Solución de problemas de verificación de certificado de servidor de tráfico de Expressway para servicios MRA introducidos por CSCwc69661 / CSCwa25108

Contenido

Introducción Prerequisites Requirements Componentes Utilizados Antecedentes Cadena de CA de confianza Comprobación de SAN o CN Cambio de comportamiento Versiones inferiores a X14.2.0 Versiones de X14.2.0 y superiores Solucionar escenarios 1. La CA que firmó el certificado remoto no es de confianza 2. La dirección de conexión (FQDN o IP) no está incluida en el certificado Cómo validarlo fácilmente Solución

Introducción

Este documento describe el cambio de comportamiento en las versiones de Expressway de X14.2.0 y posteriores vinculadas al Id. de bug Cisco <u>CSCwc6961</u> o al Id. de bug Cisco <u>CSCwa25108</u>.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- configuración básica de Expressway
- configuración básica de MRA

Componentes Utilizados

La información de este documento se basa en Cisco Expressway en la versión X14.2 y posteriores.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Antecedentes

Con este cambio de comportamiento marcado por el Id. de bug Cisco <u>CSCwc69661</u> o ID de bug de Cisco <u>CSCwa25108</u>, el servidor de tráfico de la plataforma Expressway realiza la verificación de certificados de los nodos de Cisco Unified Communication Manager (CUCM), Cisco Unified Instant Messaging & Presence (IM&P) y del servidor Unity para los servicios Mobile and Remote Access (MRA). Este cambio puede provocar errores de inicio de sesión de MRA después de una actualización en su plataforma de Expressway.

El protocolo seguro de transferencia de hipertexto (HTTPS) es un protocolo de comunicación segura que utiliza la seguridad de la capa de transporte (TLS) para cifrar la comunicación. Crea este canal seguro mediante el uso de un certificado TLS que se intercambia en el intercambio de señales TLS. De esta manera, tiene dos propósitos: autenticación (para saber a quién se conecta la persona remota) y privacidad (cifrado). La autenticación protege frente a ataques de intrusos y la privacidad evita que los atacantes intercepten y manipulen la comunicación.

La verificación de TLS (certificado) se realiza a la vista de la autenticación y le permite asegurarse de que se ha conectado a la parte remota correcta. La verificación consta de dos elementos individuales:

- 1. Cadena de autoridad certificadora de confianza (CA)
- 2. Nombre alternativo del sujeto (SAN) o nombre común (CN)

Cadena de CA de confianza

Para que Expressway-C confíe en el certificado que envía CUCM / IM&P / Unity, debe poder establecer un vínculo desde ese certificado a una entidad de certificación (CA) de nivel superior (raíz) en la que confíe. Este vínculo, una jerarquía de certificados que vincula un certificado de entidades a un certificado de CA raíz, se denomina cadena de confianza. Para poder verificar dicha cadena de confianza, cada certificado contiene dos campos : Emisor (o 'Emitido por') y Asunto (o 'Emitido para').

Los certificados de servidor, como el que CUCM envía a Expressway-C, tienen en el campo "Asunto" su nombre de dominio completo (FQDN) en el 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 Ejemplo de certificado de servidor para CUCM.cucm.vngtp.lab. Tiene el FQDN en el atributo CN del campo Asunto junto con otros atributos como País (C), Estado (ST), Ubicación (L), ... También podemos ver que el certificado del servidor es entregado (emitido) por una CA llamada vngtp-ACTIVE-DIR-CA.

Las CA de nivel superior (CA raíz) también pueden emitir un certificado para identificarse. En dicho certificado de CA raíz, vemos que el emisor y el sujeto tienen el mismo valor :

Issuer: DC=lab, DC=vngtp, CN=vngtp-ACTIVE-DIR-CA Subject: DC=lab, DC=vngtp, CN=vngtp-ACTIVE-DIR-CA Es un certificado emitido por una CA raíz para identificarse.

En una situación típica, las CA raíz no emiten directamente certificados de servidor. En su lugar, emiten certificados para otras CA. Estas otras CA se denominan CA intermedias. A su vez, las CA intermedias pueden emitir directamente certificados de servidor o certificados para otras CA intermedias. Podemos tener una situación en la que un certificado de servidor es emitido por la CA 1 intermedia, que a su vez obtiene un certificado de la CA 2 intermedia y así sucesivamente. Hasta que finalmente la CA intermedia obtiene su certificado directamente de la CA raíz :

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

Ahora, para que Expressway-C confíe en el certificado de servidor que envía CUCM, debe poder generar la cadena de confianza desde ese certificado de servidor hasta un certificado de CA raíz. Para que esto ocurra, necesitamos cargar el certificado de CA raíz y también todos los certificados de CA intermedios (si los hay, lo que no es el caso si la CA raíz hubiera emitido directamente el certificado de servidor de CUCM) en el almacén de confianza de Expressway-C.

Nota: Aunque los campos Emisor y Asunto son fáciles de crear en la cadena de confianza de una manera legible por las personas, CUCM no utiliza estos campos en el certificado. En su lugar, utiliza los campos 'Identificador de clave de autoridad X509v3' e 'Identificador de clave de asunto X509v3' para crear la cadena de confianza. Esas claves contienen identificadores para los certificados que son más precisos que para utilizar los campos Asunto/Emisor : puede haber 2 certificados con los mismos campos Asunto/Emisor, pero uno de ellos ha caducado y otro sigue siendo válido. Ambos tendrían un identificador de clave de asunto X509v3 diferente, por lo que CUCM aún puede determinar la cadena de confianza correcta.

Este no es el caso de Expressway, aunque según el identificador de error de Cisco <u>CSCwa12905</u> y no es posible cargar dos certificados diferentes (autofirmados, por ejemplo) en el almacén de confianza de Expressway que tienen el mismo nombre común (CN). La manera de corregir esto es utilizar certificados firmados por CA o utilizar nombres comunes diferentes para ello o ver que utiliza siempre el mismo certificado (potencialmente a través de la función de certificado de reutilización en CUCM 14).

Comprobación de SAN o CN

En el paso 1 se desprotege el almacén de confianza; sin embargo, cualquier persona que tenga

un certificado firmado por una CA en el almacén de confianza será válida en ese momento. Esto claramente no es suficiente. Por lo tanto, hay una comprobación adicional que valida que el servidor al que se conecta específicamente es el correcto. Lo hace basándose en la dirección para la que se formuló la solicitud.

El mismo tipo de operación ocurre en su navegador, así que vamos a ver esto a través de un ejemplo. Si navega hasta <u>https://www.cisco.com</u> verá un icono de candado junto a la URL que ingresó y significa que se trata de una conexión confiable. Esto se basa tanto en la cadena de confianza de la CA (desde la primera sección) como en la comprobación de SAN o CN. Si abrimos el certificado (a través del navegador haciendo clic en el icono de candado), verá que el nombre común (que aparece en el campo 'Emitido para:') está configurado en <u>www.cisco.com</u> y que corresponde exactamente a la dirección a la que deseábamos conectarnos. De esta manera, podemos estar seguros de que nos conectamos al servidor correcto (porque confiamos en la CA que firmó el certificado y que realiza la verificación antes de entregar el certificado).

