

Risoluzione dei problemi relativi alla verifica dei certificati del server Expressway Traffic per i servizi MRA

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Catena CA attendibile](#)

[Controllo SAN o CN](#)

[Modifica del comportamento](#)

[Versioni inferiori a X14.2.0](#)

[Versioni di X14.2.0 e successive](#)

[Risoluzione dei problemi](#)

[1. La CA che ha firmato il certificato remoto non è attendibile](#)

[2. L'indirizzo di connessione \(FQDN o IP\) non è contenuto nel certificato](#)

[Come convalidarlo in modo semplice](#)

[Soluzione](#)

[Informazioni correlate](#)

Introduzione

Questo documento descrive la modifica del comportamento nelle versioni Expressway di X14.2.0 e versioni successive collegata all'ID bug Cisco [CSCwc69661](#) o all'ID bug Cisco [CSCwa25108](#).

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

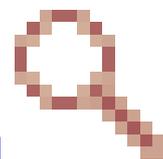
- Configurazione di base di Expressway
- Configurazione di base MRA

Componenti usati

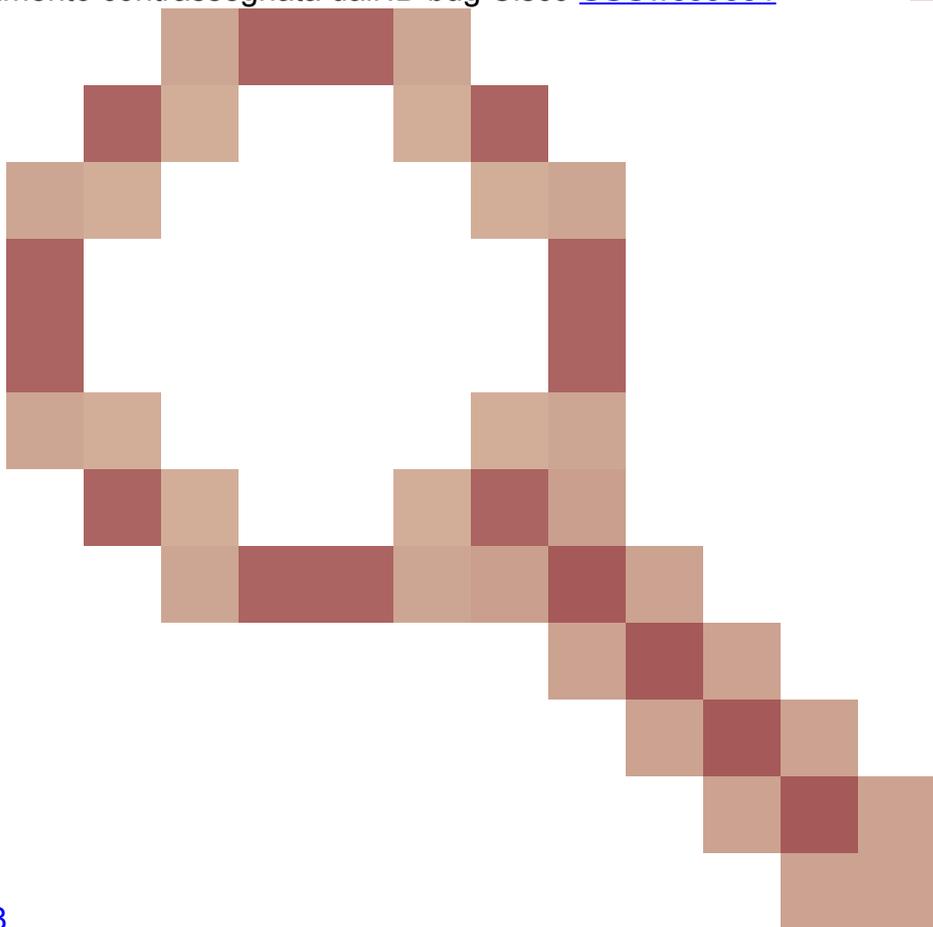
Le informazioni di questo documento si basano su Cisco Expressway versione X14.2 e successive.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse



Con questa modifica di comportamento contrassegnata dall'ID bug Cisco [CSCwc69661](#)



o dall'ID bug Cisco [CSCwa25108](#)

, il server del traffico sulla piattaforma Expressway esegue la verifica dei certificati di Cisco Unified Communications Manager (CUCM), Cisco Unified Instant Messaging & Presence (IM&P) e dei nodi server Unity per i servizi Mobile e Remote Access (MRA). Questa modifica può causare errori di accesso MRA dopo un aggiornamento della piattaforma Expressway.

HTTPS (Hypertext Transfer Protocol Secure) è un protocollo di comunicazione sicuro che utilizza TLS (Transport Layer Security) per crittografare la comunicazione. Questo canale sicuro viene creato mediante l'utilizzo di un certificato TLS scambiato nell'handshake TLS. Questo server ha due finalità: autenticazione (per sapere a chi si sta connettendo la parte remota) e privacy (la crittografia). L'autenticazione protegge dagli attacchi man-in-the-middle e la privacy impedisce agli aggressori di intercettare e manomettere la comunicazione.

La verifica TLS (Certificato) viene eseguita con l'obiettivo dell'autenticazione e consente di essere certi di aver effettuato la connessione alla parte remota corretta. La verifica è costituita da due

elementi distinti:

1. Catena di Autorità di certificazione (CA) attendibili
2. Nome alternativo del soggetto (SAN) o nome comune (CN)

Catena CA attendibile

Affinché Expressway-C consideri attendibile il certificato inviato da CUCM / IM&P / Unity, è necessario che sia in grado di stabilire un collegamento da tale certificato a un'Autorità di certificazione (CA) di livello superiore (principale) considerata attendibile. Tale collegamento, ovvero una gerarchia di certificati che collega un certificato di entità a un certificato CA radice, è denominato catena di attendibilità. Per poter verificare tale catena di attendibilità, ogni certificato contiene due campi: Emittente (o 'Rilasciato da') e Oggetto (o 'Rilasciato a').

I certificati server, ad esempio quello inviato da CUCM a Expressway-C, hanno in genere nel campo 'Oggetto' il nome di dominio completo (FQDN) della CN:

```
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
```

Esempio di certificato server per CUCM.vngtp.lab. Il nome di dominio completo (FQDN) è presente nell'attributo CN del campo Oggetto insieme ad altri attributi, quali Paese (C), Stato (ST), Posizione (L), ... Si noti inoltre che il certificato del server viene rilasciato da una CA denominata vngtp-ACTIVE-DIR-CA.

