c xConfiguration Zones Zone 6 DNS SIP Record Route Address Type: "IP"
*c xConfiguration Zones Zone 6 DNS SIP SearchAutoResponse: "Off"
*c xConfiguration Zones Zone 6 DNS SIP TLS Verify InboundClassification: "On"
*c xConfiguration Zones Zone 6 DNS SIP TLS Verify Mode: "On"
*c xConfiguration Zones Zone 6 DNS SIP TLS Verify Subject Name: "callservice.ciscospark.com"
```

```
*c xConfiguration Zones Zone 6 DNS SIP UDP BFCP Filter Mode: "Off"
*c xConfiguration Zones Zone 6 DNS SIP UDP IX Filter Mode: "Off"
```

In the xConfiguration of the Expressway-E, you can see there are two particular values of interest that relate to DNS lookups: **DNSOverride Name** and **DNSOverride Override**. Based off this xConfiguration the DNSOverride Override is set to Off, therefore the DNSOverride Name would not take effect. To better understand what these values do, you can use the Expressway Web UI to look up the definition of the values.

### Modify DNS request (Translates to DnsOverride Override in xConfig)

Routes outbound SIP calls from this zone to a manually specified SIP domain instead of the domain in the dialed destination. This option is primarily intended for use with Cisco Webex Call Service. See [www.cisco.com/go/hybrid-services](http://www.cisco.com/go/hybrid-services).

### Domain to search for (Translates to DnsOverride Name in xConfig)

Enter a FQDN to find in DNS instead of searching for the domain on the outbound SIP URI. The original SIP URI is not affected.

Now that you have these definitions, it's clear that these values if set correctly would be entirely relevant for our DNS lookup logic. If you couple this with the statements from the Deployment Guide for Cisco Webex Hybrid Call Services, you would find that the Modify DNS Request must be set to **On** and the Domain to search for should be set to **callservice.ciscospark.com**. If you were to change these values to specify the correct information, the DNS SRV lookup logic would be entirely different. Below is a snippet of what you could expect from the Expressway-E diagnostic logging perspective

```
2017-09-19T10:18:35.048-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:18:35,048"
Module="network.dns" Level="DEBUG": Detail="Sending DNS query"
Name="_sips._tcp.callservice.ciscospark.com" Type="SRV (IPv4 and IPv6)"
2017-09-19T10:18:35.126-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:18:35,126"
Module="network.dns" Level="DEBUG": Detail="Resolved hostname to:
['IPv4','TCP','146.20.193.70:5062'] (A/AAAA) ['IPv4','TCP','146.20.193.64:5062'] (A/AAAA)
Hostname:'12sip-cfa-02.wbx2.com' Port:'5062' Priority:'5' TTL:'300' Weight:'10' (SRV)
Hostname:'12sip-cfa-01.wbx2.com' Port:'5062' Priority:'5' TTL:'300' Weight:'10' (SRV) Number of
relevant records retrieved: 4"
```

### Solution

1. Log into the Expressway-E
2. Navigate to **Configuration Zones > Zones**
3. Select the Webex Hybrid DNS Zone that has been configured
4. Set the Modify DNS request to **On**
5. Set the Domain to search for value to **callservice.ciscospark.com**
6. Save the changes

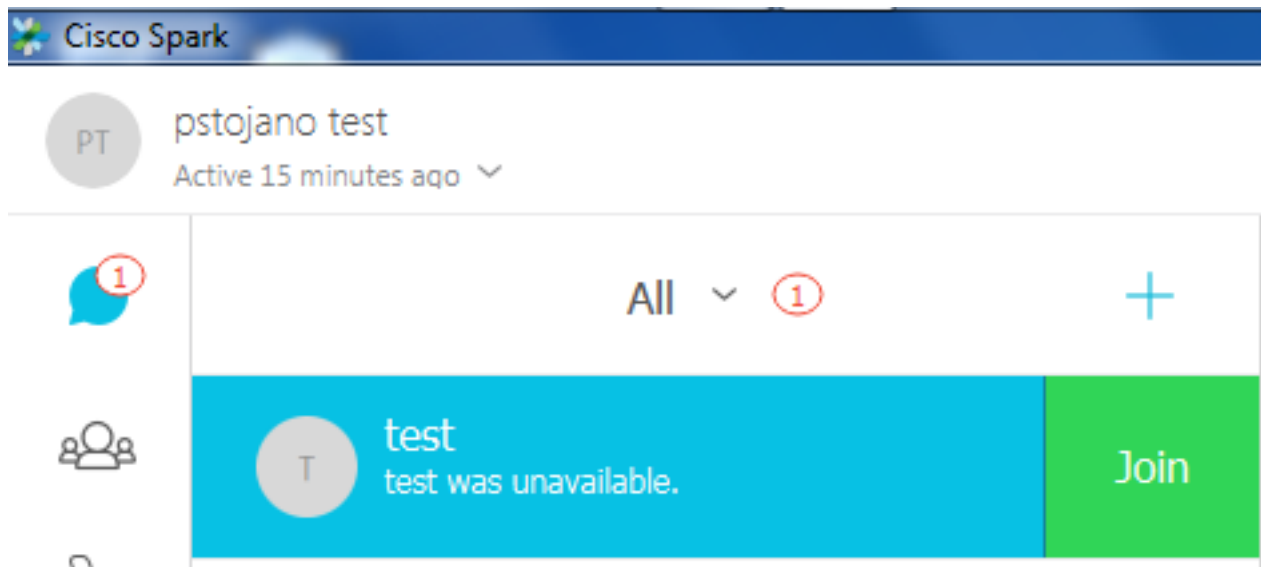
**Note:** If there is only one DNS Zone being used on the Expressway, a separate DNS Zone should be configured to be used with Hybrid Call Service that can take advantage of these values.

### Issue 2. Port 5062 is blocked outbound to Cisco Webex

One thing that is unique about the forked outbound call failures to Cisco Webex is that the called

party's Cisco Webex app will present a Join button on their app although the client never rings. Like the scenario above, for this issue you will again need to use the same tools and logging to best understand where the failure exists. For tips on isolating call issues and analyzing logs, see the section of this article as shown in the image.

Illustration of the Join button being presented



Like Outbound call Issue #1, you can start analysis at the Expressway-E diagnostic logging, because you've used the Search History on the Expressway to determine that the call is getting that far. As before, start out with the initial INVITE that comes into the Expressway-E from the Expressway-C. Remember the things you want to look for are:

1. Whether the Expressway-E receives the INVITE
2. Whether Search Rule logic passes the call to the Hybrid DNS Zone
3. Whether the DNS Zone performs the DNS Lookup and on the correct domain
4. Whether the system attempted and correctly established a TCP Handshake for Port 5062
5. Whether the Mutual TLS Handshake succeeded

```
2017-09-19T10:18:35.023-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:18:35,017"
Module="network.sip" Level="DEBUG": Action="Received" Local-ip="192.168.1.6" Local-port="7003"
Src-ip="192.168.1.5" Src-port="26513" Msg-Hash="3732376649380137405"
SIPMSG:
|INVITE sip:pstoiano-test@dmzlab.call.ciscospark.com SIP/2.0
Via: SIP/2.0/TLS 192.168.1.5:5061;egress-
zone=HybridCallServiceTraversal;branch=z9hG4bK57d8d5c823824bcd62f6ff7e09f9939482.899441b6d60c
444e4ed58951d07b5224;proxy-call-id=696f6f1c-9abe-47f3-96a4-e26f649fb76f;rport
Via: SIP/2.0/TCP 192.168.1.21:5065;branch=z9hG4bK12d4b77c97a64;received=192.168.1.21;ingress-
zone=CUCM11
Call-ID: 6a48de80-9c11273a-12d08-1501a8c0@192.168.1.21
CSeq: 101 INVITE
Call-Info: <urn:x-cisco-remotecc:callinfo>;x-cisco-video-traffic-class=DESKTOP
Remote-Party-ID: "Jonathan Robb"
<sip:5010@rtp.ciscotac.net>;party=calling;screen=yes;privacy=off
Contact: <sip:5010@192.168.1.21:5065;transport=tcp>;video;audio
From: "Jonathan Robb" <sip:5010@rtp.ciscotac.net>;tag=328867~c9cc7ddc-9592-49e8-a13c-
79e26f48eebc-30106829
To: <sip:pstoiano-test@dmzlab.call.ciscospark.com>
Max-Forwards: 15
Record-Route: <sip:proxy-call-id=696f6f1c-9abe-47f3-96a4-
e26f649fb76f@192.168.1.5:5061;transport=tls;lr>
```

```
Record-Route: <sip:proxy-call-id=696f6f1c-9abe-47f3-96a4-
e26f649fb76f@192.168.1.5:5060;transport=tcp;lr>
Allow: INVITE,OPTIONS,INFO,BYE,CANCEL,ACK,PRACK,UPDATE,REFER,SUBSCRIBE,NOTIFY
User-Agent: Cisco-CUCM11.5
Expires: 180
Date: Tue, 19 Sep 2017 14:18:34 GMT
Supported: timer,resource-priority,replaces,X-cisco-srtp-fallback,X-cisco-original-called
Session-Expires: 1800
Min-SE: 1800
Allow-Events: presence
X-TAATag: b2967a3b-93fb-4ca4-b0d7-131f75335684
Session-ID: 75957d4fb66a13e835c10737aa328865;remote=00000000000000000000000000000000
Cisco-Guid: 1783160448-0000065536-0000000126-0352430272
Content-Type: application/sdp
Content-Length: 714
<SDP Omitted>
```

As you can see in the INVITE above, the INVITE is received as normal. This is a "received" action and it is coming from the Expressway-C IP address. You can now move onto the Search Rule Logic

```
2017-09-19T10:18:35.023-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:18:35,022"
Module="network.search" Level="DEBUG": Detail="Search rule 'B2B calls to VCS-C' did not match
destination alias 'pstojano-test@dmzlab.call.ciscospark.com'"
2017-09-19T10:18:35.023-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:18:35,022"
Module="network.search" Level="DEBUG": Detail="Search rule 'Webex Hybrid' ignored due to source
filtering"
2017-09-19T10:18:35.023-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:18:35,022"
Module="network.search" Level="DEBUG": Detail="Search rule 'Calls to Webex' did not match
destination alias 'pstojano-test@dmzlab.call.ciscospark.com'"
2017-09-19T10:18:35.023-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:18:35,022"
Module="network.search" Level="DEBUG": Detail="Considering search rule 'Webex Hybrid - to Webex
Cloud' towards target 'Hybrid Call Services DNS' at priority '90' with alias 'pstojano-
test@dmzlab.call.ciscospark.com'"
```

Based on the log snippet above, you can see that the Expressway-E parsed through four Search Rules however only one (*Webex Hybrid - to Webex Cloud*) was considered. The Search Rule had a priority of 90 and was targeted to go to the *Hybrid Call Services DNS Zone*. Now that the call is being sent to a DNS Zone, you can review the DNS SRV Lookups that are occurring on the Expressway-E. All of this is entirely normal. Now you can focus on the DNS Lookup logic

