Expresswayトラフィックサーバ証明書の検証のトラブルシューティング

内容

[はじめに](#)

[前提条件](#)

- [要件](#)
- [使用するコンポーネント](#)

[背景説明](#)

- [信頼済みCAチェーン](#)
- [SANまたはCNの確認](#)

[動作の変更](#)

- [X14.2.0より前のバージョン](#)
- [X14.2.0以降のバージョン](#)

[シナリオのトラブルシューティング](#)

- [1. リモート証明書に署名したCAが信頼されていない](#)
- [2. 証明書に接続アドレス（FQDNまたはIP）が含まれていない](#)

[簡単に検証する方法](#)

[解決方法](#)

[関連情報](#)

はじめに

このドキュメントでは、Cisco Bug ID CSCwc69661またはCisco Bug ID CSCwa25108に関連する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。

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

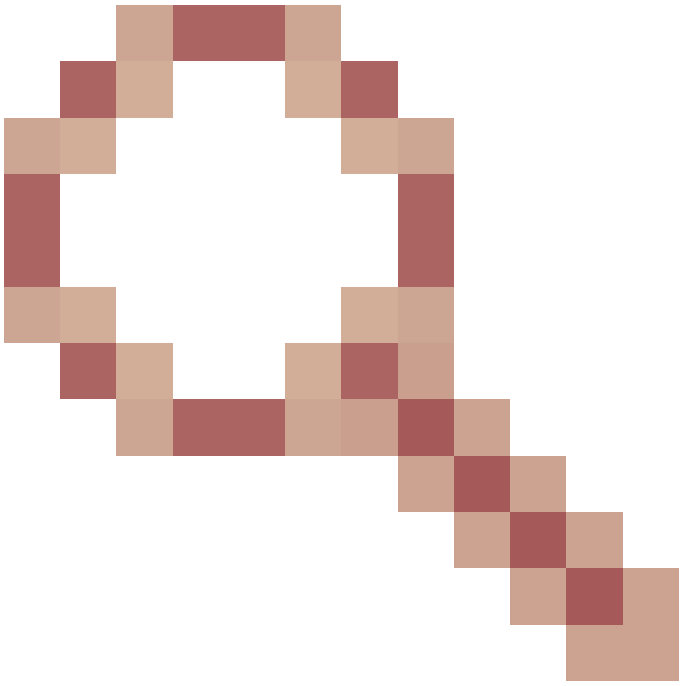
使用するコンポーネント

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

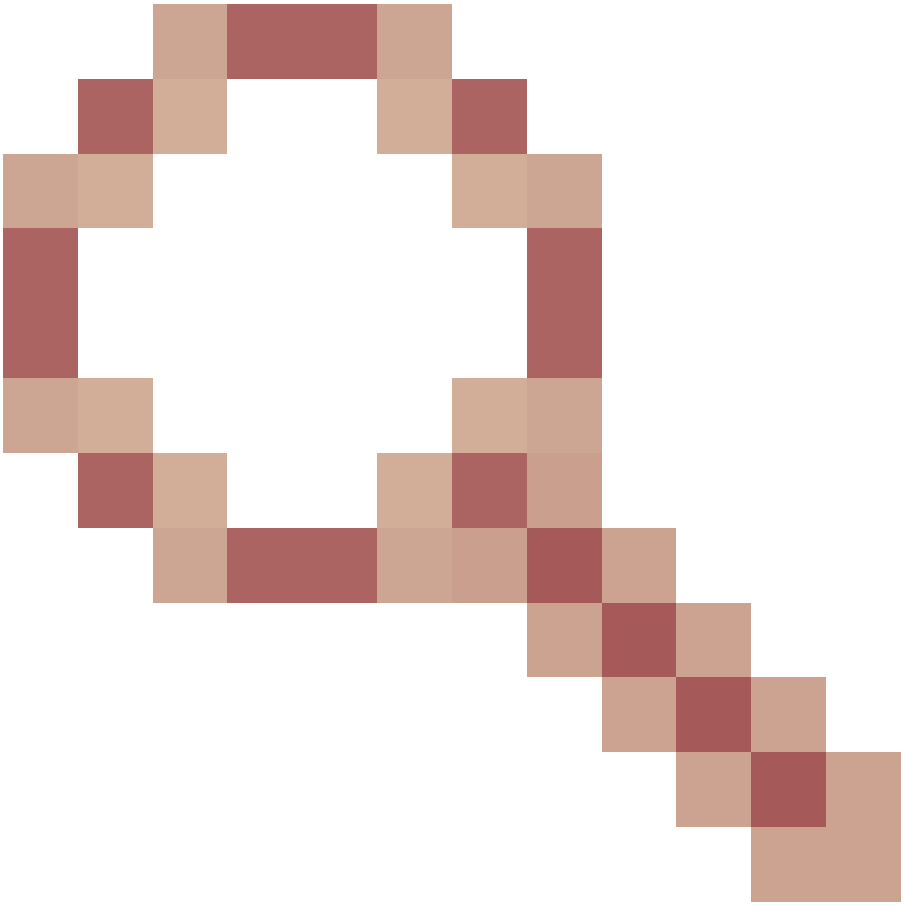
このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始していま

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



この動作変更に伴い、Cisco Bug ID CSCwc69661



またはCisco Bug ID CSCwa25108
は、Expresswayプラットフォーム上のトラフィックサーバが、Mobile and Remote Access(MRA)サービスのCisco Unified Communication Manager(CUCM)、Cisco Unified Instant Messaging & Presence(IM&P)、およびUnityサーバノードの証明書検証を実行します。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。

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。これは、認証（リモート側の接続先を知ること）とプライバシー（暗号化）の2つの目的に使用されます。 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たり改ざんしたりするのを防ぎます。

TLS（証明書）検証は認証の観点で実行され、正しいリモートパーティに接続した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)

信頼済み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、発行者（または発行者）とサブジェクト（または発行先）の2つのフィールドが含まれています。

CUCMがExpressway-Cに送信する証明書などのサーバ証明書は、通常、サブジェクト(FQDN)フィールドがCNの完全修飾ドメイン名(FQDN)になっています。

```
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、Country(C)、State(ST)、Location(L)などの他の属性とともにFQDNが含まれています。また、サーバ証明書がvngtp-ACTIVE-DIR-CAという名前のCAによって配布（発行）されていることも確認できます。

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が同じ値を持ってい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の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用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ような状況が考えられます。最後まで、中間CAはルートCAから自身の証明書を直接取得します。

```
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から送信され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。そのためには、ルートCA証明書およびすべての中間CA証明書（中間CA証明書が存在する場合は、ルートCAがCUCMのサーバ証明書を直接発行した場合を除く）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、信頼のチェーンを人間が読める方法で簡単に構築できますが、CUCMでは、証明書に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より正確な証明書の識別子が含まれています。同じSubject/Issuerフィールドを持つ2つの証明書が存在できますが、そのうちの1つは期限切れであり、1つは有効です。両方とも異なるX509v3のSubject Key Identifier(PKID)を持つため、CUCMは正しい信頼チェーンを引き続き判別できます。