Le CA di livello superiore (CA radice) possono inoltre rilasciare un certificato per identificarsi. In questo certificato CA radice, è possibile notare che l'autorità emittente e l'oggetto hanno lo stesso valore:

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

Si tratta di un certificato rilasciato da una CA radice per identificarsi.

In una situazione tipica, le CA radice non rilasciano direttamente certificati server. Al contrario, emettono certificati per altre CA. Tali altre CA vengono quindi definite CA intermedie. Le CA intermedie possono a loro volta emettere direttamente certificati server o certificati per altre CA intermedie. Si può verificare una situazione in cui un certificato server viene rilasciato dalla CA intermedia 1, che a sua volta ottiene un certificato dalla CA intermedia 2 e così via. Finché la CA intermedia non ottiene il proprio certificato direttamente dalla CA radice:

```
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
```

Ora, affinché Expressway-C consideri attendibile il certificato server inviato da CUCM, è necessario che sia in grado di creare la catena di attendibilità da tale certificato server fino a ottenere un certificato CA radice. A tale scopo, è necessario caricare il certificato CA radice e tutti i certificati CA intermedi (se presenti, il che non è il caso se la CA radice avrebbe rilasciato direttamente il certificato server di CUCM) nell'archivio di attendibilità di Expressway-C.

 Nota: sebbene i campi Emittente e Oggetto siano facili da creare e leggibili, CUCM non utilizza questi campi nel certificato. Vengono invece utilizzati i campi 'Identificatore chiave autorità X509v3' e 'Identificatore chiave oggetto X509v3' per creare la catena di attendibilità. Tali chiavi contengono identificatori per i certificati più accurati rispetto a quelli utilizzati nei campi Oggetto/Emittente: possono esistere 2 certificati con gli stessi campi Oggetto/Emittente, ma uno di essi è scaduto e uno è ancora valido. Entrambi avrebbero un identificatore di chiave del soggetto X509v3 diverso, in modo che CUCM possa ancora determinare la corretta catena di fiducia.

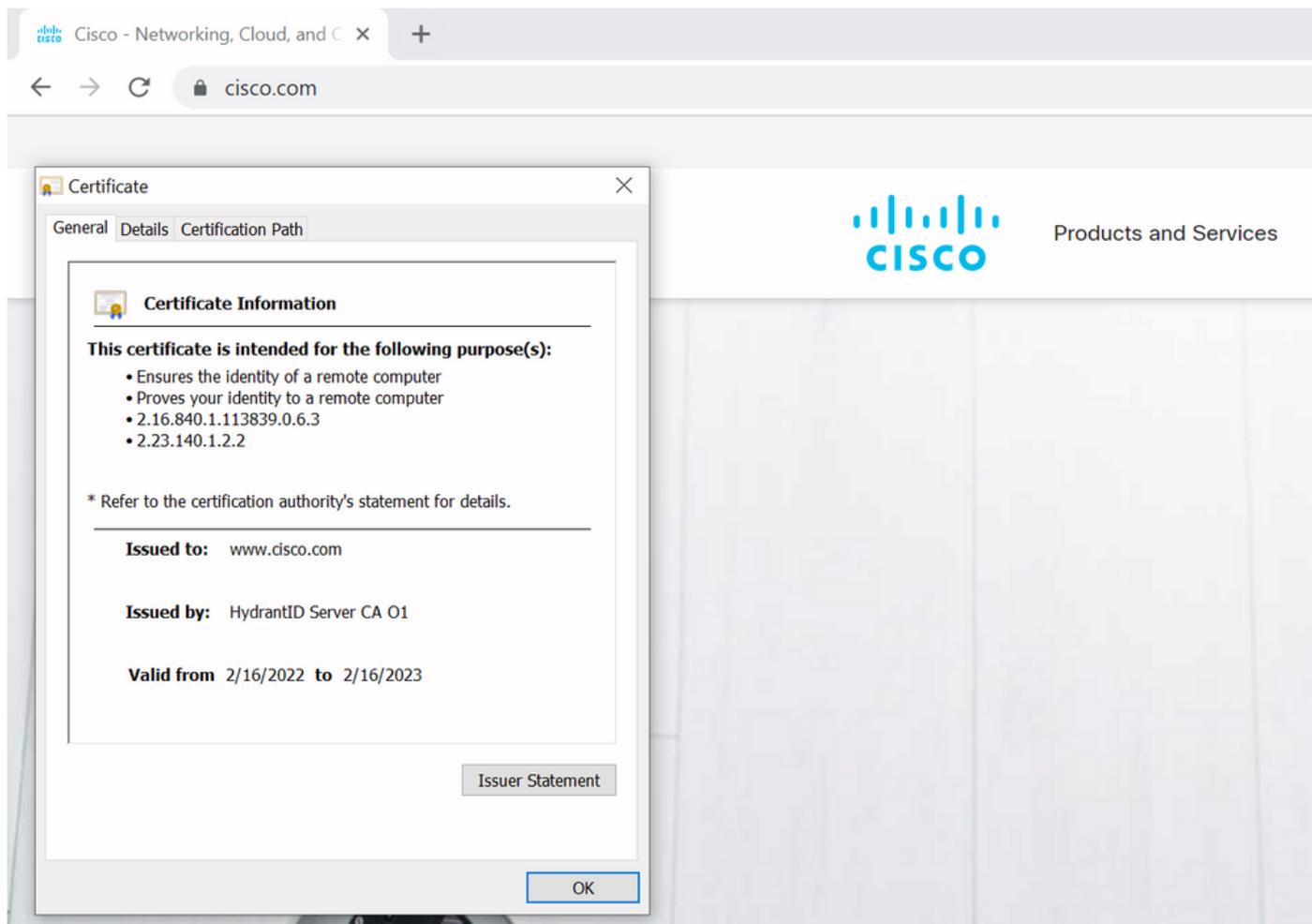
Questo non è il caso di Expressway, tuttavia come per l'ID bug Cisco [CSCwa12905](#), e non è possibile caricare due certificati diversi, ad esempio autofirmati, nell'archivio di attendibilità di Expressway con lo stesso nome comune (CN). Per risolvere il problema, è possibile utilizzare certificati firmati dall'autorità di certificazione o nomi comuni diversi per tali certificati o verificare che utilizzi sempre lo stesso certificato (potenzialmente tramite la funzionalità di riutilizzo dei certificati di CUCM 14).

