CSCwc69661引入的MRA服务的Expressway流量服务器证书验证故障排除

目录

简介

本文档介绍链接到Cisco Bug ID CSCwc6961的Expressway版本X14.2.0及更高版本上的行为更改。通过此更改，Expressway平台上的流量服务器执行移动和远程访问(MRA)服务的Cisco Unified Communication Manager(CUCM)、Cisco Unified Instant Messaging & Presence(IM&P)和Unity服务器节点的证书验证。在Expressway平台上升级后，此更改可能会导致MRA登录失败。

先决条件

要求

Cisco 建议您了解以下主题：

- Expressway基本配置
- MRA基本配置

使用的组件

本文档中的信息基于X14.2及更高版本上的Cisco Expressway。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

安全超文本传输协议(HTTPS)是一种使用传输层安全(TLS)加密通信的安全通信协议。它通过使用在TLS握手过程中交换的TLS证书来创建此安全通道。这样，它就实现了两个目的：身份验证（了解您连接的远程方）和隐私（加密）。 身份验证可防止中间人攻击，并且隐私可防止攻击者窃听和篡改通信。

TLS（证书）验证在看到身份验证时执行，并允许您确保已连接到正确的远程方。验证包括两个单独的项目：

1.受信任证书颁发机构(CA)链

2.主题备用名称(SAN)或公用名称(CN)

可信CA链

为了使Expressway-C信任CUCM/IM&P/Unity发送的证书，它需要能够建立从该证书到其信任的顶级（根）证书颁发机构(CA)的链接。此类链接是将实体证书链接到根CA证书的证书层次结构，称为信任链。为了能够验证此类信任链，每个证书包含两个字段：Issuer（或“Issued by”）和Subject（或“Issued To”）。

服务器证书（例如CUCM发送到Expressway-C的那个）在“Subject”字段中通常在CN中具有其完全限定域名(FQDN):

```
        Issuer: DC=lab, DC=vngtp, CN=vngtp-ACTIVE-DIR-CA
        Subject: C=BE, ST=Flamish-Brabant, L=Diegem, O=Cisco, OU=TAC, CN=cucm.vngtp.lab
```

CUCM cucm.vngtp.lab的服务器证书示例。它在“主题”(Subject)字段的CN属性中具有FQDN，同时还具有其他属性，例如国家(C)、州(ST)、位置(L)。.我们还可以看到服务器证书由名为vngtp-ACTIVE-DIR-CA的CA分发（颁发）。

顶级CA（根CA）也可以颁发证书来标识自己。在这样的根CA证书中，我们看到颁发者和使用者具有相同的值：

```
        Issuer:  DC=lab, DC=vngtp, CN=vngtp-ACTIVE-DIR-CA
        Subject: DC=lab, DC=vngtp, CN=vngtp-ACTIVE-DIR-CA
```

它是根CA分发的用于标识自己的证书。

在典型情况下，根CA不会直接颁发服务器证书。相反，它们会为其他CA颁发证书。这些其它CA然后称为中间CA。反过来，中间CA可以直接为其他中间CA颁发服务器证书或证书。我们可能会遇到中间的CA 1颁发服务器证书，而中间的CA 1又从中间的CA 2获得证书，以此类推。直到最终中间CA直接从根CA获取其证书：

```
Server certificate :
Issuer: DC=lab, DC=vngtp, CN=vngtp-intermediate-CA-1 Subject: C=BE, ST=Flamish-Brabant,
L=Diegem, O=Cisco, OU=TAC, CN=cucm.vngtp.lab
Intermediate CA 1 certificate :
Issuer: DC=lab, DC=vngtp, CN=vngtp-intermediate-CA-2
Subject: DC=lab, DC=vngtp, CN=vngtp-intermediate-CA-1
Intermediate CA 2 certificate :
Issuer: DC=lab, DC=vngtp, CN=vngtp-intermediate-CA-3
Subject: DC=lab, DC=vngtp, CN=vngtp-intermediate-CA-2
```

```
...
Intermediate CA n certificate :
Issuer: DC=lab, DC=vngtp, CN=vngtp-ACTIVE-DIR-CA
Subject: DC=lab, DC=vngtp, CN=vngtp-intermediate-CA-n
Root CA certificate :
Issuer: DC=lab, DC=vngtp, CN=vngtp-ACTIVE-DIR-CA
Subject: DC=lab, DC=vngtp, CN=vngtp-ACTIVE-DIR-C
```

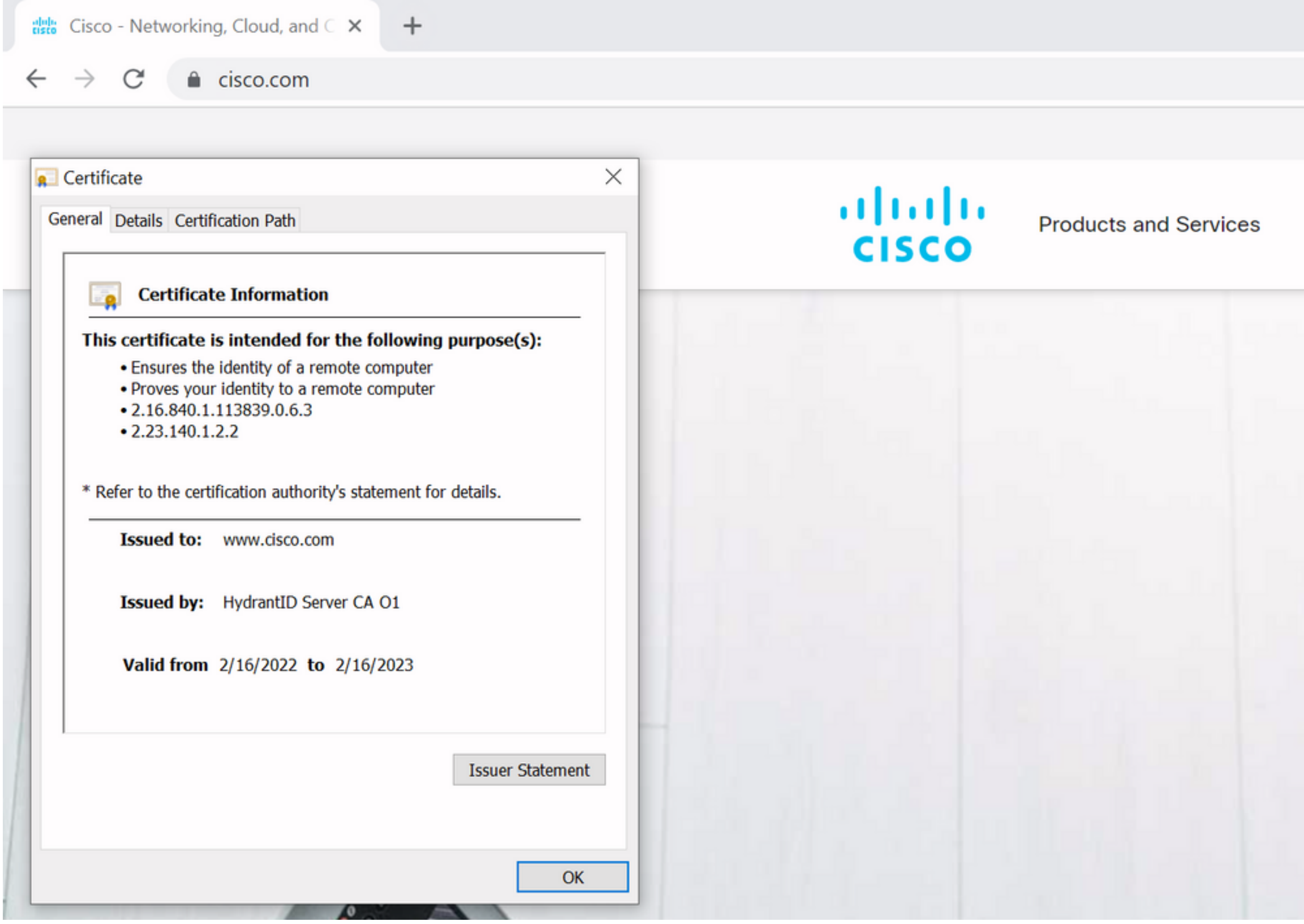
现在，为了让Expressway-C信任CUCM发送的服务器证书，它需要能够从该服务器证书构建信任链，直到根CA证书。为此，我们需要在Expressway-C的信任存储中上传根CA证书和所有中间CA证书（如果有，如果根CA直接颁发CUCM的服务器证书则不会出现这种情况）。

> **注意：虽然Issuer和Subject字段易于以易于阅读的方式构建信任链，但Expressway-C和CUCM在证书中不使用这些字段。相反，它们使用“X509v3授权密钥标识符”和“X509v3主题密钥标识符”字段构建信任链。这些密钥包含更准确的证书标识符，然后使用Subject/Issuer字段：可以有2个具有相同Subject/Issuer字段的证书，但其中一个证书已过期，另一个证书仍然有效。它们都有不同的X509v3主题密钥标识符，因此Expressway/CUCM仍可确定正确的信任链。**

SAN或CN检查

第1步检查信任库，但拥有信任库中的CA签名的证书的任何人在此时都是有效的。这显然是不够的。因此，另外会进行检查，以验证您专门连接的服务器是否正确。它根据发出请求的地址执行此操作。

在浏览器中也会发生同样的操作，因此让我们通过一个示例来了解这一点。如果浏览到https://www.cisco.com，您会在输入的URL旁边看到一个锁图标，这意味着它是受信任连接。这既基于CA信任链（来自第一部分），也基于SAN或CN检查。如果我们打开证书（通过浏览器单击锁定图标），您会看到“公用名”（在“Issued to：”字段中看到）设置为www.cisco.com，并且完全对应于要连接的地址。这样可以确保我们连接到正确的服务器（因为我们信任签署证书并在分发证书之前执行验证的CA）。



Certificate Information

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- Proves your identity to a remote computer
- 2.16.840.1.113839.0.6.3
- 2.23.140.1.2.2

