CSCwc69661によって導入されたMRAサービスのためのExpresswayトラフィックサーバ証明書検証のトラブルシューティング

内容

概要

このドキュメントでは、Cisco Bug ID CSCwc69661にリンクされているExpresswayバージョンX14.2.0以降での動作の変更について説明します。この変更により、Expresswayプラットフォームのトラフィックサーバで、Mobile and Remote Access(MRA)サービス用のCisco Unified Communication Manager(CUCM)、Cisco Unified Instant Messaging & Presence(IM&P)、およびUnityサーバノードのの証明書検証がが実行されます。この変更により、Expresswayプラットフォームでのアップグレード後にMRAログインが失敗する可能性があります。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Expressway基本設定
- MRA基本設定

使用するコンポーネント

このドキュメントの情報は、バージョンX14.2以降のCisco Expresswayに基づくものです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

Hypertext Transfer Protocol Secure(HTTPS)は、Transport Layer Security(TLS)を使用して通信を暗号化するセキュアな通信プロトコルです。このセキュアチャネルは、TLSハンドシェイクで交換されるTLS証明書を使用して作成されます。この方法では、次の2つの目的を果たします。認証（接続先のリモートパーティを知るため）とプライバシー（暗号化）。 認証は中間者攻撃から保護し、プライバシーは攻撃者が通信を傍受して改ざんすることを防ぎます。

TLS（証明書）検証は認証の観点から実行され、適切なリモートパーティに接続していることを確認できます。検証は、次の2つの個別の項目で構成されます。

1.信頼された認証局(CA)チェーン

2.サブジェクト代替名(SAN)または共通名(CN)

信頼されたCAチェーン

Expressway-CがCUCM/IM&P/Unityが送信する証明書を信頼するには、その証明書から、信頼するトップレベル（ルート）認証局(CA)へのリンクを確立できる必要があります。このようなリンクは、エンティティ証明書をルートCA証明書にリンクする証明書の階層であり、信頼のチェーンと呼ばれます。このような信頼のチェーンを検証できるように、各証明書には2つのフィールド（証明書と証明書）が含まれています。Issuer（または'Issued by'）およびSubject（または'Issued To'）。

CUCMがExpressway-Cに送信するサーバ証明書などのサーバ証明書は、通常、[Subject]フィールドにCN内の完全修飾ドメイン名(FQDN)を持っています。

```
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のサーバ証明書の例SubjectフィールドのCN属性には、Country(C)、State(ST)、Location(L)、...などの他の属性と共にFQDNが含まれています。また、サーバ証明書がvngtp-ACTIVE-DIR-CAというCAによって配布（発行）されていることがわかります。

トップレベルCA（ルートCA）は、自身を識別する証明書を発行することもできます。このようなルートCA証明書では、IssuerとSubjectの値が同じであることがわかります。

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

これは、ルートCAが自身を識別するために配布する証明書です。

一般的な状況では、ルートCAはサーバ証明書を直接発行しません。代わりに、他のCAに対して証明書を発行します。このような他のCAは、中間CAと呼ばれます。中間CAは、サーバ証明書または他の中間CAの証明書を直接発行できます。サーバ証明書が中間CA 1によって発行され、次に中間CA 2から証明書が取得される状況が考えられます。最終的に中間CAがルートCAから直接その証明書を取得するまで（CAがルートCAから直接ルートCAを取得するまで）、