Controllo SAN o CN

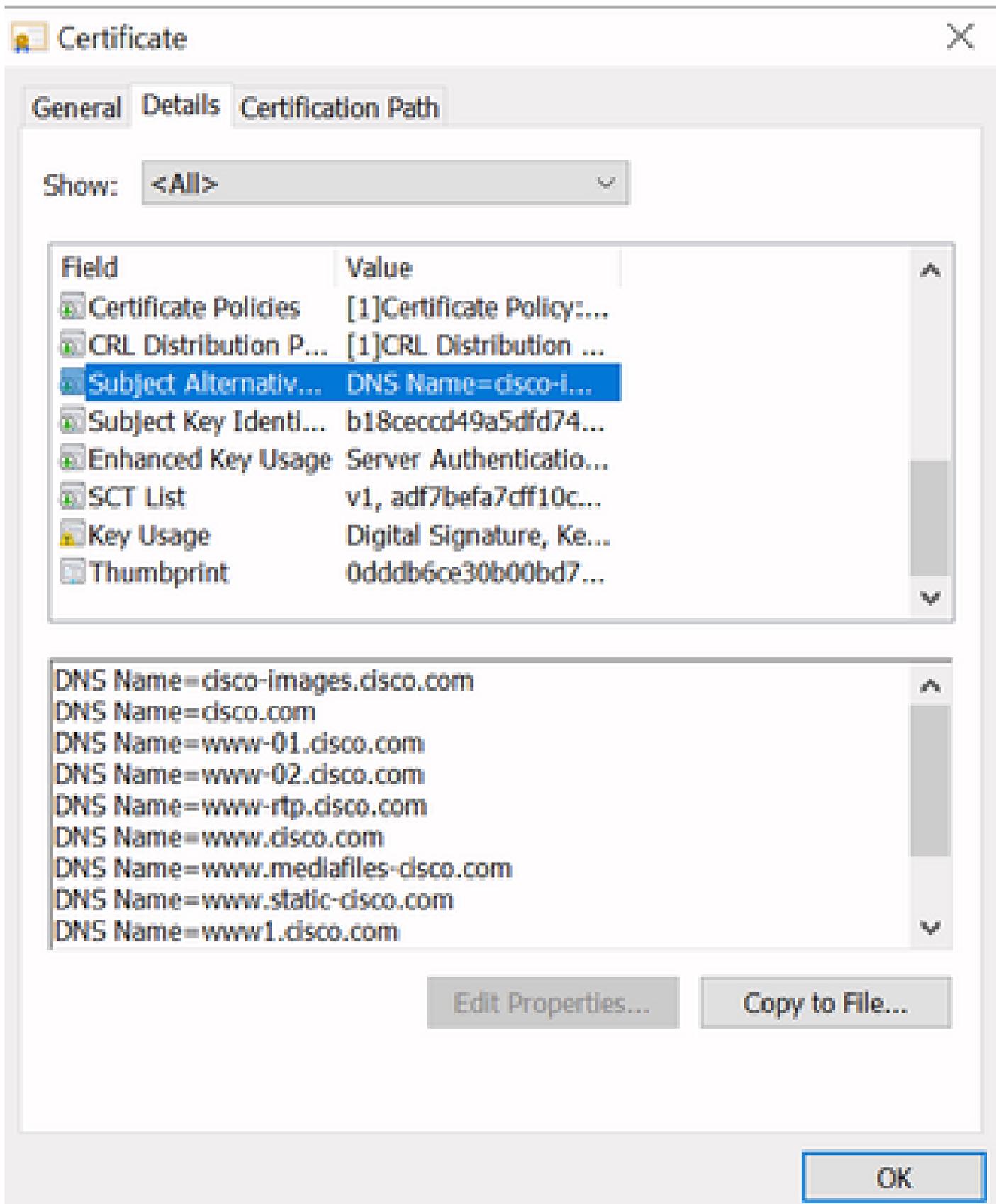
Il passaggio 1 consente di estrarre l'archivio di attendibilità. In questo caso, tuttavia, chiunque disponga di un certificato firmato da una CA nell'archivio di attendibilità sarà valido. Questo chiaramente non è sufficiente. Pertanto, è disponibile un ulteriore controllo che consente di verificare che il server a cui ci si connette sia effettivamente quello corretto. L'indirizzo a cui si riferisce la richiesta è quello indicato nella richiesta.

Lo stesso tipo di operazione si verifica nel browser, quindi esaminiamo questo aspetto attraverso

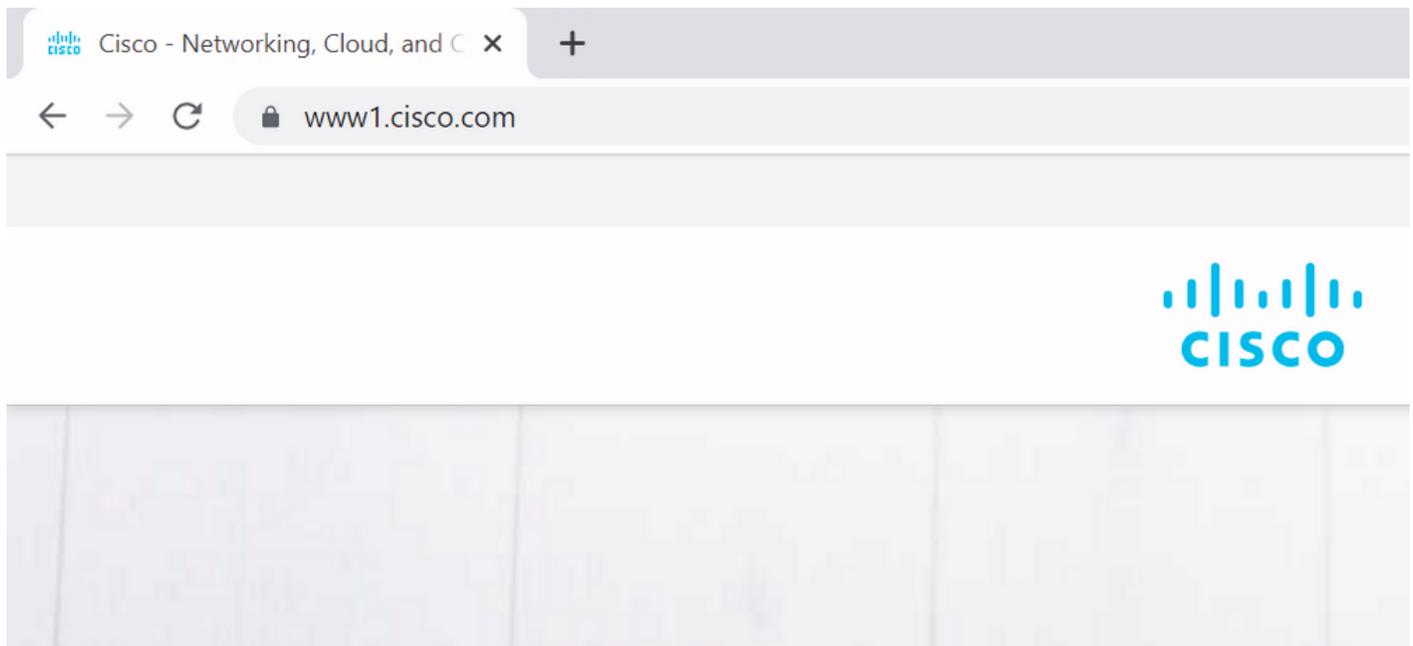
un esempio. Se si passa a <https://www.cisco.com>, accanto all'URL immesso viene visualizzata un'icona a forma di lucchetto che indica che la connessione è attendibile. Questo si basa sia sulla catena di attendibilità della CA (dalla prima sezione) sia sul controllo della SAN o della CN. Se si apre il certificato (tramite il browser facendo clic sull'icona a forma di lucchetto), si noterà che il campo Nome comune (visualizzato nel campo 'Rilasciato a:') è impostato su www.cisco.com e corrisponde esattamente all'indirizzo a cui si desidera connettersi. In questo modo, è possibile essere certi di connettersi al server corretto, in quanto la CA che ha firmato il certificato e che esegue la verifica prima della distribuzione del certificato è considerata attendibile.



