CSCwc에서 소개한 MRA 서비스에 대한 Expressway Traffic Server 인증서 확인 문제 해결 69661

목차

소개

이 문서에서는 Cisco 버그 ID CSCwc69661에 연결된 Expressway 버전 X14.2.0 이상의 동작 변경에 대해 설명합니다. 이 변경을 통해 Expressway 플랫폼의 트래픽 서버는 CUCM(Cisco Unified Communication Manager), Cisco IM&P(Unified Instant Messaging & Presence) 및 MRA(Mobile and Remote Access) 서비스용 Unity 서버 노드의 인증서 확인을 수행합니다. 이러한 변경으로 인해 Expressway 플랫폼에서 업그레이드한 후 MRA 로그인이 실패할 수 있습니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Expressway 기본 컨피그레이션
- MRA 기본 컨피그레이션

사용되는 구성 요소

이 문서의 정보는 버전 X14.2 이상의 Cisco Expressway를 기반으로 합니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

HTTPS(Hypertext Transfer Protocol Secure)는 TLS(Transport Layer Security)를 사용하여 통신을 암호화하는 보안 통신 프로토콜입니다. TLS 핸드셰이크에서 교환되는 TLS 인증서를 사용하여 이 보안 채널을 생성합니다. 이를 통해 IT는 두 가지 목적을 달성할 수 있습니다. 인증(원격 상대방이 누구에 연결되었는지 확인) 및 프라이버시(암호화) 인증은 중간자 공격(man-in-the-middle attack)으로부터 보호하며 프라이버시는 공격자가 통신을 도청하고 변조하는 것을 방지합니다.

TLS(인증서) 확인은 인증을 통해 수행되며, 올바른 원격 상대방에 연결했는지 확인할 수 있습니다. 검증은 두 개의 개별 항목으로 구성됩니다.

1. 신뢰할 수 있는 CA(인증 기관) 체인

2. 주체 대체 이름(SAN) 또는 공통 이름(CN)

신뢰할 수 있는 CA 체인

Expressway-C가 CUCM/IM&P/Unity가 보내는 인증서를 신뢰하려면 해당 인증서에서 신뢰하는 최상위(루트) CA(Certification Authority)로의 링크를 설정할 수 있어야 합니다. 엔터티 인증서를 루트 CA 인증서에 연결하는 인증서 계층 구조인 이러한 링크를 신뢰 체인이라고 합니다. 이러한 신뢰 체인을 확인할 수 있도록 각 인증서에는 두 개의 필드가 있습니다. 발급자(또는 '발급자') 및 제목(또는 '발급자')

CUCM이 Expressway-C로 전송하는 것과 같은 서버 인증서는 'Subject' 필드에 일반적으로 CN의 FQDN(정규화된 도메인 이름)이 있습니다.

```
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.lab에 대한 서버 인증서의 예. Subject(주체) 필드의 CN 특성에 FQDN을 비롯하여 국가(C), 상태(ST), 위치(L) 등의 다른 특성이 있습니다. 또한 vngtp-ACTIVE-DIR-CA라는 CA에서 서버 인증서를 전달(발급)하는 것을 볼 수 있습니다.

최상위 CA(루트 CA)는 스스로를 식별하기 위해 인증서를 발급할 수도 있습니다. 이러한 루트 CA 인증서에는 발급자와 주체가 동일한 값을 갖습니다.

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

이는 자신을 식별하기 위해 루트 CA에서 제공하는 인증서입니다.

일반적인 상황에서는 루트 CA가 서버 인증서를 직접 발급하지 않습니다. 대신 다른 CA에 대한 인증서를 발급합니다. 그런 다음 이러한 다른 CA를 중간 CA라고 합니다. 중간 CA는 다른 중간 CA에 대한 서버 인증서 또는 인증서를 직접 발급할 수 있습니다. 중간 CA 1에서 서버 인증서를 발급하여 중간 CA 2에서 인증서를 발급받는 등의 상황이 발생할 수 있습니다. 최종 중간 CA가 루트 CA에서 인증서를 직접 가져올 때까지:

```
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
```

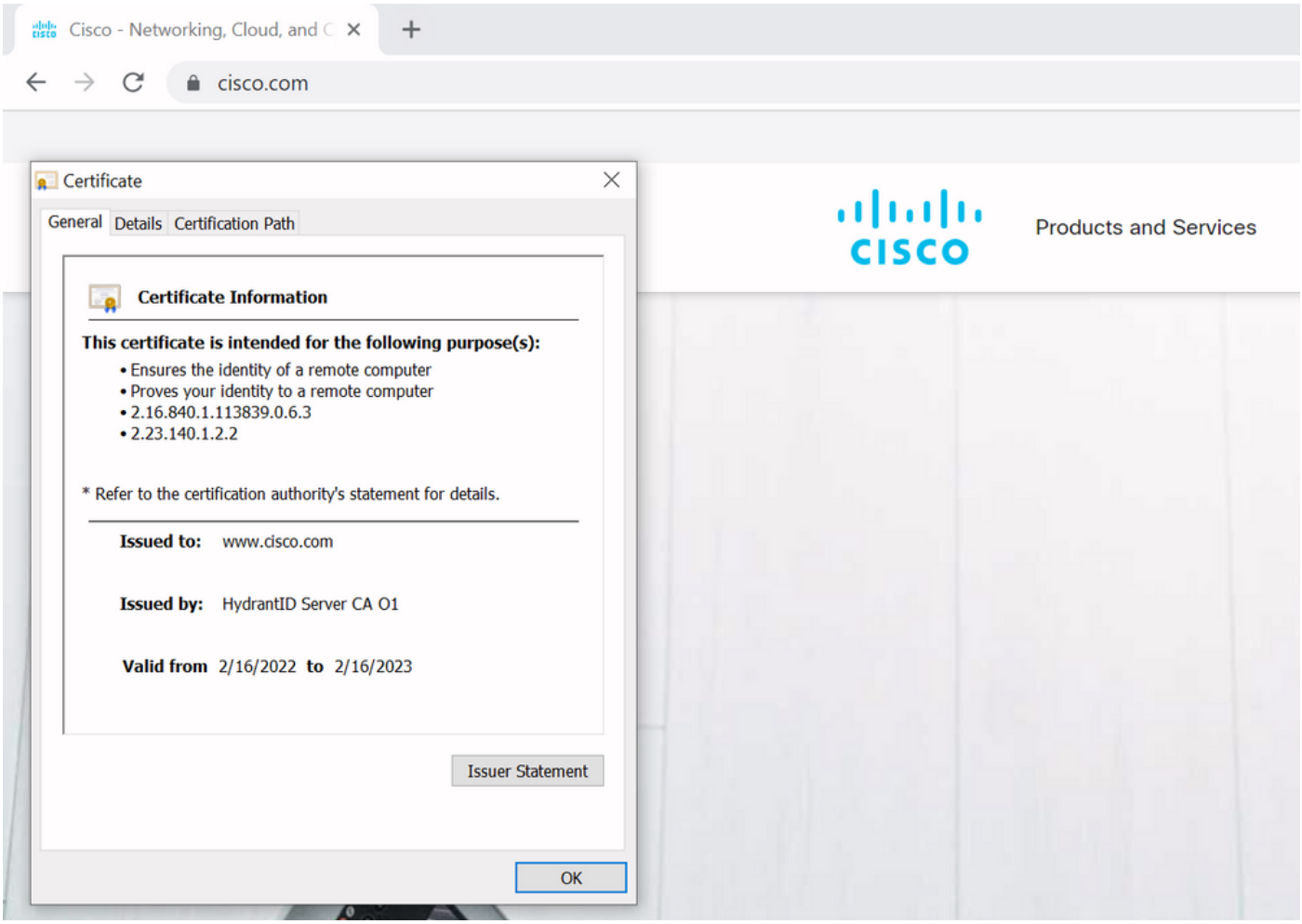
이제 Expressway-C가 CUCM이 전송하는 서버 인증서를 신뢰하려면 해당 서버 인증서에서 루트 CA 인증서까지 신뢰 체인을 구축할 수 있어야 합니다. 그러기 위해서는 루트 CA 인증서와 모든 중간 CA 인증서(있는 경우, 루트 CA가 CUCM의 서버 인증서를 직접 발급한 경우는 제외)를 Expressway-C의 트러스트 저장소에 업로드해야 합니다.

> **참고: Issuer(발급자) 및 Subject(주체) 필드는 사람이 읽을 수 있는 방식으로 Trust 체인을 구축하기 쉽지만, Expressway-C 및 CUCM은 인증서에서 이러한 필드를 사용하지 않습니다. 대신 'X509v3 권한 키 식별자' 및 'X509v3 주체 키 식별자' 필드를 사용하여 신뢰 체인을 작성합니다. 이러한 키에는 Subject/Issuer 필드를 사용하는 것보다 더 정확한 인증서의 식별자가 포함됩니다. 동일한 Subject/Issuer(주체/발급자) 필드가 있는 인증서가 2개 있을 수 있지만 그중 하나는 만료되었고 나머지 하나는 여전히 유효합니다. Expressway/CUCM이 올바른 신뢰 체인을 계속 확인할 수 있도록 둘 다 X509v3 주체 키 식별자가 다릅니다.**

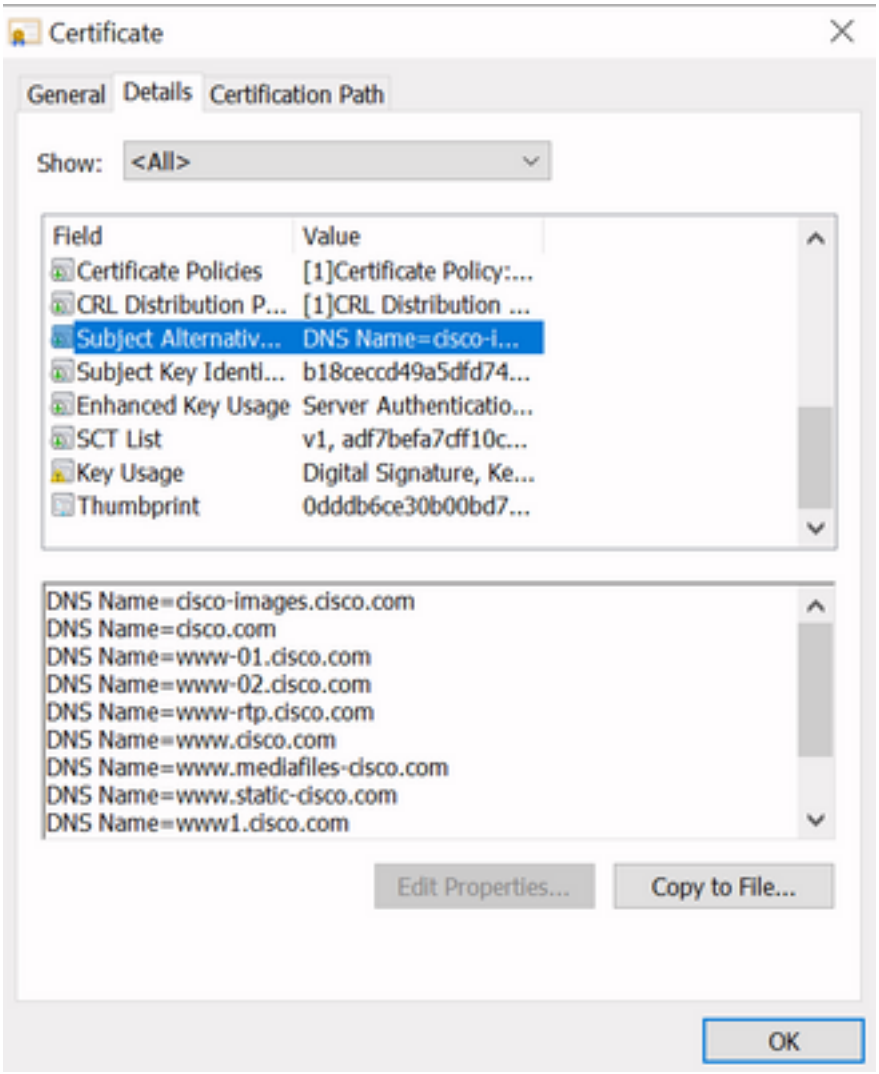
SAN 또는 CN 확인

1단계에서는 트러스트 스토어를 체크 아웃하지만, 트러스트 스토어에서 CA가 서명한 인증서가 있는 모든 사용자는 유효합니다. 이것은 분명히 충분하지 않다. 따라서 특별히 연결한 서버가 실제로 올바른 서버인지 확인하는 추가 검사가 있습니다. 이 작업은 요청이 수행된 주소를 기반으로 수행됩니다.

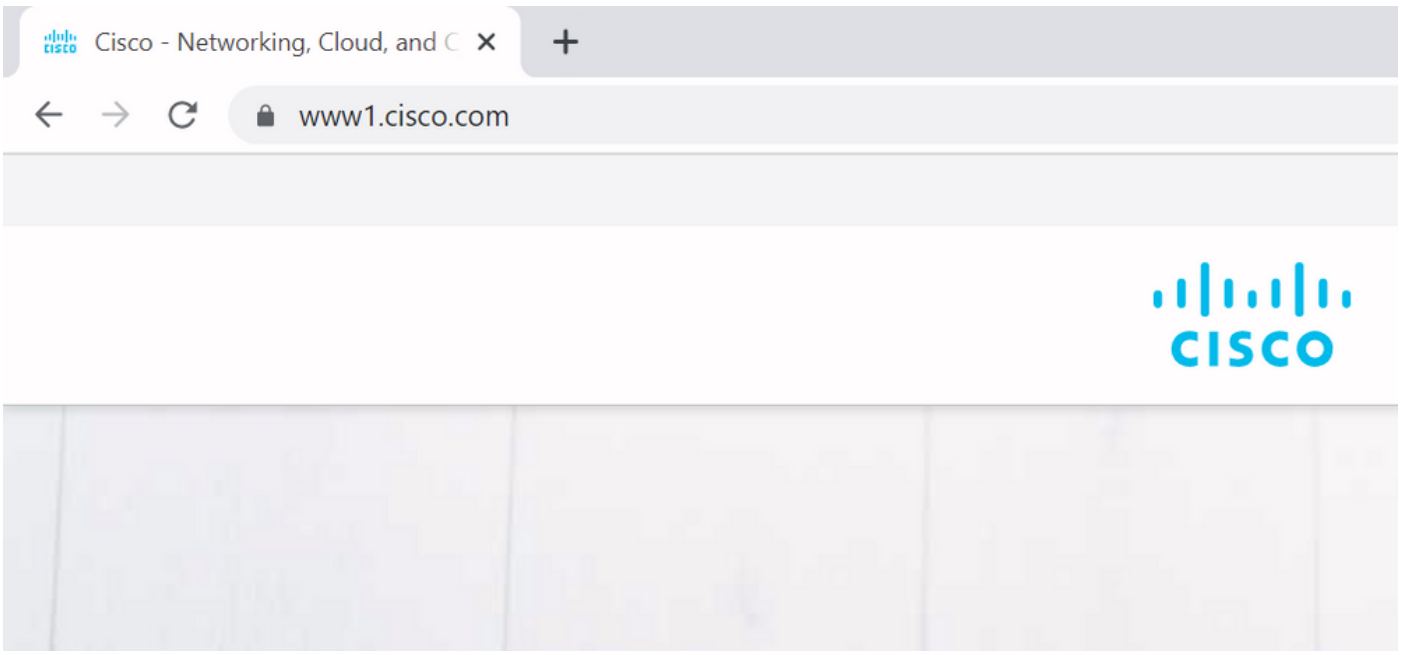
브라우저에서 같은 종류의 작업이 수행되므로 예를 통해 살펴보겠습니다. https://www.cisco.com으로 이동하면 입력한 URL 옆에 잠금 아이콘이 표시되며 이는 해당 URL이 신뢰할 수 있는 연결임을 의미합니다. 이는 CA 신뢰 체인(첫 번째 섹션)과 SAN 또는 CN 검사를 모두 기반으로 합니다. 잠금 아이콘을 클릭하여 브라우저를 통해 인증서를 열면 공용 이름(Issued to:' 필드에 표시됨)이 www.cisco.com으로 설정되어 있으며 연결하려는 주소에 정확히 해당하는 것을 알 수 있습니다. 이러한 방법으로 올바른 서버에 연결할 수 있습니다(인증서를 배포하기 전에 인증서에 서명하고 확인을 수행하는 CA를 신뢰하기 때문).