```
2017-09-19T10:18:35.048-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:18:35,048"
Module="network.dns" Level="DEBUG": Detail="Sending DNS query"
Name="_sips._tcp.callservice.ciscospark.com" Type="SRV (IPv4 and IPv6)"
2017-09-19T10:18:35.126-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:18:35,126"
Module="network.dns" Level="DEBUG": Detail="Resolved hostname to:
['IPv4','TCP','146.20.193.70:5062'] (A/AAAA) ['IPv4','TCP','146.20.193.64:5062'] (A/AAAA)
Hostname:'l2sip-cfa-02.wbx2.com' Port:'5062' Priority:'5' TTL:'300' Weight:'10' (SRV)
Hostname:'l2sip-cfa-01.wbx2.com' Port:'5062' Priority:'5' TTL:'300' Weight:'10' (SRV) Number of
relevant records retrieved: 4"
```

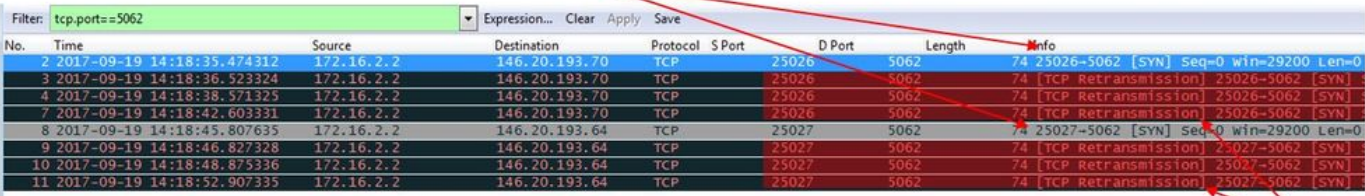
You can clearly see that in this instance, the callservice.ciscospark.com SRV record is resolved. The response is four different valid records all of which use port 5062. This is normal behavior. At this point, you can now analyze the TCP handshake that should come next. As mentioned earlier in the document, you can search the diagnostic logs for "TCP Connecting" and look for the line item that lists the Dst-port="5062". Below is a sample of what you'll see in this scenario:

```
2017-09-19T10:18:35.474-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:18:35,474"
Module="network.tcp" Level="DEBUG": Src-ip="172.16.2.2" Src-port="25026" Dst-ip="146.20.193.70"
Dst-port="5062" Detail="TCP Connecting"
```

2017-09-19T10:28:35.295-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:28:35,289"  
Module="network.tcp" Level="ERROR": Src-ip="172.16.2.2" Src-port="25026" Dst-ip="146.20.193.70"  
Dst-port="5062" Detail="TCP Connection Failed"

You can also use the tcpdump that was included with the diagnostic logging bundle to get some more detailed information about the TCP handshake as shown in the image.

Expressway-E attempts TCP Connection twice



No.	Time	Source	Destination	Protocol	S Port	D Port	Length	Info
2	2017-09-19 14:18:35.474312	172.16.2.2	146.20.193.70	TCP	25026	5062	74	25026->5062 [SYN] Seq=0 win=29200 Len=0
3	2017-09-19 14:18:36.522324	172.16.2.2	146.20.193.70	TCP	25026	5062	74	[TCP Retransmission] 25026->5062 [SYN]
4	2017-09-19 14:18:38.571325	172.16.2.2	146.20.193.70	TCP	25026	5062	74	[TCP Retransmission] 25026->5062 [SYN]
7	2017-09-19 14:18:42.603331	172.16.2.2	146.20.193.70	TCP	25026	5062	74	[TCP Retransmission] 25026->5062 [SYN]
8	2017-09-19 14:18:45.807635	172.16.2.2	146.20.193.64	TCP	25027	5062	74	25027->5062 [SYN] Seq=0 win=29200 Len=0
9	2017-09-19 14:18:46.827328	172.16.2.2	146.20.193.64	TCP	25027	5062	74	[TCP Retransmission] 25027->5062 [SYN]
10	2017-09-19 14:18:48.875336	172.16.2.2	146.20.193.64	TCP	25027	5062	74	[TCP Retransmission] 25027->5062 [SYN]
11	2017-09-19 14:18:52.907335	172.16.2.2	146.20.193.64	TCP	25027	5062	74	[TCP Retransmission] 25027->5062 [SYN]

The Expressway-E doesn't receive a SYN-ACK so it attempts to retransmit.

At this point, you can conclude that the Expressway-E is routing the call correctly. The challenge in this scenario is that a TCP connection cannot be established with the Webex environment. This could be happening because the Webex environment is not responding to the TCP SYN packet however that would be unlikely considering the server handling the connection is shared between many customers. The more likely cause in this scenario is some type of intermediary device (firewall, IPS, etc) is not allowing the traffic out.

### Solution

Because the issue was isolated, this data should be provided to the customer's network administrator. Additionally, if they need more information, you can take a capture off the outside interface of the edge device and/or firewall for further proof. From the Expressway perspective, there is no further action to perform since the issue doesn't reside on that device.

### Issue 3. Expressway Search rule misconfiguration

Search rule misconfiguration is one of the largest configuration related issues on the Expressways. Search rule configuration issues can be bi-directional, because you need Search rules for inbound calls and you need Search rules for outbound calls. As you walk through this issue, you'll discover that while regex issues are quite common on the Expressway, they are not always the cause of a search rule issue. In this particular segment, you will walk through an outbound call that is failing. Like all of our other outbound forked call scenarios, the symptoms remain the same:

- The Called user's Cisco Webex app presented Join button
- The Calling phone was playing a ring back
- The Called user's on-premises phone was ringing
- The Called user's Cisco Webex app never rang

Like all of the other scenarios, you will also want to leverage CUCM SDL traces along with Expressway-C and E diagnostic logs. As before, you should reference the for leveraging Search History and tips for identifying a call in the diagnostic logs. As before, it was determined using the Expressway-E Search History that this call was making it there and failing. Below is the beginning of the analysis for which we take a look at the initial SIP INVITE coming into the Expressway-E from the Expressway-C.

2017-09-25T11:26:02.959-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 15:26:02,959"



Module="network.sip" Level="DEBUG": **Action="Received"** Local-ip="192.168.1.6" Local-port="7003" Src-ip="192.168.1.5" Src-port="25675" Msg-Hash="1536984498381728689"  
SIPMSG:  
|**INVITE sip:pstojano-test@dmzlab.call.ciscospark.com** SIP/2.0  
Via: SIP/2.0/TLS 192.168.1.5:5061;egress-  
**zone=HybridCallServiceTraversal**;branch=z9hG4bK1c7bf93ff08014ca5e00bb0b5f8b184b272412.a81f2992e3863ac202a000a3dd599763;proxy-call-id=f79b8631-947b-46d4-a888-911bf0150bfe;rport  
Via: SIP/2.0/TCP 192.168.1.21:5065;branch=z9hG4bK1c8c419938648;received=192.168.1.21;ingress-  
zone=CUCM11  
**Call-ID: d58f2680-9c91200a-1c7ba-1501a8c0@192.168.1.21**  
CSeq: 101 INVITE  
Call-Info: <urn:x-cisco-remotecall:callinfo>;x-cisco-video-traffic-class=DESKTOP  
Remote-Party-ID: "Jonathan Robb"  
<sip:5010@rtp.ciscotac.net>;party=calling;screen=yes;privacy=off  
Contact: <sip:5010@192.168.1.21:5065;transport=tcp>;video;audio  
**From: "Jonathan Robb" <sip:5010@rtp.ciscotac.net>;tag=505817~c9cc7ddc-9592-49e8-a13c-79e26f48eebc-30106972**  
**To: <sip:pstojano-test@dmzlab.call.ciscospark.com>**  
Max-Forwards: 15  
Record-Route: <sip:proxy-call-id=f79b8631-947b-46d4-a888-911bf0150bfe@192.168.1.5:5061;transport=tls;lr>  
Record-Route: <sip:proxy-call-id=f79b8631-947b-46d4-a888-911bf0150bfe@192.168.1.5:5060;transport=tcp;lr>  
Allow: INVITE,OPTIONS,INFO,BYE,CANCEL,ACK,PRACK,UPDATE,REFER,SUBSCRIBE,NOTIFY  
**User-Agent: Cisco-CUCM11.5**  
Expires: 180  
Date: Mon, 25 Sep 2017 15:26:02 GMT  
Supported: timer,resource-priority,replaces,X-cisco-srtp-fallback,X-cisco-original-called  
Session-Expires: 1800  
Min-SE: 1800  
Allow-Events: presence  
X-TAATag: 8e8c014d-5d01-4581-8108-5cb096778fc5  
Session-ID: 75957d4fb66a13e835c10737aa505813;remote=00000000000000000000000000000000  
Cisco-Guid: 3582928512-0000065536-0000000240-0352430272  
Content-Type: application/sdp  
Content-Length: 714

<SDP Omitted>

Using the Call-ID (**d58f2680-9c91200a-1c7ba-1501a8c0**) from the SIP header, you can quickly search down all messages associated to this dialog. When looking at the third hit in the logs for the Call-ID, you can see that the Expressway-E immediately sends a **404 Not Found** to the Expressway-C.

2017-09-25T11:26:13.286-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 15:26:13,286"  
Module="network.sip" Level="DEBUG": **Action="Sent"** Local-ip="192.168.1.6" Local-port="7003" Dst-ip="192.168.1.5" Dst-port="25675" Msg-Hash="12372154521012287279"  
SIPMSG:  
|**SIP/2.0 404 Not Found**  
Via: SIP/2.0/TLS 192.168.1.5:5061;egress-  
zone=HybridCallServiceTraversal;branch=z9hG4bK1c7bf93ff08014ca5e00bb0b5f8b184b272412.a81f2992e3863ac202a000a3dd599763;proxy-call-id=f79b8631-947b-46d4-a888-911bf0150bfe;received=192.168.1.5;rport=25675;ingress-zone=HybridCallServiceTraversal  
Via: SIP/2.0/TCP 192.168.1.21:5065;branch=z9hG4bK1c8c419938648;received=192.168.1.21;ingress-  
zone=CUCM11  
**Call-ID: d58f2680-9c91200a-1c7ba-1501a8c0@192.168.1.21**  
CSeq: 101 INVITE  
**From: "Jonathan Robb" <sip:5010@rtp.ciscotac.net>;tag=505817~c9cc7ddc-9592-49e8-a13c-79e26f48eebc-30106972**  
**To: <sip:pstojano-test@dmzlab.call.ciscospark.com>;tag=10d2cfbc45e4373f**  
**Server: TANDBERG/4135 (X8.10.2)**  
**Warning: 399 192.168.1.6:7003 "Policy Response"**

Session-ID: 00000000000000000000000000000000;remote=75957d4fb66a13e835c10737aa505813  
Content-Length: 0

This data tells you two things:

1. The Expressway-E never attempted to send the INVITE to Cisco Webex
2. The Expressway-E was the responsible party for making the logic decision to reject the call with a 404 Not Found error.

A 404 Not Found error generally means the Expressway is not able to find the destination address. Since the Expressways use Search Rules to route calls between themselves and to different environments, start by focusing on the xConfiguration of the Expressway-E. Within this xConfiguration, you can look for the Search Rule that should pass the call out to the Webex Hybrid DNS Zone. To find the search rules configured on the Expressway from the xConfiguration perspective, you can search for "**xConfiguration Zones Policy SearchRules Rule**" By doing this, you'll see a list of Search Rule configuration for each Search Rule created on the Expressway. The number that comes after the "Rule" will increase based on the search rule that was created first being marked 1. If you're having trouble finding the search rule. you can use commonly used naming values such as "Webex" to better locate the Search Rule. Another way to identify the rule is finding the Pattern String value that is set to "**.\*@.\*\ciscospark\.com**". That is the Pattern String that is suppose to be configured. *(Assuming the Pattern String is configured correctly)*

After reviewing the xConfiguration from this scenario, you can see that Search Rule 6 is the correct rule to pass the call out to Cisco Webex.