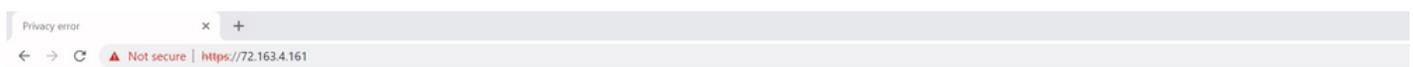
Quando si esaminano i dettagli del certificato e in particolare le voci SAN, si osserva che lo stesso vale per altri FQDN:



Ciò significa che quando si richiede la connessione a <https://www1.cisco.com>, ad esempio, questa viene visualizzata anche come connessione protetta, in quanto è inclusa nelle voci SAN.



Tuttavia, se si sceglie <https://www.cisco.com> ma si accede direttamente all'indirizzo IP (<https://72.163.4.161>), la connessione non risulterà sicura in quanto l'autorità di certificazione che l'ha firmata non è considerata attendibile, ma il certificato presentato non contiene l'indirizzo (72.163.4.161) utilizzato per la connessione al server.



```
Command Prompt - nslookup
C:\Users\stejans>
C:\Users\stejans>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](#)

Hide advanced

Back to safety

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 \(unsafe\)](#)

Nel browser è possibile ignorare questa impostazione, ma è possibile abilitarla sulle connessioni TLS in modo che non sia consentito ignorarla. È pertanto importante che i certificati contengano i nomi CN o SAN corretti che la parte remota intende utilizzare per connettersi.

Modifica del comportamento

I servizi MRA si basano molto su diverse connessioni HTTPS su Expressways verso i server

CUCM / IM&P / Unity per autenticarsi correttamente e raccogliere le informazioni appropriate specifiche per il client che esegue il login. Questa comunicazione in genere ha luogo sulle porte 8443 e 6972.

Versioni inferiori a X14.2.0

Nelle versioni precedenti a X14.2.0, il server di traffico su Expressway-C che gestisce queste connessioni HTTPS sicure non ha verificato il certificato presentato dall'estremità remota. Questo potrebbe portare ad attacchi di tipo man-in-the-middle. Nella configurazione MRA è disponibile un'opzione per la verifica del certificato TLS mediante la configurazione di 'Modalità verifica TLS' su 'Attivata' quando si aggiungono server CUCM / IM&P / Unity in Configurazione > Comunicazioni unificate > Server CM unificati / Nodi IM e Servizio presenza / Server Unity Connection. L'opzione di configurazione e la casella delle informazioni rilevanti vengono mostrate come esempio, a indicare che non verifica l'FQDN o l'IP nella SAN, nonché la validità del certificato e se è firmato da una CA attendibile.



Cisco Expressway-C

Status > System > **Configuration >** Applications > Users > Maintenance >

Unified CM servers You are here: [Configuration](#) >

Unified CM server lookup

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

Save Delete Cancel

Questo controllo di verifica del certificato TLS viene eseguito solo al rilevamento dei server CUCM/IM&P/Unity e non al momento dell'accesso MRA ai vari server. Un primo inconveniente di questa configurazione è che la verifica solo per l'indirizzo dell'autore aggiunto. Non verifica se il certificato nei nodi del sottoscrittore è stato impostato correttamente in quanto recupera le informazioni sul nodo del sottoscrittore (FQDN o IP) dal database del nodo del server di pubblicazione. Un secondo inconveniente di questa configurazione è che ciò che viene annunciato ai client MRA come informazioni di connessione può essere diverso dall'indirizzo dell'autore che è

stato messo nella configurazione Expressway-C. Ad esempio su CUCM, in Sistema > Server è possibile annunciare il server all'esterno con un indirizzo IP (ad esempio 10.48.36.215) e questo viene poi utilizzato dai client MRA (tramite la connessione Expressway proxy), tuttavia è possibile aggiungere il CUCM su Expressway-C con il FQDN di cucm.steven.lab. Si supponga quindi che il certificato tomcat di CUCM contenga cucm.steven.lab come voce SAN ma non l'indirizzo IP. La ricerca con 'Modalità di verifica TLS' impostata su 'Attivata' avrà esito positivo, ma le comunicazioni effettive provenienti dai client MRA possono avere come destinazione un FQDN o un IP diverso e pertanto non superano la verifica TLS.

Versioni di X14.2.0 e successive

A partire dalla versione X14.2.0, il server Expressway esegue la verifica del certificato TLS per ogni singola richiesta HTTPS effettuata tramite il server di traffico. Ciò significa che questa operazione viene eseguita anche quando 'TLS Verify Mode' è impostato su 'Off' durante il rilevamento dei nodi CUCM / IM&P / Unity. Se la verifica non ha esito positivo, l'handshake TLS non viene completato e la richiesta non riesce. Ciò può causare, ad esempio, la perdita di funzionalità quali problemi di ridondanza o failover o errori di accesso completi. Inoltre, se 'TLS Verify Mode' è impostata su 'On', non è possibile garantire che tutte le connessioni funzionino correttamente, come illustrato nell'esempio riportato di seguito.

I certificati esatti che Expressway controlla verso i nodi CUCM / IM&P / Unity sono come mostrato nella sezione della [guida MRA](#).

Oltre all'impostazione predefinita per la verifica TLS, in X14.2 è stata introdotta una modifica che potrebbe annunciare un ordine di preferenza diverso per l'elenco di cifratura, a seconda del percorso di aggiornamento. Ciò può causare connessioni TLS impreviste dopo un aggiornamento software, perché può accadere che prima dell'aggiornamento richiesto per il certificato Cisco Tomcat o Cisco CallManager da CUCM (o qualsiasi altro prodotto che abbia un certificato separato per l'algoritmo ECDSA) ma che dopo l'aggiornamento richieda per la variante ECDSA (che è la variante di cifratura più sicura in realtà di RSA). È possibile che i certificati Cisco Tomcat-ECDSA o Cisco CallManager-ECDSA siano firmati da un'autorità di certificazione diversa o semplicemente autofirmati (impostazione predefinita).

La modifica dell'ordine delle preferenze di cifratura non è sempre rilevante in quanto dipende dal percorso di aggiornamento, come illustrato nelle [note di rilascio di Expressway X14.2.1](#). In breve, è possibile vedere da Manutenzione > Sicurezza > Cifre per ciascuno dei cifrari se è preceduto o meno "ECDHE-RSA-AES256-GCM-SHA384:". In caso contrario, preferisce la cifratura ECDSA più recente a quella RSA. In caso affermativo, si avrà il comportamento precedente di RSA che ha la preferenza più alta.

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**).

La verifica TLS potrebbe non riuscire in questo scenario in due modi, descritti in dettaglio più

avanti:

1. La CA che ha firmato il certificato remoto non è attendibile
 - a. Certificato autofirmato
 - b. Certificato firmato da una CA sconosciuta
2. L'indirizzo di connessione (FQDN o IP) non è contenuto nel certificato

Risoluzione dei problemi

Gli scenari successivi mostrano uno scenario simile in un ambiente lab in cui l'accesso MRA non è riuscito dopo un aggiornamento di Expressway da X14.0.7 a X14.2. Condividono analogie nei registri, ma la risoluzione è diversa. I registri vengono raccolti dalla registrazione diagnostica (da Manutenzione > Diagnostica > Registrazione diagnostica) avviata prima dell'accesso all'MRA e interrotta dopo che l'accesso all'MRA non è riuscito. Non è stata abilitata alcuna registrazione di debug aggiuntiva.

1. La CA che ha firmato il certificato remoto non è attendibile

Il certificato remoto potrebbe essere firmato da una CA non inclusa nell'archivio di attendibilità di Expressway-C oppure potrebbe essere un certificato autofirmato (in sostanza anche una CA) che non viene aggiunto nell'archivio di attendibilità del server Expressway-C.

Nell'esempio riportato di seguito, è possibile osservare che le richieste inviate a CUCM (10.48.36.215 - cucm.steven.lab) vengono gestite correttamente sulla porta 8443 (risposta 200 OK) ma genera un errore (risposta 502) sulla porta 6972 per la connessione TFTP.