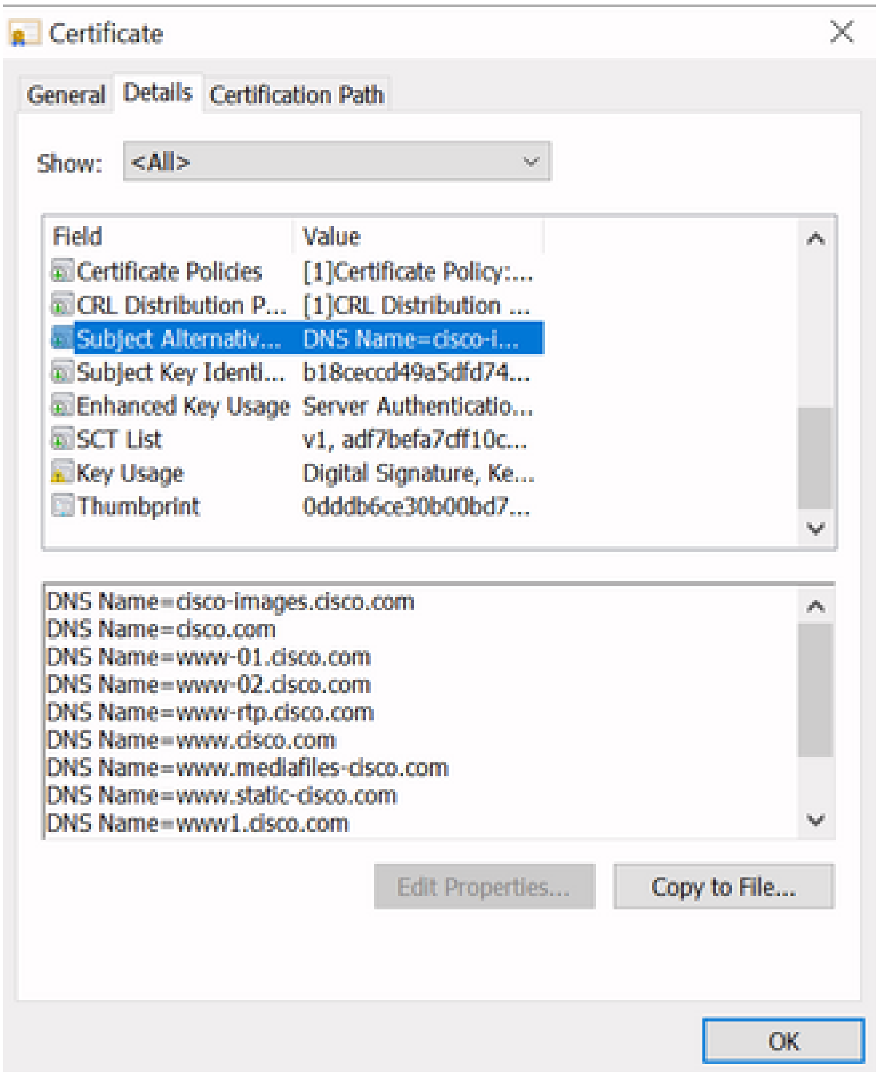
* Refer to the certification authority's statement for details.

Issued to: www.cisco.com

Issued by: HydrantID Server CA O1

Valid from 2/16/2022 **to** 2/16/2023

当我们查看证书的详细信息（尤其是SAN条目）时，我们会看到该详细信息与某些其他FQDN一样：

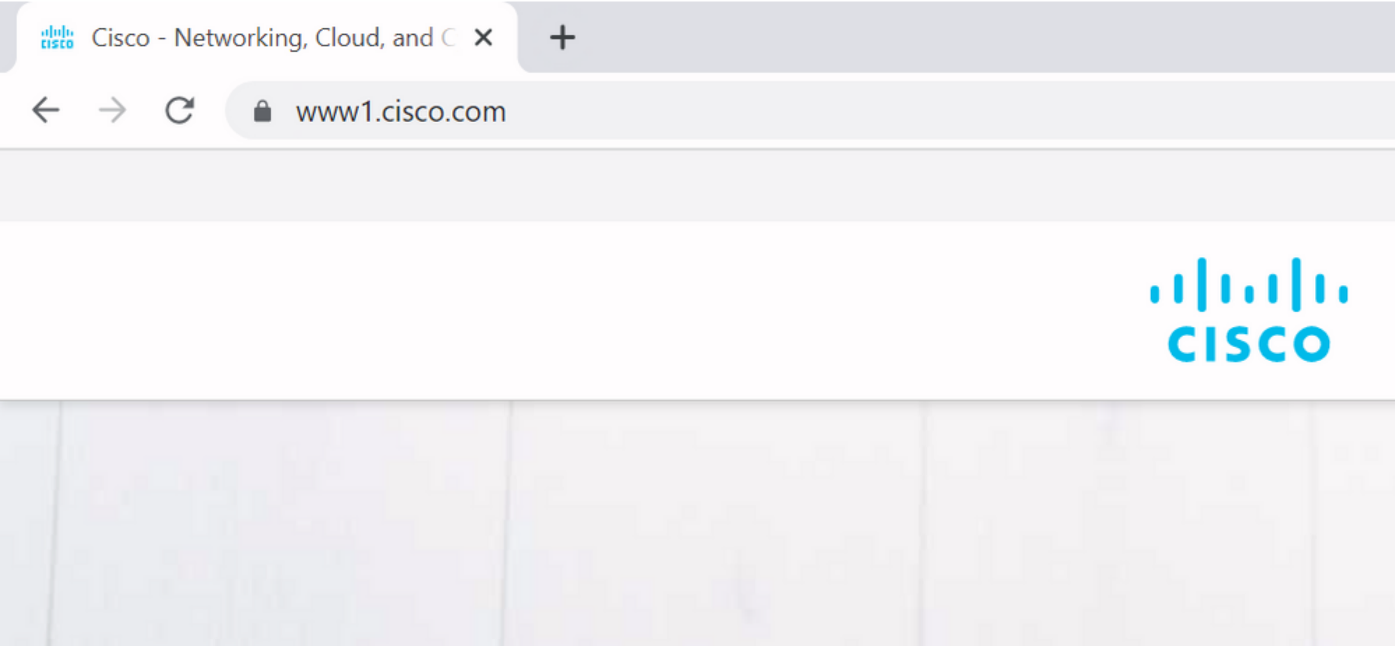


Show: <All>

Field	Value
Certificate Policies	[1]Certificate Policy:...
CRL Distribution P...	[1]CRL Distribution ...
Subject Alternativ...	DNS Name=cisco-i...
Subject Key Identi...	b18ceccd49a5dfd74...
Enhanced Key Usage	Server Authenticatio...
SCT List	v1, adf7befa7cff10c...
Key Usage	Digital Signature, Ke...
Thumbprint	0dddb6ce30b00bd7...

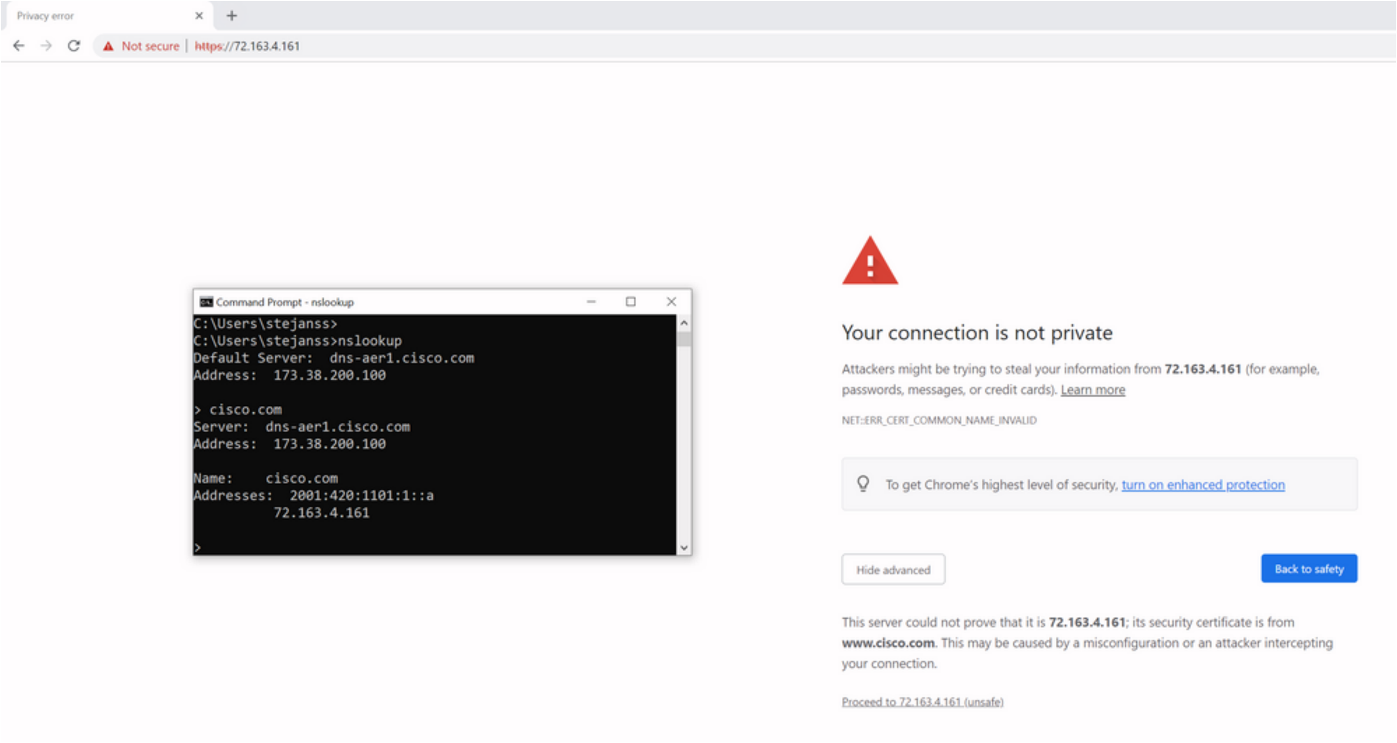
```
DNS Name=cisco-images.cisco.com
DNS Name=cisco.com
DNS Name=www-01.cisco.com
DNS Name=www-02.cisco.com
DNS Name=www-rtp.cisco.com
DNS Name=www.cisco.com
DNS Name=www.mediafiles-cisco.com
DNS Name=www.static-cisco.com
DNS Name=www1.cisco.com
```

这意味着，例如，当我们请求连接到https://www1.cisco.com时，它也会显示为安全连接，因为它包含在SAN条目中。



www1.cisco.com

但是，如果我们不浏览https://www.cisco.com，而是直接浏览到IP地址(https://72.163.4.161)，则不会显示安全连接，因为它确实信任签名它的CA，但是提供给我们的证书不包含我们用于连接到服务器的地址(72.163.4.161)。



```
C:\Users\stejanss>
C:\Users\stejanss>nslookup
Default Server:  dns-aer1.cisco.com
Address:  173.38.200.100

> cisco.com
Server:  dns-aer1.cisco.com
Address:  173.38.200.100

Name:    cisco.com
Addresses:  2001:420:1101:1::a
          72.163.4.161

>
```

Your connection is not private

Attackers might be trying to steal your information from **72.163.4.161** (for example, passwords, messages, or credit cards). Learn more

NET::ERR_CERT_COMMON_NAME_INVALID

To get Chrome's highest level of security, turn on enhanced protection

This server could not prove that it is **72.163.4.161**; its security certificate is from **www.cisco.com**. This may be caused by a misconfiguration or an attacker intercepting your connection.