```
*c xConfiguration Zones Policy SearchRules Rule 6 Authentication: "No"
*c xConfiguration Zones Policy SearchRules Rule 6 Description: "Outbound calls to Webex"
*c xConfiguration Zones Policy SearchRules Rule 6 Mode: "AliasPatternMatch"
*c xConfiguration Zones Policy SearchRules Rule 6 Name: "Webex Hybrid - to Webex Cloud"
*c xConfiguration Zones Policy SearchRules Rule 6 Pattern Behavior: "Leave"
*c xConfiguration Zones Policy SearchRules Rule 6 Pattern Replace:
*c xConfiguration Zones Policy SearchRules Rule 6 Pattern String: ".*@.*\ciscospark\.com"
*c xConfiguration Zones Policy SearchRules Rule 6 Pattern Type: "Regex"
*c xConfiguration Zones Policy SearchRules Rule 6 Priority: "101"
*c xConfiguration Zones Policy SearchRules Rule 6 Progress: "Stop"
*c xConfiguration Zones Policy SearchRules Rule 6 Protocol: "SIP"
*c xConfiguration Zones Policy SearchRules Rule 6 SIPTrafficType: "Any"
*c xConfiguration Zones Policy SearchRules Rule 6 Source Mode: "Named"
*c xConfiguration Zones Policy SearchRules Rule 6 Source Name: "Hybrid Call Service Traversal"
*c xConfiguration Zones Policy SearchRules Rule 6 State: "Enabled"
*c xConfiguration Zones Policy SearchRules Rule 6 SystemGenerated: "No"
*c xConfiguration Zones Policy SearchRules Rule 6 Target Name: "Hybrid Call Services DNS"
*c xConfiguration Zones Policy SearchRules Rule 6 Target SIPVariant: "Any"
*c xConfiguration Zones Policy SearchRules Rule 6 Target Type: "Zone"
```

To test this pattern, we can use the Check pattern function described in the. The important call out here is that we will want the following values configured:

### Maintenance > Tools > Check pattern

- Alias: %Request URI in the initial INVITE% (Ex: pstojano-test@dmzlab.call.ciscospark.com)
- Pattern type: Regex
- Pattern String **.\*@.\*\ciscospark\.com**
- Pattern behavior: Leave

If the Regex for the rule is set up correct, you should see the result of this Check pattern Succeed. Below is an illustration demonstrating this as shown in the image:

### Check pattern

Alias

Alias \* pstojano-test@dmzlab.call.ciscospark.com i

Pattern

Pattern type Regex i  
 Pattern string \* \*@\*.ciscosparkl.com i  
 Pattern behavior Leave i

Check pattern

#### Result

Result	Succeeded
Details	Alias matched pattern
Alias	pstojano-test@dmzlab.call.ciscospark.com

Now that you can confirm the Search rule is present and configured correctly, you can look closer at the Search logic that the Expressway is performing to determine if it is affecting the Expressway-E that is sending the 404 Not Found. Below is a sample of the search rule logic that the Expressway was performing.

```

2017-09-25T11:26:02.966-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 15:26:02,966"
Module="network.search" Level="DEBUG": Detail="Search rule 'B2B calls to VCS-C' did not match
destination alias 'pstojano-test@dmzlab.call.ciscospark.com'"
2017-09-25T11:26:02.966-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 15:26:02,966"
Module="network.search" Level="DEBUG": Detail="Search rule 'Webex Hybrid' ignored due to source
filtering"
2017-09-25T11:26:02.966-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 15:26:02,966"
Module="network.search" Level="DEBUG": Detail="Search rule 'Calls to Webex' did not match
destination alias 'pstojano-test@dmzlab.call.ciscospark.com'"
2017-09-25T11:26:02.967-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 15:26:02,967"
Module="network.search" Level="DEBUG": Detail="Considering search rule 'to DNS' towards target
'DNS' at priority '100' with alias 'pstojano-test@dmzlab.call.ciscospark.com'"

2017-09-25T11:26:02.968-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 15:26:02,968"
Module="network.dns" Level="DEBUG": Detail="Sending DNS query" Name="dmzlab.call.ciscospark.com"
Type="NAPTR (IPv4 and IPv6)"
2017-09-25T11:26:02.982-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 15:26:02,982"
Module="network.dns" Level="DEBUG": Detail="Could not resolve hostname"
2017-09-25T11:26:02.982-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 15:26:02,982"
Module="network.dns" Level="DEBUG": Detail="Sending DNS query"
Name="_sips._tcp.dmzlab.call.ciscospark.com" Type="SRV (IPv4 and IPv6)"

```

In this sample, you can see the Expressway processed four search rules. The first 3 were not considered because of various reasons, however the 4th was considered. The interesting piece of data is that immediately after consideration the Expressway jumps straight to DNS lookup logic. If you recall what we had seen in the xConfiguration the Search rule configured for Webex Hybrid was named Webex Hybrid - to Webex Cloud and it wasn't even considered in this Search rule logic above. At this point, it is worth looking into how the considered search rule (to DNS) was implemented so that you can better understand if it is impacting the use of the Webex Hybrid Search rule. To do that, you can revisit the xConfig this time looking for the Search Rule named "to DNS"

```

*c xConfiguration Zones Policy SearchRules Rule 1 Authentication: "No"
*c xConfiguration Zones Policy SearchRules Rule 1 Description:

```

```

*c xConfiguration Zones Policy SearchRules Rule 1 Mode: "AliasPatternMatch"
*c xConfiguration Zones Policy SearchRules Rule 1 Name: "to DNS"
*c xConfiguration Zones Policy SearchRules Rule 1 Pattern Behavior: "Leave"
*c xConfiguration Zones Policy SearchRules Rule 1 Pattern Replace:
*c xConfiguration Zones Policy SearchRules Rule 1 Pattern String: "(?!.*@%localdomains%.*$).*"
*c xConfiguration Zones Policy SearchRules Rule 1 Pattern Type: "Regex"
*c xConfiguration Zones Policy SearchRules Rule 1 Priority: "100"
*c xConfiguration Zones Policy SearchRules Rule 1 Progress: "Stop"
*c xConfiguration Zones Policy SearchRules Rule 1 Protocol: "Any"
*c xConfiguration Zones Policy SearchRules Rule 1 SIPTrafficType: "Any"
*c xConfiguration Zones Policy SearchRules Rule 1 Source Mode: "Any"
*c xConfiguration Zones Policy SearchRules Rule 1 Source Name: "Please Select"
*c xConfiguration Zones Policy SearchRules Rule 1 State: "Enabled"
*c xConfiguration Zones Policy SearchRules Rule 1 SystemGenerated: "No"
*c xConfiguration Zones Policy SearchRules Rule 1 Target Name: "DNS"
*c xConfiguration Zones Policy SearchRules Rule 1 Target SIPVariant: "Any"
*c xConfiguration Zones Policy SearchRules Rule 1 Target Type: "Zone"

```

After review of this Search rule, you can conclude the following:

- The pattern string would match the Cisco Webex Request URI
- The Priority is set to 100
- The Progress (Pattern behavior) is set to Stop.

What this information tells us is that the Cisco Webex Request URI being called would match this rule and if the rule was matched the Expressway would stop searching (Considering) other Search rules. With this understanding, the Rule Priority becomes a key factor. The way the Expressway Search rule priority works is the lowest priority rule is attempted first. Below is an example.

Search Rule: Local

Pattern behavior: Continue

Priority 1

Search Rule: Neighbor

Pattern behavior: Continue

Priority 10

Search Rule: DNS

Pattern behavior: Stop

Priority 50

In this example, the Search rule named Local (1) would be attempted first and if a match was found it would move to Search rule Neighbor (10) because of the Pattern behavior being set to Continue. If search rule Neighbor wasn't matched, it will still continue to Search Rule DNS (50) and consider that last. If Search Rule DNS was matched, the search would stop regardless of whether there was another Search Rule with a priority higher than 50, because the Pattern behavior was set to Stop.

With this understanding, you can take a look at the Search Rule priorities between the **"to DNS"** and **"Webex Hybrid - to Webex Cloud"** rules.