Certificate	×		
neral Details Certification Path		CISCO	Products and Servio
Certificate Information			
This certificate is intended for the following p • Ensures the identity of a remote computer • Proves your identity to a remote computer • 2.16.840.1.113839.0.6.3 • 2.23.140.1.2.2	purpose(s):		
* Refer to the certification authority's statement for	details.		
Issued to: www.cisco.com			
Issued by: HydrantID Server CA O1			
Valid from 2/16/2022 to 2/16/2023			

Cuando observamos los detalles del certificado y, en particular, las entradas de SAN, vemos que se repite lo mismo, así como algunos otros FQDN:

Certificate		>
General Details Certifica	ation Path	
Show: <all></all>	~	
Field Certificate Policies CRL Distribution P Subject Alternativ Subject Key Identi Enhanced Key Usage	Value [1]Certificate Policy: [1]CRL Distribution DNS Name=disco-i b18ceccd49a5dfd74 Server Authenticatio v1_adf2befa2cff10c	^
DNS Name=cisco-imag	olddb6ce30b00bd7	~
DNS Name=osco.com DNS Name=www-01.ci DNS Name=www-02.ci DNS Name=www-rtp.ci DNS Name=www.cisco DNS Name=www.cisco DNS Name=www.statici DNS Name=www.statici	sco.com sco.com sco.com .com afiles-cisco.com -cisco.com o.com	~
	Edit Properties	Copy to File
		ОК

Esto significa que cuando solicitáramos la conexión a <u>https://www1.cisco.com</u>, por ejemplo, también se mostraría como una conexión segura porque está contenida en las entradas de SAN.



Sin embargo, cuando no navegamos a <u>https://www.cisco.com</u> sino directamente a la dirección IP (<u>https://72.163.4.161</u>), no aparece una conexión segura porque sí confía en la CA que la firmó, pero el certificado que se nos presentó no contiene la dirección (72.163.4.161) que usamos para conectarnos hacia el servidor.



En el navegador, puede omitir esto, sin embargo, es una configuración que puede habilitar en las conexiones TLS que no permite una omisión. Por lo tanto, es importante que sus certificados contengan los nombres CN o SAN correctos que la parte remota planea utilizar para conectarse a ella.

Cambio de comportamiento

Los servicios MRA dependen en gran medida de varias conexiones HTTPS a través de Expressway hacia los servidores CUCM / IM&P / Unity para autenticarse correctamente y recopilar la información correcta específica para el cliente que inicia sesión. Esta comunicación suele ocurrir en los puertos 8443 y 6972.

Versiones inferiores a X14.2.0

En versiones inferiores a X14.2.0, el servidor de tráfico de Expressway-C que administra esas conexiones HTTPS seguras no verificó el certificado presentado por el extremo remoto. Esto podría llevar a ataques de intrusos. En la configuración de MRA, hay una opción para la verificación de certificados TLS mediante la configuración del 'Modo de verificación TLS' a 'Activado' cuando agregaría servidores CUCM / IM&P / Unity en **Configuración > Unified Communications > Servidores Unified CM / nodos IM and Presence Service / Servidores Unity Connection**. La opción de configuración y el cuadro de información relevante se muestran como ejemplo, lo que indica que sí verifica el FQDN o la IP en la SAN, así como la validez del certificado y si está firmado por una CA de confianza.

Cisco Expressway-C

Status >	System >	Configuration >	Applications >	Users >	Maintenance >	
Unified (CM servers				You are here: Configuration	<u>1</u> •
Unified C	M server lookup]			
Unified CM	publisher address		cucmpub.vngtp.lab			
Username			* administrator		(i)	
Password			*		(i)	
TLS verify	mode		On 🗸 (į			
Deploymer	it		Default deployme	nt 🗸 i		
AES GCM	support		0ff 🗸 (į́)			
SIP UPDAT	E for session refre	sh	Off v			
ICE Passth	rough support		Off v			
AES GCM SIP UPDAT ICE Passth	support E for session refre rough support	sh	Off v (i) Off v (i) Off v (i)			

Save Delete Cancel

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

Esta comprobación de verificación del certificado TLS solo se realiza cuando se detectan los servidores CUCM / IM&P / Unity y no en el momento en que se consultan los distintos servidores durante el inicio de sesión de MRA. Un primer inconveniente de esta configuración es que sólo la comprueba para la dirección del editor que agrega. No valida si el certificado de los nodos del suscriptor se ha configurado correctamente, ya que recupera la información del nodo del suscriptor (FQDN o IP) de la base de datos del nodo del editor. Un segundo inconveniente de esta configuración también es que lo que se anuncia a los clientes MRA como la información de conexión puede ser diferente de la dirección del editor que se ha colocado en la configuración de Expressway-C. Por ejemplo, en CUCM, en **System > Server** podría anunciar el servidor con una dirección IP (10.48.36.215, por ejemplo) y los clientes de MRA lo utilizan (a través de la conexión de Expressway con proxy); sin embargo, podría agregar CUCM en Expressway-C con el FQDN de cucm.steven.lab. Por lo tanto, suponga que el certificado de Tomcat de CUCM contiene cucm.steven.lab como entrada de SAN pero no la dirección IP; a continuación, la detección con

'TLS Verify Mode' establecido en 'On' se realiza correctamente, pero las comunicaciones reales de los clientes de MRA pueden dirigirse a un FQDN o IP diferente y, por lo tanto, no pasar la verificación de TLS.

Versiones de X14.2.0 y superiores

A partir de la versión X14.2.0, el servidor de Expressway realiza la <u>verificación de certificado TLS</u> para cada solicitud HTTPS que se realiza a través del servidor de tráfico. Esto significa que también realiza esto cuando el 'Modo de verificación de TLS' se establece en 'Desactivado' durante la detección de los nodos CUCM / IM&P / Unity. Cuando la verificación no tiene éxito, el intercambio de señales TLS no se completa y la solicitud falla, lo que puede llevar a la pérdida de funcionalidad como problemas de redundancia o conmutación por fallas o fallas de inicio de sesión completas, por ejemplo. Además, con 'TLS Verify Mode' establecido en 'On', no garantiza que todas las conexiones funcionen correctamente como se describe en el ejemplo posterior.

Los certificados exactos que verifica Expressway hacia los nodos CUCM / IM&P / Unity son como se muestra en la sección de la <u>guía MRA</u>.

Aparte de la verificación predeterminada de TLS, también hay un cambio introducido en X14.2 que podría anunciar un <u>orden de preferencia diferente para la lista de cifrado</u>, que depende de su trayectoria de actualización. Esto puede causar conexiones TLS inesperadas después de una actualización de software porque puede suceder que antes de la actualización solicitó el certificado de Cisco Tomcat o Cisco CallManager de CUCM (o cualquier otro producto que tenga un certificado independiente para el algoritmo ECDSA) pero que después de la actualización solicita la variante ECDSA (que es la variante de cifrado más seguro en realidad que RSA). Los certificados de Cisco Tomcat-ECDSA o Cisco CallManager-ECDSA podrían estar firmados por una CA diferente o simplemente ser certificados autofirmados (el valor predeterminado).

Este cambio en el orden de preferencia de cifrado no siempre es relevante para usted, ya que depende de la ruta de actualización, como se muestra en las <u>notas de la versión de</u> Expressway X14.2.1. En resumen, puede ver en **Mantenimiento > Seguridad > Cifras** para cada una de las listas cifradas si antepone "ECDHE-RSA-AES256-GCM-SHA384:" o no. Si no es así, prefiere el cifrado ECDSA más reciente sobre el cifrado RSA. Si es así, entonces tiene el comportamiento anterior con RSA que tiene la preferencia más alta.