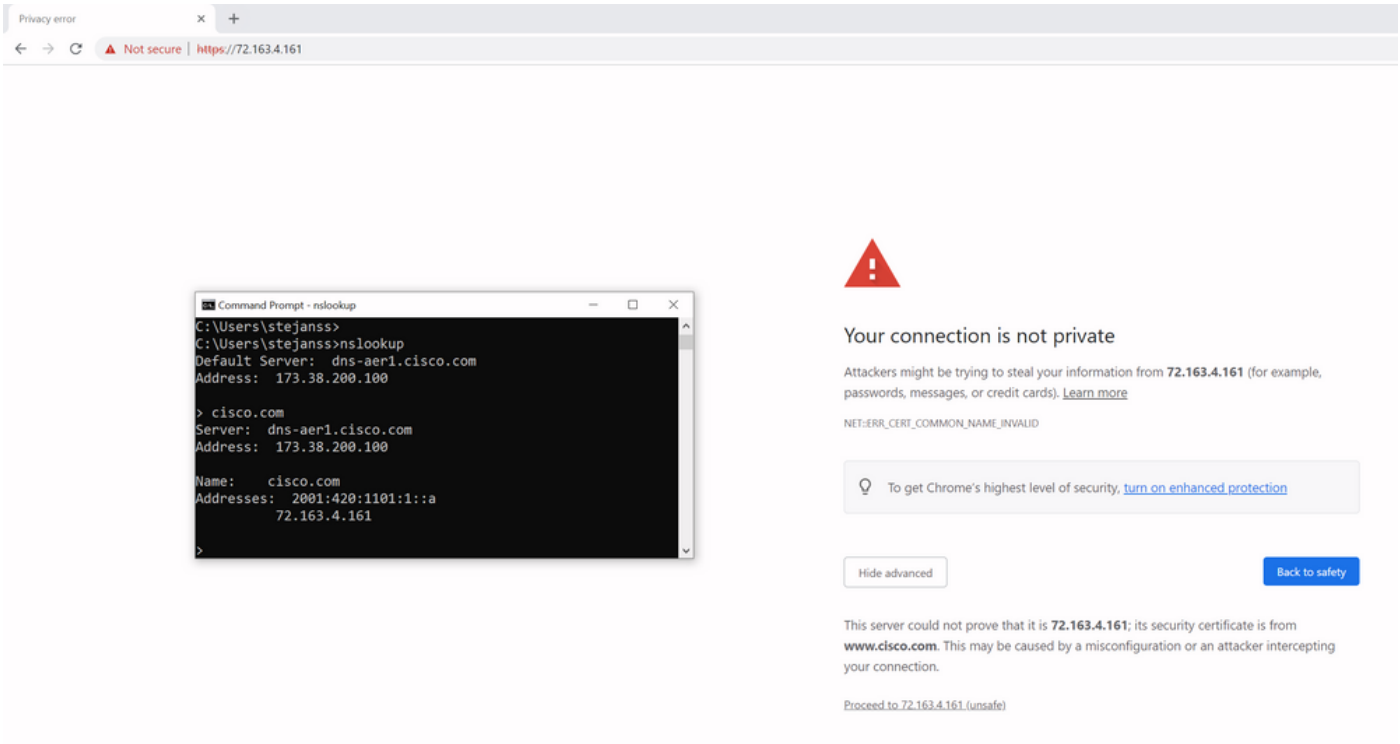
인증서의 세부사항 및 특히 SAN 항목을 보면 동일한 내용이 반복되고 다른 FQDN도 반복되는 것을 알 수 있습니다.



즉, https://www1.cisco.com에 대한 연결을 요청하는 경우 SAN 항목에 포함되어 있으므로 보안 연결로도 표시됩니다.



그러나 https://www.cisco.com으로 이동하지 않고 IP 주소(https://72.163.4.161)로 직접 이동하면 해당 CA가 서명한 CA를 신뢰하지만 Cisco에 제공된 인증서는 서버에 연결하는 데 사용한 주소(72.163.4.161)를 포함하지 않기 때문에 보안 연결이 표시되지 않습니다.



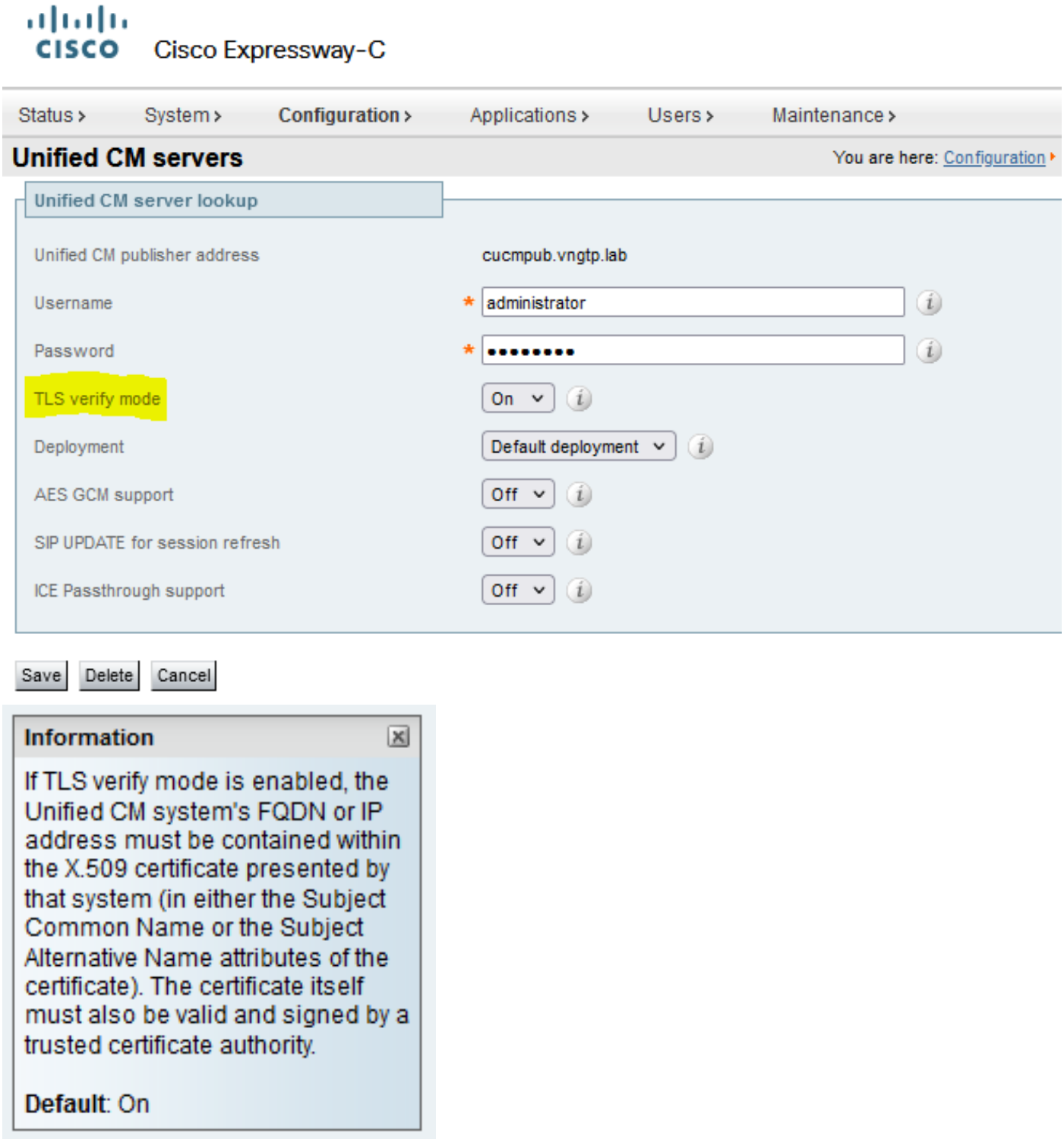
브라우저에서 이를 무시할 수 있지만, TLS 연결에서 활성화할 수 있는 설정이며, 이 설정에서는 우회가 허용되지 않습니다. 따라서 인증서에 연결할 때 원격 상대방이 사용할 올바른 CN 또는 SAN 이름이 포함되어 있어야 합니다.

동작 변경

MRA 서비스는 CUCM/IM&P/Unity 서버를 향하는 Expressway를 통한 여러 HTTPS 연결에 크게 의존하여 올바르게 인증하고 로그인하는 클라이언트에 해당하는 올바른 정보를 수집합니다. 이 통신은 일반적으로 포트 8443 및 6972를 통해 발생합니다.

X14.2.0 이하 버전

X14.2.0 이전 버전에서는 이러한 보안 HTTPS 연결을 처리하는 Expressway-C의 트래픽 서버가 원격 쪽에서 제공한 인증서를 확인하지 않았습니다. 이는 중간자 공격(man-in-the-middle attack)으로 이어질 수 있습니다. MRA 컨피그레이션에는 'TLS 확인 모드' 컨피그레이션에 의한 TLS 인증서 확인을 위한 옵션이 있습니다. 이 경우 Configuration(컨피그레이션) > Unified **Communications > Unified CM servers / IM and Presence Service nodes / Unity Connection servers(Unified CM 서버 / IM and Presence 서비스 노드 / Unity 연결 서버) 아래**에서 CUCM / IM&P / Unity servers(CUCM / IM&P / Unity 서버)를 추가하면 'On(켜기)'으로 바뀝니다. 