```
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
```

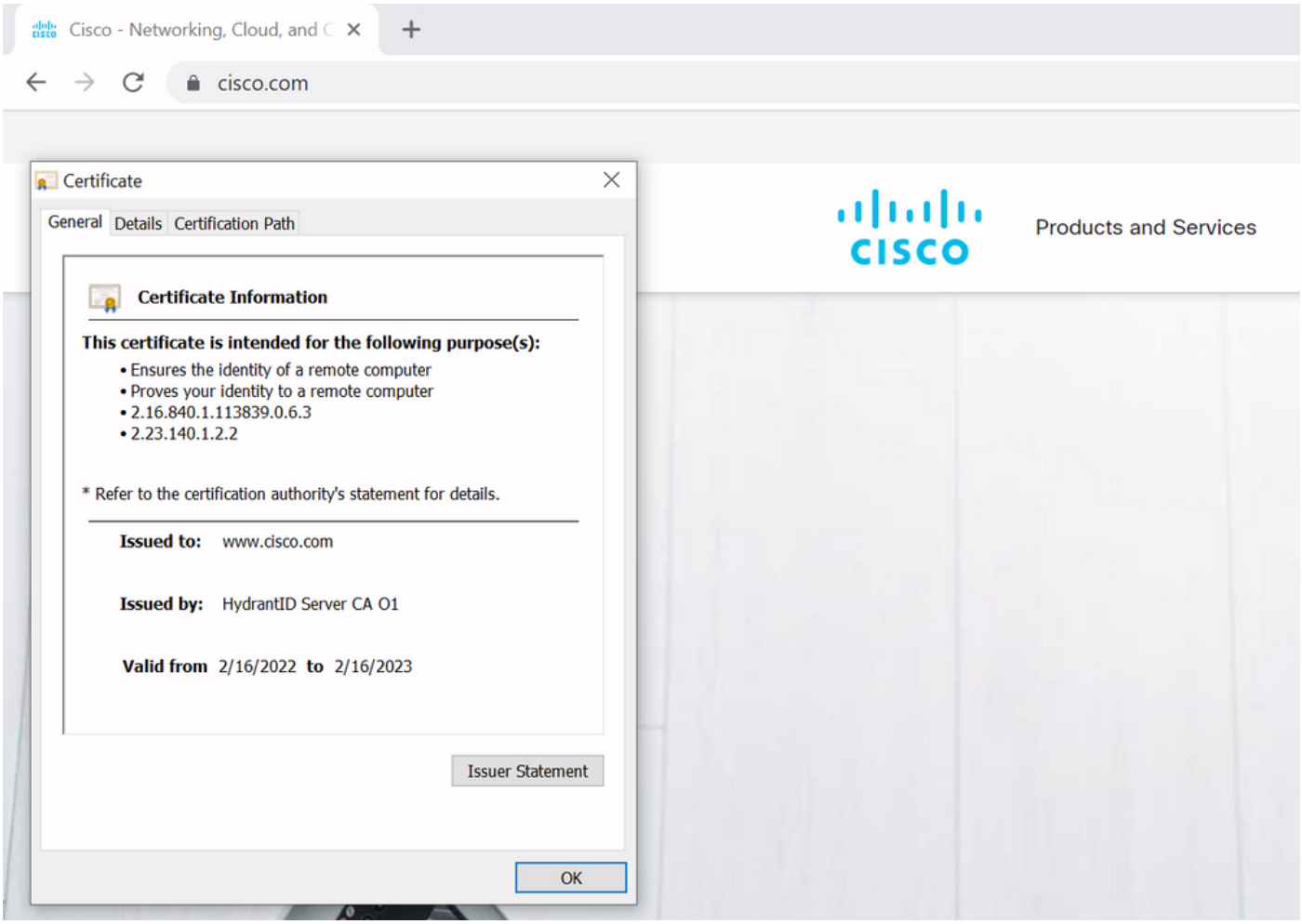
ここで、Expressway-CがCUCMが送信するサーバ証明書を信頼するためには、そのサーバ証明書からルートCA証明書までの信頼チェーンを構築できる必要があります。これを行うには、ルートCA証明書およびすべての中間CA証明書（存在する場合。ルートCAがCUCMのサーバ証明書を直接発行した場合は該当しません）をExpressway-Cの信頼ストアにアップロードする必要があります。

> **注：IssuerフィールドとSubjectフィールドは、人間が読める方法で信頼のチェーンを構築するのは簡単ですが、Expressway-CとCUCMは、証明書でこれらのフィールドを使用しません。代わりに、「X509v3 Authority Key Identifier」フィールドと「X509v3 Subject Key Identifier」フィールドを使用して信頼のチェーンを構築します。これらのキーには、Subject/Issuerフィールドを使用する方がより正確な証明書のIDが含まれています（CA証明書のIDはSubject/Issuerフィールドを使用する方が正確です）。同じSubject/Issuerフィールドを持つ2つの証明書を使用できますが、そのうち1つは期限切れであり、残りの1つは有効です。両方とも異なるX509v3 Subject Key IDを持つため、Expressway/CUCMは引き続き正しい信頼チェーンを決定できます。**

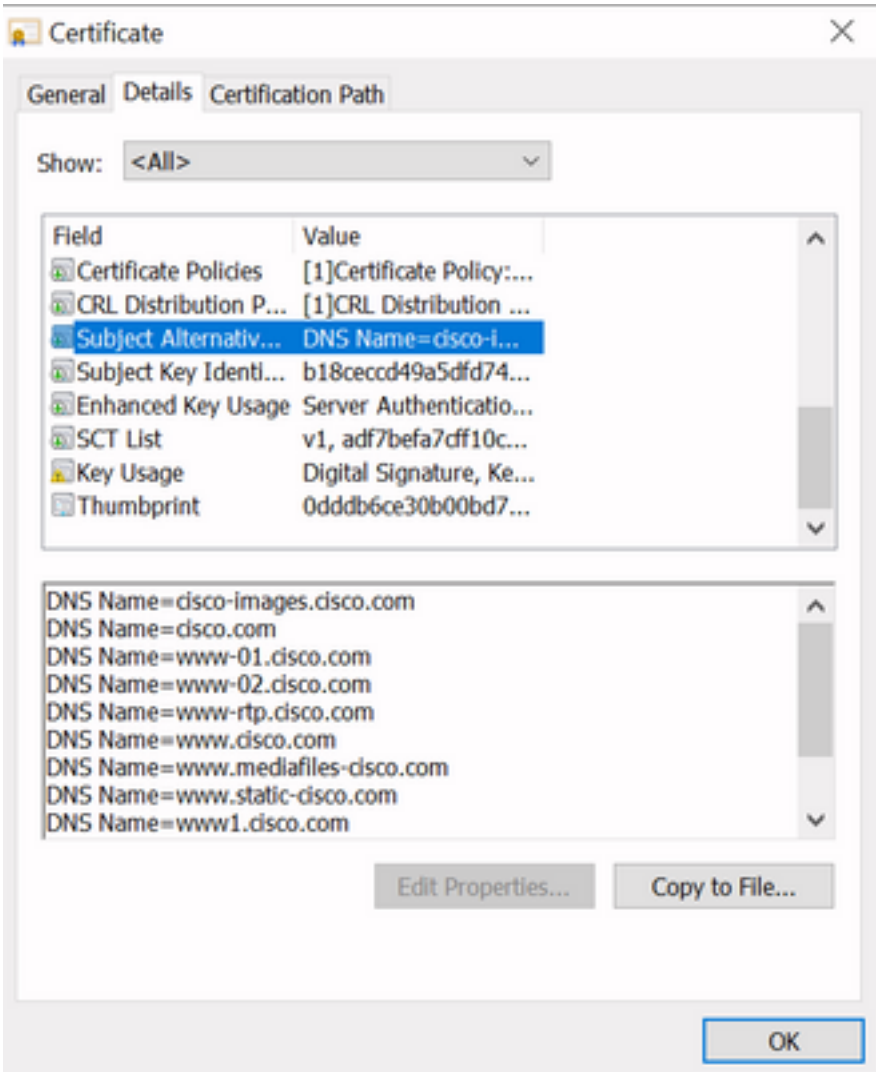
SANまたはCNのチェック

ステップ1は信頼ストアをチェックアウトしますが、信頼ストア内のCAによって署名された証明書を持つユーザは誰でも有効です。これは明らかに十分ではありません。したがって、具体的に接続するサーバが実際に正しいものであるかどうかを検証する追加のチェックがあります。これは、要求が行われたアドレスに基づいて行われます。

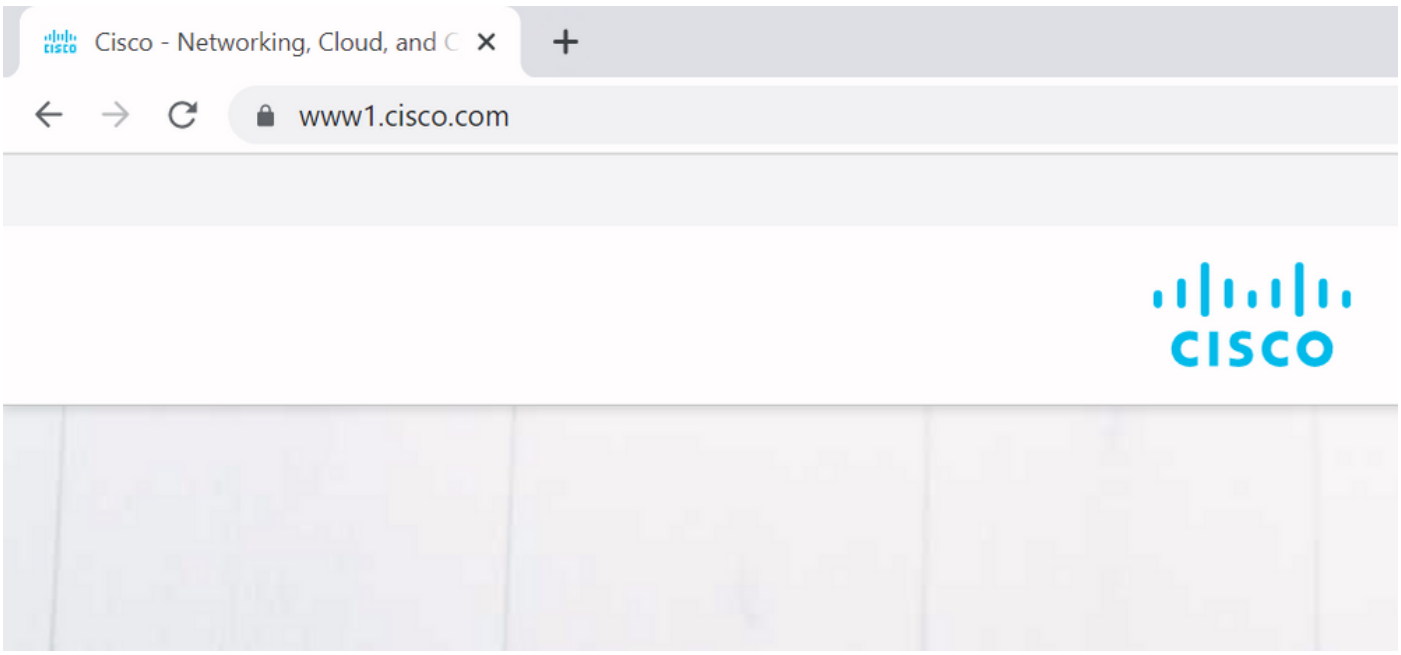