```
*c xConfiguration Zones Policy SearchRules Rule 1 Name: "to DNS"
*c xConfiguration Zones Policy SearchRules Rule 1 Priority: "100"
*c xConfiguration Zones Policy SearchRules Rule 1 Progress: "Stop"

*c xConfiguration Zones Policy SearchRules Rule 6 Name: "Webex Hybrid - to Webex Cloud"
*c xConfiguration Zones Policy SearchRules Rule 6 Priority: "101"
*c xConfiguration Zones Policy SearchRules Rule 6 Progress: "Stop"
```

Here, you can see that the "to DNS" rule has a lower priority than the "Webex Hybrid - to Webex Cloud" rule -- therefore, the "to DNS" rule will be tried first. Given that the Pattern behavior (Progress) is set to Stop, the Expressway-E never considers the Webex Hybrid - to Webex Cloud rule and the call ultimately fails.

## Solution

This type of problem is increasingly common with Hybrid Call Service Connect. Many times when the solution is deployed, people create a high priority rule to use for the Cisco Webex searches. Many times this rule that is created isn't getting invoked because of existing lower priority rules are being matched and it results in a failure. This issue happens on both inbound and outbound calls to Cisco Webex. To resolve this, you'll need to follow these steps:

1. Log into the Expressway-E
2. Navigate to **Configuration > Dial Plan > Search rules**
3. Find the Webex Hybrid Search rule and click it (*Ex: Name: Webex Hybrid - to Webex Cloud*)
4. Set the Priority value to something lower than other Search rules, yet high enough so that it won't impact others. (*Ex: Priority: 99*)

The general rule of thumb with Search rules is the more specific the Pattern string, the lower it can be placed in the Search rule priority list. Generally a DNS Zone is configured with a Pattern string that is going to catch anything that is not a local domain and send it to the Internet. Due to this, we recommend that you set that type of Search rule to a high priority so it's invoked last.

## Issue 4. Expressway CPL misconfiguration

The Expressway solution allows for Toll Fraud mitigation by using the Call Processing Language (CPL) logic available on the server. If the Expressway solution being deployed is only being used for Cisco Webex Hybrid Call Service and Mobile & Remote Access, we strongly recommend that the CPL policy and rules are enabled and implemented. While the CPL configuration on the Expressway for Cisco Webex Hybrid is fairly straightforward, if misconfigured it can easily block call attempts from happening. The scenarios below show you how to use the diagnostic logging to identify a CPL misconfiguration.

Like all of other outbound forked call scenarios, the symptoms remained the same:

- The called user's Cisco Webex app presented a Join button
- The calling phone was playing a ring back
- The called user's on-premises phone was ringing
- The called user's app never rang

Like all of the other scenarios, you can use the CUCM SDL traces along with Expressway-C and E diagnostic logs. As before, you should reference the for using Search History and tips for identifying a call in the diagnostic logs. As before, it was determined using the Expressway-E Search History that this call was arriving there and failing. Below is the beginning of the analysis in

which you can take a look at the initial SIP INVITE coming into the Expressway-E from the Expressway-C.

```
2017-09-25T16:54:43.722-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 20:54:43,722"
Module="network.sip" Level="DEBUG": Action="Received" Local-ip="192.168.1.6" Local-port="7003"
Src-ip="192.168.1.5" Src-port="26404" Msg-Hash="17204952472509519266"
SIPMSG:
|INVITE sip:pstojano-test@dmzlab.call.ciscospark.com SIP/2.0
Via: SIP/2.0/TLS 192.168.1.5:5061;egress-
zone=HybridCallServiceTraversal;branch=z9hG4bK781a130d234ed9aaec86834368739430283256.34216c32a0d
e36e16590bae36df388b6;proxy-call-id=3bbbf94a-082e-4088-8f5a-5ea7e82f8aac;rport
Via: SIP/2.0/TCP 192.168.1.21:5065;branch=z9hG4bK1cf344a8b117e;received=192.168.1.21;ingress-
zone=CUCM11
Call-ID: c030f100-9c916d13-1cdcb-1501a8c0@192.168.1.21
CSeq: 101 INVITE
Call-Info: <urn:x-cisco-remotecall:callinfo>;x-cisco-video-traffic-class=DESKTOP
Remote-Party-ID: "Jonathan Robb"
<sip:5010@rtp.ciscotac.net>;party=calling;screen=yes;privacy=off
Contact: <sip:5010@192.168.1.21:5065;transport=tcp>;video;audio
From: "Jonathan Robb" <sip:5010@rtp.ciscotac.net>;tag=512579~c9cc7ddc-9592-49e8-a13c-79e26f48eebc-30107000
To: <sip:pstojano-test@dmzlab.call.ciscospark.com>
Max-Forwards: 15
Record-Route: <sip:proxy-call-id=3bbbf94a-082e-4088-8f5a-5ea7e82f8aac@192.168.1.5:5061;transport=tls;lr>
Record-Route: <sip:proxy-call-id=3bbbf94a-082e-4088-8f5a-5ea7e82f8aac@192.168.1.5:5060;transport=tcp;lr>
Allow: INVITE,OPTIONS,INFO,BYE,CANCEL,ACK,PRACK,UPDATE,REFER,SUBSCRIBE,NOTIFY
User-Agent: Cisco-CUCM11.5
Expires: 180
Date: Mon, 25 Sep 2017 20:54:43 GMT
Supported: timer,resource-priority,replaces,X-cisco-srtp-fallback,X-cisco-original-called
Session-Expires: 1800
Min-SE: 1800
Allow-Events: presence
X-TAATag: 4ffffefed-0512-4067-ac8c-35828f0a1150
Session-ID: 75957d4fb66a13e835c10737aa512577;remote=00000000000000000000000000000000
Cisco-Guid: 3224432896-0000065536-000000264-0352430272
Content-Type: application/sdp
Content-Length: 714
```

<SDP Omitted>

Using the Call-ID (**c030f100-9c916d13-1cdcb-1501a8c0**) from the SIP header, you quickly search down all messages associated to this dialog. When looking at the third hit in the logs for the Call-ID, you can see that the Expressway-E immediately sends a **403 Forbidden** to the Expressway-C.

```
2017-09-25T16:54:43.727-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 20:54:43,727"
Module="network.sip" Level="DEBUG": Action="Sent" Local-ip="192.168.1.6" Local-port="7003" Dst-
ip="192.168.1.5" Dst-port="26404" Msg-Hash="9195436101110134622"
SIPMSG:
|SIP/2.0 403 Forbidden
Via: SIP/2.0/TLS 192.168.1.5:5061;egress-
zone=HybridCallServiceTraversal;branch=z9hG4bK781a130d234ed9aaec86834368739430283256.34216c32a0d
e36e16590bae36df388b6;proxy-call-id=3bbbf94a-082e-4088-8f5a-5ea7e82f8aac;received=192.168.1.5;rport=26404;ingress-zone=HybridCallServiceTraversal
Via: SIP/2.0/TCP 192.168.1.21:5065;branch=z9hG4bK1cf344a8b117e;received=192.168.1.21;ingress-
zone=CUCM11
Call-ID: c030f100-9c916d13-1cdcb-1501a8c0@192.168.1.21
CSeq: 101 INVITE
```

**From:** "Jonathan Robb" <sip:5010@rtp.ciscotac.net>;tag=512579~c9cc7ddc-9592-49e8-a13c-79e26f48eebc-30107000  
**To:** <sip:pstojano-test@dmzlab.call.ciscospark.com>;tag=64fe7f9eab37029d  
 Server: TANDBERG/4135 (X8.10.2)  
**Warning:** 399 192.168.1.6:7003 "Policy Response"  
 Session-ID: 00000000000000000000000000000000;remote=75957d4fb66a13e835c10737aa512577  
 Content-Length: 0

To understand why the Expressway-E denied this call and sent a 403 Forbidden error to the Expressway-C, you want to analyze the log entries between the 403 Forbidden and the original SIP INVITE that entered into the Expressway. By analyzing these log entries, you can typically see all the logic decisions that are being made. Note that you do not see any Search rules being invoked but do see Call Process Language (CPL) logic being invoked. Below is a snippet of that.

```

2017-09-25T16:54:43.725-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 20:54:43,725"
Module="network.cpl" Level="DEBUG": Remote-ip="192.168.1.5" Remote-port="26404" Detail="CPL:
<routed> "
2017-09-25T16:54:43.725-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 20:54:43,725"
Module="network.cpl" Level="DEBUG": Remote-ip="192.168.1.5" Remote-port="26404" Detail="CPL:
<rule-switch> "
2017-09-25T16:54:43.725-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 20:54:43,726"
Module="network.cpl" Level="DEBUG": Remote-ip="192.168.1.5" Remote-port="26404" Detail="CPL:
<rule unauthenticated-origin=".*" destination=".*@dmzlab\call\ciscospark\com.*" message-
regex=""> matched "
2017-09-25T16:54:43.725-04:00 amer-expressway01 tvcs: UTCTime="2017-09-25 20:54:43,726"
Module="network.cpl" Level="DEBUG": Remote-ip="192.168.1.5" Remote-port="26404" Detail="CPL:
<reject/> "
  
```

Based on the log analysis above, you can make the determination that the CPL is rejecting the call.