Proceed to 72.163.4.161 (unsafe)

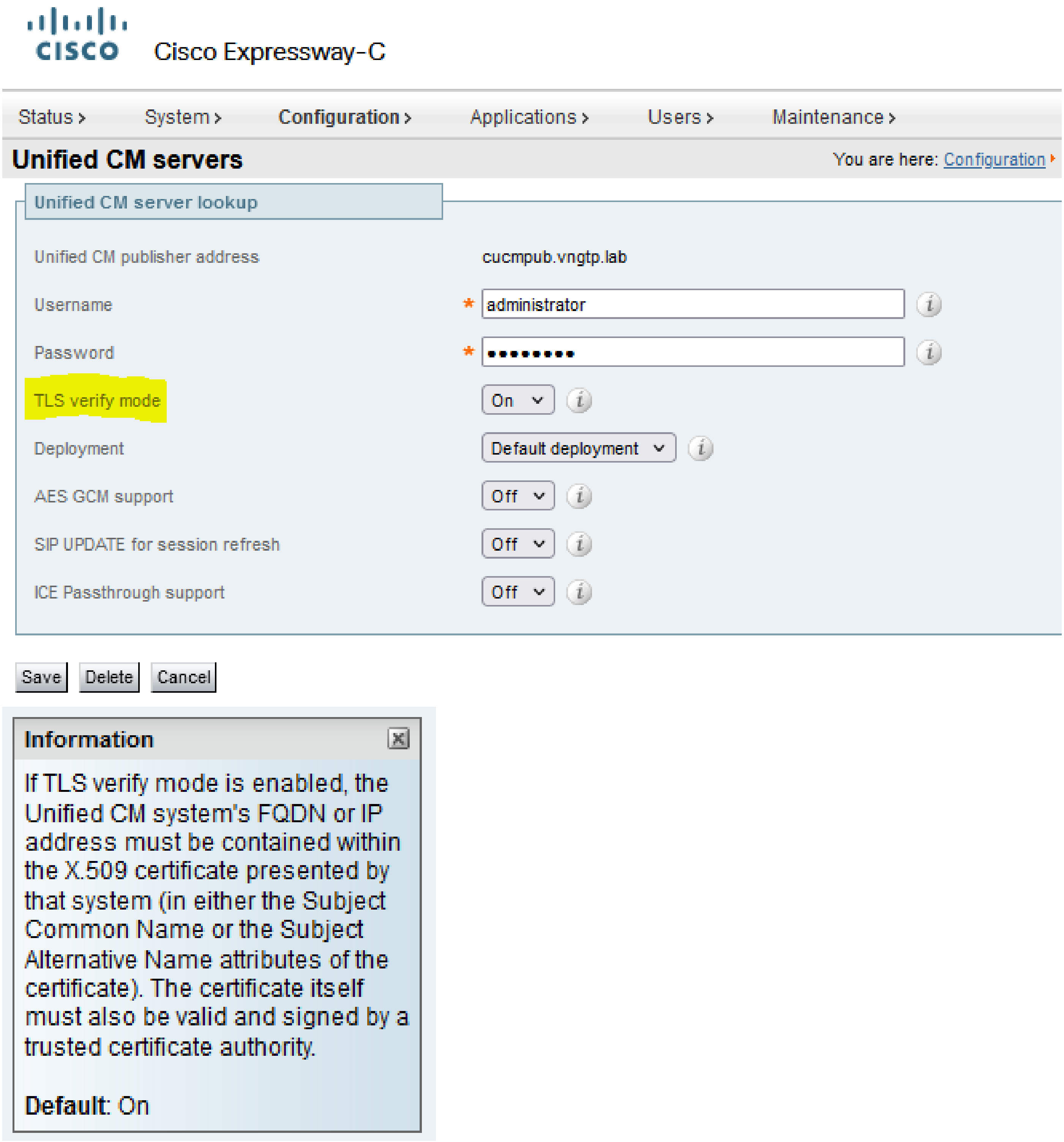
在浏览器中，您可以绕过此设置，但它是可以在TLS连接上启用的设置，不允许绕行。因此，您的证书必须包含远程方计划用于连接它的正确CN或SAN名称。

行为更改

MRA服务严重依赖通过到CUCM/IM&P/Unity服务器的Expressway上的几个HTTPS连接，以便正确进行身份验证并收集特定于登录客户端的正确信息。此通信通常发生在端口8443和6972上。

低于X14.2.0的版本

在低于X14.2.0的版本中，Expressway-C上处理这些安全HTTPS连接的流量服务器不会验证远程端提供的证书。这可能导致中间人攻击。在MRA配置上，当您将任一CUCM / IM&P / Unity服务器添加到**Configuration > Unified Communications > Unified CM servers / IM and Presence Service nodes / Unity Connection servers**下，则有一个选项可用于通过“TLS验证模式”配置到“开”进行TLS证书验证。配置选项和相关信息框以示例形式显示，表示它确实验证了SAN中的FQDN或IP，以及证书的有效性以及证书是否由受信任CA签名。



Unified CM servers

Unified CM server lookup

Unified CM publisher address	cucmpub.vngtp.lab
Username	administrator
Password	●●●●●●●●●
TLS verify mode	On
Deployment	Default deployment
AES GCM support	Off
SIP UPDATE for session refresh	Off
ICE Passthrough support	Off

Information

If TLS verify mode is enabled, the Unified CM system's FQDN or IP address must be contained within the X.509 certificate presented by that system (in either the Subject Common Name or the Subject Alternative Name attributes of the certificate). The certificate itself must also be valid and signed by a trusted certificate authority.

Default: On

此TLS证书验证检查仅在发现CUCM/IM&P/Unity服务器时完成，而不是在MRA登录期间查询各种服务器时完成。此配置的第一个缺点是，它仅验证您添加的发布者地址。它不会验证用户节点上的证书是否设置正确，因为它从发布者节点的数据库中检索用户节点信息（FQDN或IP）。此配置的第二个缺点是，由于连接信息可能不同于Expressway-C配置中的发布方地址，因此通告给MRA客户端的内容可能不同。例如，在CUCM上，在**System > Server**下，可以使用IP地址（例如10.48.36.215）向外通告服务器，然后由MRA客户端使用（通过代理的Expressway连接），但您可以在Expressway-C上使用FQDN cucm.steven.lab添加CUCM。因此，假设CUCM的tomcat证书包含cucm.steven.lab作为SAN条目而不是IP地址，则将“TLS验证模式”设置为“打开”的发现成功，但来自MRA客户端的实际通信可以针对不同的FQDN或IP，从而无法通过TLS验证。

X14.2.0及更高版本

从X14.2.0版本开始，Expressway服务器会对通过流量服务器发出的每个HTTPS请求执行TLS证书验证。这意味着在发现CUCM/IM&P/Unity节点期间，当“TLS验证模式”设置为“关闭”时，它也会执行此操作。如果验证失败，则TLS握手不会完成，并且请求失败，这可能导致功能丢失，例如冗余或故障转移问题或完全登录失败。此外，如果将“TLS验证模式”设置为“开”，则不能保证所有连接都能正常运行，如以下示例所述。

除了默认的TLS验证，X14.2中还引入了一个更改，它通告了密码列表的不同的首选顺序。这可能会导致软件升级后出现意外的TLS连接，因为在升级之前，它请求从CUCM（或任何具有单独的ECDSA算法证书的其他产品）获取Cisco Tomcat或Cisco CallManager证书，但在升级之后，它请求获取ECDSA变体。Cisco Tomcat-ECDSA或Cisco CallManager-ECDSA证书可以由其他CA签名，也可以仅由自签名证书签名（默认）。

在此场景中，TLS验证有两种可能失败，稍后将详细介绍：

1.签署远程证书的CA不受信任

a.自签名证书

b.由未知CA签名的证书

2.证书中不包含连接地址（FQDN或IP）

故障排除场景

下面的场景显示实验室环境中的类似场景，其中Expressway从X14.0.7升级到X14.2后，MRA登录确实失败。这些场景在日志中有相似之处，但分辨率不同。日志仅通过MRA登录之前开始并在MRA登录失败之后停止的诊断日志记录(从**维护>诊断>诊断日志记录**)收集。未为其启用其他调试日志记录。

1.签署远程证书的CA不受信任

远程证书可以由未包含在Expressway-C的信任存储中的CA进行签名，也可以是未添加到Expressway-C服务器的信任存储中的自签名证书（本质上也是CA）。