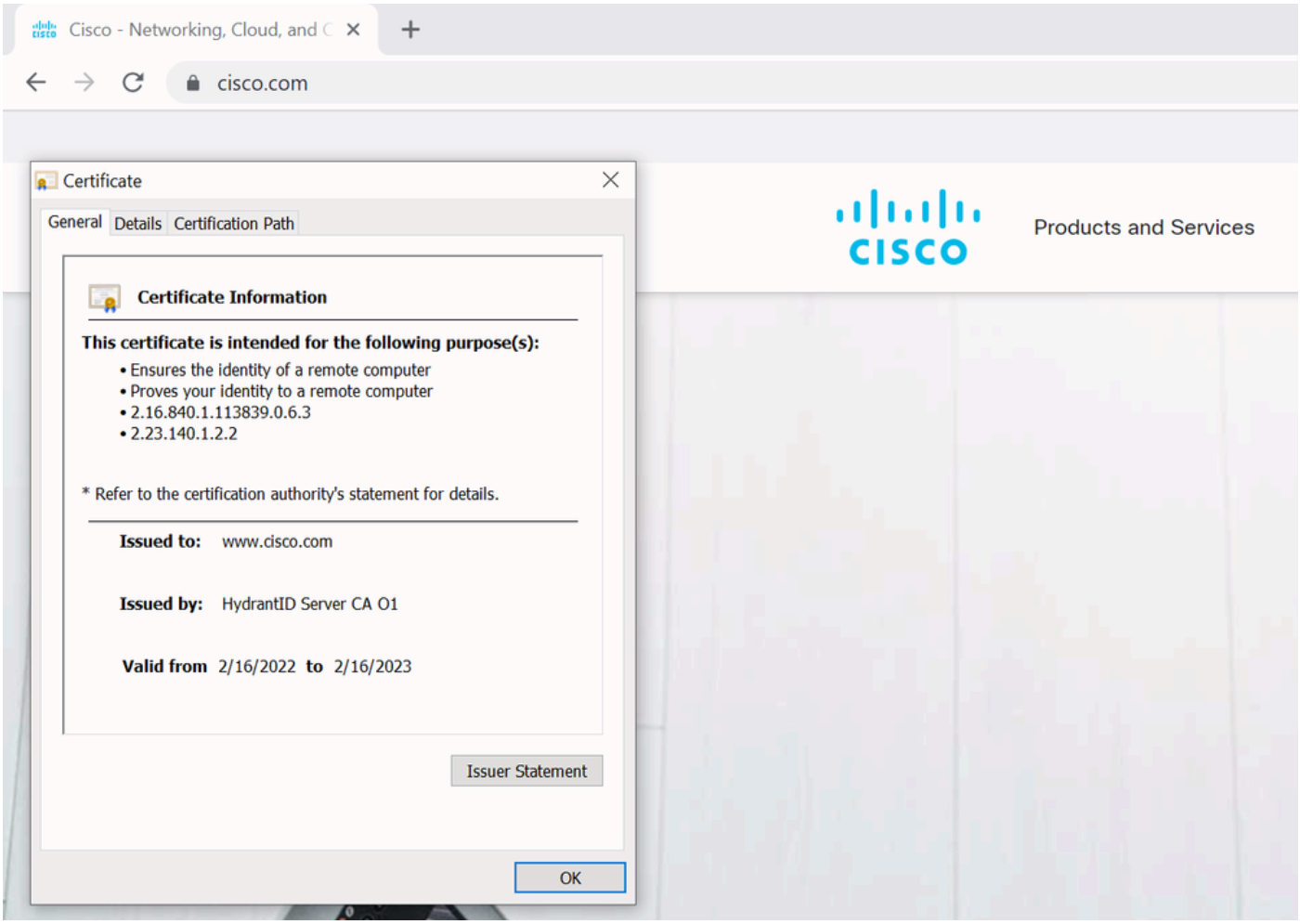
Cisco Bug ID CSCwa12905によると、これはExpresswayには当てはまらず、同じ共通名(CN)を持つ2つの異なる（たとえば自己署名された）証明書をExpresswayの信頼ストアにアップロードすることはできません。 これを修正する方法は、CA署名付き証明書を使用するか、異なる共通名(CN)を使用するか、または常に同じ証明書を使用することを確認します（CUCM 14の再使用証明書機能を通じて可能です）。

SANまたはCNの確認

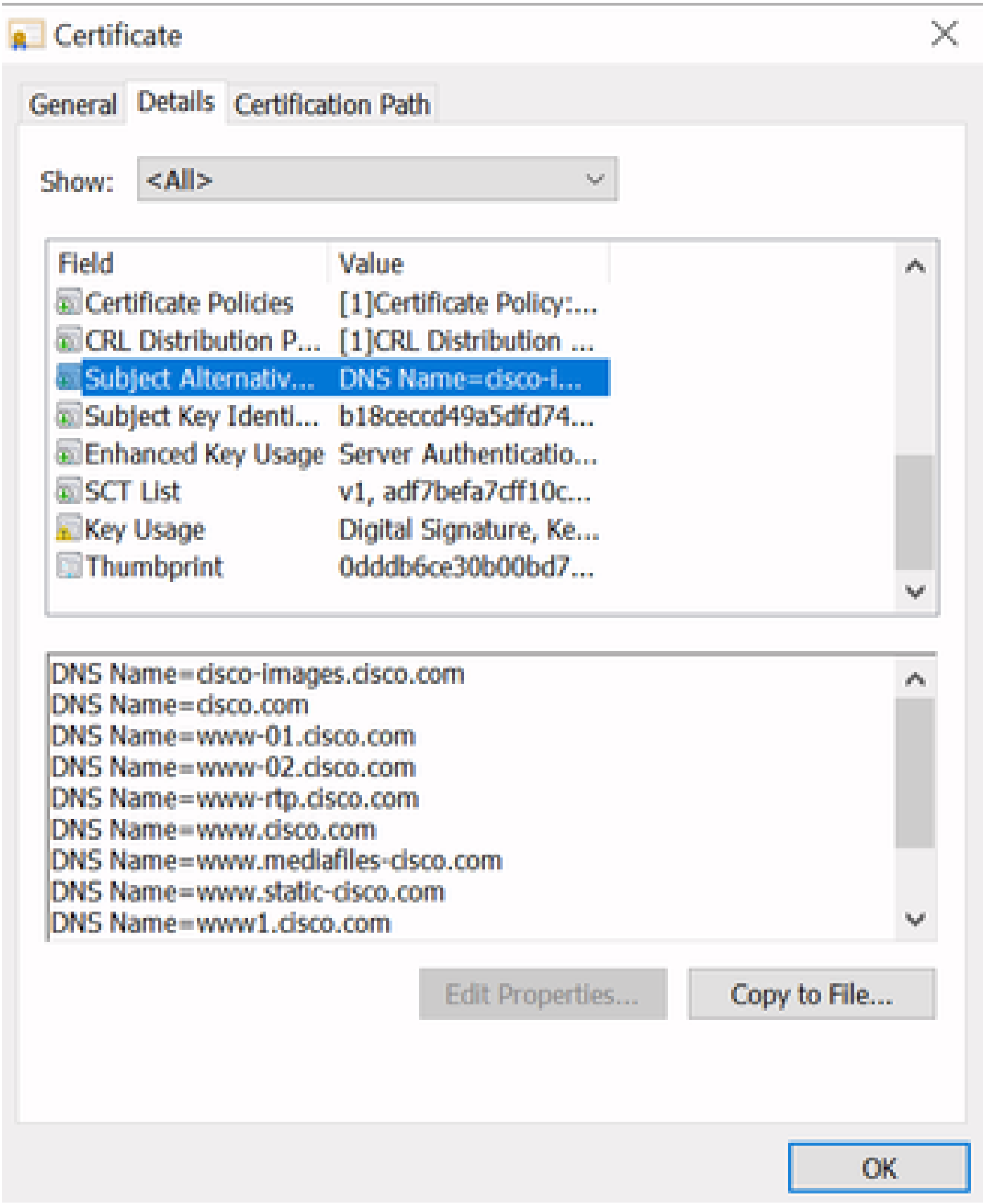
ステップ1は信頼ストアをチェックアウトしますが、信頼ストア内のCAによって署名された証明書を持つユーザは誰でも有効になります。これは明らかに十分ではありません。したがって、特に接続しているサーバが正しいサーバであることを検証する追加のチェックがあります。これは