```

2017-09-25T16:54:43.725-04:00 amer-expressway01 tvcs: Event="Search Completed"
Reason="Forbidden" Service="SIP" Src-alias-type="SIP" Src-alias="5010@rtp.ciscotac.net" Dst-
alias-type="SIP" Dst-alias="sip:pstojano-test@dmzlab.call.ciscospark.com" Call-serial-
number="48c80582-ec79-4d89-82e2-e5546f35703c" Tag="4ffffefed-0512-4067-ac8c-35828f0a1150"
Detail="found:false, searchtype:INVITE, Info:Policy Response" Level="1" UTCTime="2017-09-25
20:54:43,726"
2017-09-25T16:54:43.725-04:00 amer-expressway01 tvcs: Event="Call Rejected" Service="SIP" Src-
ip="192.168.1.5" Src-port="26404" Src-alias-type="SIP"
  
```

*Note: In this situation you will not see Search rules being invoked because CPLs, FindMe, and Transforms are all processed before a Search rule*

In most circumstances, you can leverage the xConfig of the Expressway to better understand the circumstances. However, for CPLs, you cannot see the Rules that are defined, only if the policy is enabled. Below is the portion of the xConfig that shows us this Expressway-E is using the Local CPL logic.

\*c xConfiguration Policy AdministratorPolicy Mode: "LocalCPL"

To better understand the rule configuration, you need to log in to the Expressway-E and navigate to **Configuration > Call Policy > Rules** as shown in the image.

Call Policy rules			
Source	Destination	Action	Rearrange
.	*@dmzlab.call.ciscospark.com	Reject	↓

When reviewing this configuration, you can see the following is configured

Source: .\*

Destination: `.*@dmzlab\call\ciscopark\com.*`

Action: **Reject**

Compared to what's been documented in the [Cisco Webex Hybrid Call Service Deployment Guide](#), you can see that the Source and Destination were configured backwards.

Field	Setting
Source Type	<b>From address</b>
Rule applies to	<b>Unauthenticated callers</b>
Source pattern	<code>.*@example\call\ciscopark\com.*</code> , where <b>example</b> is your company's subdomain.
Destination pattern	<code>.*</code>
Action	<b>Reject</b>

## Solution

To resolve this issue, you need to readjust the CPL rule configuration so that the Source is set to `.*@%Webex_subdomain%\call\ciscopark\com.*` and the Destination Pattern is `.*`

1. Log into the Expressway-E
2. Navigate to **Configuration > Call Policy > Rules**
3. Select the rule that was setup for the Cisco Webex Hybrid Call service
4. Enter the Source Pattern as `.*@%Webex_subdomain%\call\ciscopark\com.*` (Ex: `.*@dmzlab\call\ciscopark\com.*`)
5. Enter the Destination Pattern as `.*`
6. Select **Save**

For more information on the CPL implementation for Webex Hybrid refer to the [Cisco Webex Hybrid Design Guide](#).

## Bidirectional: Cisco Webex to On-Premises or On-Premises to Cisco Webex

### Issue 1. IP Phone/Collaboration Endpoint is offering an audio codec other than G.711, G.722, or AAC-LD.

Hybrid Call Service Connect supports three different audio codecs: **G.711, G.722, and AAC-LD**. To successfully establish a call with the Cisco Webex environment, one of these audio codecs must be used. The on-premises environment can be setup to use many types of audio codecs but at the same time it can be setup to restrict them. This can happen intentionally or unintentionally by the use of custom and/or default region settings on the Unified CM. For this particular behavior, the logging patterns can differ based on the direction of the call and if the Unified CM was configured to use Early or Delayed Offer. Below are examples of a few different situations where this behavior could present itself:

1. Cisco Webex sends an inbound INVITE w/ SDP that offers G.711, G.722, or AAC-LD. The Expressway-C sends this message to Unified CM but Unified CM is configured to only allow



- G.729 for this call. So, Unified CM will reject the call due to no available codec.
- Unified CM attempts the outbound call as *Early Offer* to Cisco Webex which means the initial INVITE sent to the Expressway-C will contain SDP ONLY supporting G.729 audio. Cisco Webex then sends a 200 OK w/ SDP that zeros out the audio (*m=audio 0 RTP/SAVP*) because it doesn't support G.729. Once the Expressway-C passes this INVITE to the Unified CM, the Unified CM terminates the call because there isn't an available codec.
  - Unified CM attempts the outbound call as *Delayed Offer* to Cisco Webex which means the initial INVITE sent to the Expressway-C will not contain SDP. Cisco Webex then sends a 200 OK w/ SDP containing all the supported audio codecs Cisco Webex supports. The Expressway-C sends this 200 OK to Unified CM but Unified CM is only configured to only allow G.729 for this call. So, Unified CM will reject the call due to no available codec.

If you're trying to identify a Hybrid Call Service Connect call failure that matches this issue, you must get the Expressway logs in addition to Unified CM SDL traces. The example log snippets below match situation #2 where Unified CM is attempting the outbound call as *Early Offer*. Because we know that the call is getting out to Cisco Webex, the log analysis starts on the Expressway-E.

Here is a snippet of the initial INVITE out to Cisco Webex. You can see that the preferred audio codec is set to G.729 (Payload 18). The 101 is for DTMF and for this particular scenario isn't relevant.

```
2017-09-19T10:46:10.488-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:46:10,488"
Module="network.sip" Level="DEBUG": Action="Sent" Local-ip="172.16.2.2" Local-port="25034" Dst-
ip="146.20.193.64" Dst-port="5062" Msg-Hash="4309505007645007056"
SIPMSG:
INVITE sip:pstojano-test@dmzlab.call.ciscospark.com SIP/2.0
Via: SIP/2.0/TLS 64.102.241.236:5062;egress-
zone=HybridCallServicesDNS;branch=z9hG4bK323e6b15ad0cbbf409751f67848136fa1115;proxy-call-
id=a3a78ee2-c01b-4741-b29b-55aede256d2;rport
Via: SIP/2.0/TLS 172.16.2.2:5073;branch=z9hG4bK350703fe46645f0acddef05b35adc5c157;x-cisco-local-
service=nettle;received=172.16.2.2;rport=41511;ingress-zone=DefaultZone
Via: SIP/2.0/TLS 192.168.1.6:5061;egress-
zone=DefaultZone;branch=z9hG4bKf71f2bf47233d6ca52b579364594ac6c1114.a402e3f25603f5a77b60b17ea47d
bf72;proxy-call-id=be17a470-0bca-4ad5-8a6c-14872e007efb;received=192.168.1.6;rport=25025
Via: SIP/2.0/TLS 192.168.1.5:5061;egress-