컨피그레이션 옵션 및 관련 정보 상자가 예로 표시됩니다. 이는 SAN의 FQDN 또는 IP는 물론 인증서의 유효성 및 신뢰할 수 있는 CA에서 서명했는지 여부를 확인하는 것입니다.



이 TLS 인증서 확인 확인은 CUCM / IM&P / Unity 서버 검색 시에만 수행되며 MRA 로그인 시 다양한 서버가 쿼리되는 시점에는 수행되지 않습니다. 이 구성의 첫 번째 단점은 추가한 게시자 주소에 대해서만 검증한다는 것입니다. 게시자 노드의 데이터베이스에서 구독자 노드 정보(FQDN 또는 IP)를 검색하므로 구독자 노드의 인증서가 올바르게 설정되었는지 확인하지 않습니다. 이 컨피그레이션의 두 번째 단점은 연결 정보가 Expressway-C 컨피그레이션에 입력된 게시자 주소와 다를 수 있으므로 MRA 클라이언트에 광고된다는 점입니다. 예를 들어 CUCM의 **System > Server**에서 IP 주소(예: 10.48.36.215)로 서버를 광고할 수 있으며, 이 주소는 (프록시 Expressed Expressway 연결을 통해) MRA 클라이언트에서 사용되지만 Expressway-C의 CUCM에서 cucm.steven.lab의 FQDN을 사용하여 추가할 수 있습니다. 따라서 CUCM의 tomcat 인증서에 SAN 항목으로 cucm.steven.lab이 포함되지만 IP 주소는 포함되지 않는다고 가정합니다. 그러면 'TLS Verify Mode'가 'On'으로 설정된 검색이 성공하지만 MRA 클라이언트의 실제 통신은 다른 FQDN 또는 IP를 대상으로 할 수 있으므로 TLS 확인에 실패합니다.

X14.2.0 이상 버전

X14.2.0 버전 이상에서는 Expressway 서버가 트래픽 서버를 통해 생성되는 모든 HTTPS 요청에 대해 TLS 인증서 확인을 수행합니다. 즉, CUCM / IM&P / Unity 노드를 검색하는 동안 'TLS Verify Mode(TLS 확인 모드)'가 'Off(해제)'로 설정된 경우에도 이 작업을 수행합니다. 검증에 성공하지 못하면 TLS 핸드셰이크가 완료되지 않고 요청이 실패하기 때문에 리던던시, 장애 조치 문제 등의 기능이 손실되거나 로그인 실패가 완료됩니다. 또한 'TLS Verify Mode'가 'On'으로 설정된 경우 모든 연결이 나중에 예에서 설명한 대로 제대로 작동한다는 보장은 없습니다.