ブラウザでも同様の操作が行われるので、例を見てみます。https://www.cisco.comを参照すると、入力したURLの横にロックアイコンが表示され、そのURLが信頼できる接続であることを示します。これは、CA信頼チェーン（最初のセクションから）とSANまたはCNチェックの両方に基づいています。証明書を開くと（ブラウザでロックアイコンをクリックして）、共通名（[Issued to:]フィールドに表示）がwww.cisco.comに設定されており、接続先のアドレスに正確に対応していることがわかります。このようにして、正しいサーバに確実に接続できます（証明書に署名し、証明書を配布する前に検証を実行するCAを信頼するため）。



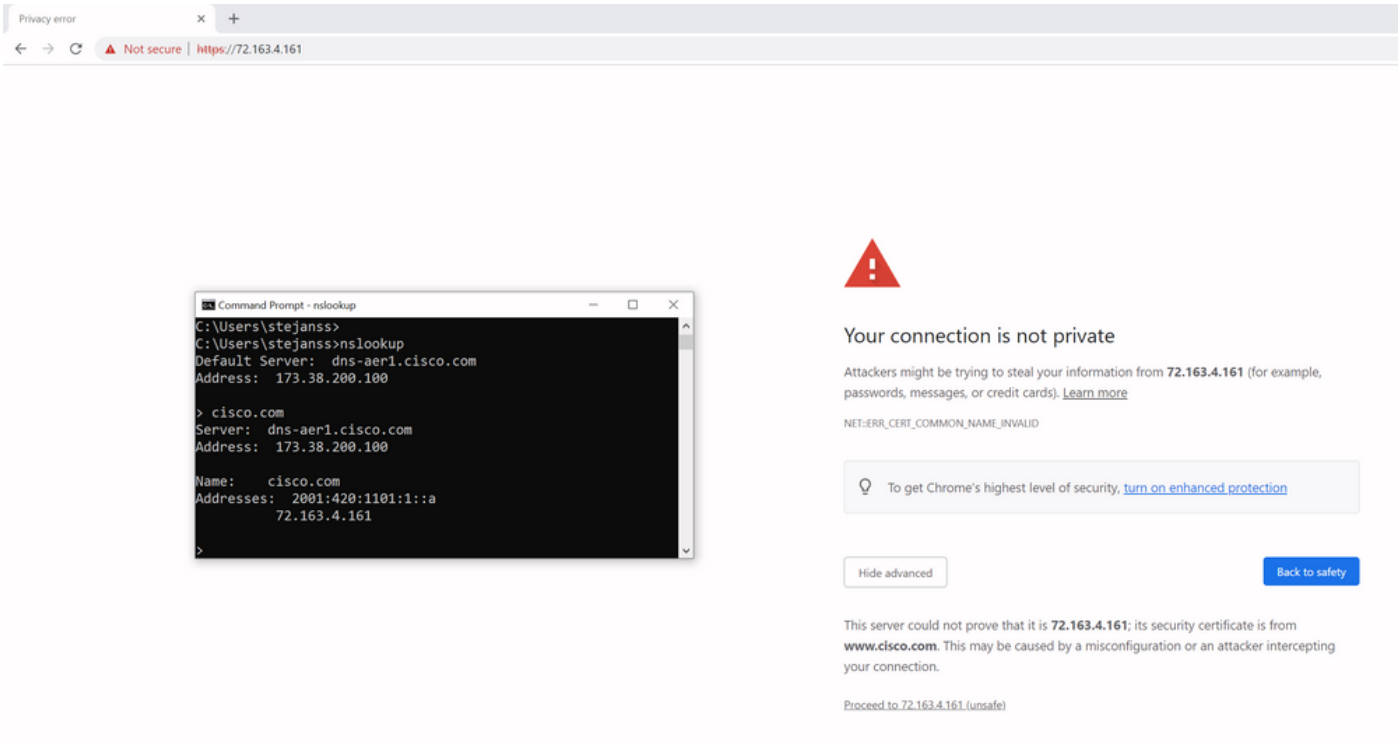
証明書、特にSANエントリの詳細を見ると、同じことが他のFQDNと同様に繰り返されていることがわかります。



つまり、たとえばhttps://www1.cisco.comへの接続を要求すると、SANエントリに含まれているため、セキュアな接続として表示されます。



ただし、https://www.cisco.comをブラウズせず、直接IPアドレス(https://72.163.4.161)をブラウズする場合、署名したCAを信頼しますが、提示された証明書にはサーバへの接続に使用したアドレス(72.163.4.161)が含まれないため、セキュアな接続は表示されません。



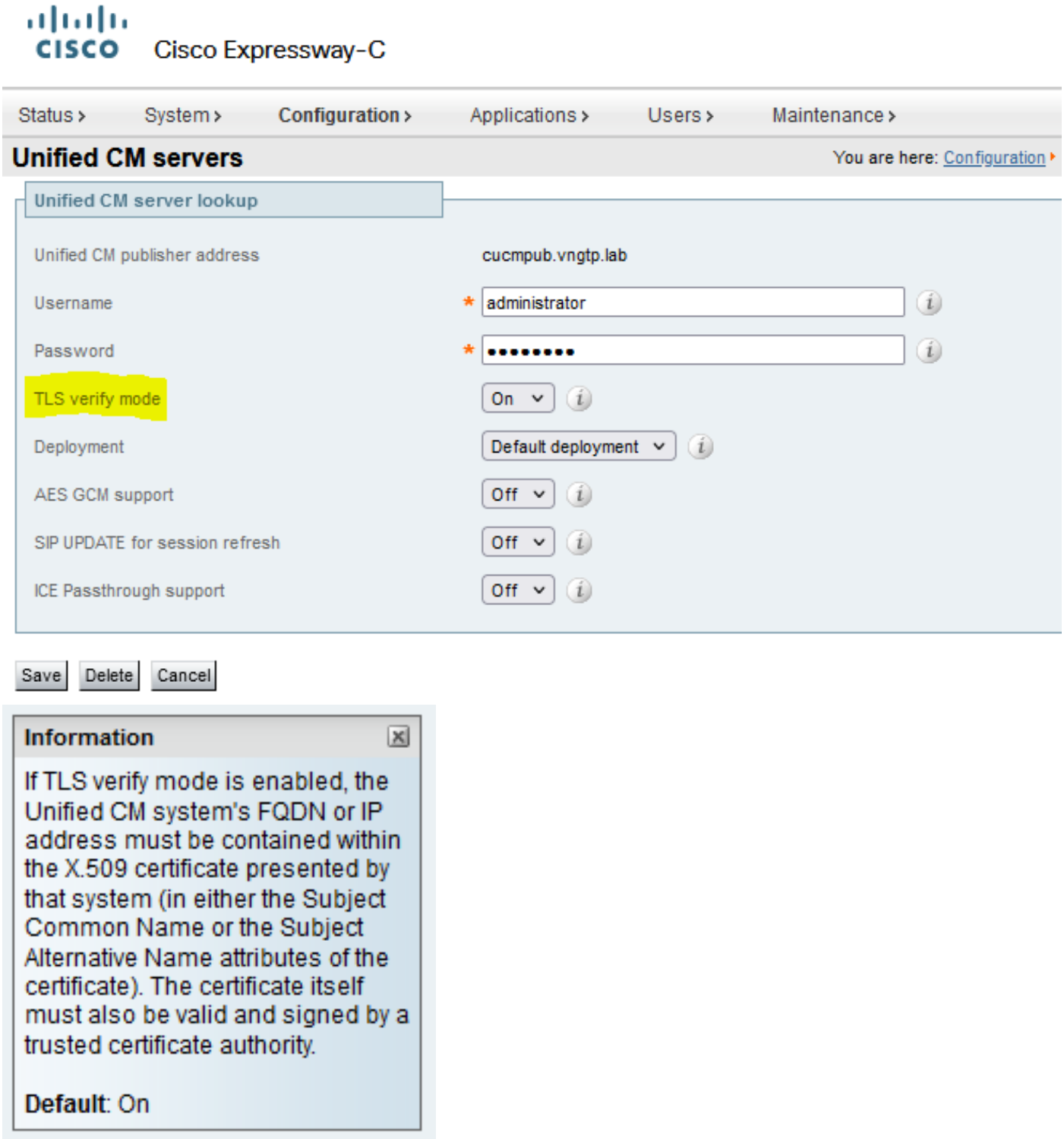
ブラウザでは、これをバイパスできますが、バイパスが許可されていないTLS接続で有効にできる設定です。したがって、証明書には、リモート側が接続に使用する予定の正しいCNまたはSAN名が含まれていることが重要です。

動作の変更

MRAサービスは、CUCM/IM&P/Unityサーバに対するExpressway経由の複数のHTTPS接続に大きく依存して、適切に認証し、ログインするクライアントに固有の適切な情報を収集します。通常、この通信はポート8443および6972で行われます。

X14.2.0より前のバージョン