、要求が行われたアドレスに基づいて行われます。

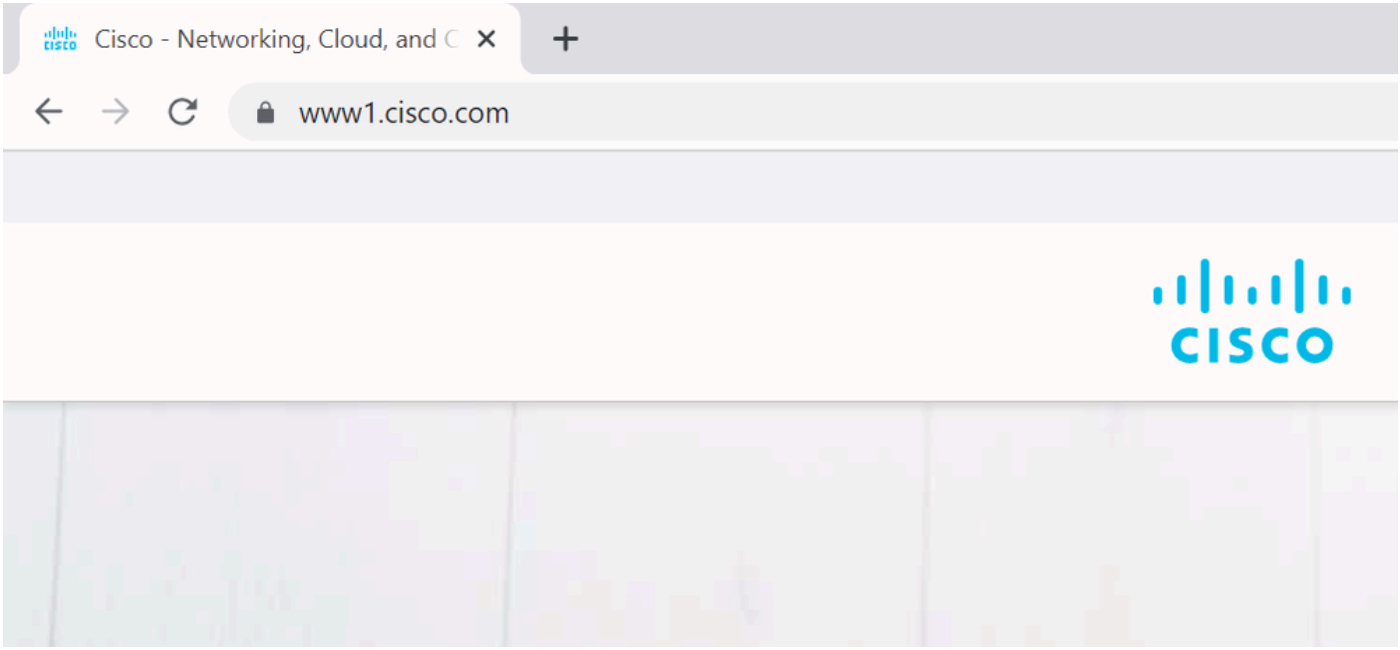
ブラウザでも同様の操作が行われるため、例で確認できます。Cisco.comをブラウズする場合、入力したURLの横にロックアイコンが表示され、これは信頼できる接続であることを意味します。これは、（最初のセクションの）CA信頼チェーンと、SANまたはCNのチェックの両方に基づいています。ブラウザのロックアイコンをクリックして証明書を開くと、共通名（Issued to：フィールドに表示）がCisco.comに設定されており、接続先のアドレスに正確に対応していることがわかります。この方法で、正しいサーバに確実に接続できます（証明書に署名したCAを信頼し、証明書を配布する前に検証を実行するため）。