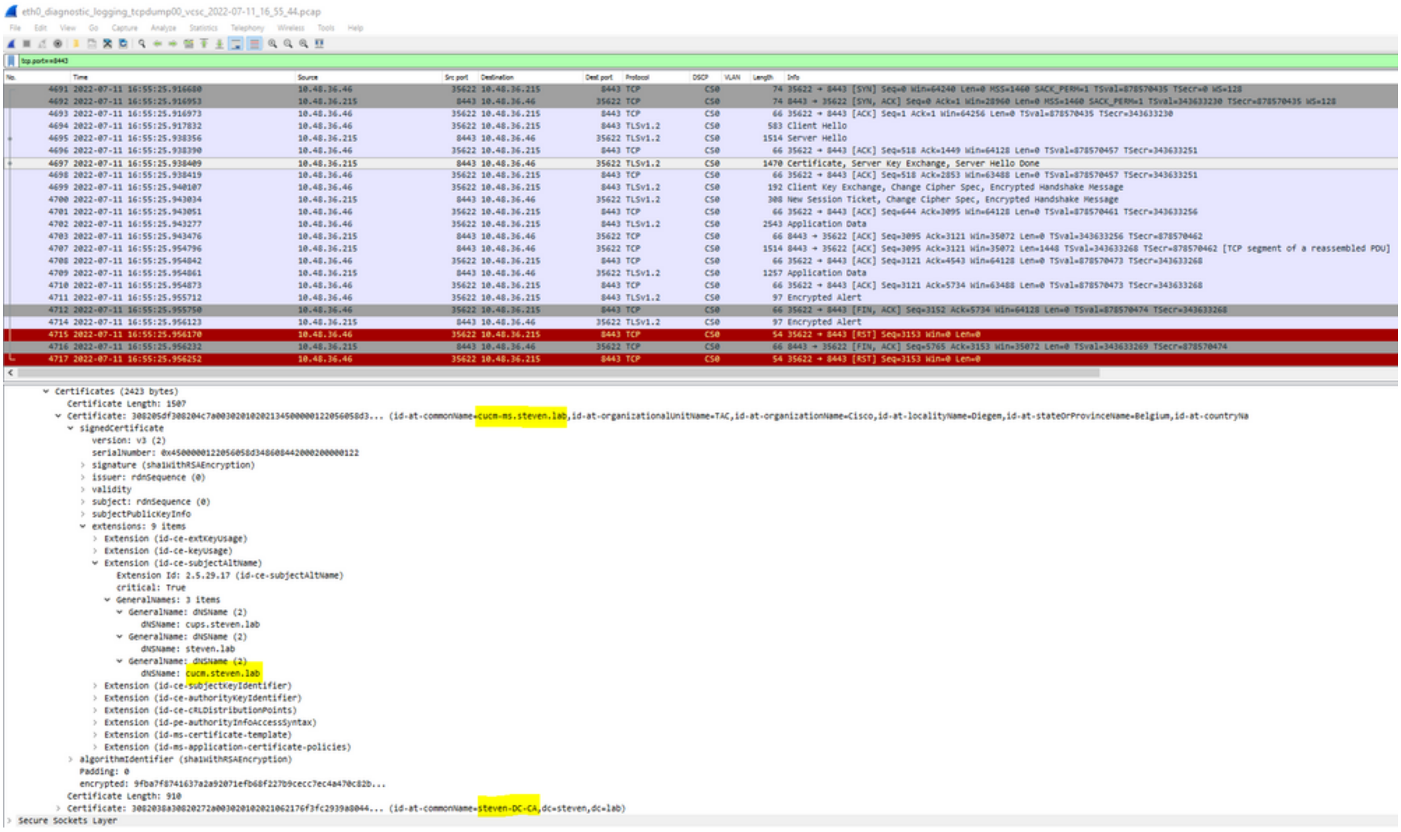
在本例中，您会发现，发往CUCM(10.48.36.215 - cucm.steven.lab)的请求在端口8443（200 OK响应）上得到正确处理，但是在TFTP连接的端口6972上引发错误（502响应）。

```
===Success connection on 8443===

2022-07-11T18:55:25.910+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:25,910"
Module="network.http.trafficserver" Level="INFO": Detail="Receive Request" Txn-id="189"
TrackingID="6af9a674-9ebc-41ea-868e-90e7309a758c" Src-ip="127.0.0.1" Src-port="35764" Last-via-
addr="" Msg="GET
http://vcs_control.steven.lab:8443/c3RldmVuLmxhYi9odHRwcy9jdWNtLnN0ZXZlbi5sYWIvODQ0Mw/cucm-
uds/user/emusk/devices HTTP/1.1"

2022-07-11T18:55:25.917+02:00 vcsc traffic_server[18242]: Event="Request Allowed" Detail="Access
allowed" Reason="In allow list" Username="emusk" Deployment="1" Method="GET"
Request="https://cucm.steven.lab:8443/cucm-uds/user/emusk/devices"
Rule="https://cucm.steven.lab:8443/cucm-uds/user/" Match="prefix" Type="Automatically generated
rule for CUCM server" UTCTime="2022-07-11 16:55:25,916"
2022-07-11T18:55:25.917+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:25,916"
Module="network.http.trafficserver" Level="INFO": Detail="Sending Request" Txn-id="189"
```

```
TrackingID="6af9a674-9ebc-41ea-868e-90e7309a758c" Dst-ip="10.48.36.215" Dst-port="8443" Msg="GET
/cucm-uds/user/emusk/devices HTTP/1.1"
2022-07-11T18:55:25.955+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:25,955"
Module="network.http.trafficserver" Level="INFO": Detail="Receive Response" Txn-id="189"
TrackingID="" Src-ip="10.48.36.215" Src-port="8443" Msg="HTTP/1.1 200 "
2022-07-11T18:55:25.956+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:25,955"
Module="network.http.trafficserver" Level="INFO": Detail="Sending Response" Txn-id="189"
TrackingID="" Dst-ip="127.0.0.1" Dst-port="35764" Msg="HTTP/1.1 200 "

===Failed connection on 6972===

2022-07-11T18:55:26.000+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:26,000"
Module="network.http.trafficserver" Level="INFO": Detail="Receive Request" Txn-id="191"
TrackingID="bb0c8492-8c15-4537-a7d1-082dde781dbd" Src-ip="127.0.0.1" Src-port="35766" Last-via-
addr="" Msg="GET
http://vcs_control.steven.lab:8443/c3RldmVuLmxhYi9odHRwcy9jdWNtLnN0ZXZlbi5sYWIvNjk3Mg/CSFemusk.c
nf.xml HTTP/1.1"

2022-07-11T18:55:26.006+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:26,006"
Module="network.http.trafficserver" Level="INFO": Detail="Sending Request" Txn-id="191"
TrackingID="bb0c8492-8c15-4537-a7d1-082dde781dbd" Dst-ip="10.48.36.215" Dst-port="6972" Msg="GET
/CSFemusk.cnf.xml HTTP/1.1"
2022-07-11T18:55:26.016+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:26,016"
Module="network.http.trafficserver" Level="INFO": Detail="Sending Request" Txn-id="191"
TrackingID="bb0c8492-8c15-4537-a7d1-082dde781dbd" Dst-ip="10.48.36.215" Dst-port="6972" Msg="GET
/CSFemusk.cnf.xml HTTP/1.1"
2022-07-11T18:55:26.016+02:00 vcsc traffic_server[18242]: [ET_NET 0] WARNING: Core server
certificate verification failed for (cucm.steven.lab). Action=Terminate Error=self signed
certificate server=cucm.steven.lab(10.48.36.215) depth=0
2022-07-11T18:55:26.016+02:00 vcsc traffic_server[18242]: [ET_NET 0] ERROR: SSL connection
failed for 'cucm.steven.lab': error:1416F086:SSL
routines:tls_process_server_certificate:certificate verify failed
2022-07-11T18:55:26.024+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:26,024"
Module="network.http.trafficserver" Level="INFO": Detail="Sending Response" Txn-id="191"
TrackingID="" Dst-ip="127.0.0.1" Dst-port="35766" Msg="HTTP/1.1 502 connect failed"
```

“certificate verify failed”错误表示Expressway-C无法验证TLS握手的事实。原因显示在警告行上，因为它表示自签名证书。如果深度显示为0，则为自签名证书。当深度大于0时，这意味着它有一个证书链，因此由未知CA签名（从Expressway-C的角度而言）。

当我们查看在文本日志中提及的时间戳处收集的pcap文件时，您可以看到CUCM将带有CN的证书显示为cucm-ms.steven.lab（和cucm.steven.lab作为SAN），由steven-DC-CA签署，并发送到端口8443上的Expressway-C。