X14.2.0より前のバージョンでは、これらのセキュアなHTTPS接続を処理するExpressway-Cのトラフィックサーバは、リモートエンドから提示された証明書を確認しませんでした。これは、中間者攻撃につながる可能性があります。MRA設定では、[Configuration] > [Unified Communications] > [Unified CM servers] > [IM and Presence Service nodes] / [Unity Connection servers] でCUCM / IM&P / Unityサーバを追加するときに、[TLS Verify Mode]の設定でTLS証明書の検証を[On]にするオプションがあります。例として、設定オプションと関連情報ボックスが示されています。この例では、証明書の有効性と、信頼できるCAによって署名されているかどうかだけでなく、SAN内のFQDNまたはIPを確認することが示されています。



このTLS証明書検証チェックは、CUCM/IM&P/Unityサーバの検出時にのみ実行され、MRAログイン時にさまざまなサーバが照会されることはありません。この設定の最初の欠点は、追加したパブリッシャアドレスに対してだけ確認されることです。サブスクライバノード上の証明書が正しく設定されているかどうかを検証しません。これは、サブスクライバノード情報（FQDNまたはIP）をパブリッシャノードのデータベースから取得するためです。この設定の2つ目の欠点は、接続情報がExpressway-C設定に設定されたパブリッシャアドレスと異なる場合があるため、MRAクライアントにアドバタイズされる情報が異なる場合があることです。たとえば、CUCMの[System] > [Server] で、IPアドレス（10.48.36.215など）を使用してサーバをアドバタイズし、これがMRAクライアントによって（プロキシExpressway接続を介して）使用されますが、CUCM.steven.labのFQDNを使用してExpressway-CのCUCMに追加できます。CUCMのtomcat証明書にSANエントリとしてcucm.steven.labが含まれていて、IPアドレスが含まれていないと仮定すると、「TLS Verify Mode」が「On」に設定されたディスカバリは成功しますが、MRAクライアントからの実際の通信は異なるFQDNまたはIPを対象とすることができ、TLS検証は失敗しま

す。

X14.2.0以降のバージョン