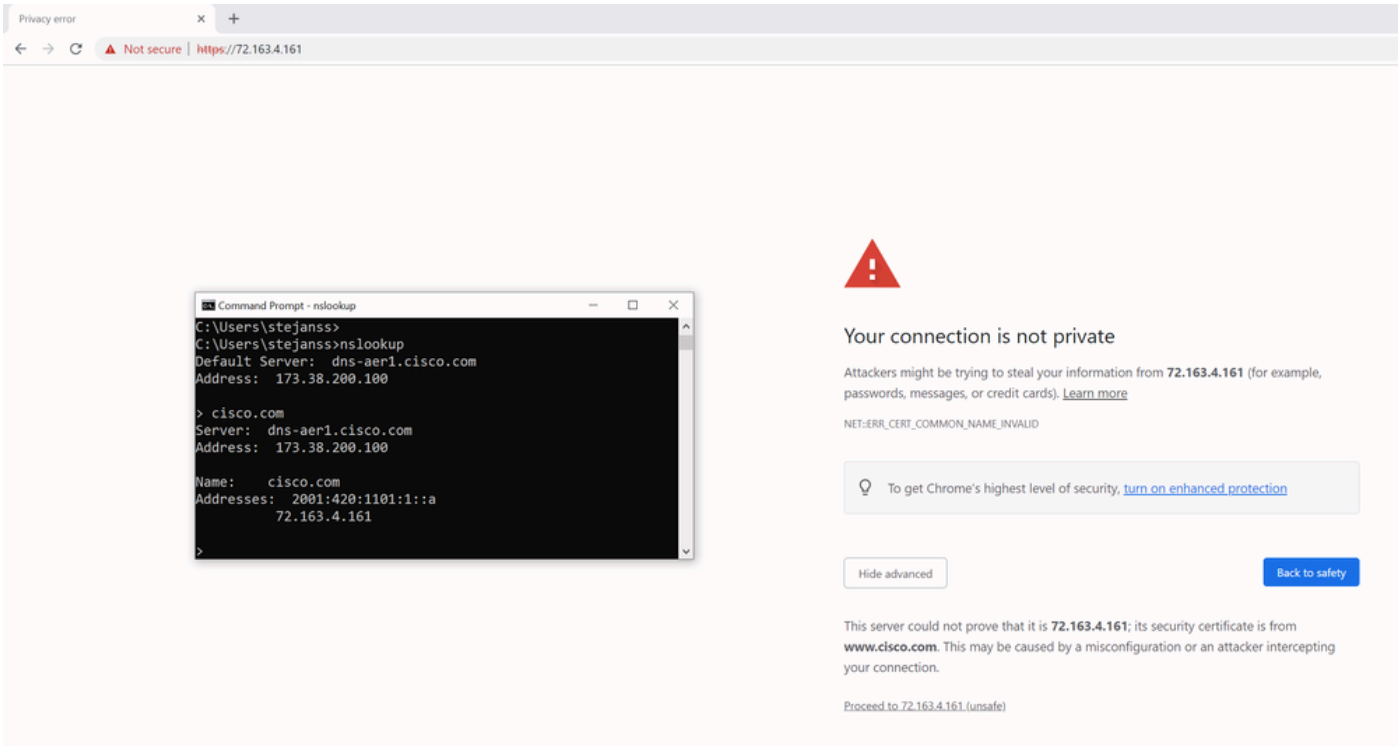
証明書の詳細、特にSANエントリを調べると、同じことが他のFQDNと同様に繰り返されていることがわかります。



つまり、たとえばCisco.comへの接続を要求する場合、この接続はSANエントリに含まれているため、セキュアな接続としても表示されます。



ただし、Cisco.comをブラウズせず、IPアドレス(番号付きWebアドレス)を直接参照する場合、署名したCAを信頼しないため、セキュアな接続は表示されません。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)が含まれていません。



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。そのため、証明書には、リモートパーティ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。

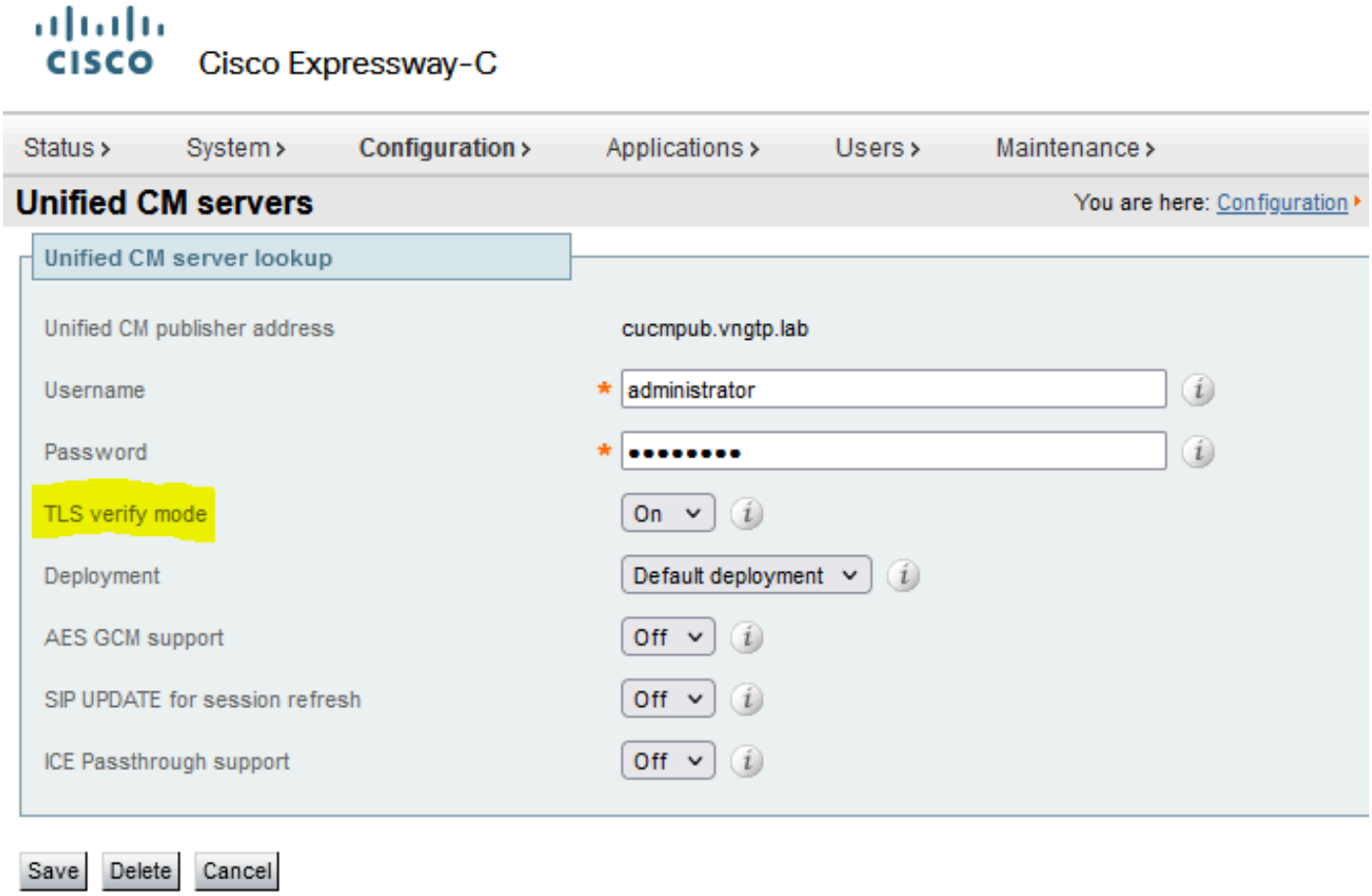
動作の変更

MRAサービスは、CUCM/IM&P/Unityサーバへの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、リモートエンドから提示された証明書を確認しなかったセキュアなHTTPS接続を処理するExpressway-C上のトラフィックサーバ。これにより、中間者攻撃が発生す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'to Onの設定でTLS証明書を検証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する情報ボックスを示します。この例では、SAN内のFQDNまたはIP、証明書の有効性、および証明書が信頼できるCAによって署名されているかどうかを確認します。