zone=HybridCallServiceTraversal;branch=z9hG4bKf4cfd09d213a88bd2331cef0bc82b540559.494a140082bd
66357134b9eed4335df8;proxy-call-id=d4d4e950-babc-45d5-a4a7-
c60a8b17a8bd;received=192.168.1.5;rport=26513;ingress-zone=HybridCallServiceTraversal
Via: SIP/2.0/TCP 192.168.1.21:5065;branch=z9hG4bK12dd82194c4f7;received=192.168.1.21;ingress-
zone=CUCM11
Call-ID: 44bdd400-9c112db1-12d95-1501a8c0@192.168.1.21
CSeq: 101 INVITE
Remote-Party-ID: "Jonathan Robb" <sip:5010@rtp.ciscotac.net>;privacy=off;screen=no;party=calling
Contact: <sip:172.16.2.2:5073;transport=tls>;video;audio
From: "Jonathan Robb" <sip:5010@rtp.ciscotac.net>;tag=329447-c9cc7ddc-9592-49e8-a13c-
79e26f48eebc-30106833
To: <sip:pstojano-test@dmzlab.call.ciscospark.com>
Max-Forwards: 14
Record-Route: <sip:proxy-call-id=a3a78ee2-c01b-4741-b29b-
55aede256d2@64.102.241.236:5062;transport=tls;lr>
Record-Route: <sip:proxy-call-id=a3a78ee2-c01b-4741-b29b-
55aede256d2@172.16.2.2:5061;transport=tls;lr>
Allow: INVITE,ACK,BYE,CANCEL,INFO,OPTIONS,REFER,SUBSCRIBE,NOTIFY
User-Agent: TANDBERG/4352 (X8.10.2-b2bua-1.0)
Supported: X-cisco-srtp-fallback,replaces,timer
Session-Expires: 1800;refresher=uac
Min-SE: 500
```

X-TAATag: 14a0bd87-1825-4ecf-9f3d-4a23cfa69725  
Session-ID: 75957d4fb66a13e835c10737aa329445;remote=00000000000000000000000000000000  
Content-Type: application/sdp  
Content-Length: 1407

v=0  
o=tandberg 0 1 IN IP4 64.102.241.236  
s=-  
c=IN IP4 64.102.241.236  
b=AS:384  
t=0 0  
**m=audio 52668 RTP/SAVP 18 101 <-- CUCM is only supporting G.729 for this call**  
**a=rtpmap:18 G729/8000**  
a=fmtp:18 annexb=no  
a=rtpmap:101 telephone-event/8000  
a=fmtp:101 0-15  
a=crypto:1 AES\_CM\_128\_HMAC\_SHA1\_80 inline:.....  
a=crypto:2 AES\_CM\_128\_HMAC\_SHA1\_80 inline:.....  
UNENCRYPTED\_SRTCP  
a=crypto:3 AES\_CM\_128\_HMAC\_SHA1\_32 inline:.....  
a=crypto:4 AES\_CM\_128\_HMAC\_SHA1\_32 inline:.....  
UNENCRYPTED\_SRTCP  
a=sendrecv  
a=rtcp:52669 IN IP4 64.102.241.236  
m=video 52670 RTP/SAVP 126 97  
b=TIAS:384000  
a=rtpmap:126 H264/90000  
a=fmtp:126 profile-level-id=42801e;packetization-mode=1;level-asymmetry-allowed=1  
a=rtpmap:97 H264/90000  
a=fmtp:97 profile-level-id=42801e;packetization-mode=0;level-asymmetry-allowed=1  
a=rtcp-fb:\* nack pli  
a=crypto:1 AES\_CM\_128\_HMAC\_SHA1\_80 inline:.....  
a=crypto:2 AES\_CM\_128\_HMAC\_SHA1\_80 inline:.....  
UNENCRYPTED\_SRTCP  
a=crypto:3 AES\_CM\_128\_HMAC\_SHA1\_32 inline:.....  
a=crypto:4 AES\_CM\_128\_HMAC\_SHA1\_32 inline:.....  
UNENCRYPTED\_SRTCP  
a=sendrecv  
a=content:main  
a=label:11  
a=rtcp:52671 IN IP4 64.102.241.236

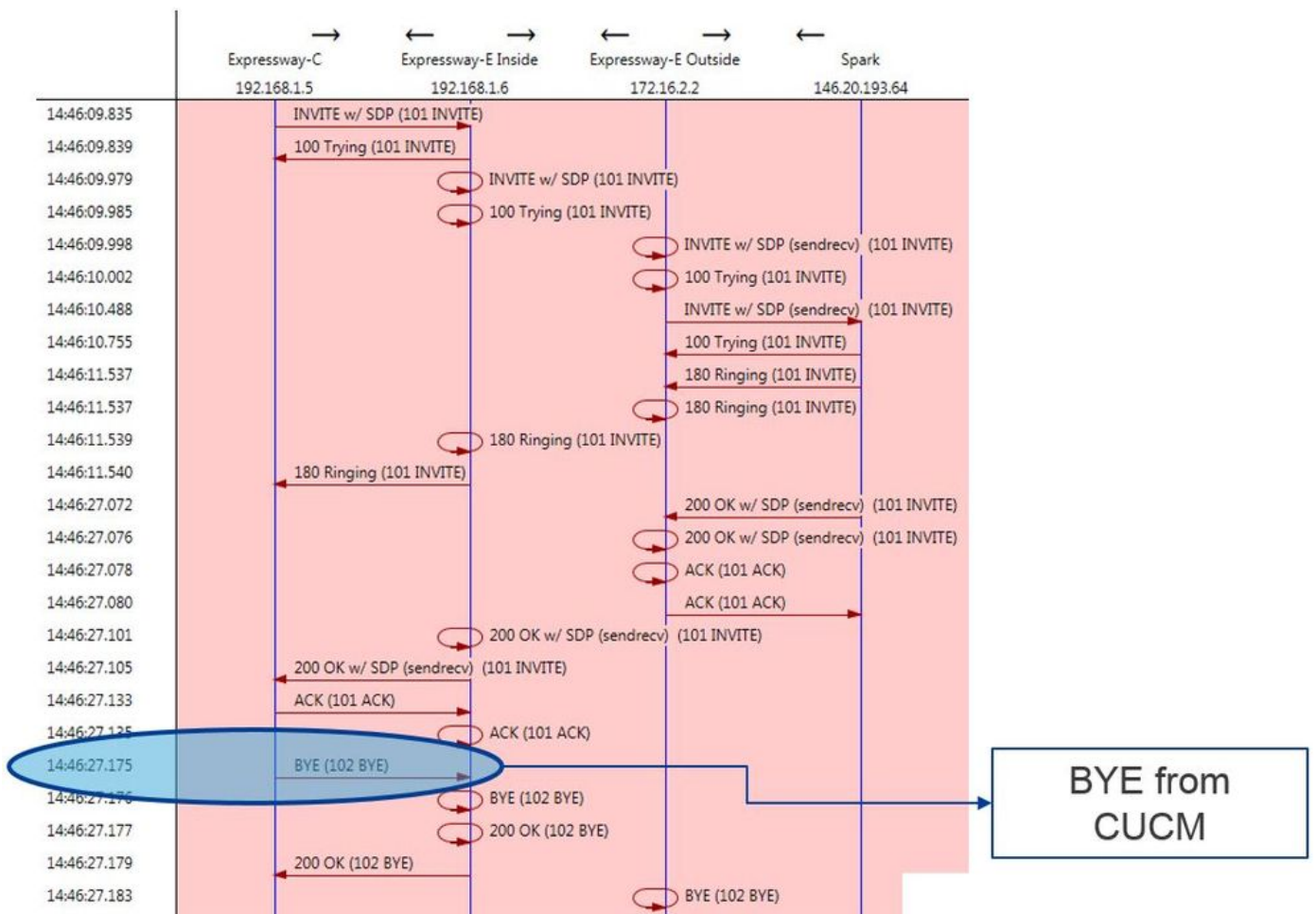
In response to this initial INVITE, Cisco Webex responds with a 200 OK message. If you take a closer look at this message, you can see that the audio codec was zeroed out. This is problematic because without an audio port assigned, the call will not be able to negotiate that stream.

2017-09-19T10:46:27.073-04:00 amer-expressway01 tvcs: UTCTime="2017-09-19 14:46:27,072"  
Module="network.sip" Level="DEBUG": **Action="Received"** Local-ip="172.16.2.2" Local-port="25034"  
**Src-ip="146.20.193.64" Src-port="5062"** Msg-Hash="5236578200712291002"  
SIPMSG:  
SIP/2.0 200 OK  
Via: SIP/2.0/TLS 64.102.241.236:5062;egress-  
zone=HybridCallServicesDNS;branch=z9hG4bK323e6b15ad0cbbf409751f67848136fa1115;proxy-call-  
id=a3a78ee2-c01b-4741-b29b-55aede256d2;rport=38245;received=192.168.5.26,SIP/2.0/TLS  
172.16.2.2:5073;branch=z9hG4bK350703fe46645f0acddef05b35adc5c157;x-cisco-local-  
service=nettle;received=172.16.2.2;rport=41511;ingress-zone=DefaultZone,SIP/2.0/TLS  
192.168.1.6:5061;egress-  
zone=DefaultZone;branch=z9hG4bKf71f2bf47233d6ca52b579364594ac6c1114.a402e3f25603f5a77b60b17ea47d  
bf72;proxy-call-id=be17a470-0bca-4ad5-8a6c-  
14872e007efb;received=192.168.1.6;rport=25025,SIP/2.0/TLS 192.168.1.5:5061;egress-  
zone=HybridCallServiceTraversal;branch=z9hG4bKf4cfd09d213a88bd2331cef0bc82b540559.494a140082bd  
66357134b9eed4335df8;proxy-call-id=d4d4e950-babc-45d5-a4a7-  
c60a8b17a8bd;received=192.168.1.5;rport=26513;ingress-

zone=HybridCallServiceTraversal,SIP/2.0/TCP  
192.168.1.21:5065;branch=z9hG4bK12dd82194c4f7;received=192.168.1.21;ingress-zone=CUCM11  
Call-ID: 44bdd400-9c112db1-12d95-1501a8c0@192.168.1.21  
CSeq: 101 INVITE  
Contact: "l2sip-UA" <sip:l2sip-UA@l2sip-cfa-01.wbx2.com:5062;transport=tls>  
**From: "Jonathan Robb" <sip:5010@rtp.ciscotac.net>;tag=329447-c9cc7ddc-9592-49e8-a13c-79e26f48eebc-30106833**  
**To: <sip:pstojano-test@dmzlab.call.ciscospark.com>;tag=1311451760**  
Record-Route: <sip:l2sip-cfa-01.wbx2.com:5062;transport=tls;lr>,<sip:proxy-call-id=a3a78ee2-c01b-4741-b29b-55aede256d2@64.102.241.236:5062;transport=tls;lr>,<sip:proxy-call-id=a3a78ee2-c01b-4741-b29b-55aede256d2@172.16.2.2:5061;transport=tls;lr>  
Allow: INVITE,ACK,CANCEL,BYE,REFER,INFO,OPTIONS,NOTIFY,SUBSCRIBE  
User-Agent: Cisco-L2SIP  
Supported: replaces  
Accept: application/sdp  
Allow-Events: kpml  
Session-ID: ed35426ed3ade6fdc3b058792333df2b;remote=75957d4fb66a13e835c10737aa329445  
Locus: 4711a33f-9d49-11e7-9bf6-dea12d0f2127  
Locus-Type: CALL  
Content-Type: application/sdp  
Content-Length: 503

v=0  
o=linus 0 1 IN IP4 146.20.193.109  
s=-  
c=IN IP4 146.20.193.109  
b=TIAS:384000  
t=0 0  
**m=audio 0 RTP/SAVP \* <-- Webex is zeroing this port out**  
m=video 33512 RTP/SAVP 108  
c=IN IP4 146.20.193.109  
b=TIAS:384000  
a=content:main  
a=sendrecv  
a=rtpmap:108 H264/90000  
a=fmtp:108 profile-level-id=42001E;packetization-mode=1;max-mps=40500;max-fs=1620;max-fps=3000;max-br=10000;max-dpb=3037;level-asymmetry-allowed=1  
a=rtcp-fb:\* nack pli  
a=crypto:1 AES\_CM\_128\_HMAC\_SHA1\_80 inline:.....  
a=label:200

You can now use TranslatorX to review the remainder of the dialog. You can see that the dialog itself completes with an ACK. The problem is immediately after the dialog completes there is a BYE that comes from the direction of the Expressway-C as shown in the image.



Here is a detailed sample of the BYE message. You can clearly see that the User-Agent is Cisco-CUCM11.5 which means that the message was generated by the Unified CM. Another thing to point out is that the Reason code is set to cause=47. The common translation for this is No resource available.