X14.2.0バージョン以降、Expresswayサーバは、トラフィックサーバを介して行われるすべてのHTTPS要求に対して、TLS証明書の検証を実行します。つまり、CUCM/IM&P/Unityノードの検出中に「TLS Verify Mode」が「Off」に設定されている場合にも、これが実行されます。検証が成功しない場合、TLSハンドシェイクは完了せず、要求が失敗します。これにより、冗長性やフェールオーバーの問題などの機能が失われたり、ログインが完全に失敗したりする可能性があります。また、[TLS Verify Mode]を[On]に設定しても、すべての接続が後の例で説明するように正常に動作するとは保証されません。

TLS検証のデフォルト以外にも、X14.2で導入された変更があり、暗号リストの異なる優先順位をアドバタイズします。アップグレード前にCUCM（またはECDSAアルゴリズム用の個別の証明書を持つその他の製品）からCisco TomcatまたはCisco CallManager証明書を要求し、アップグレード後にECDSAバリアントを要求することが原因で、ソフトウェアアップグレード後に予期しないTLS接続が発生する可能性があります。Cisco Tomcat-ECDSA証明書またはCisco CallManager-ECDSA証明書は、別のCAによって署名することも、自己署名証明書だけで署名することもできます（デフォルト）。

このシナリオでTLS検証が失敗する可能性がある方法は2つあります。これについては後で詳しく説明します。

1.リモート証明書に署名したCAが信頼されていない

a.自己署名証明書

b.不明なCAによって署名された証明書

2.接続アドレス（FQDNまたはIP）が証明書に含まれていない

シナリオのトラブルシューティング

次のシナリオは、ExpresswayをX14.0.7からX14.2にアップグレードした後にMRAログインが失敗したラボ環境での同様のシナリオを示しています。これらのシナリオではログの類似点が共有されていますが、解決策は異なります。ログは、MRAログイン前に開始され、MRAログインが失敗した後に停止された診断ログ([Maintenance] > [Diagnostics] > [Diagnostic logging] から)によって収集されます。追加のデバッグロギングは有効になっていません。