Cipher Pref	erence	es - ECDSA Cipher Preference Over RSA
ECDSA cert	ificates	are preferred over RSA.
N	The fo	Illowing points lists the various upgrade path(s) that are mandatory for upgrading ciphers.
Important	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).

Hay dos maneras en que la verificación de TLS podría fallar en esta situación, que se tratan en detalle más adelante:

- 1. La CA que firmó el certificado remoto no es de confianza
- a. Certificado con firma automática
- b. Certificado firmado por CA desconocida
- 2. La dirección de conexión (FQDN o IP) no está incluida en el certificado

Solucionar escenarios

Los siguientes escenarios muestran un escenario similar en un entorno de laboratorio donde el inicio de sesión de MRA falló después de una actualización de Expressway de X14.0.7 a X14.2. Comparten similitudes en los registros, sin embargo la resolución es diferente. Los registros se recopilan mediante el registro de diagnóstico (de **Mantenimiento > Diagnóstico > Registro de diagnóstico**) que se inició antes del inicio de sesión de MRA y se detuvo después de que fallara el inicio de sesión de MRA. No se ha habilitado ningún registro de depuración adicional para él.

1. La CA que firmó el certificado remoto no es de confianza

El certificado remoto podría estar firmado por una CA que no está incluida en el almacén de confianza de Expressway-C o podría ser un certificado autofirmado (en esencia, una CA también) que no se agrega en el almacén de confianza del servidor de Expressway-C.