```

2017-09-19T10:46:27.175-04:00 amer-expressway01 tvcs: UTCtime="2017-09-19 14:46:27,175"
Module="network.sip" Level="DEBUG": Action="Received" Local-ip="192.168.1.6" Local-port="7003"
Src-ip="192.168.1.5" Src-port="26513" Msg-Hash="237943800593485079"
SIPMSG:
BYE sip:192.168.1.6:5071;transport=tls SIP/2.0
Via: SIP/2.0/TLS 192.168.1.5:5061;egress-
zone=HybridCallServiceTraversal;branch=z9hG4bK90a666b3461356f8cd605cec91e4538240575.494a140082bd
66357134b9eed4335df8;proxy-call-id=d4d4e950-babc-45d5-a4a7-c60a8b17a8bd;rport
Via: SIP/2.0/TCP 192.168.1.21:5065;branch=z9hG4bK12ddd10269d39;received=192.168.1.21;ingress-
zone=CUCM11
Call-ID: 44bdd400-9c112db1-12d95-1501a8c0@192.168.1.21
CSeq: 102 BYE
From: "Jonathan Robb" <sip:5010@rtp.ciscotac.net>;tag=329447~c9cc7ddc-9592-49e8-a13c-
79e26f48eebc-30106833
To: <sip:pstojano-test@dmzlab.call.ciscospark.com>;tag=f3734601fb0eb541
Max-Forwards: 69
Route: <sip:proxy-call-id=be17a470-0bca-4ad5-8a6c-
14872e007efb@192.168.1.6:7003;transport=tls;lr>, <sip:proxy-call-id=be17a470-0bca-4ad5-8a6c-
14872e007efb@192.168.1.6:5061;transport=tls;lr>
User-Agent: Cisco-CUCM11.5
Date: Tue, 19 Sep 2017 14:46:09 GMT
X-TAATag: 14a0bd87-1825-4ecf-9f3d-4a23cfa69725
Reason: Q.850 ;cause=47
Session-ID: 75957d4fb66a13e835c10737aa329445;remote=ed35426ed3ade6fdc3b058792333df2b
Content-Length: 0

```

Because the Cisco Webex component zeroed out the audio codec for this call sample, the focus must be on:

- a. The initial INVITE that was sent to Cisco Webex and
- b. What was the logic Cisco Webex used to zero out that port.

Now looking at what is unique about the initial INVITE what can be noticed is it only contains G.729. Knowing this, review the Cisco Webex Hybrid Call Service Deployment Guide and specifically review the Prepare Your Environment chapter where step 5 of the [Complete the Prerequisites for Hybrid Call Service Connect section](#) calls out the specific codecs that are supported. There we would see this:

Cisco Webex supports the following codecs:

- Audio—G.711, G.722, AAC-LD
- Video—H.264

*Note: Opus is not used on the on-premise leg of the call for Cisco Webex Hybrid Call.*

With this information at hand, you can conclude that Unified CM is sending an unsupported audio codec which is the reason the Cisco Webex is zeroing out the port.

Solution:

To address this particular situation, you may need to review the region configuration between the Cisco Webex RD that is anchoring the call on-premises and the SIP Trunk for the Expressway-C. To do so, determine which Device Pool those two elements are in. The Device Pool contains the mappings to the Regions.

To determine the Device Pool of the Expressway-C SIP Trunk:

1. Log in to the Unified CM.
2. Navigate to **Device > Trunk**.
3. Search for the **Trunk** name or click **Find**.
4. Select the Expressway-C trunk.
5. Record the name of the **Device Pool**.

To Determine the Device Pool of the CTI-RD or Cisco Webex-RD that Anchored the Call:

1. Navigate to **Device > Phone**.
2. When searching you can select Device Type contains Webex or CTI Remote Device (depending on what the customer is using).
3. Record the name of the **Device Pool**.

Determine the Region attached to each Device Pool:

1. Navigate to **System > Device Pool**.
2. Search for the Device Pool used for the Expressway-C SIP Trunk.
3. Click on the **Device Pool**.
4. Record the **Region** name.
5. Search for the Device Pool used for the Webex-RD or CTI-RD.
6. Click on the **Device Pool**.

## 7. Record the **Region** name.

Determine the Region Relationship:

1. Navigate to **System > Region information > Region**.
2. Search on one of the Regions identified.
3. Determine if there is a Region relation between both regions that are using G.729.

At this point, if you identify the relationship that is using G.729, you'll need to adjust the relationship to support of the supported audio codecs that Cisco Webex uses or use a different Device Pool that has a Region that supports this. In the scenario documented above, the following was determined:

Expressway-C Trunk Region: ReservingBandwidth

Webex-RD Region: RTP-Devices

Here is a graphical illustration of the relationship between the RTP-Devices and ReservingBandwidth regions as shown in the image.

Region	Audio Codec Preference List	Maximum Audio Bit Rate	Maximum Session Bit Rate for Video Calls	Maximum Session Bit Rate for Immersive Video Calls
Default	Use System Default (Factory Default low loss)	256 kbps (L16, AAC-LD)	32000 kbps	32000 kbps
ReservingBandwidth	Use System Default (Factory Default low loss)	8 kbps (G.729)	384 kbps	384 kbps
RTP-Devices	Use System Default (Factory Default low loss)	256 kbps (L16, AAC-LD)	32000 kbps	32000 kbps
RTP-Infrastructure	Use System Default (Factory Default low loss)	256 kbps (L16, AAC-LD)	32000 kbps	32000 kbps

By changing the Device Pool the Expressway-C trunk was in, you change the Region relationship. The new Device Pool had a Region set to RTP-Infrastructure, therefore the new region relationship between the Cisco Webex-RD and Expressway-C trunk was RTP-Devices and RTP-Infrastructure. As pictured, you can see this relationship supports AAC-LD which is one of the supported audio codecs for Cisco Webex and so the call will set up correctly.

## Issue 2. Unified CM Max Incoming Message Size Exceeded

Because video has become more prevalent within the enterprise, the size of SIP messages that contain SDP has grown substantially. The servers that process these messages must be configured in such a way that they can accept a large packet. On many call control servers, the default values are fine. With the Cisco Unified Communications Manager (Unified CM), the default values to handle a large SIP message containing SDP in past releases were not. In later releases of Unified CM, the value size allowed for a SIP message have been increased however this value is only set on new installs, not upgrades. With this, all said, customers who are upgrading their older releases of Unified CM to support Hybrid Call Service Connect might be affected by the Max Incoming Message Size on Unified CM being too low.

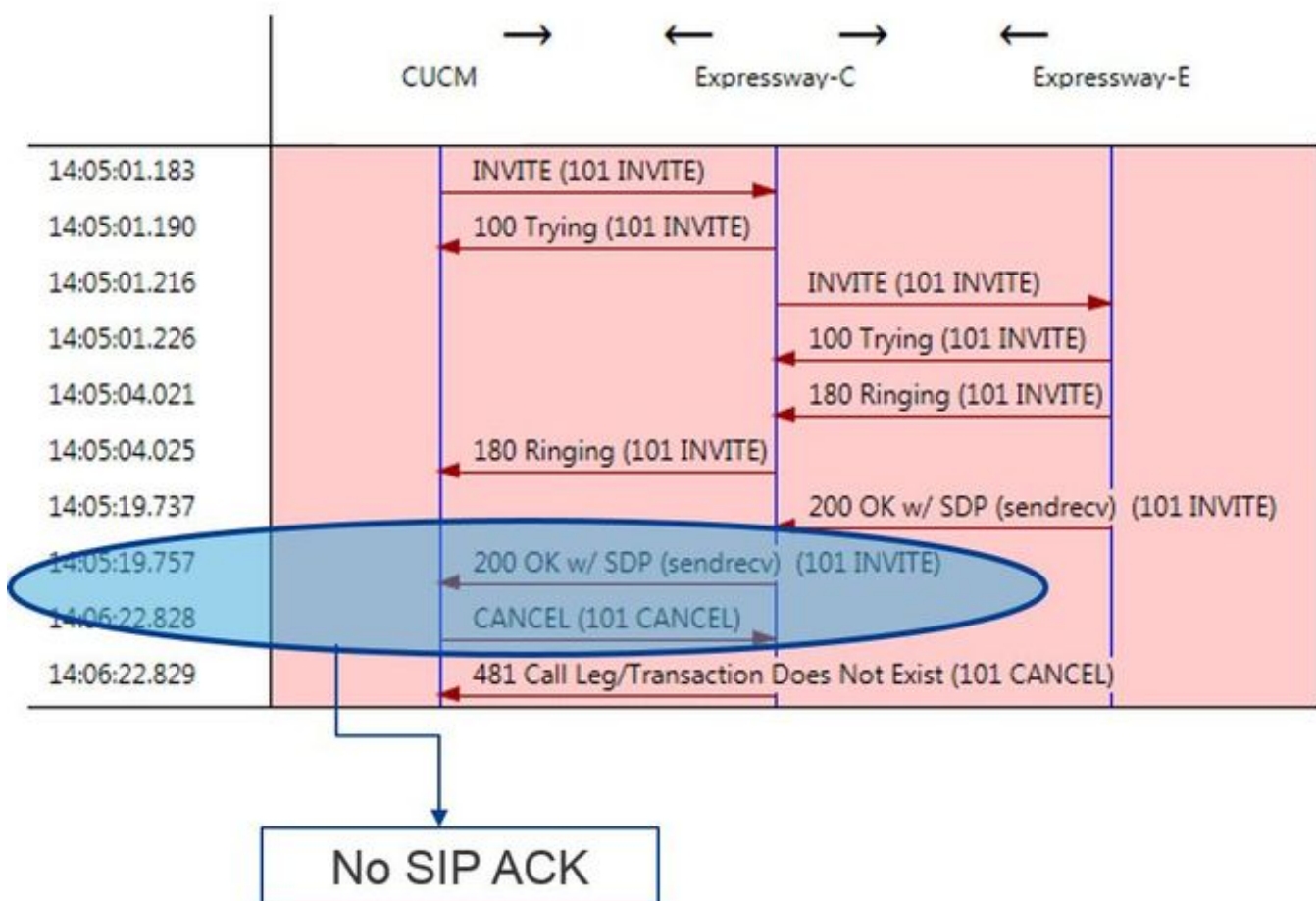
If you're trying to identify a Hybrid Call Service Connect call failure that matches this issue, you must get the Expressway logs in addition to Unified CM SDL traces. In order to identify the failure, first, understand what happens and then the types of scenarios in which the failure can occur.

To answer the question of what happens, you must know that once the Unified CM receives a SIP message that is too large, it simply closes the TCP socket and does not respond to the Expressway-C.

With this said, there are many situations and ways this could occur:

1. Cisco Webex sends an inbound INVITE w/ SDP that is too large. The Expressway-C passes this onto the Unified CM and Unified CM closes the TCP socket then the SIP dialog will time out.
2. Unified CM attempts the outbound call as Early Offer to Webex which means the initial INVITE sent to the Expressway-C will contain SDP. Cisco Webex then sends a 200 OK w/ SDP in response and the 200 OK response when passed from the Expressway-C to the Unified CM is too large. Unified CM closes the TCP socket then the SIP dialog will time out.
3. Unified CM attempts the outbound call as Delayed Offer to Webex which means the initial INVITE sent to the Expressway-C will not contain SDP. Cisco Webex then sends a 200 OK w/ SDP and the 200 OK offer when passed from the Expressway-C to the Unified CM is too large. Unified CM closes the TCP socket then the SIP dialog will time out.

Looking through the Expressway-C logs for this particular condition helps you understand the message flow. If you were to use a program like [TranslatorX](#), you could see that the Expressway-C is passing the Cisco Webex 200 OK w/ SDP to Unified CM. The challenge is that the Unified CM never responds back with a SIP ACK as shown in the image.



Since the Unified CM is the responsible party for not responding, it is worth reviewing the SDL traces to see how the Unified CM is handling this condition. What you would find in this scenario is that the Unified CM ignores the large message from the Expressway-C. A logline item such as this will be printed.

### CUCM Traces

```
53138762.000 |09:05:19.762 |AppInfo |SIPSocketProtocol(5,100,14,707326)::handleReadComplete
send SdlReadRsp: size 5000
53138763.000 |09:05:19.762 |SdlSig |SdlReadRsp |wait
|SIPTcp(5,100,71,1) |SdlTCPConnection(5,100,14,707326)
|5,100,14,707326.4^10.36.100.140^* |*TraceFlagOverrode
53138763.001 |09:05:19.762 |AppInfo |SIPTcp - SdlRead bufferLen=5000
53138763.002 |09:05:19.762 |AppInfo |//SIP/Stack/Error/0x0/httpish_cache_header_val: DROPPING
unregistered header Locus: c904ecb1-d286-11e6-bfdf-b60ed914549d
53138763.003 |09:05:19.762 |AppInfo |//SIP/Stack/Info/0x0/httpish_msg_process_network_msg:
Content Length 4068, Bytes Remaining 3804
53138763.004 |09:05:19.762 |AppInfo |//SIP/Stack/Info/0x0/ccsip_process_network_message:
process_network_msg: not complete
53138763.005 |09:05:19.762 |AppInfo |SIPTcp - Ignoring large message from %Expressway-
C_IP%:[5060]. Only allow up to 5000 bytes. Resetting connection.
```

After the SIP dialog times out, Cisco Webex will send an Inbound SIP 603 Decline message to the Expressway-E as noted in the log sample.

### Expressway-E Traces