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に照会するときには実行されません。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のみ確認が行われることです。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は検証されません。これは、パブリッシャノードのデータベースからサブスクライバノード情報（FQDNまたはIP）を取得するためです。この設定の2つ目の欠点は、接続情報としてMRAクライアントにアドバタイズされるものが、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ことです。たとえばCUCMの場合、System > Serverで、IPアドレス（たとえば10.48.36.215）を使用してサーバをアドバタイズし（プロキシされたExpressway接続を介して）、これがMRAクライアントによって使用されますが、FQDNがcucm.steven.labであるExpressway-CのCUCMに追加できます。したがって、CUCMのtomcat証

明書にIPアドレスではなくSANエントリとしてcucm.steven.labが含まれている場合、TLS検証モードをオンに設定した検出は成功しますが、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るため、TLS検証は失敗します。

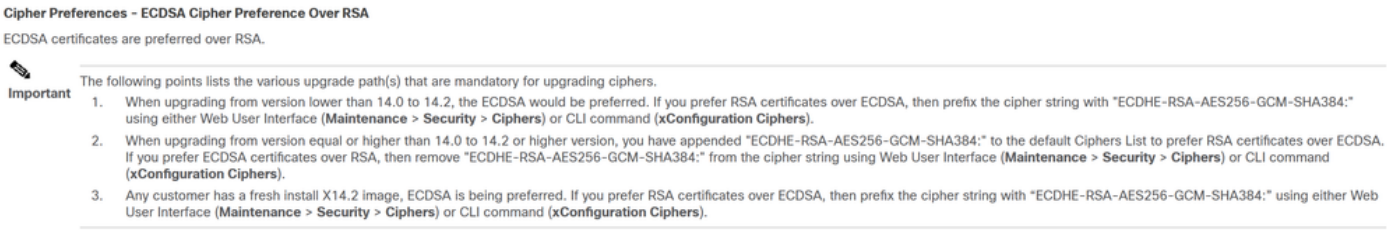
X14.2.0以降のバージョン

X14.2.0バージョン以降では、Expresswayサーバは、トラフィックサーバを介して行われる個々の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検証モードがオフ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ハンドシェイクが完了せず、要求が失敗します。その結果、冗長性などの機能の損失、フェールオーバーの問題、または完全なログイン障害などが発生する可能性があります。また、TLS Verify ModeをOnに設定しても、すべての接続が正常に動作するとは保証されません。これについては、後の例で説明します。

ExpresswayがCUCM / IM&P / Unityノードに対してチェックする正確な証明書は、『MRAガイド』のセクションに示すとおりです。

TLS検証のデフォルトの他に、X14.2で導入された変更もあり、アップグレードパスに応じて、暗号リストに異なる優先順位をアドバタイズする可能性があります。これは、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他の製品）からCisco Tomcatまたは Cisco CallManager証明書を要求する一方で、アップグレード後にECDSAバリアント（実際にはRSAよりも安全な暗号バリアント）を要求することが発生する可能性があるため、ソフトウェアアップグレード後に予期しないTLS接続を引き起こす可能性があります。 Cisco Tomcat-ECDSA証明書またはCisco CallManager-ECDSA証明書は、別のCAによって署名するか、または自己署名証明書のみで署名することができます（デフォルト）。

この暗号の優先順位の変更は、Expressway X14.2.1のリリースノートで示されているようにアップグレードパスによって異なるため、常に関連するわけではありません。つまり、各暗号リストに対してECDHE-RSA-AES256-GCM-SHA384を付加するかどうかが、Maintenance > Security > Ciphersの順に選択されます。そうでない場合は、RSA暗号よりも新しいECDSA暗号が優先されます。存在する場合は、以前と同様にRSAの方が優先度が高い動作になります。


Cipher Preferences - ECDSA Cipher Preference Over RSA

ECDSA certificates are preferred over RSA.

Important

The following points lists the various upgrade path(s) that are mandatory for upgrading ciphers.

1. When upgrading from version lower than 14.0 to 14.2, the ECDSA would be preferred. If you prefer RSA certificates over ECDSA, then prefix the cipher string with "ECDHE-RSA-AES256-GCM-SHA384:" using either Web User Interface (**Maintenance** > **Security** > **Ciphers**) or CLI command (**xConfiguration Ciphers**).
2. When upgrading from version equal or higher than 14.0 to 14.2 or higher version, you have appended "ECDHE-RSA-AES256-GCM-SHA384:" to the default Ciphers List to prefer RSA certificates over ECDSA. If you prefer ECDSA certificates over RSA, then remove "ECDHE-RSA-AES256-GCM-SHA384:" from the cipher string using Web User Interface (**Maintenance** > **Security** > **Ciphers**) or CLI command (**xConfiguration Ciphers**).
3. Any customer has a fresh install X14.2 image, ECDSA is being preferred. If you prefer RSA certificates over ECDSA, then prefix the cipher string with "ECDHE-RSA-AES256-GCM-SHA384:" using either Web User Interface (**Maintenance** > **Security** > **Ciphers**) or CLI command (**xConfiguration Ciphers**).

このシナリオでTLS検証が失敗する可能性のある方法は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は信頼されていません。

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. 証明書に接続アドレス（FQDNまたはIP）が含まれていません。

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も同様のシナリオを示しています。ログの類似性は共有されますが、解像度は異なります。ログは、(Maintenance > Diagnostics > Diagnostic loggingで)MRAログイン前に開始し、MRAログイン失敗後に停止した診断ログ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も）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、TFTP接続に対してポート6972でエラー（502応答）をスローすることを確認できます。