```
<#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="net
```

```
2022-07-11T18:55:25.917+02:00 vcsc traffic_server[18242]: Event="Request Allowed" Detail="Access allow
```

```
2022-07-11T18:55:25.917+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:25,916" Module="net
```

```
2022-07-11T18:55:25.955+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:25,955" Module="net
```

```
2022-07-11T18:55:25.956+02:00 vcsc traffic_server[18242]: UTCTime="2022-07-11 16:55:25,955" Module="net
```

```
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="net
```

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

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

```
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 vscs 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 vscs traffic_server[18242]: UTCTime="2022-07-11 16:55:26,024" Module="net

502 connect failed

"

L'errore di 'verifica certificato non riuscita' indica che Expressway-C non è stato in grado di convalidare l'handshake TLS. Il motivo è indicato nella riga di avviso in quanto indica un certificato autofirmato. Se la profondità è indicata da 0, si tratta di un certificato autofirmato. Quando la profondità è maggiore di 0, significa che dispone di una catena di certificati e pertanto è firmata da un'autorità di certificazione sconosciuta (dal punto di vista di Expressway-C).

Quando si controlla il file pcap che è stato raccolto in corrispondenza dei timestamp indicati dai log di testo, si può notare che CUCM presenta il certificato con CN come cucm-ms.steven.lab (e cucm.steven.lab come SAN) firmato da steven-DC-CA per Expressway-C sulla porta 8443.