1.リモート証明書に署名したCAが信頼されていない

リモート証明書は、Expressway-Cの信頼ストアに含まれていないCAによって署名されているか、Expressway-Cサーバの信頼ストアに追加されていない自己署名証明書（本質的にはCA）である可能性があります。

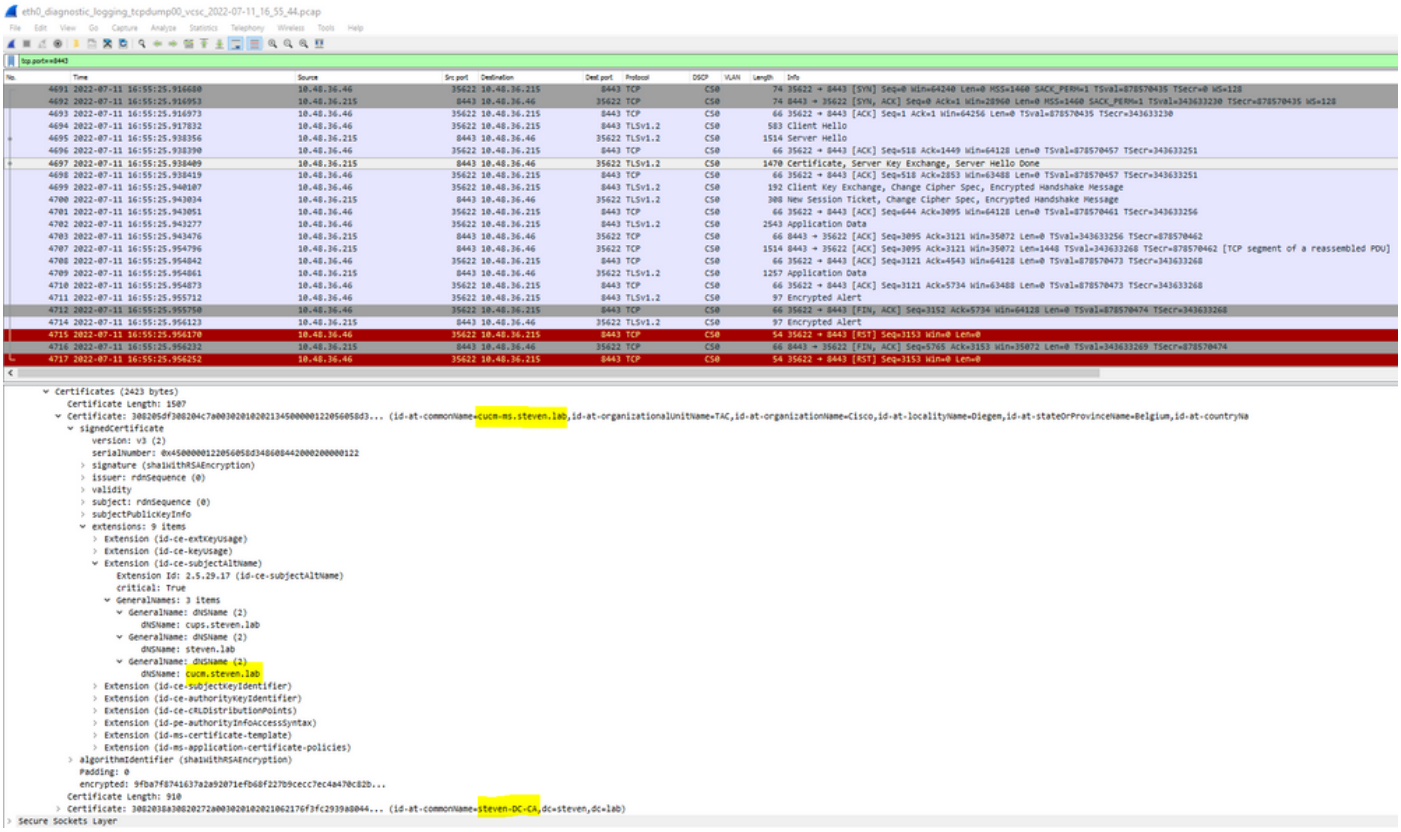
次の例では、CUCMに送信される要求(10.48.36.215 - cucm.steven.lab)がポート8443（200 OK応答）で正しく処理されるが、ポート6972でTFTP接続に対してエラー（502応答）がスローされることを確認できます。

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

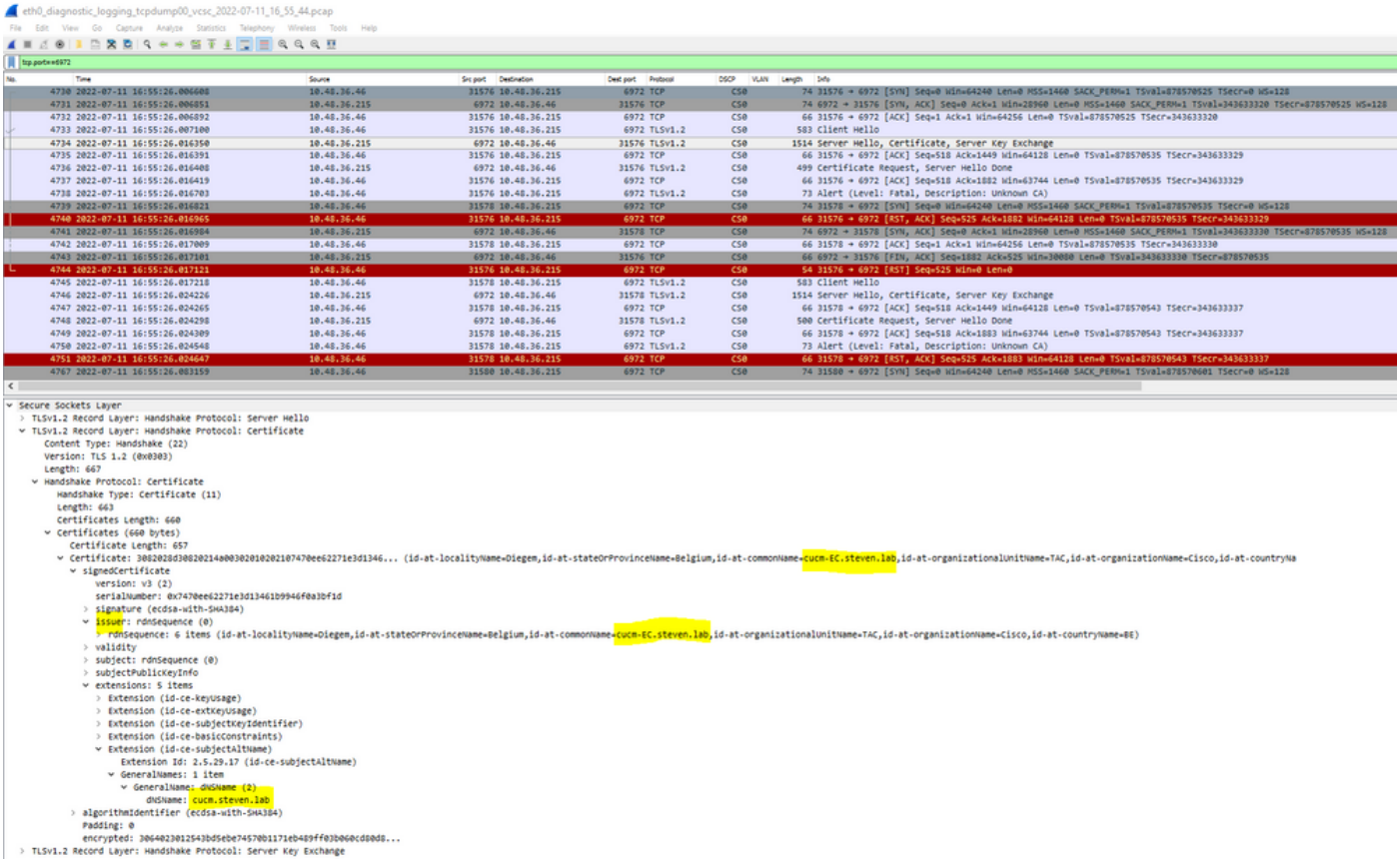
```
2022-07-11T18:55:25.910+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:25,910"
Module="network.http.trafficserver" Level="INFO": Detail="Receive Request" Txn-id="189"
TrackingID="6af9a674-9ebc-41ea-868e-90e7309a758c" Src-ip="127.0.0.1" Src-port="35764" Last-via-
addr="" Msg="GET
http://vcs_control.steven.lab:8443/c3RldmVuLmxhYi9odHRwcy9jdWNtLnN0ZXZlbi5sYWIvODQ0Mw/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」というエラーは、Expressway-CがTLSハンドシェイクを検証できなかったことを示しています。その理由は、自己署名証明書を示すため、警告行に表示されます。深さが0と表示されている場合は、自己署名証明書です。深さが0より大きい場合は、証明書チェーンが存在し、不明なCAによって署名されていることを意味します（Expressway-Cの観点から）。