<#root>

```
===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="netw

2022-07-11T18:55:25.917+02:00 vcsc traffic_server[18242]:  Event="Request Allowed" Detail="Access allowe
2022-07-11T18:55:25.917+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:25,916" Module="netw
2022-07-11T18:55:25.955+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:25,955" Module="netw
2022-07-11T18:55:25.956+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:25,955" Module="netw

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="netw

2022-07-11T18:55:26.006+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:26,006" Module="netw
2022-07-11T18:55:26.016+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:26,016" Module="netw
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

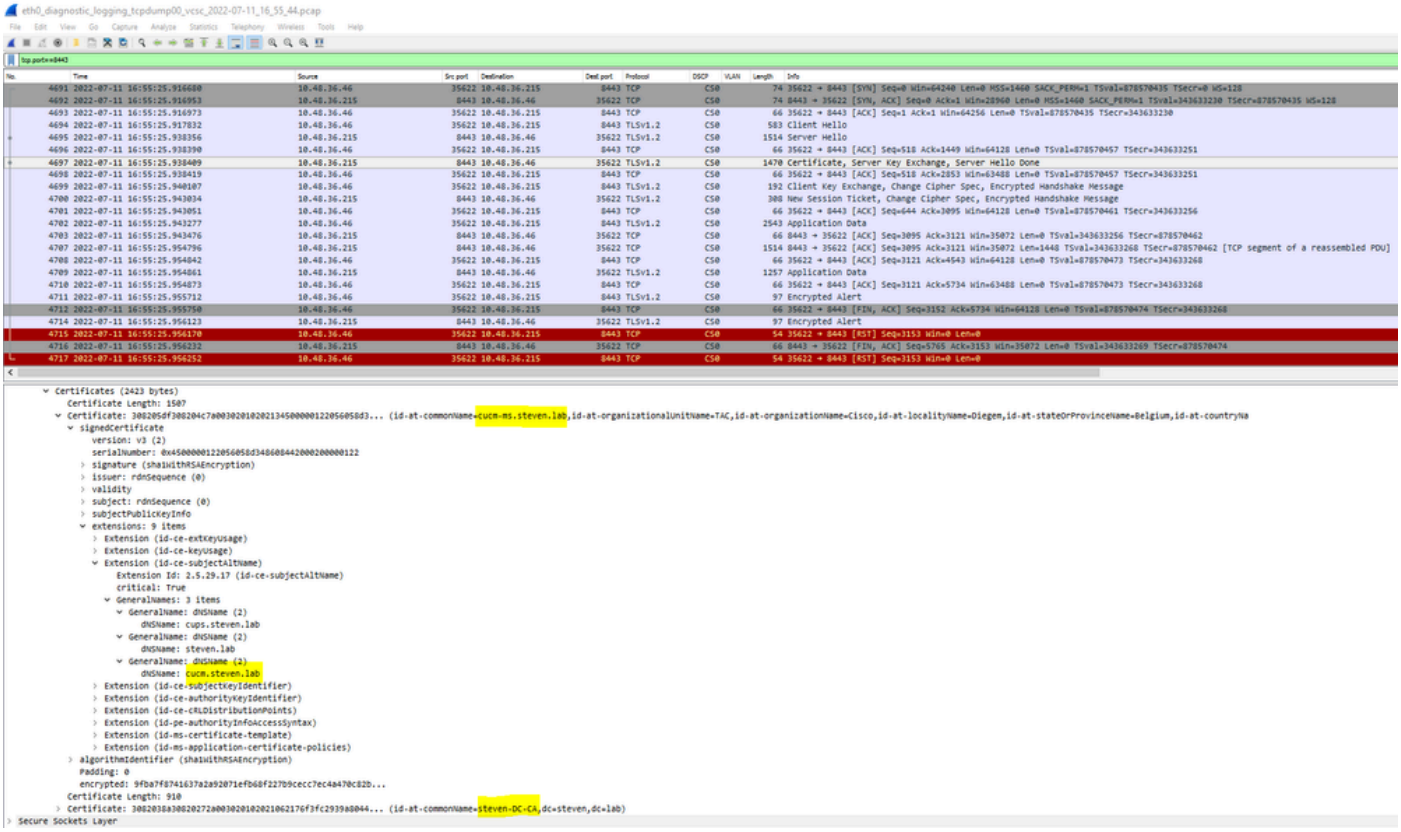
2022-07-11T18:55:26.024+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:26,024" Module="netw

502 connect failed