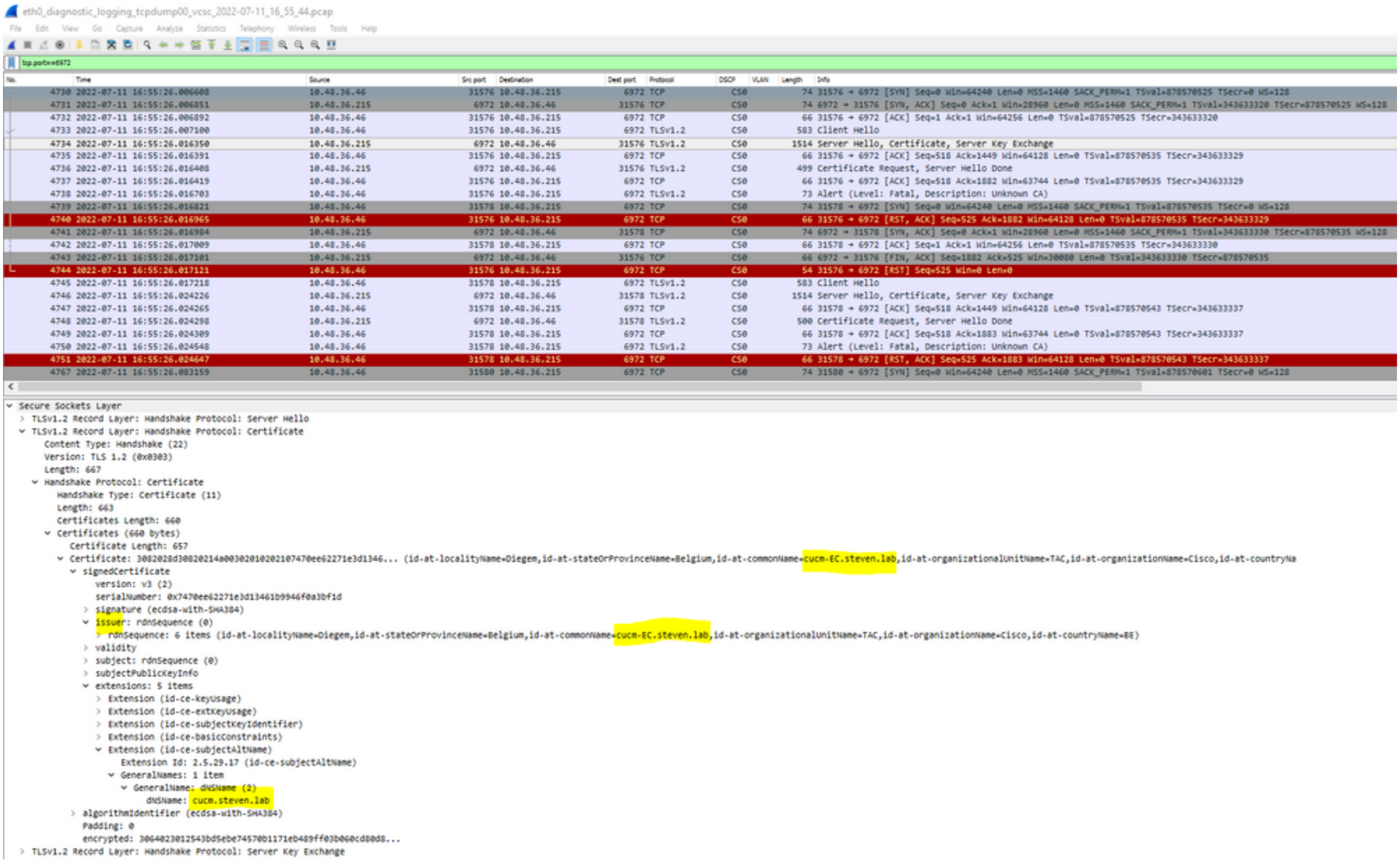


tcp.port==8443

```
No.   Time                        Source         Src port  Destination   Dest port  Protocol  DSCP  VLAN  Length  Info
4691  2022-07-11 16:55:25.916680  10.48.36.46    35622     10.48.36.215  8443       TCP       CS0         74      35622 → 8443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=878570435 TSecr=0 WS=128
4692  2022-07-11 16:55:25.916953  10.48.36.215   8443      10.48.36.46   35622      TCP       CS0         74      8443 → 35622 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=343633230 TSecr=878570435 WS=128
4693  2022-07-11 16:55:25.916973  10.48.36.46    35622     10.48.36.215  8443       TCP       CS0         66      35622 → 8443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=878570435 TSecr=343633230
4694  2022-07-11 16:55:25.917832  10.48.36.46    35622     10.48.36.215  8443       TLSv1.2   CS0         583     Client Hello
4695  2022-07-11 16:55:25.938356  10.48.36.215   8443      10.48.36.46   35622      TLSv1.2   CS0         1514    Server Hello
4696  2022-07-11 16:55:25.938390  10.48.36.46    35622     10.48.36.215  8443       TCP       CS0         66      35622 → 8443 [ACK] Seq=518 Ack=1449 Win=64128 Len=0 TSval=878570457 TSecr=343633251
4697  2022-07-11 16:55:25.938409  10.48.36.215   8443      10.48.36.46   35622      TLSv1.2   CS0         1470    Certificate, Server Key Exchange, Server Hello Done
4698  2022-07-11 16:55:25.938419  10.48.36.46    35622     10.48.36.215  8443       TCP       CS0         66      35622 → 8443 [ACK] Seq=518 Ack=2853 Win=63488 Len=0 TSval=878570457 TSecr=343633251
4699  2022-07-11 16:55:25.940107  10.48.36.46    35622     10.48.36.215  8443       TLSv1.2   CS0         192     Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
4700  2022-07-11 16:55:25.943034  10.48.36.215   8443      10.48.36.46   35622      TLSv1.2   CS0         308     New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
4701  2022-07-11 16:55:25.943051  10.48.36.46    35622     10.48.36.215  8443       TCP       CS0         66      35622 → 8443 [ACK] Seq=644 Ack=3095 Win=64128 Len=0 TSval=878570461 TSecr=343633256
4702  2022-07-11 16:55:25.943277  10.48.36.46    35622     10.48.36.215  8443       TLSv1.2   CS0         2543    Application Data
4703  2022-07-11 16:55:25.943476  10.48.36.215   8443      10.48.36.46   35622      TCP       CS0         66      8443 → 35622 [ACK] Seq=3095 Ack=3121 Win=35072 Len=0 TSval=343633256 TSecr=878570462
4707  2022-07-11 16:55:25.954796  10.48.36.215   8443      10.48.36.46   35622      TCP       CS0         1514    8443 → 35622 [ACK] Seq=3095 Ack=3121 Win=35072 Len=1448 TSval=343633268 TSecr=878570462 [TCP segment of a reassembled PDU]
4708  2022-07-11 16:55:25.954842  10.48.36.46    35622     10.48.36.215  8443       TCP       CS0         66      35622 → 8443 [ACK] Seq=3121 Ack=4543 Win=64128 Len=0 TSval=878570473 TSecr=343633268
4709  2022-07-11 16:55:25.954861  10.48.36.215   8443      10.48.36.46   35622      TLSv1.2   CS0         1257    Application Data
4710  2022-07-11 16:55:25.954873  10.48.36.46    35622     10.48.36.215  8443       TCP       CS0         66      35622 → 8443 [ACK] Seq=3121 Ack=5734 Win=63488 Len=0 TSval=878570473 TSecr=343633268
4711  2022-07-11 16:55:25.955712  10.48.36.46    35622     10.48.36.215  8443       TLSv1.2   CS0         97      Encrypted Alert
4712  2022-07-11 16:55:25.955750  10.48.36.46    35622     10.48.36.215  8443       TCP       CS0         66      35622 → 8443 [FIN, ACK] Seq=3152 Ack=5734 Win=64128 Len=0 TSval=878570474 TSecr=343633268
4714  2022-07-11 16:55:25.956123  10.48.36.215   8443      10.48.36.46   35622      TLSv1.2   CS0         97      Encrypted Alert
4715  2022-07-11 16:55:25.956170  10.48.36.46    35622     10.48.36.215  8443       TCP       CS0         54      35622 → 8443 [RST] Seq=3153 Win=0 Len=0
4716  2022-07-11 16:55:25.956232  10.48.36.215   8443      10.48.36.46   35622      TCP       CS0         66      8443 → 35622 [FIN, ACK] Seq=5765 Ack=3153 Win=35072 Len=0 TSval=343633269 TSecr=878570474
4717  2022-07-11 16:55:25.956252  10.48.36.46    35622     10.48.36.215  8443       TCP       CS0         54      35622 → 8443 [RST] Seq=3153 Win=0 Len=0
```

```
    ˅ Certificates (2423 bytes)
        Certificate Length: 1507
      ˅ Certificate: 308205df308204c7a003020102021345000001220560 58d3... (id-at-commonName=cucm-ms.steven.lab,id-at-organizationalUnitName=TAC,id-at-organizationName=Cisco,id-at-localityName=Diegem,id-at-stateOrProvinceName=Belgium,id-at-countryNa
        ˅ signedCertificate
            version: v3 (2)
            serialNumber: 0x450000012205605 8d348608442000200000122
          > signature (sha1WithRSAEncryption)
          > issuer: rdnSequence (0)
          > validity
          > subject: rdnSequence (0)
          > subjectPublicKeyInfo
          ˅ extensions: 9 items
            > Extension (id-ce-extKeyUsage)
            > Extension (id-ce-keyUsage)
            ˅ Extension (id-ce-subjectAltName)
                Extension Id: 2.5.29.17 (id-ce-subjectAltName)
                critical: True
              ˅ GeneralNames: 3 items
                ˅ GeneralName: dNSName (2)
                    dNSName: cups.steven.lab
                ˅ GeneralName: dNSName (2)
                    dNSName: steven.lab
                ˅ GeneralName: dNSName (2)
                    dNSName: cucm.steven.lab
            > Extension (id-ce-subjectKeyIdentifier)
            > Extension (id-ce-authorityKeyIdentifier)
            > Extension (id-ce-cRLDistributionPoints)
            > Extension (id-pe-authorityInfoAccessSyntax)
            > Extension (id-ms-certificate-template)
            > Extension (id-ms-application-certificate-policies)
        > algorithmIdentifier (sha1WithRSAEncryption)
          Padding: 0
          encrypted: 9fba7f8741637a2a92071efb68f227b9cecc7ec4a470c82b...
        Certificate Length: 910
      > Certificate: 3082038a30820272a00302010202106217 6f3fc2939a8044... (id-at-commonName=steven-DC-CA,dc=steven,dc=lab)
> Secure Sockets Layer
```

但是，当我们检查端口6972上提供的证书时，您可以看到它是自签证书（颁发者自身），其CN设置为cucm-EC.steven.lab。-EC扩展指明这是CUCM上设置的ECDSA证书。