TLS 확인에 대한 기본 설정 외에, X14.2에서 소개된 변경 사항도 있는데, 이는 암호 목록에 대한 다른 기본 설정 순서를 광고한다. 이 경우 소프트웨어 업그레이드 후 예기치 않은 TLS 연결이 발생할 수 있습니다. 업그레이드 전에는 CUCM의 Cisco Tomcat 또는 Cisco CallManager 인증서(또는 ECDSA 알고리즘에 대해 별도의 인증서가 있는 다른 제품)를 요청했지만 업그레이드 후에는 ECDSA 변형을 요청할 수 있기 때문입니다. Cisco Tomcat-ECDSA 또는 Cisco CallManager-ECDSA 인증서는 다른 CA에서 서명하거나 자체 서명 인증서(기본값)만 사용할 수 있습니다.

이 시나리오에서 TLS 확인이 실패할 수 있는 두 가지 방법은 뒤에서 자세히 설명합니다.

1. 원격 인증서를 서명한 CA가 신뢰되지 않음

a. 자체 서명 인증서

b. 알 수 없는 CA에서 서명한 인증서

2. 연결 주소(FQDN 또는 IP)가 인증서에 포함되어 있지 않습니다.

시나리오 트러블슈팅

다음 시나리오는 Expressway를 X14.0.7에서 X14.2로 업그레이드한 후 MRA 로그인이 실패하는 랩 환경에서 유사한 시나리오를 보여줍니다. 로그에서 유사성을 공유하지만 해상도는 다릅니다. 로그는 MRA 로그인 전에 시작되고 MRA 로그인 실패 후 중지된 진단 로깅(**Maintenance(유지 관리) > Diagnostics(진단)** > Diagnostic logging(진단 로깅)에서)에 의해 수집됩니다. 추가 디버그 로깅이 활성화되지 않았습니다.

1. 원격 인증서를 서명한 CA가 신뢰되지 않음

원격 인증서는 Expressway-C의 트러스트 저장소에 포함되지 않은 CA에 의해 서명되거나 Expressway-C 서버의 트러스트 저장소에 추가되지 않은 자체 서명 인증서(기본적으로 CA도)일 수 있습니다.

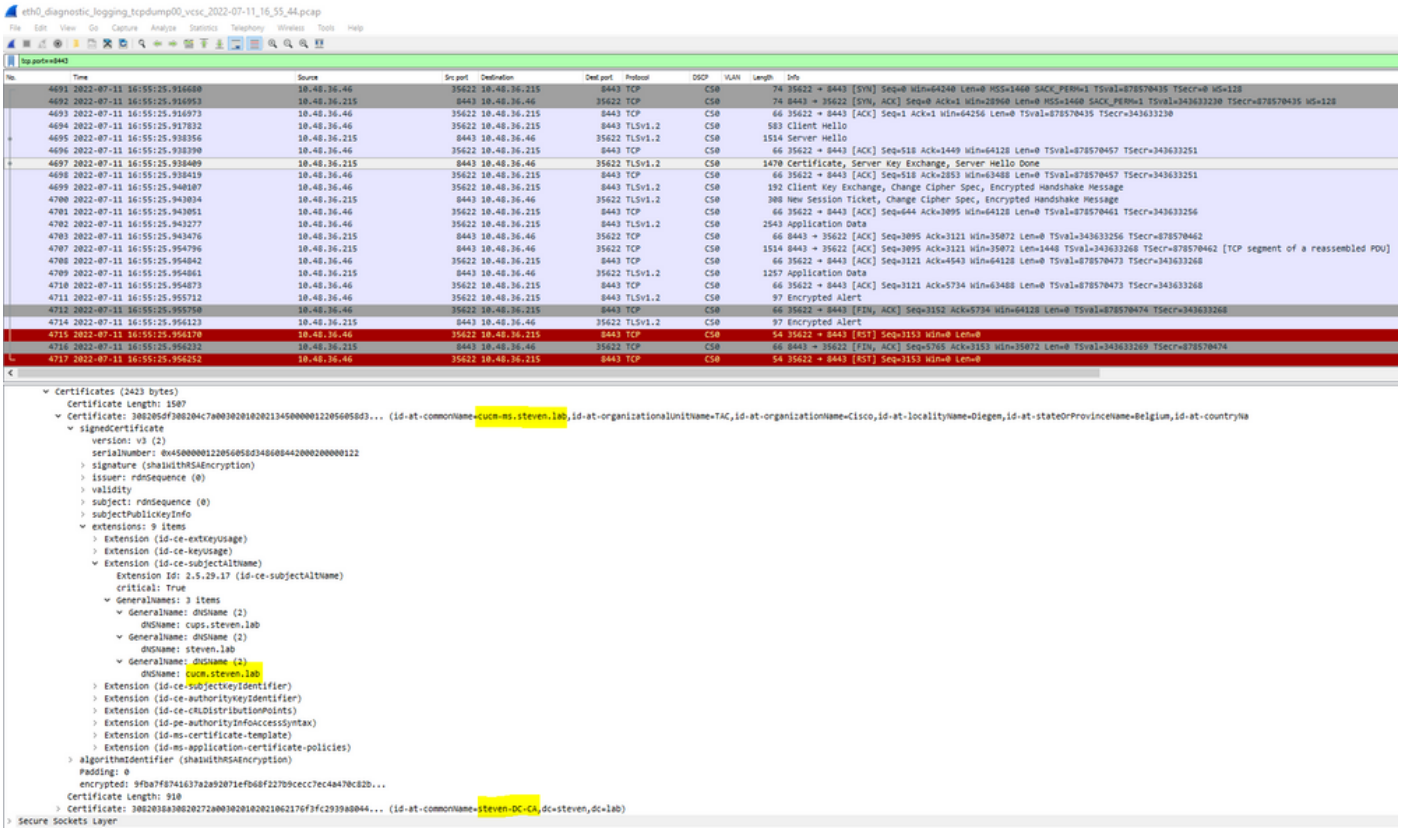
이 예에서는 CUCM(10.48.36.215 - cucm.steven.lab)으로 이동하는 요청이 포트 8443(200 OK 응답)에서 올바르게 처리되지만 TFTP 연결에 대해 포트 6972에서 오류(502 응답)가 발생함을 확인할 수 있습니다.