En el ejemplo aquí, puede observar que las solicitudes que van a CUCM (10.48.36.215 - cucm.steven.lab) se manejan correctamente en el puerto 8443 (respuesta 200 OK) pero arroja un error (respuesta 502) en el puerto 6972 para la conexión TFTP.

```
===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-viaaddr="" Msg="GET http://vcs_control.steven.lab:8443/c3RldmVuLmxhYi9odHRwcy9jdWNtLnN0ZXZlbi5sYWIvODQ0Mw/cucmuds/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/" 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="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=" 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-viaaddr="" 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" El error 'certificate verify failed' indica el hecho de que Expressway-C no pudo validar el intercambio de señales TLS. La razón de esto se muestra en la línea de advertencia, ya que indica un certificado autofirmado. Si la profundidad se muestra como 0, es un certificado autofirmado. Cuando la profundidad es mayor que 0, significa que tiene una cadena de certificados y, por lo tanto, está firmado por una CA desconocida (desde la perspectiva de Expressway-C).

Cuando observamos el archivo pcap que se recopiló en las marcas de tiempo mencionadas en los registros de texto, puede ver que CUCM presenta el certificado con CN como cucmms.steven.lab (y cucm.steven.lab como SAN) firmado por steven-DC-CA a Expressway-C en el puerto 8443.

(● 1 □ 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0					
	🗧 લ લ લ 👖				
CX #0443					
Time	Source	Src port Destination	Deal port Protocol	DSCP VLAN	Langth Drife
4691 2022-07-11 16:55:25.916680	10.48.36.46	35622 10.48.36.215	\$443 TCP	CSB	74 35622 + 8443 [SYN] Seq+0 Win+64240 Len+0 MSS+1460 SACK_PERM+1 TSval+878570435 TSecr+0 MS+128
4692 2022-07-11 16:55:25.916953	10.48.36.215	8443 10.48.36.46	35622 TCP	cse	74 8443 → 35622 [SYN, ACK] Seque Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=343633230 TSecr=878570435 WS=128
4693 2022-07-11 16:55:25.916973	10.48.36.46	35622 10.48.36.215	8443 TCP	C58	66 35622 + 8443 [ACK] Seq=1 ACK=1 H1n=64256 Len=0 TSval=878578435 TSecr=343633230
4694 2022-07-11 16:55:25.91/852	10.48.50.40	35622 10.48.56.215	8443 TLSV1.2	CS0	SSS CIERT HEID
4696 2022-07-11 16:55:25.938390	10.40.30.215	35622 10.40.36.215	5443 TCP	cse	4544 361761 FELAN 66 36622 + 8433 (ACK) SecuSIS ACK-1440 Win-64128 Lenut TSVal+878578457 TSecr=343633251
4697 2022-07-11 16:55:25,938409	10.48.36.215	8443 10.48.36.46	35622 TLSv1.2	CSB	1470 Certificate, Server Key Exchange, Server Hello Done
4698 2022-07-11 16:55:25.938419	10.48.36.46	35622 10.48.36.215	8443 TCP	CS0	66 35622 + 8443 [ACK] Seq=518 Ack=2853 Win=63488 Len=0 TSval=878570457 TSecr=343633251
4699 2022-07-11 16:55:25.940107	10.48.36.46	35622 10.48.36.215	8443 TLSv1.2	CS0	192 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
4700 2022-07-11 16:55:25.943034	10.48.36.215	8443 10.48.36.46	35622 TLSv1.2	CS0	300 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	cse	66 35622 + 8443 [ACK] Seqw644 Ackw3095 Winw64128 Lenw0 TSval=878570461 TSecr=343633256
4702 2022-07-11 16:55:25.943277	10.45.36.46	35622 10.48.36.215	8443 TL5V1.2	CSB	2543 Application Data
4783 2822-87-11 16:55:25.943476	18.48.36.215	8443 10.48.56.46	35622 TCP	CS8	66 8443 + 35622 [ACK] 568-3895 ACK+3121 H3n+35872 L6H-0 T3V81+345653256 T56CF+878578462
4/0/ 2022-0/-11 10:55:25.954/96	10.48.50.215	3445 10.48.50.40	55622 TCP	CSO	1514 8445 + 55622 [AKK] SEQ4005 AKK+S121 WIN-55972 LEM-1448 ISVAIA54565258 ISECT-8785/9642 [ICP Segment of a reasseno. 262 5622 + 4023 [ArX] Seq.3121 Ark-5421 Win-65972 Lem-1448 ISVAIA54565258 ISECT-8785/9642 [ICP Segment of a reasseno.
4709 2022-07-11 16:55:25.954861	10.48.36.215	8443 10.48.36.46	35622 TLSV1.2	cse	1957 Acolication Data
4718 2022-07-11 16:55:25,954873	18,48,36,46	35622 10.48.36.215	8443 TCP	CSB	66 35622 + 8443 [ACK] Seg=3121 ACK=5734 Win=63488 Len=0 TSval=878570473 TSecr=343633268
4711 2022-07-11 16:55:25.955712	10.48.36.46	35622 10.48.36.215	8443 TLSv1.2	CS0	97 Encrypted Alert
4712 2022-07-11 16:55:25.955750	10.48.36.46	35622 10.48.36.215	8443 TCP	CS0	66 35622 + 8443 [FIN, ACK] Seq=3152 Ack=5734 Win=64128 Len=0 TSval=878570474 TSecr=343633268
4714 2022-07-11 16:55:25.956123	10.48.36.215	8443 10.48.36.46	35622 TLSV1.2	CSB	97 Encrypted Alert
4715 2022-07-11 16:55:25.956170	10.48.36.46	35622 10.48.36.215	8443 TCP	CS8	54 35622 + 8443 [RST] Sequ3153 Win+0 Len+0
4716 2022-07-11 16:55:25.956232	10.48.36.215	8443 10.48.36.46	35622 TCP	CS0	66 8443 → 35622 [FIN, ACK] Seq=5765 ACK=8153 Hin=35872 Len=0 TSval=343633269 TSecF=878570474
<pre>version: v3 (2) serialNumber: 0x4500000122056051 > signature (shalkithRSAEncryption > issuer: rdnSequence (0) > validity > validity</pre>	1d348668442000200000122 1)			nitwame=TAC,10	d-at-organizationName-Cisco,id-at-localityName+Diegen,id-at-stateOrProvinceName=Belgium,id-at-countryNa
<pre>version: v3 (2) serialwuber: exsequences > signature (shalith#Sifecryption > issuer: indicequence (0) > validity > subjectrWollickeyInfo extension: 9 items > Extension (id-ce-exteguisage) > Extension (id-ce-exteguisage) > Extension (id-ce-subjectaltum Extension (id-ce-subjectaltum) extension (id-ce-subjectaltum) Ex</pre>	d34848842000200000122) nc) -cc-subjectAltName) ab			nitvæme-TAC,14	-at-organizationName-Cisco,id-at-localityName-Diegen,id-at-stateOrProvinceName-Belgium,id-at-countryNa
<pre>version: v3 (2) serialNumber: dev6000012056655 signaTure (shalidthSinecrythol issuer: rddsequence (0) validity validity validity validity validity validity validity validity validitation (di-ce-subjection) validity validitation (di-ce-subjection) va</pre>	<pre>sd34646442000200000122) mc) sc=-subjectAltName) ab sc=isiter) isonifier) isonifier isonifier) isonifier isonifier) isonifier isonifie</pre>			litvane + 7 AC, 14	s.at-organizationName-Cisco,id-at-localityName-Diegen,id-at-stateOrProvinceName-Belgium,id-at-countryNa

Pero cuando inspeccionamos el certificado presentado en el puerto 6972, puede ver que es un certificado autofirmado (el emisor es él mismo) con CN configurado como cucm-EC.steven.lab. La extensión -EC indica que se trata del certificado ECDSA configurado en CUCM.

eth0_diagnostic_logging_tcpdump00_vcsc_2022-07-11_16_55_44.pcap	
---	--

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools

Imp Imp <th>trp.por</th> <th>x=6972</th> <th></th> <th></th> <th></th> <th></th> <th></th>	trp.por	x=6972					
	No.	Time	Source	Srs port Destination	Cest port Protocol	DSCP VLAN	Lengh 3rb
min 000000000000000000000000000000000000	Г	4730 2022-07-11 16:55:26.006608	10.48.36.46	31576 10.48.36.215	6972 TCP	CS0	74 31576 → 6972 [SYN] Seque Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=878570525 TSecr=0 MS=128
4/3 2021-0-11 151513.400020 18.4.1.5.4.4 1151 14.5.1.4.4 1151 151.5.1.1.1.1 151.5.1.1.1 <t< td=""><td></td><td>4731 2022-07-11 16:55:26.006851</td><td>10.48.36.215</td><td>6972 10.48.36.46</td><td>31576 TCP</td><td>CS0</td><td>74 6972 + 31576 [SYN, ACK] Seq+0 Ack+1 Win+28960 Len+0 MSS=1460 SACK_PERM=1 TSval=343633320 TSecr+878570525 WS=128</td></t<>		4731 2022-07-11 16:55:26.006851	10.48.36.215	6972 10.48.36.46	31576 TCP	CS0	74 6972 + 31576 [SYN, ACK] Seq+0 Ack+1 Win+28960 Len+0 MSS=1460 SACK_PERM=1 TSval=343633320 TSecr+878570525 WS=128