The screenshot displays a network capture in Wireshark. The top pane shows a list of packets, with packet 4717 highlighted in red, indicating an error. The bottom pane shows the details of the certificate, which is self-signed and issued by 'cucm-ms.steven.lab'. The certificate's subject is 'cucm.steven.lab' and it includes 'cucm.steven.lab' as a Subject Alternative Name (SAN).

```
eth0_diagnostic_logging_tcpdump00_vscs_2022-07-11_16_55_44.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.port=8443
No. Time Source Src port Destination Dest port Protocol DSCP VLAN Length Info
4691 2022-07-11 16:55:25.916080 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 16:55:25.916083 10.48.36.215 8443 10.48.36.46 35622 TCP CS0 74 8443 → 35622 [SYN, ACK] Seq=0 Ack=1 Wm=2098 Len=0 MSS=1460 SACK_PERM=1 TSval=343633238 TSecr=878570435 WS=128
4693 2022-07-11 16:55:25.916093 10.48.36.46 35622 10.48.36.215 8443 TCP CS0 66 35622 → 8443 [ACK] Seq=1 Ack=1 Wm=64256 Len=0 TSval=878570435 TSecr=343633238
4694 2022-07-11 16:55:25.917832 10.48.36.46 8443 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 16:55:25.938398 10.48.36.46 35622 10.48.36.215 8443 TCP CS0 66 35622 → 8443 [ACK] Seq=518 Ack=1449 Wm=64128 Len=0 TSval=878570457 TSecr=343633251
4697 2022-07-11 16:55:25.938489 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 Wm=63488 Len=0 TSval=878570457 TSecr=343633251
4699 2022-07-11 16:55:25.940187 10.48.36.46 8443 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 388 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 Wm=64256 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 Wm=35072 Len=0 TSval=343633256 TSecr=878570462
4704 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 Wm=35072 Len=1448 TSval=343633268 TSecr=878570462 [TCP segment of a reassembled PDU]
4705 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 Wm=64256 Len=0 TSval=878570473 TSecr=343633268
4706 2022-07-11 16:55:25.954873 10.48.36.215 8443 10.48.36.46 35622 TLSv1.2 CS0 1257 Application Data
4707 2022-07-11 16:55:25.955712 10.48.36.46 35622 10.48.36.215 8443 TCP CS0 66 35622 → 8443 [ACK] Seq=3121 Ack=5734 Wm=63488 Len=0 TSval=878570473 TSecr=343633268
4708 2022-07-11 16:55:25.955798 10.48.36.46 35622 10.48.36.215 8443 TCP CS0 66 35622 → 8443 [FIN, ACK] Seq=3152 Ack=5734 Wm=64128 Len=0 TSval=878570474 TSecr=343633268
4709 2022-07-11 16:55:25.956123 10.48.36.215 8443 10.48.36.46 35622 TLSv1.2 CS0 97 Encrypted Alert
4710 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 Wm=0 Len=0
4711 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=5705 Ack=3153 Wm=35072 Len=0 TSval=343633269 TSecr=878570474
4712 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 Wm=0 Len=0
Certificates (2420 Bytes)
Certificate Length: 1887
Certificate: 308205df308204c7a0030201020134500000120850805... (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-countryName=)
signedCertificate
version: v3 (2)
serialNumber: 0x4500000120850805d34080844200020000112
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)
GeneralName: dNSName (2)
GeneralName: dNSName (2)
GeneralName: dNSName (3)
GeneralName: cucm.steven.lab
Extension (id-ce-subjectKeyIdentifier)
Extension (id-ce-authorityKeyIdentifier)
Extension (id-ce-certificateType)
Extension (id-ce-authorityInfoAccessSyntax)
Extension (id-ms-certificate-template)
Extension (id-ms-application-certificate-policies)
algorithmIdentifier (sha1withRSAEncryption)
padding: 0
encrypted: 9fb7f8741637a2a92071ef08f22709cc7ec48478c82d...
Certificate Length: 910
Certificate: 3082030a30820272a00302010201062176f3fc29398004... (id-at-commonName=steven-DC-CA, dc=steven, dc=lab)
Secure Sockets Layer
```

Ma quando ispezioniamo il certificato presentato sulla porta 6972, possiamo vedere che è un certificato autofirmato (l'emittente è se stesso) con CN impostato come cucm-EC.steven.lab. L'estensione -EC fornisce l'indicazione che si tratta del certificato ECDSA installato su CUCM.