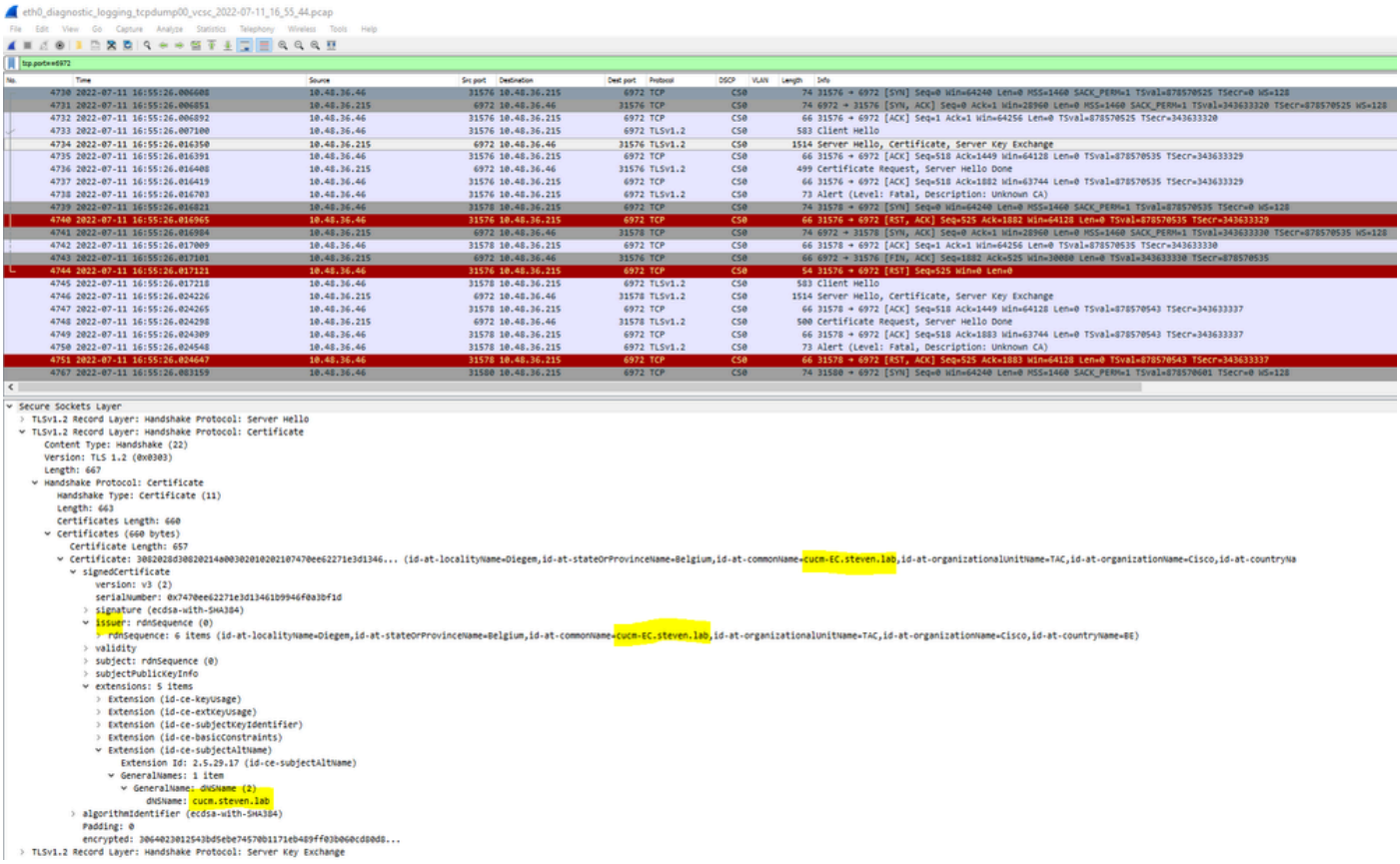
"

「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よりも大きい場合は、証明書チェーンがあることを意味し、不明なCA（Expressway-Cから見て）によって署名されています。

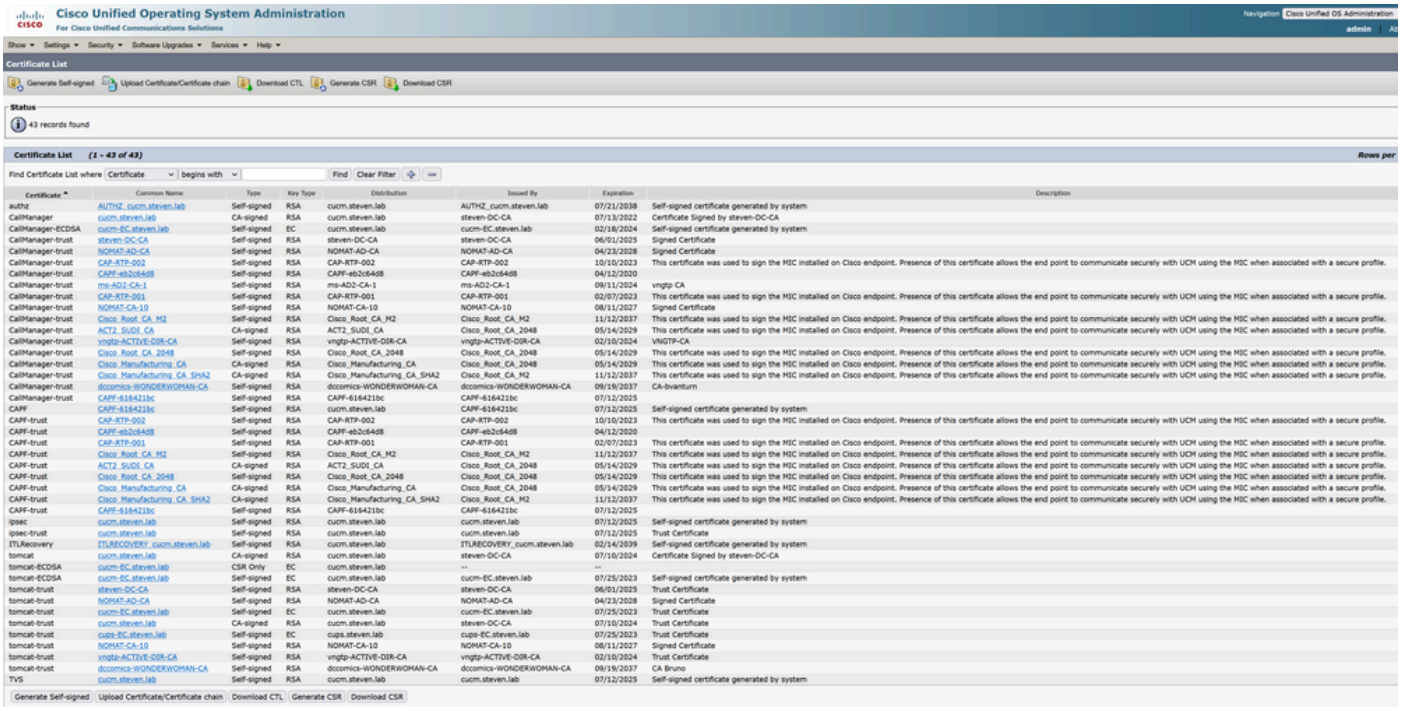
テキストログに示されているタイムスタンプで収集されたpcapファイルを見ると、CUCMがCNがcucm-ms.steven.lab（およびSANがcucm.steven.lab）で、steven-DC-CAによって署名された証明書を、ポート8443でExpressway-Cに提示していることがわかります。



ただし、ポート6972で提示された証明書を調べると、これがcucm-EC.steven.labとして設定されたCNを持つ自己署名証明書（発行者自身）であることがわかります。-EC拡張子は、これがCUCMで設定されている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。ここでは、tomcatとtomcat-ECDSA用の異なる証明書を示しています。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. 証明書に接続アドレス（FQDNまたはIP）が含まれていない

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、CUCMでIPアドレス(10.48.36.215)を使用してSystem > Server CUCMを設定している場合、Expressway-Cでは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（Expressway-C経由でプロキシ化されている）からの後続の要求はこのアドレスを対象とします。

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、502エラーがスローされます。このエラーの結果、たとえばMRAログインが失敗します。

<#root>