4.13 8.14 115 8.1 6.1 6.1 6.1 6.1 6.1 8.1		4732 2022-07-11 16:55:26.006892	10.48.36.46	31576 10.48.36.215	6972 TCP	CSB	66 31576 → 6972 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=878570525 TSecr=343633320
01 02	4	4733 2022-07-11 16:55:26.007100	10.48.36.46	31576 10.48.36.215	6972 TLSv1.2	CS0	583 Client Hello
 	_	4734 2022-07-11 16:55:26.016350	10.48.36.215	6972 10.48.36.46	31576 TLSv1.2	CS0	1514 Server Hello, Certificate, Server Key Exchange
 		4735 2022-07-11 16:55:26.016391	10.45.36.46	51576 10.48.36.215	6972 TCP	csa	66 31576 → 0972 [ACK] 540=518 ACK=1449 H1n=64128 Lefn=0 T5V01=878570535 T54CT=343633329
072 202-07-11 113-11-14 1137 12-14-17 12-14-		4736 2022-07-11 16:55:26.016408	10.48.36.215	6972 10.45.36.46	31576 TLSV1.2	CSB	499 Certificate Request, Server Helio Done
v23 202-001 1007 202-001 100		4/3/ 2022-0/-11 10:55:20.010413	10.40.30.40	31576 10.48.36.215	6972 TEP	C50	<pre>de 315/6 * 07/2 [Ark] Schelle Arkelez Hines/** Cond 15V8188/05/0535 15CC*8480633327 73 Mines (Hark) Schell Derrichten (Hines/* CO)</pre>
0 0 242222021 1552344005 1944344 1552 1642342 0 2443224 15531400 1644342 15531 1557 1557 155 155 155 1557 1557 155		4730 2022-07-11 10:35:20.010703	10.40.36.46	31570 10.48.30.213	6972 TC971.2	C50	73 Alert (Level, Mta), Description, Unknown CA) 74 Sletz - 207 (Svil Cena Wincesta Hena Mtalige Cirk Debwit Taval-Brestatit Tavala UL178
0.42 0.42 0.44 0.44 0.45 0.07 0.44 0.45 0.07 0.44 <td< td=""><td></td><td>4740 2022-07-11 16:55:26.016965</td><td>10.48.36.46</td><td>31576 10.48.36.215</td><td>6972 TCP</td><td>CSA</td><td>66 31576 + 6972 [857] ACK] Seca525 ACK 1882 Mina6128 Lena TSUBLETSORS TECH-1484533270</td></td<>		4740 2022-07-11 16:55:26.016965	10.48.36.46	31576 10.48.36.215	6972 TCP	CSA	66 31576 + 6972 [857] ACK] Seca525 ACK 1882 Mina6128 Lena TSUBLETSORS TECH-1484533270
0.42 2022-07.11 1051315.40,000 10.4.4.5.4.4 1017 10.72 COL Seq.1 ACM 2022-07.11 Signa Col		4741 2022-07-11 16:55:26.016984	10.40.36.215	6972 10,48,36,46	31578 TCP	CS0	74 6972 + 31578 [SYN, ACK] Seg+0 Ack+1 Win+28960 Len+0 MSS=1460 SACK PERM+1 TSV01=343633330 TSecr+078570535 WS+128
0.90 2022 07:11 (55):53:00.001201 104:41:64 105:10 (42:11,64:11) 104:10:64 105:10 (42:11,64:11) 104:10:64 105:10 (42:11,64:11) 104:10:64 105:10 (42:11,64:11) 104:10:64 105:10 (42:11,64:11) 104:10:64 105:10 (42:11,64:11) 104:10:64 105:10 (42:11,64:11) 104:10:64 105:10 (42:11,64:11) 104:10:64 105:10 (42:11,64:11) 104:10:64 105:10 (42:11,64:11) 104:10:64 105:10 (42:11,64:11) 104:10:64 105:10:10:10:10:10:10:10:10:10:10:10:10:10:	1	4742 2022-07-11 16:55:26.017009	10.48.36.46	31578 10.48.36.215	6972 TCP	CSB	66 31578 + 6972 [ACK] Seg=1 ACK+1 Win=64256 Len=0 TSval=878570535 TSecr=343633330
0 044 4022 e711 (1557):4.07121 044.30.44 0157 (167, 164) 057 (157, 164) 046 2022 e711 (1557):4.0713 (1537) 044.30.44 0171 (157, 164) 057 (157, 164) 057 (157, 164) 046 2022 e711 (1557):4.0424 044.30.44 0171 (157, 164) 057 (157, 164) 057 (157, 164) 057 (157, 164) 047 2022 e711 (1557):4.0424 044.30.44 0171 (157, 164) 057 (157, 164) 057 (157, 164) 057 (157, 164) 047 2022 e711 (1557):4.0424 044.30.44 0171 (157, 164) 057 (156, 164) 057 (156, 164) 057 (156, 164) 057 (156, 164) 057 (156, 164) 057 (156, 164) 057 (156, 164) 057 (156, 164) 057 (156, 164) 057 (156, 164)		4743 2022-07-11 16:55:26.017101	10.48.36.215	6972 10.48.36.46	31576 TCP	CSB	66 6972 + 31576 [FIN, ACK] Seq=1882 Ack=525 Win=30080 Len=0 TSval=343633330 TSecr=878570535
446 202 - 07-11 10.44.0.81-44 1072 10.44.0.81-25 697 10.55 10.	L	4744 2022-07-11 16:55:26.017121	10.48.36.46	31576 10.48.36.215	6972 TCP	CSB	54 31576 + 6972 [RST] Seq=525 Win+0 Len+0
444 2022-07-11 515/12-60322 10.4.4.3,6.4 3173 10.7 10.6 3134 5172 10.4.4 10.7 10.7 10.6		4745 2022-07-11 16:55:26.017218	10.48.36.46	31578 10.48.36.215	6972 TL5v1.2	CS8	S83 Client Hello
97 702 0721 071 1015132.032426 10.4.1,5.4.4 1337 18.4.3.5.35 072 70 0 C6 66 1537 1 0.4.2.5.21 0573 10.4.2.5.4 1337 10.4.3.5.4 1337 10.4.3.5.35 1337 10.4.3.5.4 1337 10.4.3.5.4 1337 10.4.3.5.4 1337 10.4.3.5.4 1337 10.4.3.5.4 1337 10.4.3.5.5 1335 10.4.5.5 1357 10.4.5.5.5 <td></td> <td>4746 2022-07-11 16:55:26.024226</td> <td>10.48.36.215</td> <td>6972 10.48.36.46</td> <td>31578 TLSv1.2</td> <td>CS0</td> <td>1514 Server Hello, Certificate, Server Key Exchange</td>		4746 2022-07-11 16:55:26.024226	10.48.36.215	6972 10.48.36.46	31578 TLSv1.2	CS0	1514 Server Hello, Certificate, Server Key Exchange
474 2021-07-11 165515.6.24298 10.4.1.5.23 672 107.1.2 C58 98 0 Critificate Request, Server Hullo Done 478 2021-07-11 165515.6.24298 10.4.1.5.4.4 1071 10.4.1.5.25 672 TCP C68 64 1074 1072 10.4.1.5.25 1072 TCP C68 74 1025 1072 10.4.1.5.25 1072 TCP C68 74 1026 1072 10.5.25 1072 TCP C68 74 1026 1072 10.5.25 1072 TCP C68 74 1026 1072 TCP C68 74 1026 1072 TCP <td< td=""><td></td><td>4747 2022-07-11 16:55:26.024265</td><td>10.48.36.46</td><td>31578 10.48.36.215</td><td>6972 TCP</td><td>CS0</td><td>66 31578 → 6972 [ACK] Seq=518 Ack=1449 Win=64128 Len=0 TSVal=878570543 TSecr=343633337</td></td<>		4747 2022-07-11 16:55:26.024265	10.48.36.46	31578 10.48.36.215	6972 TCP	CS0	66 31578 → 6972 [ACK] Seq=518 Ack=1449 Win=64128 Len=0 TSVal=878570543 TSecr=343633337
479 322:07:11 15:55:26.02049 10-42.56.46 1557 10-42.56 672 170 C40 66 1557 - 672 [CC] StepEin Accharge inchesting		4748 2022-07-11 16:55:26.024298	10.48.36.215	6972 10.48.36.46	31578 TLSv1.2	CS0	500 Certificate Request, Server Hello Done
478 202-07.11 165526.004544 10.04.05.04 10.04.05.04 10.04.05.04 072 10.01.0 </td <td></td> <td>4749 2022-07-11 16:55:26.024309</td> <td>10.48.36.46</td> <td>31578 10.48.36.215</td> <td>6972 TCP</td> <td>CSB</td> <td>66 31578 + 6972 [ACK] Seq=518 Ack≈1883 Hin=63744 Len=0 TSval=878570543 TSecr=343633337</td>		4749 2022-07-11 16:55:26.024309	10.48.36.46	31578 10.48.36.215	6972 TCP	CSB	66 31578 + 6972 [ACK] Seq=518 Ack≈1883 Hin=63744 Len=0 TSval=878570543 TSecr=343633337
APX1 202-07-11 1535520.024649 10.548.28.46 11592 18.48.28.215 697 100 Cd8 66 1159 4.49.28 (Cd5, 202) 260.552 x00.1418 kine4.128 (Ln6e No.1428 (Ln6e		4750 2022-07-11 16:55:26.024548	10.48.36.46	31578 10.48.36.215	6972 TLSv1.2	CS0	73 Alert (Level: Fatal, Description: Unknown CA)
<pre></pre>		4751 2022-07-11 16:55:26.024647	10.48.36.46	31578 10.48.36.215	6972 TCP	CS8	66 31578 → 6972 [RST, ACK] Seq=525 ACk=1883 Win=64128 Len=0 TSval=878570543 TSecr=343633337
<pre>> v Secure Sockets Layer > TLSVL3. Record Layer: Handshake Protocol: Server wello > TLSVL3. Record Layer: Handshake Protocol: Server wello > TLSVL3. Record Layer: Handshake Protocol: Server wello Content Type: Handshake Protocol: Certificate Content Type: Mandshake Protocol: Certificate Handshake Type: Certificate Handshake Type: Certificate (1) Log (See Bytes) Certificate: Length: 640 v Certificate: Length: 640 v Certificate: Length: 657 v Certificate: Institutes Sort v Certificate: Length: 640 v Sugederrificate v Signederrificate v Signederrificate</pre>	(4/6/ 2022-0/-11 10:55:20.005155	10.40.00.40	31300 10.40.30.213	0772 107	0.00	14 3196 4 63/2 [318] Sche Himerice Come Hostine Sectional Interesteder Decide Hostin