No.	Time	Source	Src port	Destination	Dest port	Protocol	OSPF	VLAN	Length	Info
4730	2022-07-11 16:55:26.006608	10.48.36.46	31576	10.48.36.215	6972	TCP	C50		74	31576 → 6972 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=878578525 TSecr=0 WS=128
4731	2022-07-11 16:55:26.006851	10.48.36.215	6972	10.48.36.46	31576	TCP	C50		74	6972 → 31576 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=343633320 TSecr=878578525 WS=128
4732	2022-07-11 16:55:26.006892	10.48.36.46	31576	10.48.36.215	6972	TCP	C50		66	31576 → 6972 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=878578525 TSecr=343633320
4733	2022-07-11 16:55:26.007100	10.48.36.46	31576	10.48.36.215	6972	TLSv1.2	C50		583	Client Hello
4734	2022-07-11 16:55:26.016350	10.48.36.215	6972	10.48.36.46	31576	TLSv1.2	C50		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	C50		66	31576 → 6972 [ACK] Seq=518 Ack=1449 Win=64128 Len=0 TSval=878578535 TSecr=343633320
4736	2022-07-11 16:55:26.016408	10.48.36.215	6972	10.48.36.46	31576	TLSv1.2	C50		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	C50		66	31576 → 6972 [ACK] Seq=518 Ack=1882 Win=63744 Len=0 TSval=878578535 TSecr=343633320
4738	2022-07-11 16:55:26.016703	10.48.36.46	31576	10.48.36.215	6972	TLSv1.2	C50		73	Alert (Level: Fatal, Description: Unknown CA)
4739	2022-07-11 16:55:26.016621	10.48.36.46	31576	10.48.36.215	6972	TCP	C50		74	31576 → 6972 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=878578535 TSecr=0 WS=128
4740	2022-07-11 16:55:26.016965	10.48.36.46	31576	10.48.36.215	6972	TCP	C50		66	31576 → 6972 [RST, ACK] Seq=525 Ack=1882 Win=64128 Len=0 TSval=878578535 TSecr=343633320
4741	2022-07-11 16:55:26.016984	10.48.36.215	6972	10.48.36.46	31576	TCP	C50		74	6972 → 31576 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=343633330 TSecr=878578535 WS=128
4742	2022-07-11 16:55:26.017009	10.48.36.46	31576	10.48.36.215	6972	TCP	C50		66	31576 → 6972 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=878578535 TSecr=343633330
4743	2022-07-11 16:55:26.017101	10.48.36.215	6972	10.48.36.46	31576	TCP	C50		66	6972 → 31576 [FIN, ACK] Seq=1882 Ack=525 Win=10880 Len=0 TSval=343633330 TSecr=878578535
4744	2022-07-11 16:55:26.017121	10.48.36.46	31576	10.48.36.215	6972	TCP	C50		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	C50		583	Client Hello
4746	2022-07-11 16:55:26.024226	10.48.36.215	6972	10.48.36.46	31578	TLSv1.2	C50		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	C50		66	31578 → 6972 [ACK] Seq=518 Ack=1449 Win=64128 Len=0 TSval=878578543 TSecr=343633337
4748	2022-07-11 16:55:26.024298	10.48.36.215	6972	10.48.36.46	31578	TLSv1.2	C50		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	C50		66	31578 → 6972 [ACK] Seq=518 Ack=1883 Win=63744 Len=0 TSval=878578543 TSecr=343633337
4750	2022-07-11 16:55:26.024548	10.48.36.46	31578	10.48.36.215	6972	TLSv1.2	C50		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	C50		66	31578 → 6972 [RST, ACK] Seq=525 Ack=1883 Win=64128 Len=0 TSval=878578543 TSecr=343633337
4767	2022-07-11 16:55:26.033159	10.48.36.46	31500	10.48.36.215	6972	TCP	C50		74	31500 → 6972 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=878578061 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: 308202803020214808302010202107470ee62271e3d1346... (id-at-localityName=Oiegen,id-at-stateOrProvinceName=Belgium,id-at-commonName=cucm-EC.steven.lab,id-at-organizationalUnitName=TAC,id-at-organizationName=Cisco,id-at-countryName=BE)
          signedCertificate
            version: v3 (2)
            serialNumber: 007470ee62271e3d1346109946f8a3bf5d
            signature (ecdsa-with-SHA384)
            issuer: rdmsSequence (8)
            rdmsSequence: 6 items (id-at-localityName=Oiegen,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: rdmsSequence (8)
            subjectPublicKeyInfo
            extensions: 5 items
              Extension (id-ce-keyUsage)
              Extension (id-ce-extendedKeyUsage)
              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: dnName (2)
                      dnName: cucm.steven.lab
                    algorithmIdentifier (ecdsa-with-SHA384)
                    padding: 0
                    encrypted: 30640230125430d5e6e74570b1171eb49f8a30e6cd090d...
  TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
  
```

In CUCM in Cisco Unified OS Administration (Amministrazione del sistema operativo unificato Cisco), è possibile esaminare i certificati in uso in Protezione > Gestione certificati, come mostrato di seguito. Viene visualizzato un certificato diverso per tomcat e tomcat-ECDSA in cui il tomcat è firmato dall'autorità di certificazione (e considerato attendibile da Expressway-C) mentre il certificato tomcat-ECDSA è autofirmato e non considerato attendibile da Expressway-C.

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-ECCSA	ucm-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-RTF-002	Self-signed	RSA	CAP-RTF-002	CAP-RTF-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 UCH using the MIC when associated with a secure profile.
CallManager-trust	CAPP-4b26468	Self-signed	RSA	CAPP-4b26468	CAPP-4b26468	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-RTF-001	Self-signed	RSA	CAP-RTF-001	CAP-RTF-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 UCH 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 UCH using the MIC when associated with a secure profile.
CallManager-trust	ACT2_SU01_CA	CA-signed	RSA	ACT2_SU01_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 UCH using the MIC when associated with a secure profile.
CallManager-trust	vngtp-ACTIVE-00R-CA	Self-signed	RSA	vngtp-ACTIVE-00R-CA	vngtp-ACTIVE-00R-CA	02/10/2024	Trust Certificate
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 UCH 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 UCH 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 UCH using the MIC when associated with a secure profile.
CallManager-trust	dcomics-WONDERWOMAN-CA	Self-signed	RSA	dcomics-WONDERWOMAN-CA	dcomics-WONDERWOMAN-CA	09/19/2037	CA-Intermediate
CallManager-trust	CAPP-616421bc	Self-signed	RSA	CAPP-616421bc	CAPP-616421bc	07/12/2025	
CAPP	CAPP-616421bc	Self-signed	RSA	cucm.steven.lab	CAPP-616421bc	07/12/2025	Self-signed certificate generated by system
CAPP-trust	CAP-RTF-002	Self-signed	RSA	CAP-RTF-002	CAP-RTF-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 UCH using the MIC when associated with a secure profile.
CAPP-trust	CAPP-4b26468	Self-signed	RSA	CAPP-4b26468	CAPP-4b26468	04/12/2020	
CAPP-trust	CAP-RTF-001	Self-signed	RSA	CAP-RTF-001	CAP-RTF-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 UCH using the MIC when associated with a secure profile.
CAPP-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 UCH using the MIC when associated with a secure profile.
CAPP-trust	ACT2_SU01_CA	CA-signed	RSA	ACT2_SU01_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 UCH using the MIC when associated with a secure profile.
CAPP-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 UCH using the MIC when associated with a secure profile.
CAPP-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 UCH using the MIC when associated with a secure profile.
CAPP-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 UCH using the MIC when associated with a secure profile.
CAPP-trust	CAPP-616421bc	Self-signed	RSA	CAPP-616421bc	CAPP-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	ITLRECOVERY_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-ECCSA	ucm-EC.steven.lab	CSR Only	EC	cucm.steven.lab	--	--	
tomcat-ECCSA	ucm-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	ucm-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	ucm-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-00R-CA	Self-signed	RSA	vngtp-ACTIVE-00R-CA	vngtp-ACTIVE-00R-CA	02/10/2024	Trust Certificate
tomcat-trust	dcomics-WONDERWOMAN-CA	Self-signed	RSA	dcomics-WONDERWOMAN-CA	dcomics-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. L'indirizzo di connessione (FQDN o IP) non è contenuto nel certificato

Oltre all'archivio di attendibilità, il server del traffico verifica anche l'indirizzo di connessione verso cui il client di Autorità registrazione integrità effettua la richiesta. Ad esempio, se avete impostato su CUCM in Sistema > Server il vostro CUCM con l'indirizzo IP (10.48.36.215), Expressway-C lo annuncia come tale al client e le richieste successive dal client (proxy attraverso Expressway-C) sono destinate a questo indirizzo.

Quando l'indirizzo di connessione non è contenuto nel certificato del server, anche la verifica TLS ha esito negativo e viene generato un errore 502 che, ad esempio, genera un errore di accesso MRA.