テキストログから示されたタイムスタンプで収集されたpcapファイルを調べると、CUCMがCNを持つ証明書を、steven-DC-CAによって署名されたcucm-ms.steven.lab（およびSANとしてcucm.steven.lab）として、ポート8443のExpressway-Cに提示していることがわかります。

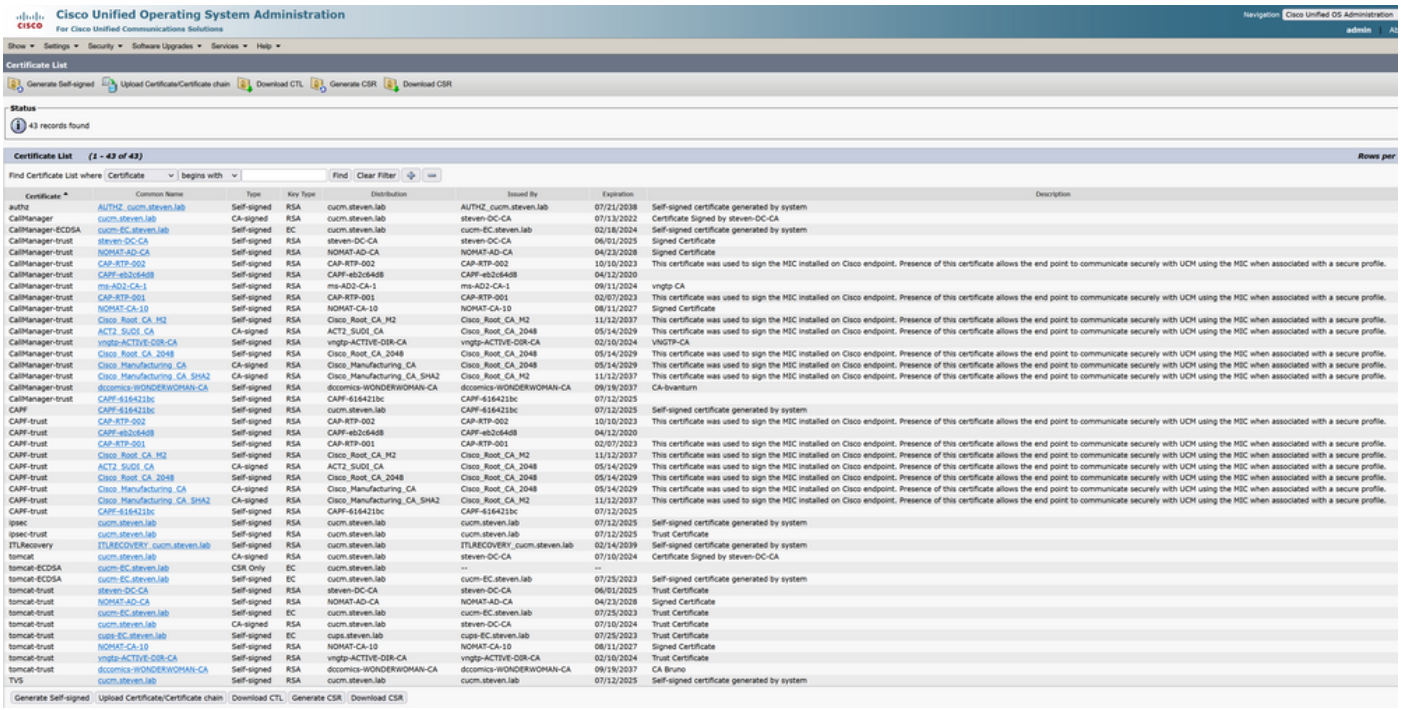


ただし、ポート6972で提示された証明書を調べると、cucm-EC.steven.labとして設定されたCNを持つ自己署名証明書（Issuer自体）であることがわかります。-EC拡張は、これがCUCMで設定されたECDSA証明書であることを示します。



CUCMの[Cisco Unified OS Administration]で、次に示すように、[Security] > [Certificate Management]の下に配置されている証明書を確認できます。tomcatとtomcat-ECDSA用の異なる証明書が表示されます。この証明書では、tomcatはCA署名され（Expressway-Cによって信頼さ

れます）、tomcat-ECDSA証明書は自己署名され、Expressway-Cによって信頼されません。



2.接続アドレス（FQDNまたはIP）が証明書に含まれていない

信頼ストアとは別に、トラフィックサーバはMRAクライアントが要求を行う接続アドレスも確認します。たとえば、[System] > [Server] のCUCMでIPアドレス(10.48.36.215)を使用してCUCMを設定すると、Expressway-Cがそのようにクライアントにアドバタイズし、クライアントからの後続の要求（Expressway-Cを介してプロキシ送信）がこのアドレスに対してターゲットになります。