```
2017-01-04T09:05:40.645-05:00 vcs-expressway tvcs: UTCTime="2017-01-04 14:05:40,645"
Module="network.sip" Level="DEBUG": Action="Received" Local-ip="%Exp-E%" Local-port="25150" Src-
ip="%Webex_IP%" Src-port="5062" Msg-Hash="2483073756671246315" SIPMSG: SIP/2.0 603 Decline
```

As mentioned, there are three different scenarios in which you could see this behaviour. For clarity, the log samples provided in this illustration matched situation 3 where the call was sent outbound to Cisco Webex as Delayed offer.

Solution:

1. Log in to the Unified CM.
2. Navigate to **System > Service Parameters**.
3. Select the Server that is running the Call Manager service.
4. Choose the **Cisco Call Manager service** when prompted for a Service selection.
5. Select the **Advanced** Option.
6. Under the Clusterwide Parameters (Device - SIP) settings change the **SIP Max Incoming Message Size to 18000**.
7. Select **Save**.
8. Repeat this process for every Unified CM node that is running the Cisco Call Manager service.

**Note:** In order for an IP Phone, Collaboration endpoint, and/or SIP Trunk to leverage this setting it must be restarted. These devices can be restarted individually to minimize the impact on the environment. DO NOT reset every device on the CUCM unless you know it is absolutely acceptable to do so.

## Appendix

### Expressway Troubleshooting Tools

#### Check Pattern Utility

The Expressway has a pattern checking utility that is useful when you want to test whether a pattern matches a particular alias and is transformed in an expected way. The utility can be found



on the Expressway under the **Maintenance > Tools > Check pattern** menu option. Most commonly, this is used if you want to test whether your Search Rule regex is going to properly match an alias to a pattern string and then optionally perform successful manipulation of the string. For Hybrid Call Service Connect, you can also test that the Unified CM Cluster FQDN is going to match the Pattern string that you set up for the Unified CM cluster FQDN. When using this utility, remember that the call will route based on the Unified CM Cluster FQDN parameter listed in the Route Header, not the Destination URI. For example if, the following invite came into the Expressway, test the Check pattern functionality against **cucm.rtp.ciscotac.net**, not **jobobb@rtp.ciscotac.net**.

```
SIPMSG:
|INVITE sip:jobobb@rtp.ciscotac.net SIP/2.0
Via: SIP/2.0/TLS 192.168.1.6:7003;egress-
zone=HybridCallServiceTraversal;branch=z9hG4bKcac6d95278590991a2b516cf57e75827371;proxy-call-
id=abcba873-eaae-4d64-83b4-c4541d4e620c;rport
Via: SIP/2.0/TLS 192.168.1.6:5073;branch=z9hG4bK837b03f2cd91b6b19be4fc58edb251bf12;x-cisco-
local-service=nettle;received=192.168.1.6;rport=41913;ingress-zone=DefaultZone
Via: SIP/2.0/TLS 64.102.241.236:5061;egress-
zone=DefaultZone;branch=z9hG4bK524f89592d00ffc45b7b53000271676c370.88b5177ac4d7cfcaeleb8f8be78da
055;proxy-call-id=2db939b2-a49b-4307-8d96-23716a2c090b;received=172.16.2.2;rport=25010
Via: SIP/2.0/TLS
192.168.4.150:5062;branch=z9hG4bK92f9ef952712e6610c3e6b72770c1230;received=148.62.40.63;rport=39
986;ingress-zone=HybridCallServicesDNS
Via: SIP/2.0/TLS 127.0.0.1:5070;branch=z9hG4bK-313634-
3d27a6f914badee6420287903c9c6a45;rport=45939
Call-ID: 3e613afb185751cdf019b056285eb574@127.0.0.1
CSeq: 1 INVITE
Contact: <sip:192.168.1.6:5073;transport=tls>
From: "pstoiano test" <sip:pstoiano-test@dmzlab.call.ciscopark.com>;tag=145765215
To: <sip:jobobb@rtp.ciscotac.net>
Max-Forwards: 15
Route: <sip:cucm.rtp.ciscotac.net;lr>
```

In order to use Check pattern to test the Hybrid Call Service Connect Route header search rule routing, follow these steps:

1. Navigate to **Maintenance > Tools > Check pattern**.
2. For the Alias, enter the **Unified CM Cluster FQDN**.
3. Set the Pattern Type to **Prefix**.
4. Set the Pattern String to **Unified CM Cluster FQDN**.
5. Set the Pattern behaviour to **Leave**.
6. Select **Check pattern**.

If the search rules on the Expressway are configured correctly, you can expect to see the Results return a Succeeded message.

Here is an example of a successful Check pattern test as shown in the image.

**Check pattern**

Alias

Alias

Pattern

Pattern type

Pattern string

Pattern behavior

**Result**

Result	Succeeded
Details	Alias matched pattern
Alias	cucm.rtp.ciscotac.net

The reason this is successful is that this Alias (cucm.rtp.ciscotac.net) matches the Prefix pattern string of (cucm.rtp.ciscotac.net). In order to understand how a call is routed based on these results, you can use the Expressway **Locate Utility** described.

## Locate Utility

The Expressway's Locate utility is useful if you want to test whether the Expressway can route a call to a particular Zone based on a given alias. All this can be completed without having to place a real call. The Locate utility can be found on the Expressway under the **Maintenance > Tools > Locate** menu. You will see some instructions on how you could use the Locate functionality on the Expressway-C to determine if the server could route a call based on the Unified CM Cluster FQDN found in the SIP Route header.

1. Navigate to **Maintenance > Tools > Locate**.
2. Enter the **Unified CM Cluster FQDN** in the Alias field.
3. Select **SIP** as the Protocol.
4. Select your **Cisco Webex Hybrid Traversal client Zone** for the Source.
5. Select **Locate**.

At the bottom of the interface, you will now see the search results. Here is an example of the sample test that was run with the matching results as shown in the image.

**Locate**

Locate

Alias

Hop count

Protocol

Source

Authenticated

Source alias

Here are the results of the Locate. Bolded are the values of interest. These results show:

- The fact that the Alias could be routed (True)
- Source information (Zone name/type)
- Destination information (alias being routed)
- Search Rule being matched (Hybrid Call Service Inbound Routing)
- The zone that the call would be sent to (CUCM11)

**Search (1)**

**State: Completed**

**Found: True**

Type: SIP (OPTIONS)

SIPVariant: Standards-based

**CallRouted: True**

CallSerial Number: ae73fb64-c305-457a-b7b3-59ea9688c630

Tag: 473a5b19-9a37-40bf-bbee-6f7bc94e7c77

**Source (1)**

Authenticated: True

Aliases (1)

Alias (1)

Type: Url

Origin: Unknown

Value: xcom-locate

**Zone (1)**

**Name: Hybrid Call Service Traversal**

**Type: TraversalClient**

Path (1)

Hop (1)

Address: 127.0.0.1

**Destination (1)**

**Alias (1)**

Type: Url

Origin: Unknown

**Value: sip:cucm.rtp.ciscotac.net**

StartTime: 2017-09-24 09:51:18

Duration: 0.01

SubSearch (1)

Type: Transforms

Action: Not Transformed

ResultAlias (1)

Type: Url

Origin: Unknown

Value: cucm.rtp.ciscotac.net

SubSearch (1)

Type: Admin Policy

Action: Proxy

ResultAlias (1)

Type: Url

Origin: Unknown

Value: cucm.rtp.ciscotac.net

SubSearch (1)

Type: FindMe

Action: Proxy

ResultAlias (1)

Type: Url

Origin: Unknown

Value: cucm.rtp.ciscotac.net

SubSearch (1)

Type: Search Rules

SearchRule (1)

Name: as is local

Zone (1)

Name: LocalZone

Type: Local  
Protocol: SIP  
Found: False  
Reason: Not Found  
StartTime: 2017-09-24 09:51:18  
Duration: 0  
Gatekeeper (1)  
Address: 192.168.1.5:0  
Alias (1)  
Type: Url  
Origin: Unknown  
Value: cucm.rtp.ciscotac.net  
Zone (2)  
Name: LocalZone  
Type: Local  
Protocol: H323  
Found: False  
Reason: Not Found  
StartTime: 2017-09-24 09:51:18  
Duration: 0  
Gatekeeper (1)  
Address: 192.168.1.5:0  
Alias (1)  
Type: Url  
Origin: Unknown  
Value: cucm.rtp.ciscotac.net  
**SearchRule (2)**  
**Name: Hybrid Call Service Inbound Routing**  
**Zone (1)**  
**Name: CUCM11**  
**Type: Neighbor**  
**Protocol: SIP**  
**Found: True**  
**StartTime: 2017-09-24 09:51:18**  
**Duration: 0**  
**Gatekeeper (1)**  
**Address: 192.168.1.21:5065**  
**Alias (1)**  
**Type: Url**  
**Origin: Unknown**  
**Value: cucm.rtp.ciscotac.net**

## Diagnostic Logging

Any time you're troubleshooting a calling or media issue for a call that traverses the Expressway solution, you must use the diagnostic logging. This Expressway capability gives an engineer a great detail of information for all the logic decisions the Expressway is going through as the call passes. You can see the full body SIP messages, how the Expressway passes that call through, and how the Expressway sets up the media channels. The diagnostic logging has a number of different modules that feed into it. The logging levels can be adjusted to show FATAL, ERROR, WARN, INFO, DEBUG, TRACE. By default, everything is set to INFO which captures almost everything you need to diagnose a problem. From time to time, you may need to adjust a logging level of a particular module from INFO to DEBUG to get a better understanding of what is happening. The steps below illustrate how you can adjust the logging levels of the `developer.ssl` module which is responsible for providing information for (mutual) TLS handshakes.

1. Log in to the Expressway server (Must be done on both the Expressway-E and C).
2. Navigate to **Maintenance > Diagnostics > Advanced > Support Log configuration**.
3. Scroll to the module you would like to adjust, in this instance **developer.ssl** and click it.
4. Next to the Level parameter, choose **DEBUG** from the menu.

5. Click **Save**.

At this point, you're prepared to capture the diagnostic logging:

1. Log into the Expressway server (Must be done on both the Expressway-E and C).
2. Navigate to **Maintenance > Diagnostics > Diagnostic logging**.
3. Click on **Start New Log** (Ensure that you check the **tcpdump** option).
4. Reproduce the issue.
5. Click **Stop Logging**.
6. Click **Download Log**.

For the Expressway diagnostic logging, keep in mind that you would start the logging from both the Expressway-C and Expressway-E in parallel: first, start the logging on the Expressway-E, then go to the Expressway-C and start it. At that point, you can then reproduce the problem.

**Note:** Currently, the Expressway/VCS diagnostic log bundle does not contain information about the Expressway Server certificate or Trusted CA list. If you have a case where having this functionality would be beneficial, please attach your case to [this defect](#).

## Related Information

- [Deployment Guide for Cisco Webex Hybrid Call Services](#)
- [Cisco Webex Hybrid Design Guide](#)
- [Cisco Expressway Administrator Guide](#)
- [Technical Support & Documentation - Cisco Systems](#)