<pre>v Secure Socket Layer > TSV:1.2 Record Layer: Handbake Protocol: Server wello Content Type: Handbake Protocol: Certificate Content Type: Handbake Protocol: Certificate (2) version: TLS 1.2 (BeNBBD) Length: 640 v Handbake Protocol: Certificate (11) Length: 640 v Certificate Length: 640 v Certificate</pre>							
<pre>> Itsv1.2 model Light: Handbake Protocol: Server Hallo V Expl.2 model Light: Handbake Protocol: Server Hallo Content Type: Handbake Protocol: Server Hallo V Handbake Protocol: Certificate Handbake Protocol: Certificate (III) Length: 64 Certificate: Light: 64 Certificate: Light: 657 V Certificate: INNOVABUREREDUIDEDIDEDIDEDIDEDIDEDIDEDIDEDIDEDIDEDI</pre>	✓ secur	e Sockets Layer					
<pre>v USV1.2 Metor Usys: Handbake Protocol: Certificate Content Type: Handbake (2) version: TLS 1.2 (000303) Length: 647 v Handbake Protocol: Certificate (2) version: VG (2) version: VG version: VG version: VG version: VG (2) version: VG</pre>	> TL	V1.2 Record Layer: Handshake Protocol: Server	Hello				
<pre>unter: jpt::msubake(z2) iveget: 12 (embody) iveget: 12 (embod</pre>	* TL	V1.2 Record Layer: Handshake Protocol: Certif	lcate				
<pre>isegit: 407 v isegit: 407 certificate (iii) isegit: 40 v certificate (int) certificate certific</pre>		Content Type: Handshake (22)					
<pre>v isststate Frotcol: Certificate andstate Frotcol: Set and certificate andstate and certificate and and certificate and and and and and and and and and and</pre>		version: (LS 1.2 (000303)					
<pre>wadsake type: (etificate (11) i.eqth: 64 certificate (10) certificate</pre>	~	Handshake Brotocol: Certificate					
<pre>Length: 647 Certificates Length: 649 Certificates Length: 647 Certificates Length: 657 Cert</pre>		Handshake Tune: Certificate (11)					
<pre>certificate (seq1): seq v certificate (seq1): seq certificate (seq1): seq v certificate (seq1): seq v signedcertificate v signedcertificate v</pre>		Length: 443					
<pre>v (cmtificate: (of byte)) Certificate: mathematics for v (cmtificate: mathematics for v (cmtificate: mathematics for math</pre>		Certificates Length: 660					
<pre>certificate inegit: 657 v certificate v signedcertificate v signedcertificate</pre>		 Certificates (660 bytes) 					
<pre>v certificate: 3020203000201202107470ee62271e301346 (id-at-localityName-Diegen,id-at-stateOrProvinceName-Belgium,id-at-commonName-cure.EC.steven.lab. v signedCertificate: 302020300202107470ee62271e3013412093ef8a3bf1d > signetry (costa-stin-SHADB) > signetry (costa-stin-SHADB) > signetry (costa-stin-SHADB) > subject: rindSequence (a) > subject: rindSequence (b) > subject: rindSequence (a) > subject: rindSequence (b) > subject: rindSequence</pre>		Certificate Length: 657					
<pre>v signedcrificate version: v3(2) serialwumber: m2/370estplails4109946f8albfid) signature (edsa-vith-Sw184) v issuer: redsequence (0) v alistiv > valistiv > valistiv > subject: ridsequence (d) > subject:</pre>		 Certificate: 3082028d30820214a00302010203 	2107470ee62271e3d1346 (id	-at-localityName=Diegem,id-at-sta	teOrProvinceName=Belgi	um,id-at-commo	nName <mark>=cucm-EC.steven.lab</mark> ,id-at-organizationalUnitName=TAC,id-at-organizationName=Cisco,id-at-countryNa
<pre>version v9 (2) sertailumeter: exit v9 (2) sertailumeter: exit v9 (2) signature (ecds-with=value) ignature (ecds-with=value)</pre>		✓ signedCertificate					
<pre>serialwater: ex/24%est27zhid144199446f83bfid > igsupr (edsa.vith=9x1344</pre>		version: v3 (2)					
<pre>> signature (ecds=with=SWAB4) > issuer: consequence (a) > voltequence: 6 items (id=at=localityname=olegem,id=at=stateorProvincemame=selgium,id=at=commonwame=cucm=EC_site=en_law,id=at=organizationalunitwame=TAC,id=at=organizationname=cisco,id=at=countryname=BE) > voltety > subject: rdnSequence (a) > subjectsholiccerystof > subjectsholiccerystof > stemsion (id=c=basis) > Extension (id=c=basis) > Extension (id=c=basis)centrylame) > Extension (id=c=basis)centrylame) > Extension (id=c=basis)centrylame) > Extension (id=c=basis)centrylame) > Extension (id=c=basis)centrylame) > Extension (id=c=basis)centrylame) > Extension (id=c=basis)centrylame)</pre>		serialNumber: 0x7470ee62271e3d134610	09946fea3bf1d				
<pre>v issuer: redsequence (0)</pre>		> signature (ecdsa-with-SHA384)					
<pre>> rdisequence: 6 items (id-at-localityname=biggen,id-at-stateorProvincemame=Belgium,id-at-commonwame=Cuen=EC_steven,lab,id-at-organizationalunituame=tc,id-at-organizationname=Cisco,id-at-countryname=EE) > subject: rdnSequence (0) > subjectPublicscyInfo > extension: 5 items > Extension (id-ce-basicscontraints) > Extension (id-ce-subjectrayInter) > Extension (id-ce-subjectrayInter) > Extension (id-ce-subjectrayInter)</pre>		issuer: rdnSequence (0)					
<pre>> valisity > subject:rindsquence (0) > subject:vindsquence (0) > subject:vindsquence (0) > subjectivalized (0) > subjectivalize</pre>		> rdnSequence: 6 items (id-at-local	ityName=Diegen,id-at-stateOr	<pre>rProvinceName=Belgium,id-at-common</pre>	Wame+cucm-EC.steven.la	b,id-at-organi	zationalUnitName=TAC,id-at-organizationName=Cisco,id-at-countryName=BE)
<pre>> subject: rdnSequence (0) > subject:vDilCxcginfo v extension: 5 items > Extension (id-ce-texpusage) > Extension (id-ce-texpusage) > Extension (id-ce-texpusage) > Extension (id-ce-texpusage) > Extension (id-ce-texpusage)</pre>		> validity					
> subjectivalickeginfo v extensions : 5 items > Extension (Li-ce-trypusage) > Extension (Li-ce-trypusage) > Extension (Li-ce-trypusage) > Extension (Li-ce-trypusage) > Extension (Li-ce-trypusage)		> subject: rdnSequence (0)					
<pre>v extension: 5 items > Extension (id-ce-kgusage) > Extension (id-ce-kgusage)</pre>		> subjectPublickeyInfo					
> Extension (Li-c-e-extra/subject/sepident/sint) > Extension (Li-c-e-extra/subject/sepident/sint) > Extension (Li-c-e-basicconstraints)		<pre>v extensions: s items</pre>					
<pre>> Extension (u=<==xxxyosger) > Extension (u=<==xybetxyosger) > Extension (u=<=xybetxyosger) > Extension (u=<=xybetx</pre>		<pre>> Extension (id co outfould and)</pre>					
 Extension (List-c-basicconstraints) Extension (List-c-basicconstraints) 		Extension (id-ce-extRe)usage)	(inc)				
A RANDOWN AND AND AND AND AND AND AND AND AND AN		 Extension (id-ce-subject/eyioentia) Extension (id-ce-subject/eyioentia) 	Taer)				
Y PRIPOSED [10:CF:SUDJECTATINEE]		v Extension (id-ce-subjectaltuane)	7				
Extension Id: 2.5.20.17 (id.cs.subjectalthame)		Extension Id: 2.5.29.17 (id.ce.	subjectAltName)				
v GeneralManes: 1 iten		✓ GeneralNames: 1 item					
v Generalisme: discusse (2)		<pre>✓ GeneralName: dVSName (2)</pre>					
dhShame: cucm.steven.lab		dNSName: cucm.steven.lab					
> algorithmIdentifier (ecdsa-with-Sw4304)		> algorithmIdentifier (ecdsa-with-SHA384	•)				
Padding: 0		Padding: 0					
encrypted: 3054023012543bd5ebe74570b1171eb489ff03b060cd80d8		encrypted: 3064023012543bd5ebe74570b11	71eb489ff03b060cd80d8				
> TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange	> TL	iv1.2 Record Layer: Handshake Protocol: Server	Key Exchange				