```
===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-
```

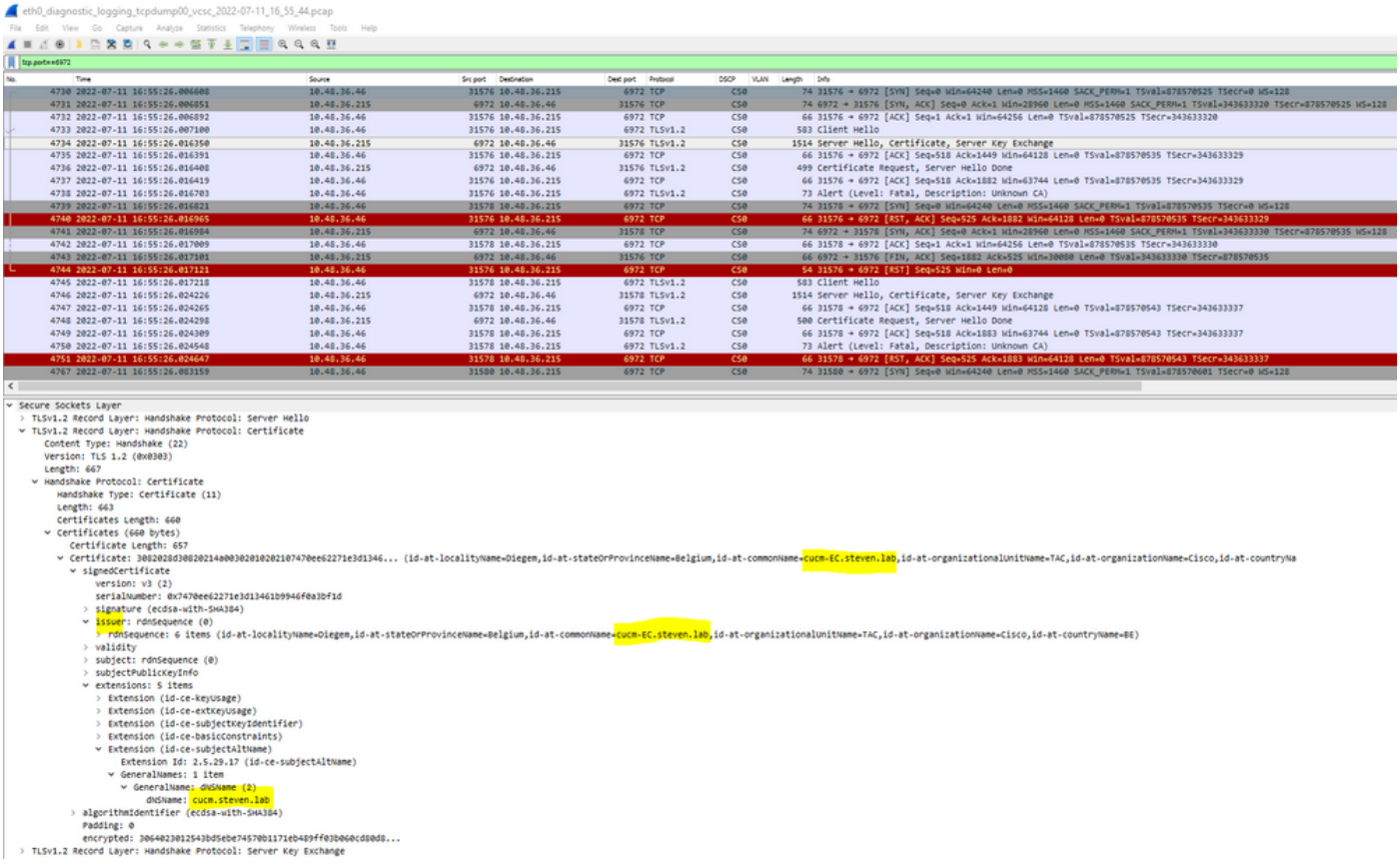
```
uds/user/emusk/devices HTTP/1.1"