その特定の接続アドレスがサーバ証明書に含まれていない場合、TLS検証も失敗し、たとえばMRAログイン失敗の原因となる502エラーがスローされます。

```
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**

c3RldmVuLmxhYi9odHRwcy8xMC40OC4zNi4yMTUvODQ0Mwは(base64 - https://www.base64decode.org/)をsteven.lab/https/10.48.36.215/8443に変換します。これは、接続アドレスとしてcucm.steven.labではなく10.48.36.215に接続する必要があることを示しています。パケットキャプチャに示されているように、CUCM tomcat証明書にはSANのIPアドレスが含まれていないため、エラーがスローされます。

簡単に検証する方法

次の手順で、この動作が簡単に変更されるかどうかを検証できます。

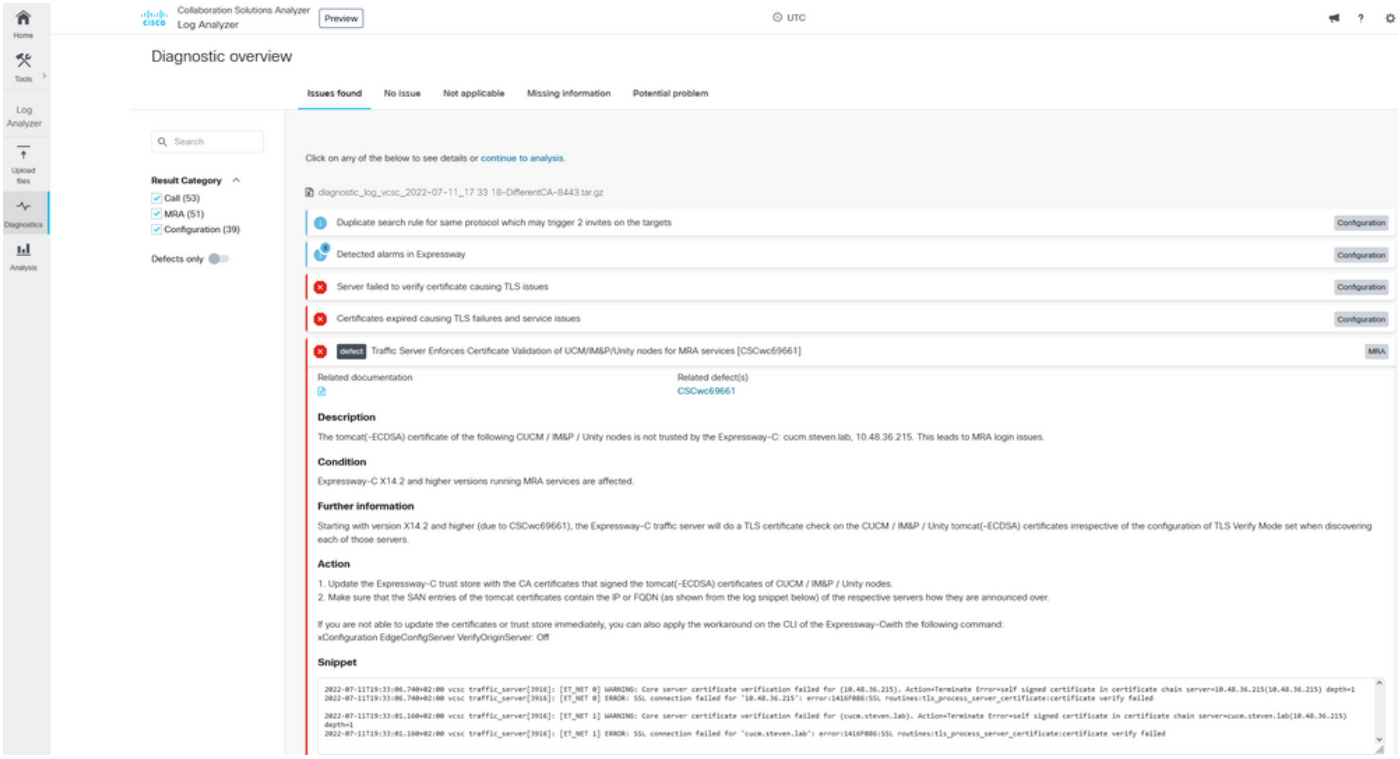
1. Expressway-EおよびCサーバで（TCPDumpsが有効な状態で）診断ロギングを[Maintenance] > [Diagnostics] > [Diagnostic Logging] から開始します（クラスタの場合は、マスターノードから開始するだけで十分です）。

2.アップグレード後にMRAログインを試行するか、中断された機能をテストします

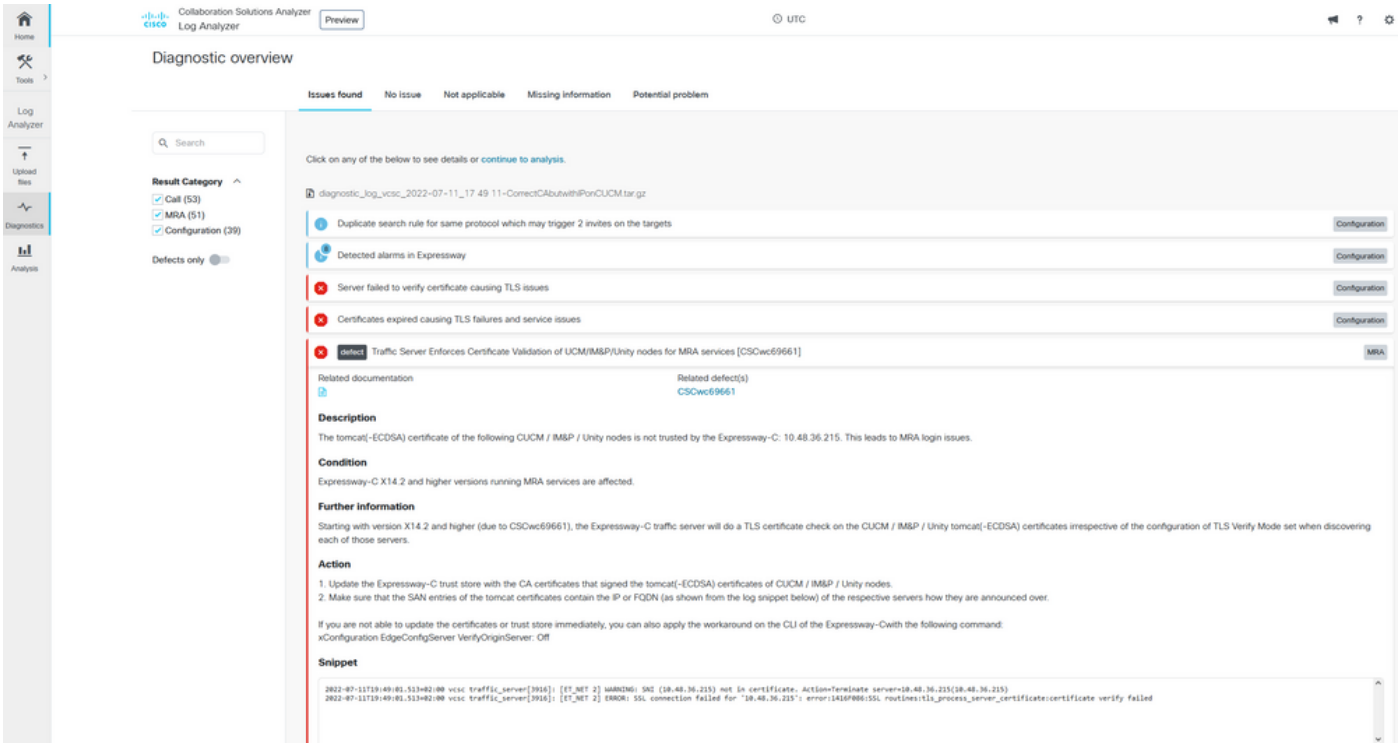
3.失敗するまで待ってから、Expressway-EおよびCサーバの診断ログを停止します（クラスタの場合は、クラスタの各ノードから個別にログを収集してください）。

4. Collaboration Solution Analyzerツールでログをアップロードして分析する。

5.問題が発生した場合は、影響を受けた各サーバについて、この変更に関連する最新の警告およびエラー行がピックアップされます



CA診断シグニチャ



SNI診断シグニチャ

解決方法

長期的な解決策は、TLS検証が正常に機能することを確認することです。実行するアクションは、表示される警告メッセージによって異なります。

WARNING:(<server-FQDN-or-IP>)のコアサーバー証明書の検証に失敗しました。 Action=Terminate Error=self signed certificate server=cucm.steven.lab(10.48.36.215) ***depth=xメッセージが表示されたら、それに従ってExpressway-Cサーバの信頼ストアを更新する必要があります。***この証明書に署名したCAチェーン(depth > 0)を使用するか、[Maintenance] > [Security] > [Trusted CA Certificate] から自己署名証明書(depth = 0)を使用します。クラスタ内のすべてのサーバでこのアクションを実行してください。別のオプションとして、Expressway-C信頼ストア上の既知のCAによってリモート証明書に署名する方法があります。

WARNING:SNI (<server-FQDN-or-IP>) not in certificate」というメッセージが表示された場合は、提示された証明書にこのサーバのFQDNまたはIPが含まれていないことを示しています。この情報を含むように証明書を適合させることも、設定を変更して（CUCMの[System] > [Server]でサーバ証明書に含まれるものに変更するなど）、Expressway-Cサーバの設定を更新して、この設定を反映させることもできます。

短期的な解決策は、文書化されている回避策を適用して、X14.2.0より前の以前の動作にフォールバックすることです。新しく導入されたコマンドを使用して、Expressway-CサーバノードのCLIから実行できます。

```
xConfiguration EdgeConfigServer VerifyOriginServer: Off
```

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。