En CUCM, en Administración de Cisco Unified OS, puede consultar los certificados en vigor en Seguridad > Administración de certificados, como se muestra, por ejemplo, aquí. Muestra un certificado diferente para tomcat y tomcat-ECDSA donde tomcat está firmado por CA (y es de confianza para Expressway-C) mientras que el certificado tomcat-ECDSA está firmado por sí mismo y no es de confianza para Expressway-C aquí.

CISCO For Cisco	Unified Communications Solution						admin
Show - Settings - 5	Security · Software Upgrades · Se	rvices + Help	•				
Certificate List							
限 Generate Self-signe	d 🐴 Upload Certificate/Certificate ch	uain 👔 Down	load CTL 🧯	Generate CSR 🔋 Download CSF	1		
Cabus.							
(1) 43 records found							
Certificate List	(1 - 43 of 43)						Rows p
Find Certificate List wh	ere Certificate v begins with	h v		Find Clear Filter 💠 🚥			
Certificate *	Common Name	Type	Key Type	Distribution	Issued By	Expiration	Description
authg	AUTHZ_cuom.steven.lab	Self-signed	RSA	cucm.steven.lab	AUTHZ_cuom.steven.lab	07/21/2038	Self-signed certificate generated by system
CaliManager	cucm.steven.lab	CA-signed	RSA	cucm.steven.lab	steven-DC-CA	07/13/2022	Certificate Signed by steven-DC-CA
CallManager-ECDSA	cucm-EC.steven.lab	Self-signed	EC	cucm.steven.lab	cucm-EC.steven.lab	02/18/2024	Self-signed certificate generated by system
CallManager-trust	steven-DC-CA	Self-signed	RSA	steven-DC-CA	steven-DC-CA	06/01/2025	Signed Certificate
CallManager-trust	NOMAT-AD-CA	Self-signed	RSA	NOMAT-AD-CA	NOMAT-AD-CA	04/23/2028	Signed Certificate
CaliManager-trust	CAP-RTP-002	Self-signed	RSA	CAP-RTP-002	CAP-RTP-002	10/10/2023	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCH using the MIC when associated with a secure profile.
CalManager-trust	CAPF-eb2c64d8	Self-signed	RSA	CAPF-eb2c64d8	CAPF-eb2c64d8	04/12/2020	
CallManager-trust	ms-AD2-CA-1	Self-signed	RSA	ms-AD2-CA-1	ms-AD2-CA-1	09/11/2024	Vingto CA
CalManager-trust	CAP-RTP-001	Self-signed	RSA	CAP-RTP-001	CAP-RTP-001	02/07/2023	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with 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
CalManager-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 SUDE CA	CA-signed	RSA	ACT2 SUDI CA	Cisco Root CA 2048	05/14/2029	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCH using the MIC when associated with a secure profile.
CalManager-trust	vnoto-ACTIVE-DIR-CA	Self-signed	RSA	vnotp-ACTIVE-DIR-CA	vngtp-ACTEVE-DIR-CA	02/10/2024	VNGTP-CA
CaliManager-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.
CaliManager-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.
CaliManager-trust	dccomics-WONDERWOMAN-CA	Self-signed	RSA	dccomics-WONDERWOMAN-CA	dccomics-WONDERWOMAN-CA	09/19/2037	CA-byantum
CalManager-trust	CAPF-616421bc	Self-signed	RSA	CAPF-616421bc	CAPF-616421bc	07/12/2025	
CAPF	CAPF-616421bc	Self-signed	RSA	cucm.steven.lab	CAPF-616421bc	07/12/2025	Self-signed certificate generated by system
CAPF-trust	CAP-RTP-002	Self-signed	RSA	CAP-RTP-002	CAP-RTP-002	10/10/2023	This certificate was used to sign the MIC installed on Cisco endpoint. Presence of this certificate allows the end point to communicate securely with UCN using the MIC when associated with a secure profile.
CAPF-trust	CAPF-eb2c64d8	Self-signed	RSA	CAPF-eb2c64d8	CAPF-eb2c64d8	04/12/2020	
CAPF-trust	CAP-RTP-001	Self-signed	RSA	CAP-RTP-001	CAP-RTP-001	02/07/2023	This certificate was used to sion the MIC installed on Cisco endooint. Presence of this certificate allows the end point to communicate securely with UCN using the MIC when associated with a secure profile.
CAPF-trust	Cisco Root CA M2	Self-signed	RSA	Cisco Root CA M2	Cisco Root CA M2	11/12/2037	This certificate was used to sion the MIC installed on Cisco endooint. Presence of this certificate allows the end point to communicate securely with UCH using the MIC when associated with a secure profile.
CAPF-trust	ACT2 SUDE CA	CA-signed	RSA	ACT2 SUDI CA	Cisco Root CA 2048	05/14/2029	This certificate was used to sign the MIC installed on Cisco endooint. Presence of this certificate allows the end point to communicate securely with UCH using the MIC when associated with a secure profile.
CAPF-trust	Cisco Root CA 2048	Self-signed	RSA	Cisco Root CA 2048	Cisco Root CA 2048	05/14/2029	This certificate was used to sion 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.
CAPF-trust	Cisco Manufacturing CA	CA-signed	RSA	Cisco Manufacturing CA	Cisco Root CA 2048	05/14/2029	This certificate was used to sion the MIC installed on Cisco endooint. Presence of this certificate allows the end point to communicate securely with UCH using the MIC when associated with a secure profile.
CAPF-trust	Cisco Manufacturino CA SHA2	CA-signed	RSA	Cisco Manufacturino CA SHA2	Cisco Root CA M2	11/12/2037	This certificate was used to sion the MIC installed on Cisco endooint. Presence of this certificate allows the end point to communicate securely with UCH using the MIC when associated with a secure profile.
CAPF-trust	CAPE-616421bc	Self-signed	RSA	CAPE-616421bc	CAPE-616421bc	07/12/2025	
insec	curren streven lab	Self-signed	RSA	ourm steven lab	cucm steven lab	07/12/2025	Salf-sionad cartificate generated by system
insec-trust	cucm steven Jab	Self-signed	RSA	ourm steven Jab	cucm steven lab	07/12/2025	Trust Certificate
ITLRecovery	ITLRECOVERY cucm steven lab	Self-signed	RSA	ourm steven lab	ITLRECOVERY cucm steven lab	02/14/2039	Self-signed certificate generated by system
tomcat	curren sheven Jah	C&-signed	RSA	ouron steven Jab	steven-DC-CA	07/10/2024	Certificate Stoned by these DC-CA
tomcat-ECDSA	cucm-EC steven lab	CSR Only	EC.	cucm.steven.lab			
tomcat-ECDSA	cucm-EC steven Jab	Self-signed	FC .	ouron steven Jab	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	Sioned Certificate
tomcat-trust	cucm-EC steven lab	Self-signed	EC	cucm.steven.lab	cucm-EC.steven.lab	07/25/2023	Trut Certificate
tomcat-brust	cucm steven lab	CA-signed	RSA	curm steven lab	steven-DC-CA	07/10/2024	Trut Certificate
tomcat-brust	cupt-EC steven Jab	Salf-sinned	EC.	curs steven lab	cups-EC steven lab	07/25/2023	Trut Certificate
homeal-brust	NOMAT-CA-10	Salf-sinned	RSA	NOMAT-CA-10	NOMAT-CA-10	08/11/2027	Group Cartificate
home all design	vente ACTIVE DIR.CA	Calf.sinned	PSA	vento-ACTIVE-DIR-CA	vente-ACTIVE-DIR-CA	02/10/2024	Trut Cartificata
home all design	dreamine and a second	Calif.sinned	PCA	decomics. WONDERWOMAN.CA	dromine WOADERWOALAN.CA	09/19/2027	CA Brunn
101111011111010	AND A REAL PROPERTY AND A	and any set	1100	And the second second second second	and a second sec	477 #37 69437	ALL IN STATE

2. La dirección de conexión (FQDN o IP) no está incluida en el certificado

Aparte del almacén de confianza, el servidor de tráfico de MRA también verifica la dirección de conexión hacia la que realiza la solicitud el cliente MRA. Por ejemplo, cuando ha configurado en

CUCM en **System > Server** su CUCM con la dirección IP (10.48.36.215), Expressway-C anuncia esto como tal al cliente y las solicitudes posteriores del cliente (procesadas a través de Expressway-C) se dirigen hacia esta dirección.

Cuando esa dirección de conexión en particular no está contenida en el certificado del servidor, la verificación de TLS también falla y se arroja un error 502 que resulta en una falla de inicio de sesión de MRA, por ejemplo.