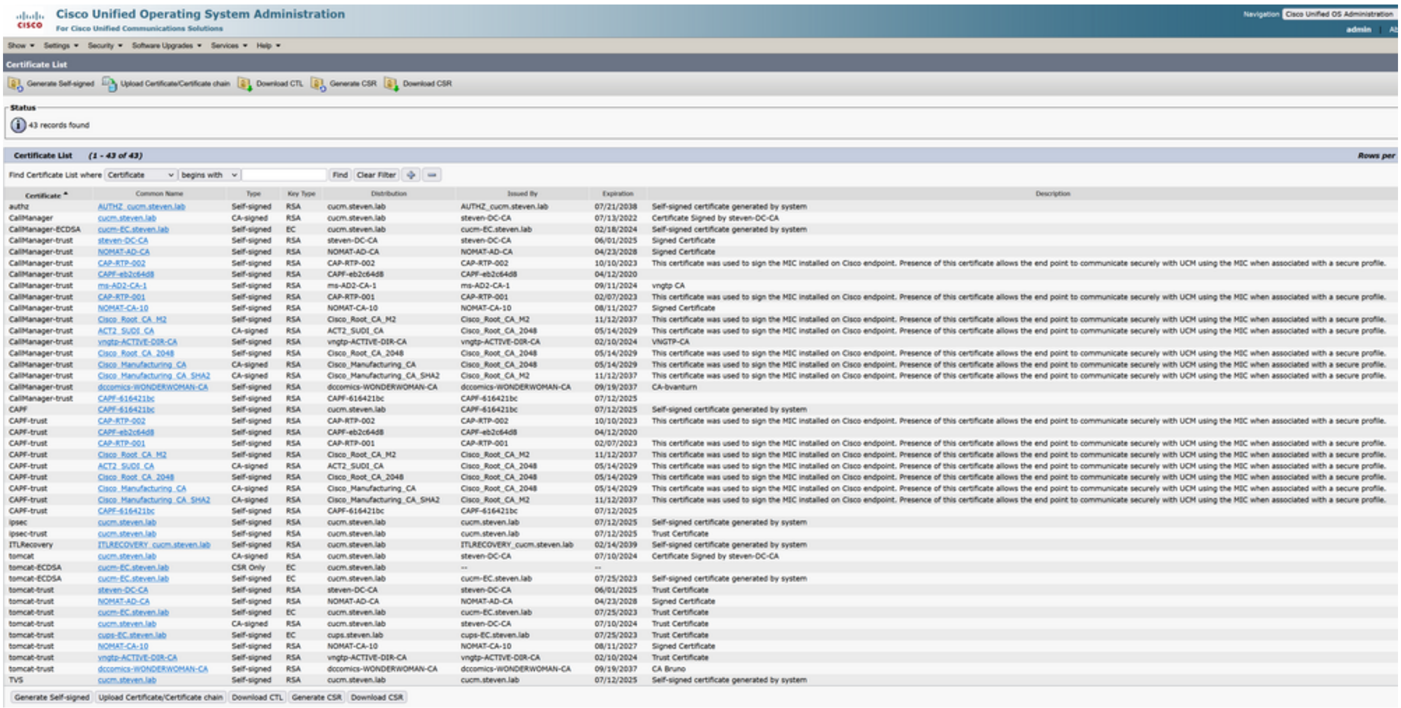


tcp.port==6972

```
No.   Time                        Source         Src port  Destination   Dest port  Protocol  DSCP  VLAN  Length  Info
4730  2022-07-11 16:55:26.006608  10.48.36.46    31576     10.48.36.215  6972       TCP       CS0         74      31576 → 6972 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=878570525 TSecr=0 WS=128
4731  2022-07-11 16:55:26.006851  10.48.36.215   6972      10.48.36.46   31576      TCP       CS0         74      6972 → 31576 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=343633320 TSecr=878570525 WS=128
4732  2022-07-11 16:55:26.006892  10.48.36.46    31576     10.48.36.215  6972       TCP       CS0         66      31576 → 6972 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=878570525 TSecr=343633320
4733  2022-07-11 16:55:26.007100  10.48.36.46    31576     10.48.36.215  6972       TLSv1.2   CS0         583     Client Hello
4734  2022-07-11 16:55:26.016350  10.48.36.215   6972      10.48.36.46   31576      TLSv1.2   CS0         1514    Server Hello, Certificate, Server Key Exchange
4735  2022-07-11 16:55:26.016391  10.48.36.46    31576     10.48.36.215  6972       TCP       CS0         66      31576 → 6972 [ACK] Seq=518 Ack=1449 Win=64128 Len=0 TSval=878570535 TSecr=343633329
4736  2022-07-11 16:55:26.016408  10.48.36.215   6972      10.48.36.46   31576      TLSv1.2   CS0         499     Certificate Request, Server Hello Done
4737  2022-07-11 16:55:26.016419  10.48.36.46    31576     10.48.36.215  6972       TCP       CS0         66      31576 → 6972 [ACK] Seq=518 Ack=1882 Win=63744 Len=0 TSval=878570535 TSecr=343633329
4738  2022-07-11 16:55:26.016703  10.48.36.46    31576     10.48.36.215  6972       TLSv1.2   CS0         73      Alert (Level: Fatal, Description: Unknown CA)
4739  2022-07-11 16:55:26.016821  10.48.36.46    31578     10.48.36.215  6972       TCP       CS0         74      31578 → 6972 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=878570535 TSecr=0 WS=128
4740  2022-07-11 16:55:26.016965  10.48.36.46    31576     10.48.36.215  6972       TCP       CS0         66      31576 → 6972 [RST, ACK] Seq=525 Ack=1882 Win=64128 Len=0 TSval=878570535 TSecr=343633329
4741  2022-07-11 16:55:26.016984  10.48.36.215   6972      10.48.36.46   31578      TCP       CS0         74      6972 → 31578 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=343633330 TSecr=878570535 WS=128
4742  2022-07-11 16:55:26.017009  10.48.36.46    31578     10.48.36.215  6972       TCP       CS0         66      31578 → 6972 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=878570535 TSecr=343633330
4743  2022-07-11 16:55:26.017101  10.48.36.215   6972      10.48.36.46   31576      TCP       CS0         66      6972 → 31576 [FIN, ACK] Seq=1882 Ack=525 Win=30080 Len=0 TSval=343633330 TSecr=878570535
4744  2022-07-11 16:55:26.017121  10.48.36.46    31576     10.48.36.215  6972       TCP       CS0         54      31576 → 6972 [RST] Seq=525 Win=0 Len=0
4745  2022-07-11 16:55:26.017218  10.48.36.46    31578     10.48.36.215  6972       TLSv1.2   CS0         583     Client Hello
4746  2022-07-11 16:55:26.024226  10.48.36.215   6972      10.48.36.46   31578      TLSv1.2   CS0         1514    Server Hello, Certificate, Server Key Exchange
4747  2022-07-11 16:55:26.024265  10.48.36.46    31578     10.48.36.215  6972       TCP       CS0         66      31578 → 6972 [ACK] Seq=518 Ack=1449 Win=64128 Len=0 TSval=878570543 TSecr=343633337
4748  2022-07-11 16:55:26.024298  10.48.36.215   6972      10.48.36.46   31578      TLSv1.2   CS0         500     Certificate Request, Server Hello Done
4749  2022-07-11 16:55:26.024309  10.48.36.46    31578     10.48.36.215  6972       TCP       CS0         66      31578 → 6972 [ACK] Seq=518 Ack=1883 Win=63744 Len=0 TSval=878570543 TSecr=343633337
4750  2022-07-11 16:55:26.024548  10.48.36.46    31578     10.48.36.215  6972       TLSv1.2   CS0         73      Alert (Level: Fatal, Description: Unknown CA)
4751  2022-07-11 16:55:26.024647  10.48.36.46    31578     10.48.36.215  6972       TCP       CS0         66      31578 → 6972 [RST, ACK] Seq=525 Ack=1883 Win=64128 Len=0 TSval=878570543 TSecr=343633337
4767  2022-07-11 16:55:26.083159  10.48.36.46    31580     10.48.36.215  6972       TCP       CS0         74      31580 → 6972 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=878570601 TSecr=0 WS=128
```

```
˅ Secure Sockets Layer
  > TLSv1.2 Record Layer: Handshake Protocol: Server Hello
  ˅ TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 667
    ˅ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 663
        Certificates Length: 660
      ˅ Certificates (660 bytes)
          Certificate Length: 657
        ˅ Certificate: 3082028d30820214a003020102021074 70ee62271e3d1346... (id-at-localityName=Diegem,id-at-stateOrProvinceName=Belgium,id-at-commonName=cucm-EC.steven.lab,id-at-organizationalUnitName=TAC,id-at-organizationName=Cisco,id-at-countryNa
          ˅ signedCertificate
              version: v3 (2)
              serialNumber: 0x7470ee62271e3d13461b9946f0a3bf1d
            > signature (ecdsa-with-SHA384)
            ˅ issuer: rdnSequence (0)
              > rdnSequence: 6 items (id-at-localityName=Diegem,id-at-stateOrProvinceName=Belgium,id-at-commonName=cucm-EC.steven.lab,id-at-organizationalUnitName=TAC,id-at-organizationName=Cisco,id-at-countryName=BE)
            > validity
            > subject: rdnSequence (0)
            > subjectPublicKeyInfo
            ˅ extensions: 5 items
              > Extension (id-ce-keyUsage)
              > Extension (id-ce-extKeyUsage)
              > Extension (id-ce-subjectKeyIdentifier)
              > Extension (id-ce-basicConstraints)
              ˅ Extension (id-ce-subjectAltName)
                  Extension Id: 2.5.29.17 (id-ce-subjectAltName)
                ˅ GeneralNames: 1 item
                  ˅ GeneralName: dNSName (2)
                      dNSName: cucm.steven.lab
          > algorithmIdentifier (ecdsa-with-SHA384)
            Padding: 0
            encrypted: 3064023012543bd5ebe74570b1171eb489ff03b060cd80d8...
  > TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
```

在Cisco Unified OS Administration下的CUCM上，您可以查看Security > Certificate Management下的现有证书，如下例所示。它显示不同的tomcat和tomcat-ECDSA证书，其中tomcat是CA签名的（并受Expressway-C信任），而tomcat-ECDSA证书是自签名的，不受Expressway-C信任。



43 records found

Certificate List (1 - 43 of 43)