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'(인증서 확인 실패) 오류는 Expressway-C가 TLS 핸드셰이크의 유효성을 검사할 수 없다는 사실을 나타냅니다. 그 이유는 자체 서명 인증서를 나타내므로 경고 줄에 표시됩니다. 깊이가 0으로 표시되면 자체 서명 인증서입니다. 깊이가 0보다 높으면 인증서 체인이 있는 것이므로 알 수 없는 CA에서 서명합니다(Expressway-C의 관점에서).

텍스트 로그에서 언급한 타임스탬프에 수집된 pcap 파일을 보면 CUCM이 포트 8443의 Expressway-C에 steven-DC-CA가 서명한 CN을 cucm-ms.steven.lab(SAN의 cucm.steven.lab)으로, 인증서를 제공합니다.

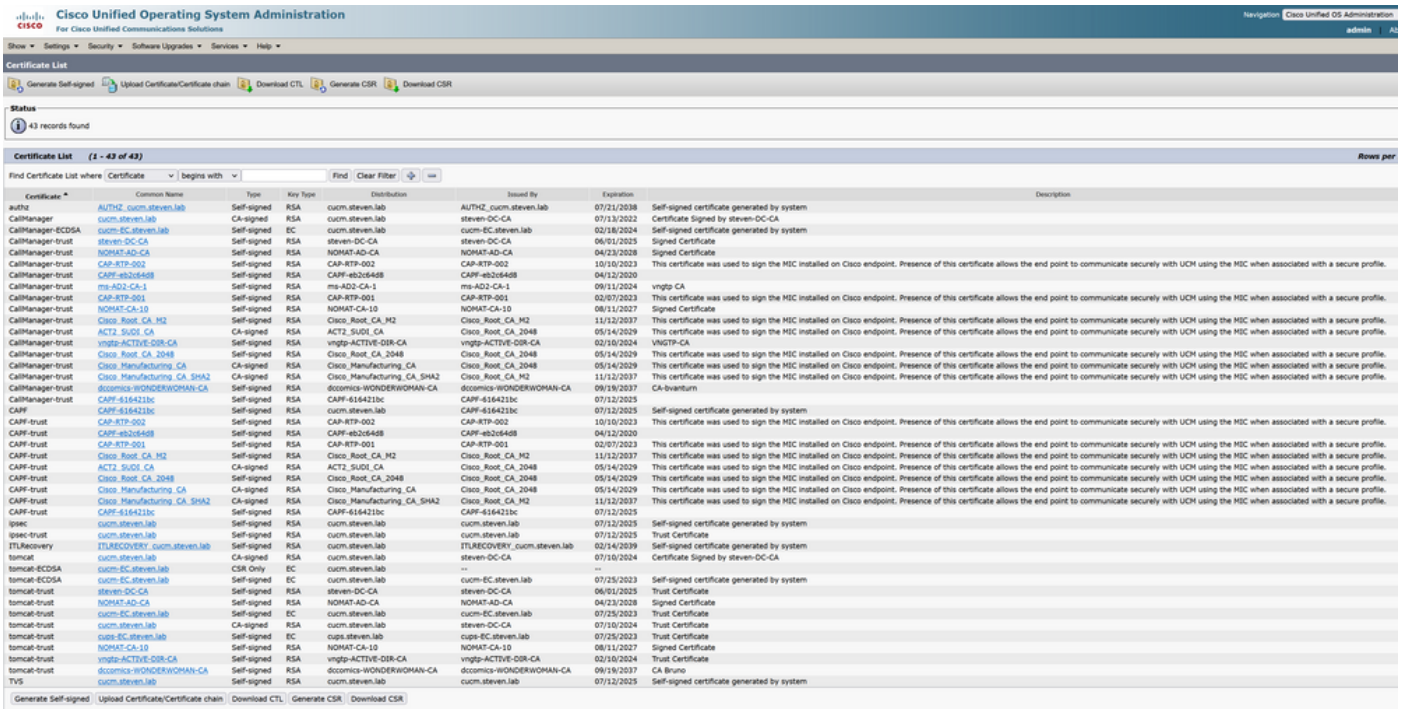


그러나 포트 6972에 표시된 인증서를 검사하면 CN이 cucm-EC.steven.lab으로 설정된 자체 서명 인증서(발급자 자체)를 확인할 수 있습니다. -EC 확장은 CUCM에 설정된 ECDSA 인증서임을 나타냅니다.



Cisco Unified OS Administration(Cisco Unified OS 관리) 아래의 CUCM에서 다음 예와 같이 Security(보안) > Certificate Management(인증서 관리) 아래의 인증서를 확인할 수 있습니다. tomcat이 CA 서명(Expressway-C에서 신뢰)된 경우 tomcat 및 tomcat-ECDSA에 대한 다른 인증서

가 표시되며, tomcat-ECDSA 인증서는 자체 서명되며 여기서 Expressway-C에서 신뢰되지 않습니다.



2. 연결 주소(FQDN 또는 IP)가 인증서에 포함되어 있지 않습니다.

신뢰 저장소 외에도, 이 트래픽 서버는 MRA 클라이언트가 요청을 보내는 연결 주소도 확인합니다. 예를 들어 **System(시스템) > Server(서버)**의 CUCM에서 IP 주소(10.48.36.215)를 사용하여 CUCM을 설정한 경우 Expressway-C는 이를 클라이언트에 알리고 Expressway-C를 통해 프록시되는 클라이언트의 후속 요청은 이 주소로 전달됩니다.