```
<#root>
```

```
2022-07-11T19:49:01.472+02:00 vcsc traffic_server[3916]: UTCTime="2022-07-11 17:49:01,472" Module="network"
HTTPMSG:
|GET http://vcs_control.steven.lab:8443/c3RldmVuLmXhYi9odHRwcy8xMC400C4zNi4yMTUvODQ0Mw/cucm-uds/user/emusk/...
...
```

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

dove c3RldmVuLmxhYi9odHRwcy8xMC40OC4zNi4yMTUvODQ0Mw converte (base64) in steven.lab/https/10.48.36.215/8443, a indicare che è necessario impostare la connessione a 10.48.36.215 come indirizzo di connessione anziché a cucm.steven.lab. Come mostrato nelle acquisizioni del pacchetto, il certificato tomcat CUCM non contiene l'indirizzo IP nella SAN e quindi viene generato l'errore.

Come convalidarlo in modo semplice

È possibile verificare se il comportamento cambia facilmente con i passaggi successivi:

1. Avviare la registrazione diagnostica sui server Expressway-E e C (idealmente con TCPDump abilitato) da Manutenzione > Diagnostica > Registrazione diagnostica (nel caso di un cluster, è sufficiente avviarlo dal nodo primario)
2. Tentare un accesso MRA o testare la funzionalità interrotta dopo l'aggiornamento
3. Attendere che si verifichi un errore e quindi arrestare la registrazione diagnostica sui server Expressway-E e C (nel caso di un cluster, assicurarsi di raccogliere i log da ogni singolo nodo del cluster)
4. Caricare e analizzare i log nello [strumento Collaboration Solution Analyzer](#)
5. Se si verifica il problema, vengono selezionate le righe di avvertenza e di errore più recenti relative alla modifica per ognuno dei server interessati

The screenshot shows the Cisco Collaboration Solutions Analyzer Log Analyzer interface. The main content area displays a diagnostic overview for the file 'diagnostic_log_vcsd_2022-07-11_17_33_18-DifferentCA-8443.tar.gz'. The interface includes a sidebar with navigation options (Home, Tools, Log Analyzer, Upload files, Diagnostics, Analysis) and a search bar. The main panel shows a list of issues found, with the selected issue being 'Traffic Server Enforces Certificate Validation of UCM/IMP/Unity nodes for MRA services [CSCw69661]'. The detailed view for this issue includes:

- Description:** The tomcat(-ECDSA) certificate of the following CUCM / IMP / 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 CSCw69661), the Expressway-C traffic server will do a TLS certificate check on the CUCM / IMP / Unity tomcat(-ECDSA) certificates irrespective of the configuration of TLS Verify Mode set when discovering each of those servers.
- Action:**
 - Update the Expressway-C trust store with the CA certificates that signed the tomcat(-ECDSA) certificates of CUCM / IMP / Unity nodes.
 - 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-C with the following command:
`xConfiguration EdgeConfigServer VerifyOriginServer: Off`
- Snippet:**

```

2022-07-11T19:33:06.740+02:00 vcsd traffic_server[3936]: [ET_NET 0] WARNING: Core server certificate verification failed for (10.48.36.215). Action:Terminate Error=ssl signed certificate in certificate chain server=10.48.36.215(10.48.36.215) depth=1
2022-07-11T19:33:06.740+02:00 vcsd traffic_server[3936]: [ET_NET 0] ERROR: SSL connection failed for "10.48.36.215": error:1416F080:SSL routines:tls_process_server_certificate:certificate verify failed
2022-07-11T19:33:06.160+02:00 vcsd traffic_server[3936]: [ET_NET 1] WARNING: Core server certificate verification failed for (cucm.steven.lab). Action:Terminate Error=ssl signed certificate in certificate chain server=cucm.steven.lab(10.48.36.215) depth=0
2022-07-11T19:33:06.160+02:00 vcsd traffic_server[3936]: [ET_NET 1] ERROR: SSL connection failed for "cucm.steven.lab": error:1416F080:SSL routines:tls_process_server_certificate:certificate verify failed

```

Firma diagnostica CA

The screenshot shows the Cisco Collaboration Solutions Analyzer Log Analyzer interface. The main content area displays a diagnostic overview for the file 'diagnostic_log_vcsd_2022-07-11_17_49_11-ConnectCAbutwithIPonCUCM.tar.gz'. The interface includes a sidebar with navigation options (Home, Tools, Log Analyzer, Upload files, Diagnostics, Analysis) and a search bar. The main panel shows a list of issues found, with the selected issue being 'Traffic Server Enforces Certificate Validation of UCM/IMP/Unity nodes for MRA services [CSCw69661]'. The detailed view for this issue includes:

- Description:** The tomcat(-ECDSA) certificate of the following CUCM / IMP / 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 CSCw69661), the Expressway-C traffic server will do a TLS certificate check on the CUCM / IMP / Unity tomcat(-ECDSA) certificates irrespective of the configuration of TLS Verify Mode set when discovering each of those servers.
- Action:**
 - Update the Expressway-C trust store with the CA certificates that signed the tomcat(-ECDSA) certificates of CUCM / IMP / Unity nodes.
 - 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-C with the following command:
`xConfiguration EdgeConfigServer VerifyOriginServer: Off`
- Snippet:**

```

2022-07-11T19:49:01.533+02:00 vcsd traffic_server[3936]: [ET_NET 2] WARNING: SNI (10.48.36.215) not in certificate. Action:Terminate server=10.48.36.215(10.48.36.215)
2022-07-11T19:49:01.533+02:00 vcsd traffic_server[3936]: [ET_NET 2] ERROR: SSL connection failed for "10.48.36.215": error:1416F080:SSL routines:tls_process_server_certificate:certificate verify failed

```

Firma diagnostica SNI

Soluzione

La soluzione a lungo termine consiste nell'assicurare che la verifica TLS funzioni correttamente. L'azione da eseguire dipende dal messaggio di avviso visualizzato.

Quando viene visualizzato il messaggio di AVVISO: verifica del certificato del server di base non

riuscita per (<server-FQDN-or-IP>). Action=Terminate Error=self-signed certificate server=cucm.steven.lab(10.48.36.215) depth=x message, quindi è necessario aggiornare l'archivio di attendibilità sui server Expressway-C di conseguenza. Con la catena di CA che ha firmato il certificato (profondità > 0) o con il certificato autofirmato (profondità = 0) da Manutenzione > Sicurezza > Certificato CA attendibile. Accertarsi di eseguire questa azione su ogni server del cluster. In alternativa, è possibile firmare il certificato remoto da una CA nota nell'archivio di attendibilità di Expressway-C.

 Nota: Expressway non consente di caricare due certificati diversi (autofirmati, ad esempio) nell'archivio di attendibilità di Expressway con lo stesso nome comune (CN) di cui all'ID bug Cisco [CSCwa12905](#). Per risolvere il problema, passare ai certificati firmati da CA o aggiornare CUCM alla versione 14, dove è possibile riutilizzare lo stesso certificato (autofirmato) per Tomcat e CallManager.

Quando si osserva il messaggio AVVISO: SNI (<server-FQDN-or-IP>) non presente nel certificato, il messaggio indica che l'FQDN o l'IP del server non è contenuto nel certificato presentato. È possibile adattare il certificato in modo da includere tali informazioni oppure modificare la configurazione (ad esempio in CUCM su Sistema > Server in base a quanto contenuto nel certificato del server) e quindi aggiornare la configurazione sul server Expressway-C in modo da tenerne conto.

Informazioni correlate

La soluzione a breve termine consiste nell'applicare la soluzione descritta nella documentazione per ripristinare il comportamento precedente prima di X14.2.0. È possibile eseguire questa operazione tramite la CLI sui nodi del server Expressway-C con il comando appena introdotto:

```
xConfiguration EdgeConfigServer VerifyOriginServer: Off
```

IT

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).