```
2022-07-11T19:49:01.472+02:00 vcsc traffic_server[3916]: UTCTime="2022-07-11 17:49:01,472" Module="netw
HTTPMSG:
|GET http://vcs_control.steven.lab:8443/c3RldmVuLmxhYi9odHRwcy8xMC40OC4zNi4yMTUvODQ0Mw/cucm-uds/user/em
...

2022-07-11T19:49:01.478+02:00 vcsc traffic_server[3916]: UTCTime="2022-07-11 17:49:01,478" Module="netw
2022-07-11T19:49:01.478+02:00 vcsc traffic_server[3916]: UTCTime="2022-07-11 17:49:01,478" Module="netw
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)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

この動作が次の手順で簡単に変更されるかどうかを検証できます。

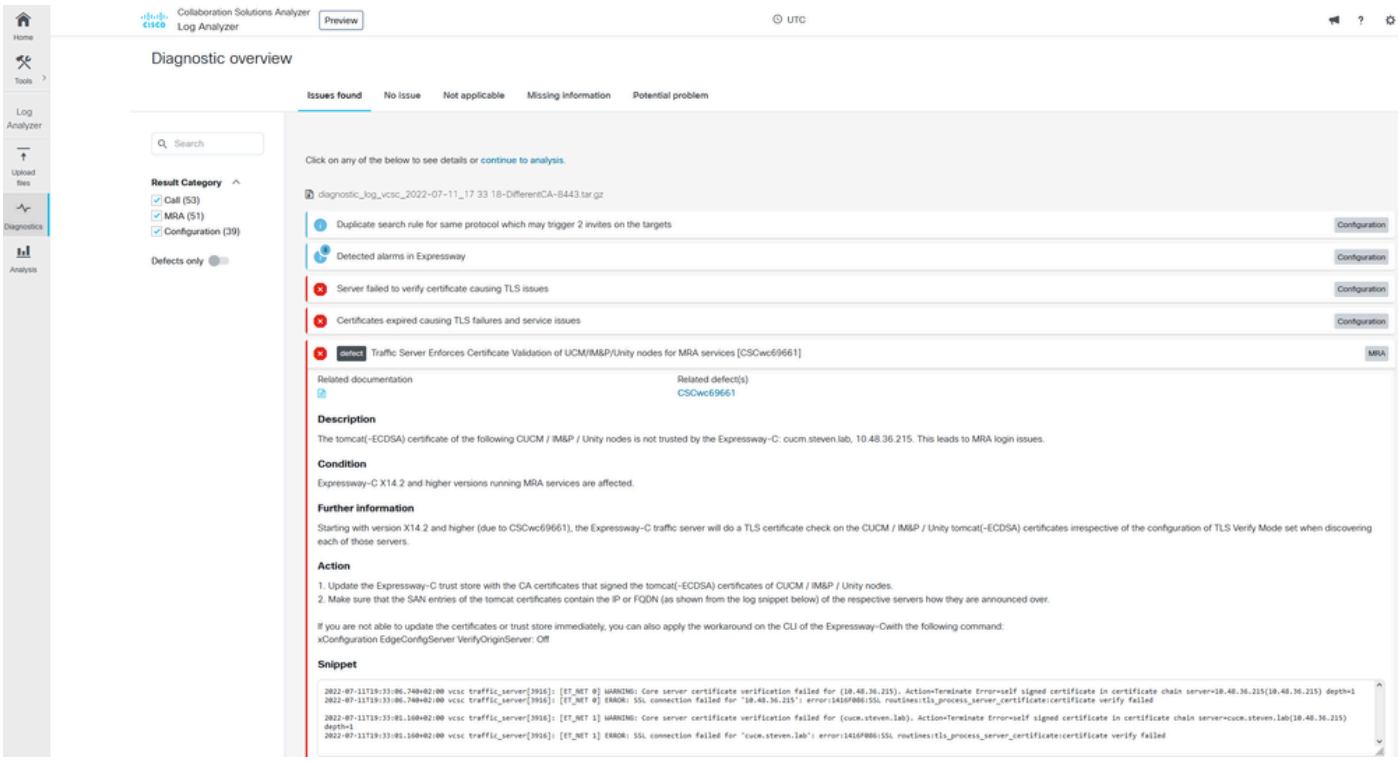
1. Expressway-EおよびCサーバ（理想的にはTCPDumpsが有効な状態）の診断ログをMaintenance > Diagnostics > Diagnostic Logging（クラスタの場合はプライマリノードから開始すれば十分）から開始します。

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

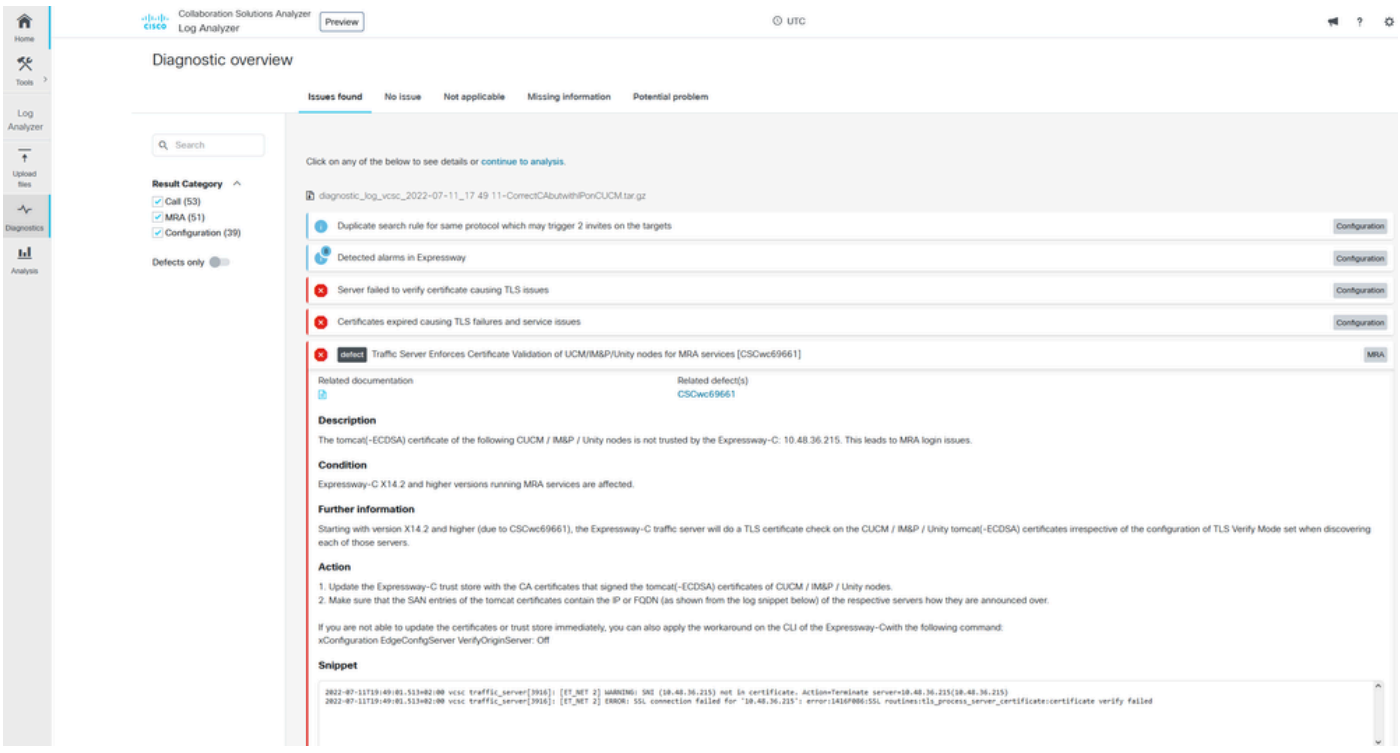
3. 障害が発生するまで待ってから、Expressway-EおよびCサーバの診断ログを停止します（クラスタの場合は、クラスタのすべてのノードからログを個別に収集してください）

4. コラボレーションソリューションアナライザツールでログをアップロードして分析します。

5. 問題が発生した場合は、影響を受ける各サーバについて、この変更に関連する最新の警告行と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検証が正常に機能してい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: Core server certificate verification failed for (<server-FQDN-or-IP>)」というエラーメッセージが表示された場合は、 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場合は、それに応じて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の順に選択し、自己署名証明書(depth = 0)を使用します。この操作は、クラスタ内のすべてのサーバで必ず実行してください。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。

注：Expresswayでは、Cisco Bug ID CSCwa12905で説明されているように、共通名(CN)が同じである2つの異なる（たとえば自己署名された）証明書をExpresswayの信頼ストアにアップロードすることはできません。これを修正するには、CA署名付き証明書に移行するか、CUCMをバージョン14にアップグレードします。このバージョンでは、TomcatとCallManagerで同じ（自己署名）証明書を再利用できます。

「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。その情報を含むように証明書を適合させるか、設定を変更し(CUCM System > Serverを使用してサーバ証明書に含まれる内容に変更するなど)、Expressway-Cサーバの設定を更新して、この設定を考慮に入れることができます。

関連情報

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。