Find Certificate List where Certificate begins with

Certificate ▲	Common Name	Type	Key Type	Distribution	Issued By	Expiration	Description
authz	AUTHZ_cucm.steven.lab	Self-signed	RSA	cucm.steven.lab	AUTHZ_cucm.steven.lab	07/21/2038	Self-signed certificate generated by system
CallManager	cucm.steven.lab	CA-signed	RSA	cucm.steven.lab	steven-DC-CA	07/13/2022	Certificate Signed by steven-DC-CA
CallManager-ECDSA	cucm-EC.steven.lab	Self-signed	EC	cucm.steven.lab	cucm-EC.steven.lab	02/18/2024	Self-signed certificate generated by system
CallManager-trust	steven-DC-CA	Self-signed	RSA	steven-DC-CA	steven-DC-CA	06/01/2025	Signed Certificate
CallManager-trust	NOMAT-AD-CA	Self-signed	RSA	NOMAT-AD-CA	NOMAT-AD-CA	04/23/2028	Signed Certificate
CallManager-trust	CAP-RTP-002	Self-signed	RSA	CAP-RTP-002	CAP-RTP-002	10/10/2023	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CallManager-trust	CAPF-eb2c64d8	Self-signed	RSA	CAPF-eb2c64d8	CAPF-eb2c64d8	04/12/2020	
CallManager-trust	ms-AD2-CA-1	Self-signed	RSA	ms-AD2-CA-1	ms-AD2-CA-1	09/11/2024	vngtp CA
CallManager-trust	CAP-RTP-001	Self-signed	RSA	CAP-RTP-001	CAP-RTP-001	02/07/2023	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CallManager-trust	NOMAT-CA-10	Self-signed	RSA	NOMAT-CA-10	NOMAT-CA-10	08/11/2027	Signed Certificate
CallManager-trust	Cisco_Root_CA_M2	Self-signed	RSA	Cisco_Root_CA_M2	Cisco_Root_CA_M2	11/12/2037	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CallManager-trust	ACT2_SUDI_CA	CA-signed	RSA	ACT2_SUDI_CA	Cisco_Root_CA_2048	05/14/2029	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CallManager-trust	vngtp-ACTIVE-DIR-CA	Self-signed	RSA	vngtp-ACTIVE-DIR-CA	vngtp-ACTIVE-DIR-CA	02/10/2024	VNGTP-CA
CallManager-trust	Cisco_Root_CA_2048	Self-signed	RSA	Cisco_Root_CA_2048	Cisco_Root_CA_2048	05/14/2029	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CallManager-trust	Cisco_Manufacturing_CA	CA-signed	RSA	Cisco_Manufacturing_CA	Cisco_Root_CA_2048	05/14/2029	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CallManager-trust	Cisco_Manufacturing_CA_SHA2	CA-signed	RSA	Cisco_Manufacturing_CA_SHA2	Cisco_Root_CA_M2	11/12/2037	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CallManager-trust	dccomics-WONDERWOMAN-CA	Self-signed	RSA	dccomics-WONDERWOMAN-CA	dccomics-WONDERWOMAN-CA	09/19/2037	CA-bvanturn
CallManager-trust	CAPF-616421bc	Self-signed	RSA	CAPF-616421bc	CAPF-616421bc	07/12/2025	
CAPF	CAPF-616421bc	Self-signed	RSA	cucm.steven.lab	CAPF-616421bc	07/12/2025	Self-signed certificate generated by system
CAPF-trust	CAP-RTP-002	Self-signed	RSA	CAP-RTP-002	CAP-RTP-002	10/10/2023	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CAPF-trust	CAPF-eb2c64d8	Self-signed	RSA	CAPF-eb2c64d8	CAPF-eb2c64d8	04/12/2020	
CAPF-trust	CAP-RTP-001	Self-signed	RSA	CAP-RTP-001	CAP-RTP-001	02/07/2023	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CAPF-trust	Cisco_Root_CA_M2	Self-signed	RSA	Cisco_Root_CA_M2	Cisco_Root_CA_M2	11/12/2037	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CAPF-trust	ACT2_SUDI_CA	CA-signed	RSA	ACT2_SUDI_CA	Cisco_Root_CA_2048	05/14/2029	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CAPF-trust	Cisco_Root_CA_2048	Self-signed	RSA	Cisco_Root_CA_2048	Cisco_Root_CA_2048	05/14/2029	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CAPF-trust	Cisco_Manufacturing_CA	CA-signed	RSA	Cisco_Manufacturing_CA	Cisco_Root_CA_2048	05/14/2029	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CAPF-trust	Cisco_Manufacturing_CA_SHA2	CA-signed	RSA	Cisco_Manufacturing_CA_SHA2	Cisco_Root_CA_M2	11/12/2037	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCM using the MIC when associated with a secure profile.
CAPF-trust	CAPF-616421bc	Self-signed	RSA	CAPF-616421bc	CAPF-616421bc	07/12/2025	
ipsec	cucm.steven.lab	Self-signed	RSA	cucm.steven.lab	cucm.steven.lab	07/12/2025	Self-signed certificate generated by system
ipsec-trust	cucm.steven.lab	Self-signed	RSA	cucm.steven.lab	cucm.steven.lab	07/12/2025	Trust Certificate
ITLRecovery	ITLRECOVERY_cucm.steven.lab	Self-signed	RSA	cucm.steven.lab	ITLRECOVERY_cucm.steven.lab	02/14/2039	Self-signed certificate generated by system
tomcat	cucm.steven.lab	CA-signed	RSA	cucm.steven.lab	steven-DC-CA	07/10/2024	Certificate Signed by steven-DC-CA
tomcat-ECDSA	cucm-EC.steven.lab	CSR Only	EC	cucm.steven.lab	--	--	
tomcat-ECDSA	cucm-EC.steven.lab	Self-signed	EC	cucm.steven.lab	cucm-EC.steven.lab	07/25/2023	Self-signed certificate generated by system
tomcat-trust	steven-DC-CA	Self-signed	RSA	steven-DC-CA	steven-DC-CA	06/01/2025	Trust Certificate
tomcat-trust	NOMAT-AD-CA	Self-signed	RSA	NOMAT-AD-CA	NOMAT-AD-CA	04/23/2028	Signed Certificate
tomcat-trust	cucm-EC.steven.lab	Self-signed	EC	cucm.steven.lab	cucm-EC.steven.lab	07/25/2023	Trust Certificate
tomcat-trust	cucm.steven.lab	CA-signed	RSA	cucm.steven.lab	steven-DC-CA	07/10/2024	Trust Certificate
tomcat-trust	cups-EC.steven.lab	Self-signed	EC	cups.steven.lab	cups-EC.steven.lab	07/25/2023	Trust Certificate
tomcat-trust	NOMAT-CA-10	Self-signed	RSA	NOMAT-CA-10	NOMAT-CA-10	08/11/2027	Signed Certificate
tomcat-trust	vngtp-ACTIVE-DIR-CA	Self-signed	RSA	vngtp-ACTIVE-DIR-CA	vngtp-ACTIVE-DIR-CA	02/10/2024	Trust Certificate
tomcat-trust	dccomics-WONDERWOMAN-CA	Self-signed	RSA	dccomics-WONDERWOMAN-CA	dccomics-WONDERWOMAN-CA	09/19/2037	CA Bruno
TVS	cucm.steven.lab	Self-signed	RSA	cucm.steven.lab	cucm.steven.lab	07/12/2025	Self-signed certificate generated by system

2.证书中不包含连接地址（FQDN或IP）

除了信任存储之外，流量服务器还验证MRA客户端向哪个连接地址发出请求。例如，当您在CUCM上的**System > Server**下设置CUCM时，您的CUCM的IP地址(10.48.36.215)，则Expressway-C会将此情况通告给客户端，并且来自客户端（通过Expressway-C代理）的后续请求会指向此地址。

当该特定连接地址未包含在服务器证书中时，TLS验证也会失败，并引发502错误，从而导致MRA登录失败。

```
2022-07-11T19:49:01.472+02:00 vcsc traffic_server[3916]: UTCTime="2022-07-11 17:49:01,472"
Module="network.http.trafficserver" Level="DEBUG": Detail="Receive Request" Txn-id="144"
TrackingID="0a334fa8-41e9-4b97-adf4-e165372c38cb" Src-ip="127.0.0.1" Src-port="30044" Last-via-
addr=""
HTTPMSG:
|GET http://vcs_control.steven.lab:8443/c3RldmVuLmxhYi9odHRwcy8xMC40OC4zNi4yMTUvODQ0Mw/cucm-
uds/user/emusk/devices?max=100 HTTP/1.1
...

2022-07-11T19:49:01.478+02:00 vcsc traffic_server[3916]: UTCTime="2022-07-11 17:49:01,478"
Module="network.http.trafficserver" Level="INFO": Detail="Sending Request" Txn-id="144"
TrackingID="0a334fa8-41e9-4b97-adf4-e165372c38cb" Dst-ip="10.48.36.215" Dst-port="8443" Msg="GET
/cucm-uds/user/emusk/devices?max=100 HTTP/1.1"
2022-07-11T19:49:01.478+02:00 vcsc traffic_server[3916]: UTCTime="2022-07-11 17:49:01,478"
Module="network.http.trafficserver" Level="DEBUG": Detail="Sending Request" Txn-id="144"
TrackingID="0a334fa8-41e9-4b97-adf4-e165372c38cb" Dst-ip="10.48.36.215" Dst-port="8443"
HTTPMSG:
|GET /cucm-uds/user/emusk/devices?max=100 HTTP/1.1
...

2022-07-11T19:49:01.491+02:00 vcsc traffic_server[3916]: [ET_NET 2] WARNING: SNI (10.48.36.215)
not in certificate. Action=Terminate server=10.48.36.215(10.48.36.215)
2022-07-11T19:49:01.491+02:00 vcsc traffic_server[3916]: [ET_NET 2] ERROR: SSL connection failed
for '10.48.36.215': error:1416F086:SSL routines:tls_process_server_certificate:certificate
verify failed
```

其中c3RldmVuLmxhYi9odHRwcy8xMC40OC4zNi4yMTUvODQ0Mw translate(base64 -

https://www.base64decode.org/)到steven.lab/https/10.48.36.215/8443，表明它必须建立到10.48.36.215的连接作为连接地址，而不是到cucm.steven.lab。如数据包捕获所示，CUCM tomcat证书不包含SAN中的IP地址，因此引发错误。

如何轻松验证

您可以通过以下步骤验证您是否容易遇到此行为更改：

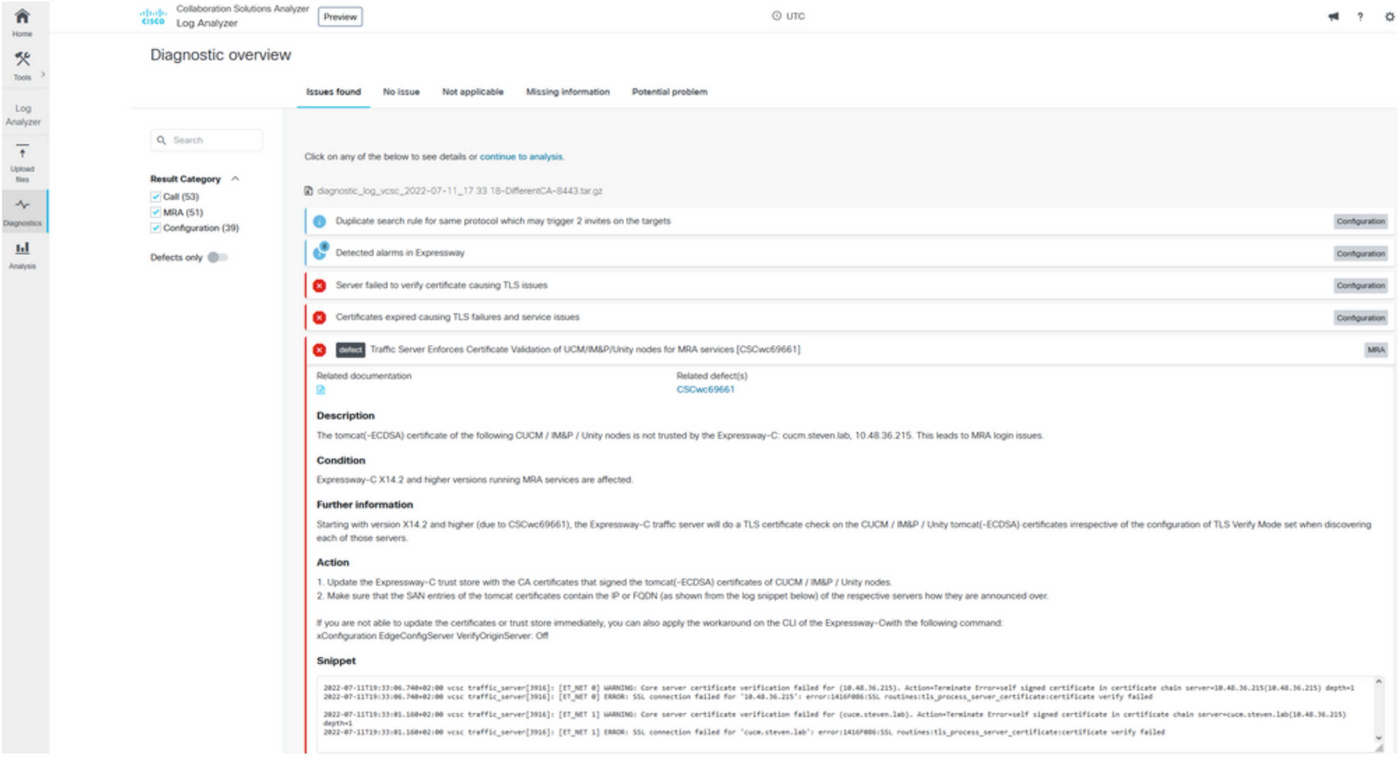
1.在Expressway-E和C服务器上启动诊断日志记录（最好启用TCPDumps），从**维护>诊断>诊断日志记录**（如果是集群，从主节点启动就足够了）