```
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/c3RldmVuLmxhYi9odHRwcy8xMC400C4zNi4yMTUvODQ0Mw/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

Donde c3RldmVuLmxhYi9odHRwcy8xMC40OC4zNi4yMTUvODQ0Mw se traduce (base64 - <u>https://www.base64decode.org/</u>) a steven.lab/https/10.48.36.215/8443, que muestra que debe hacer la conexión hacia 10.48.36.215 como la dirección de conexión en lugar de a cucm.steven.lab. Como se muestra en las capturas de paquetes, el certificado Tomcat de CUCM no contiene la dirección IP en la SAN y, por lo tanto, se produce el error.

Cómo validarlo fácilmente

Puede validar si se encuentra con este cambio de comportamiento fácilmente con los siguientes pasos:

1. Inicie el registro de diagnóstico en los servidores de Expressway-E y C (idealmente con TCPDumps habilitado) desde **Mantenimiento > Diagnóstico > Registro de diagnóstico** (en caso de un clúster, es suficiente iniciarlo desde el nodo principal)

2. Intente iniciar sesión en MRA o pruebe la funcionalidad dañada después de la actualización

3. Espere hasta que falle y luego detenga el registro de diagnóstico en los servidores de Expressway-E y C (en caso de un clúster, asegúrese de recopilar los registros de cada nodo individual del clúster)

4. Cargue y analice los registros en la herramienta Collaboration Solution Analyzer

5. Si se encuentra con el problema, recoge las líneas de error y advertencia más recientes relacionadas con este cambio para cada uno de los servidores afectados

Â	Collaboration Solutions Analyzer	Preview © UTC	r ? 0
Tools	Diagnostic overview	sues found No issue Not applicable Missing Information Potential problem	
Toda 5 Log Analyzer T Cuptorel Mark Doignostics	s Q. Search Coll Coll (53) Configuration (29) Defects only ●	a bit all wind in any deplote in deploted in any deploted in an	Contguration Contguration Contguration Contguration MBA discovering

firma de diagnóstico de CA

Â	Collaboration Solutions Analy cisco Log Analyzer	Zeer Preview © UTC	# ? 0
Tools >	Diagnostic overview	Issues found No issue Not applicable Missing information Potential problem	
Log Analyzer T Digwards Digwards Anayon	Q: Search Pecult Category P: Category MRA (51) Or Configuration (39) Defects only	Table into in the presence of the configuration into the large intermediate intermediate into the large intermediate into the large intermediate interm	Configuration Configuration Configuration Configuration MBIA en discovering

Firma de diagnóstico SNI

Solución

La <u>solución a largo plazo</u> es asegurarse de que la verificación de TLS funcione correctamente. La acción que se debe realizar depende del mensaje de advertencia que se muestre.

Cuando observe la ADVERTENCIA: Error al comprobar el certificado de servidor principal para (<FQDN-o-IP-servidor>). Action=Terminate Error=self signed certificate

server=cucm.steven.lab(10.48.36.215) **depth=x** message, entonces debe actualizar el almacén de confianza en los servidores de Expressway-C en consecuencia. Con la cadena de CA que firmó este certificado (profundidad > 0) o con el certificado autofirmado (profundidad = 0) de **Mantenimiento > Seguridad > Certificado de CA de confianza**. Asegúrese de realizar esta acción en todos los servidores del clúster. Otra opción sería firmar el certificado remoto por una CA conocida en el almacén de confianza de Expressway-C.

Nota: Expressway no permite cargar dos certificados diferentes (autofirmados, por ejemplo) en el almacén de confianza de Expressway que tienen el mismo nombre común (CN) que según el ld. de error de Cisco <u>CSCwa12905</u>. Para corregir esto, mueva a certificados firmados por CA o actualice CUCM a la versión 14 donde puede volver a utilizar el mismo certificado (autofirmado) para Tomcat y CallManager.

Cuando observe la *ADVERTENCIA: SNI (<server-FQDN-or-IP>) no está en el* mensaje del *certificado*, entonces indica que este FQDN o IP del servidor no está contenido dentro del certificado que se presentó. Puede adaptar el certificado para incluir esa información o puede modificar la configuración (como en CUCM en Sistema > Servidor para que se incluya en el certificado del servidor) y luego actualizar la configuración en el servidor de Expressway-C para que se tenga en cuenta.

La <u>solución a corto plazo</u> consiste en aplicar la solución alternativa según lo documentado para recurrir al comportamiento anterior anterior a X14.2.0. Puede realizar esto a través de la CLI en los nodos del servidor de Expressway-C con el comando recién introducido:

xConfiguration EdgeConfigServer VerifyOriginServer: Off

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).