특정 연결 주소가 서버 인증서에 포함되지 않은 경우 TLS 확인도 실패하고 502 오류가 발생하여 MRA 로그인이 실패하는 경우를 예로 들 수 있습니다.

```
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**

https://www.base64decode.org/ 여기서 c3RldmVuLmxhYi9odHRwcy8xMC40OC4zNi4yMTUvODQ0Mw는 steven.lab/https/10.48.36.215/8443으로 변환되며, 이는 cucm.steven.lab이 아닌 연결 주소로 10.48.36.215에 연결해야 함을 보여줍니다. 패킷 캡처에 표시된 것처럼 CUCM tomcat 인증서에는 SAN의 IP 주소가 포함되어 있지 않으므로 오류가 발생합니다.

쉽게 검증하는 방법

다음 단계를 통해 이 동작이 변경되는지 쉽게 확인할 수 있습니다.

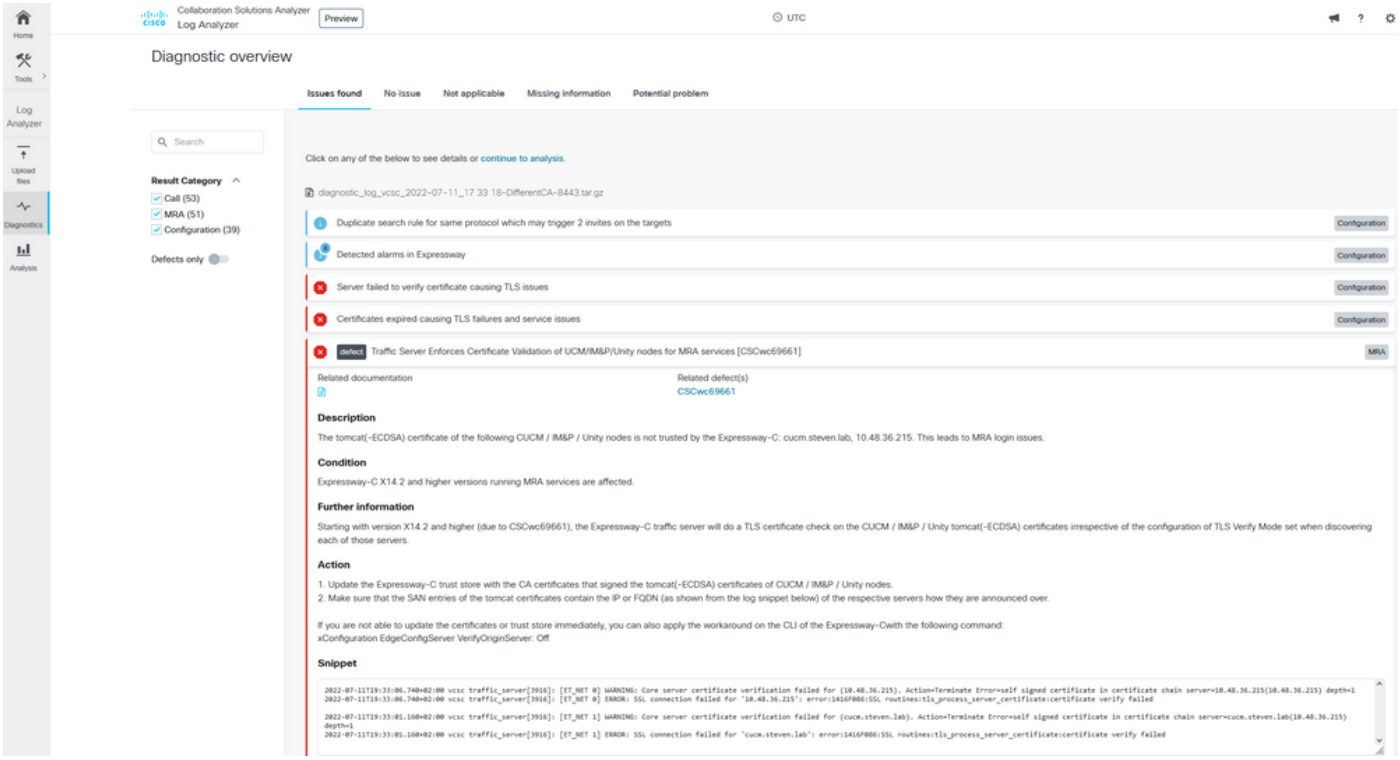
1. Expressway-E 및 C 서버에서 **Maintenance(유지 관리) > Diagnostics(진단) > Diagnostic Logging**(클러스터의 경우 마스터 노드에서 시작하기만 하면 됨)에서 진단 로깅을 시작합니다.

2. 업그레이드 후 MRA 로그인을 시도하거나 손상된 기능을 테스트합니다.

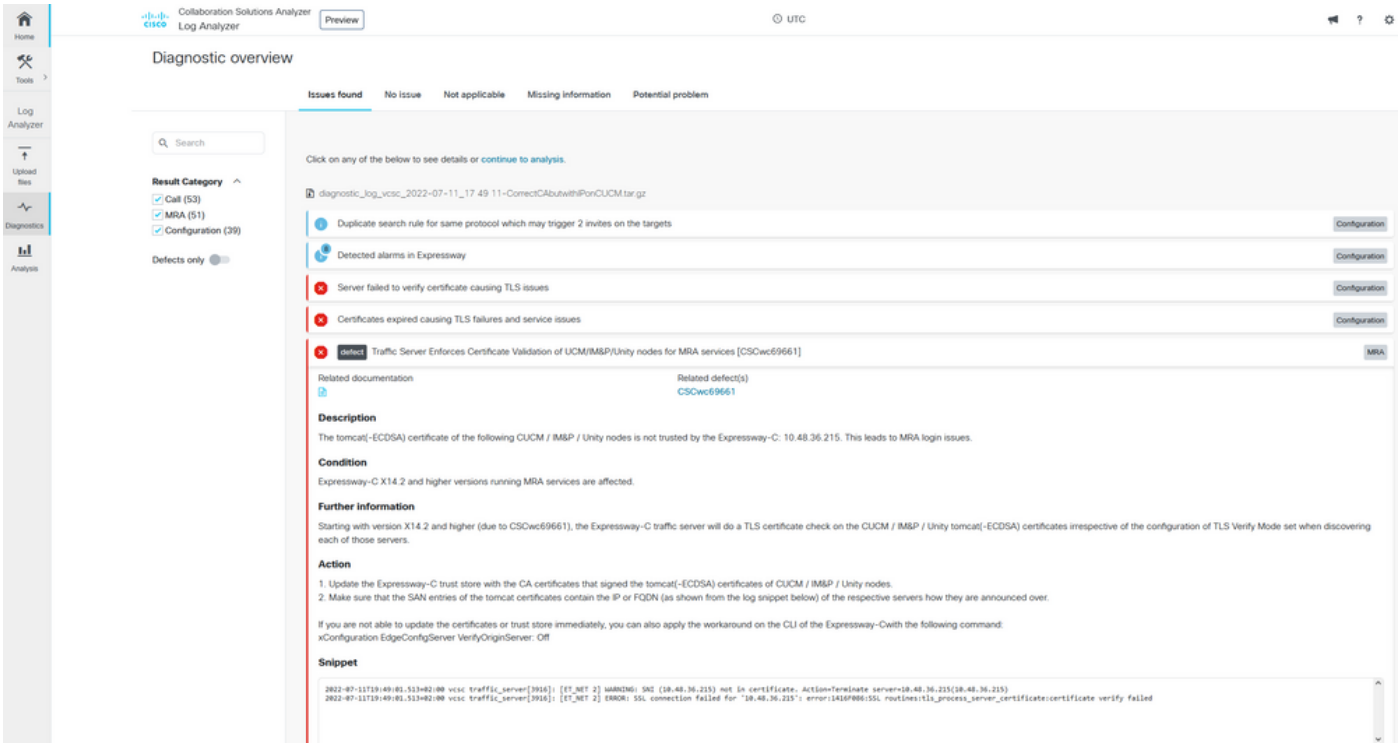
3. 오류가 발생할 때까지 기다린 다음 Expressway-E 및 C 서버에서 진단 로깅을 중지합니다(클러스터의 경우 클러스터의 각 노드에서 개별적으로 로그를 수집해야 함).

4. Collaboration Solution Analyzer 툴에서 로그 업로드 및 분석

5. 문제가 발생하면 영향을 받는 각 서버에 대해 이 변경과 관련된 가장 최근의 경고 및 오류 라인을 선택합니다



CA 진단 서명



SNI 진단 서명

솔루션

장기적으로 해결책은 TLS 검증이 잘 되게 하는 것이다. 수행할 작업은 표시되는 경고 메시지에 따라 달라집니다.

경고 표시 시 *에 대한 코어 서버 인증서를 확인하지 못했습니다(<server-FQDN-or-IP>). Action=Terminate Error=자체 서명 인증서 서버=cucm.steven.lab(10.48.36.215)* ***depth=x*** 메시지, 그러면 Expressway-C 서버에서 트러스트 저장소를 적절하게 업데이트해야 합니다. 이 인증서에 서명한 CA 체인(깊이 > 0) 또는 **Maintenance(유지 관리) > Security(보안) > Trusted CA Certificate(신뢰할 수 있는 CA 인증서)**에서 자체 서명된 인증서(깊이 = 0)로 **구성된** 경우. 클러스터의 모든 서버에서 이 작업을 수행해야 합니다. 또 다른 옵션은 Expressway-C 신뢰 저장소에서 알려진 CA가 원격 인증서를 서명하는 것입니다.

경고 표시 시 *인증서 메시지에 없는 SNI(<server-FQDN-or-IP>)*는 이 서버 FQDN 또는 IP가 제공된 인증서에 포함되지 않음을 나타냅니다. 인증서를 해당 정보를 포함하도록 조정하거나, System(시스템) > Server(서버)의 CUCM에서 서버 인증서에 포함된 컨피그레이션으로 컨피그레이션을 수정한 다음 Expressway-C 서버에서 컨피그레이션을 새로고침하여 이를 고려할 수 있습니다.

단기 솔루션은 X14.2.0 이전의 이전 동작으로 대체하도록 문서화된 해결 방법을 적용하는 것입니다. 새로 도입된 명령을 사용하여 Expressway-C 서버 노드에서 CLI를 통해 수행할 수 있습니다.

```
xConfiguration EdgeConfigServer VerifyOriginServer: Off
```

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.