2.尝试MRA登录或在升级后测试中断的功能

3.等待失败，然后停止Expressway-E和C服务器上的诊断日志记录（如果是集群，请确保分别从集群的每个节点收集日志）

4.上传并分析协作解决方案分析器工具上的日志

5.如果遇到问题，它会为每个受影响的服务器选取与此更改相关的最新警告和错误行



Diagnostic overview

Click on any of the below to see details or continue to analysis.

diagnostic_log_vcsc_2022-07-11_17 33 18-DifferentCA-8443.tar.gz

Duplicate search rule for same protocol which may trigger 2 invites on the targets

Detected alarms in Expressway

Server failed to verify certificate causing TLS issues

Certificates expired causing TLS failures and service issues

Traffic Server Enforces Certificate Validation of UCM/IM&P/Unity nodes for MRA services [CSCwc69661]

Related documentation

Related defect(s)
CSCwc69661

Description

The tomcat(-ECDSA) certificate of the following CUCM / IM&P / Unity nodes is not trusted by the Expressway-C: cucm.steven.lab, 10.48.36.215. This leads to MRA login issues.

Condition

Expressway-C X14.2 and higher versions running MRA services are affected.

Further information

Starting with version X14.2 and higher (due to CSCwc69661), the Expressway-C traffic server will do a TLS certificate check on the CUCM / IM&P / Unity tomcat(-ECDSA) certificates irrespective of the configuration of TLS Verify Mode set when discovering each of those servers.

Action

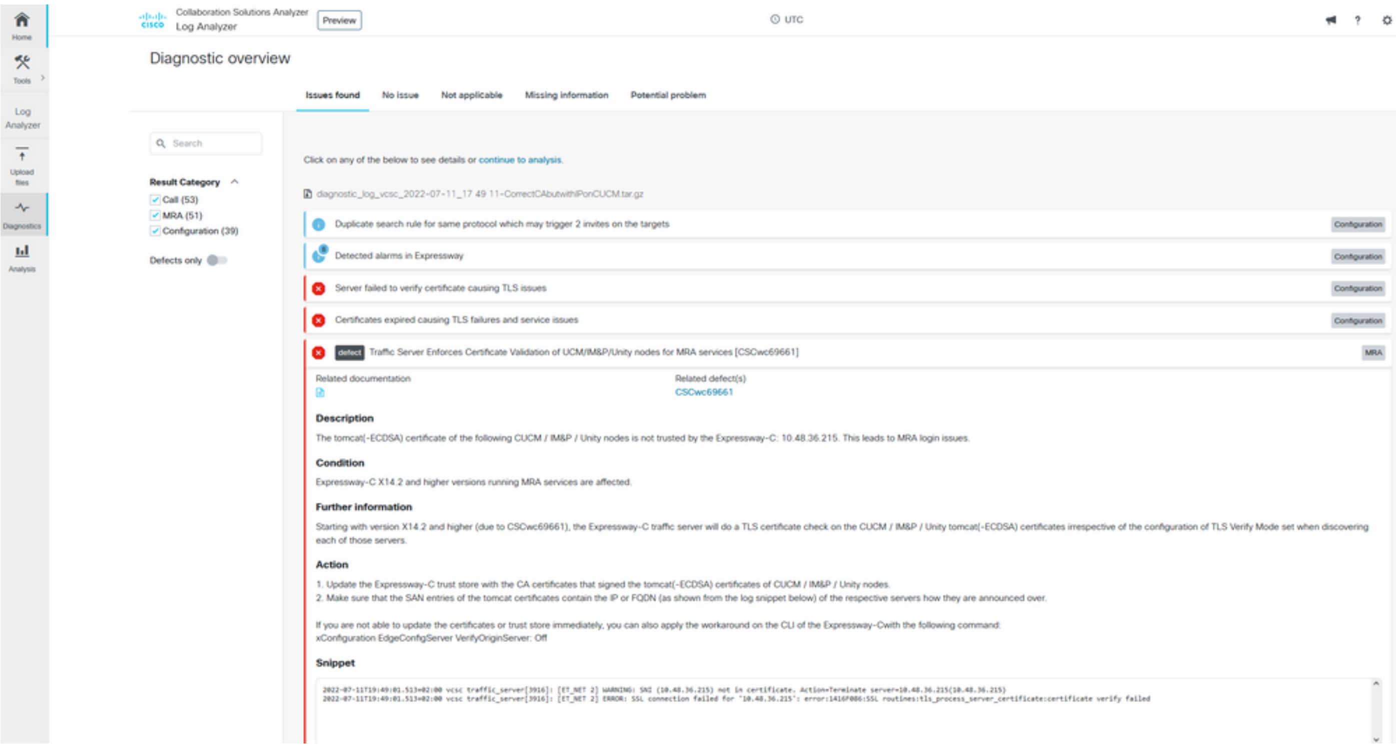
1. Update the Expressway-C trust store with the CA certificates that signed the tomcat(-ECDSA) certificates of CUCM / IM&P / Unity nodes.
2. Make sure that the SAN entries of the tomcat certificates contain the IP or FQDN (as shown from the log snippet below) of the respective servers how they are announced over.

If you are not able to update the certificates or trust store immediately, you can also apply the workaround on the CLI of the Expressway-Cwith the following command:
xConfiguration EdgeConfigServer VerifyOriginServer: Off

Snippet

```
2022-07-11T19:33:06.740+02:00 vcsc traffic_server[3916]: [ET_NET 0] WARNING: Core server certificate verification failed for (10.48.36.215). Action=Terminate Error=self signed certificate in certificate chain server=10.48.36.215(10.48.36.215) depth=1
2022-07-11T19:33:06.740+02:00 vcsc traffic_server[3916]: [ET_NET 0] ERROR: SSL connection failed for '10.48.36.215': error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed

2022-07-11T19:33:01.160+02:00 vcsc traffic_server[3916]: [ET_NET 1] WARNING: Core server certificate verification failed for (cucm.steven.lab). Action=Terminate Error=self signed certificate in certificate chain server=cucm.steven.lab(10.48.36.215)
depth=1
2022-07-11T19:33:01.160+02:00 vcsc traffic_server[3916]: [ET_NET 1] ERROR: SSL connection failed for 'cucm.steven.lab': error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed
```

CA诊断签名



Diagnostic overview

Issues found | No issue | Not applicable | Missing information | Potential problem

Result Category

- ☑ Call (53)
- ☑ MRA (51)
- ☑ Configuration (39)

Defects only

Click on any of the below to see details or continue to analysis.

diagnostic_log_vcsc_2022-07-11_17 49 11-CorrectCAbutwithIPonCUCM.tar.gz

- Duplicate search rule for same protocol which may trigger 2 invites on the targets
- Detected alarms in Expressway
- Server failed to verify certificate causing TLS issues
- Certificates expired causing TLS failures and service issues
- Traffic Server Enforces Certificate Validation of UCM/IM&P/Unity nodes for MRA services [CSCwc69661]

Related documentation

Related defect(s)
CSCwc69661

Description

The tomcat(-ECDSA) certificate of the following CUCM / IM&P / Unity nodes is not trusted by the Expressway-C: 10.48.36.215. This leads to MRA login issues.

Condition

Expressway-C X14.2 and higher versions running MRA services are affected.

Further information

Starting with version X14.2 and higher (due to CSCwc69661), the Expressway-C traffic server will do a TLS certificate check on the CUCM / IM&P / Unity tomcat(-ECDSA) certificates irrespective of the configuration of TLS Verify Mode set when discovering each of those servers.

Action

1. Update the Expressway-C trust store with the CA certificates that signed the tomcat(-ECDSA) certificates of CUCM / IM&P / Unity nodes.
2. Make sure that the SAN entries of the tomcat certificates contain the IP or FQDN (as shown from the log snippet below) of the respective servers how they are announced over.

If you are not able to update the certificates or trust store immediately, you can also apply the workaround on the CLI of the Expressway-Cwith the following command:
xConfiguration EdgeConfigServer VerifyOriginServer: Off

Snippet

```
2022-07-11T19:49:01.513+02:00 vcsc traffic_server[3916]: [ET_NET 2] WARNING: SNI (10.48.36.215) not in certificate. Action=Terminate server=10.48.36.215(10.48.36.215)
2022-07-11T19:49:01.513+02:00 vcsc traffic_server[3916]: [ET_NET 2] ERROR: SSL connection failed for '10.48.36.215': error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed
```

SNI诊断签名

解决方案

长远的解决方法是确保TLS验证正常工作。要执行的操作取决于显示的警告消息。

当您观察到*警告：(<server-FQDN-or-IP>)的核心服务器证书验证失败。 Action=Terminate Error=self signed certificate server=cucm.steven.lab(10.48.36.215)**depth=x*** message，然后您需要相应地更新Expressway-C服务器上的信任存储。使用签署此证书的CA链（深度> 0）或使用**Maintenance > Security > Trusted CA Certificate**中的自签名证书（深度= 0）。确保在群集中的每个服务器上执行此操作。另一种方法是，通过Expressway-C信任存储上的已知CA对远程证书进行签名。

当您观察到*警告：SNI(<server-FQDN-or-IP>)不在证书消息*中，则表示此服务器FQDN或IP未包含在提供的证书中。您可以调整证书以包含此信息，或者可以修改配置（例如在System > Server上的CUCM上，修改为服务器证书中包含的内容），然后刷新Expressway-C服务器上的配置以将其考虑在内。

短期解决方案是应用所记录的解决方法，以回退到X14.2.0之前的先前行为。您可以通过Expressway-C服务器节点上的CLI使用新引入的命令对此执行操作：

```
xConfiguration EdgeConfigServer VerifyOriginServer: Off
```

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。