Ejemplo de configuración unificado de TLS del SORBO del elemento de la frontera

Contenido

Introducción

Cisco Unified Border Element (CUBE) soporta Session Initiation Protocol (SIP) para llamadas SIP con Transport Layer Security (TLS). TLS proporciona privacidad e integridad de datos de los mensajes de señalización SIP entre dos aplicaciones que se comunican. TLS se ubica sobre un Reliable Transport Protocol, como puede ser TCP.

TLS en el CUBO se puede configurar sobre una base de la por-pierna para permitir TLS a la llamada del SORBO de NON-TLS. Semejantemente, el CUBO utiliza el IPSec para asegurar la señalización y las llamadas al servicio técnico de H.323 PARA SORBER con la pierna de H.323, mientras que la pierna del SORBO utiliza TLS.

prerrequisitos

Requisitos

Asegúrese de cumplir estos requisitos antes de intentar esta configuración:

- Conocimiento básico de cómo configurar y utilizar la voz del Cisco IOS (tal como dial-peers)
- Conocimiento básico de cómo configurar y utilizar el CUBO
- Familiaridad con los conceptos de la seguridad básica tales como cifrado, certificación, autoridades de certificación, PKI (claves), y autenticación

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- CUBIQUE la versión en un ISR que utilice el Cisco IOS Release 12.4T
- Router del Cisco IOS configurado como Certificate Authority (CA)

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Convenciones

Consulte Convenciones de Consejos Técnicos de Cisco para obtener más información sobre las convenciones sobre documentos.

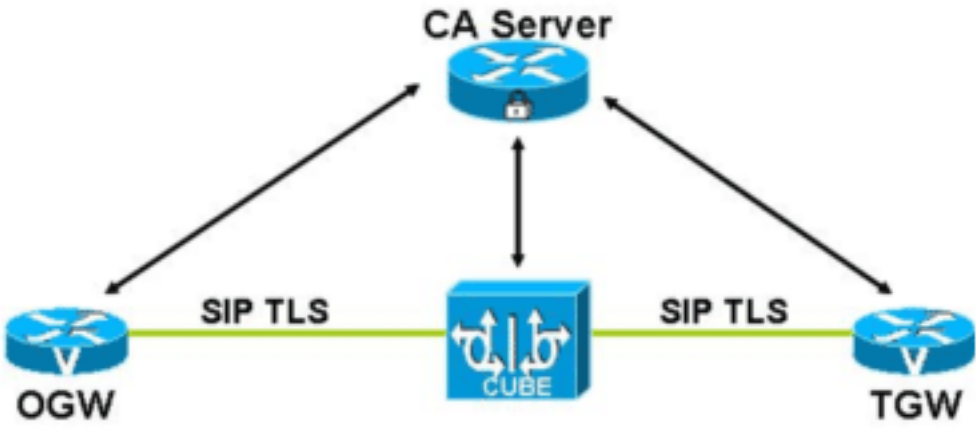
Configurar

En esta sección encontrará la información para configurar las funciones descritas en este documento.

Nota: Utilice la herramienta Command Lookup Tool (clientes registrados solamente) para obtener más información sobre los comandos utilizados en esta sección.

Diagrama de la red

Esta figura muestra un ejemplo del CUBO con las conexiones TLS del SORBO.



- El gateway de origen (OGW), el gateway de terminación (TGW), y los dispositivos del CUBO autentican y alistan con un servidor de CA. Los Certificados son firmados por el servidor de CA.
- Cuando se hace una llamada, una entrada en contacto TLS se inicia entre los dispositivos (por ejemplo, el OGW y el CUBO) y la infraestructura IOS PKI se utiliza para intercambiar los Certificados firmados por CA confiado en campo común durante el apretón de manos.
- Durante la entrada en contacto TLS, una clave simétrica dinámicamente generada y los

algoritmos de la cifra se negocian entre los dispositivos.
- Después de que la entrada en contacto TLS sea acertada, los dispositivos establecen una sesión del SORBO entre ellos. Las claves intercambiadas durante el proceso de la entrada en contacto TLS se utilizan para cifrar o para desencriptar todos los mensajes de señalización del SORBO. Sorbos del esquema de URI “: ” se utiliza para los mensajes de TLS del SORBO.

Soporte RFC para TLS en el CUBO

Las habitaciones de la cifra requeridas para TLS según el RFC 3261 del SORBO incluyen:

- TLS_RSA_WITH_AES_128_CBC_SHA (obligatorio)
- TLS_RSA_WITH_3DES_EDE_CBC_SHA (opcional) — Requerido para los servidores de red (tales como proxys y servidores de redirección para la compatibilidad descendente)

Solamente la habitación TLS_RSA_WITH_AES_128_CBC_SHA es aplicable CUBICAR y se soporta. Semejantemente, la implementación de TLS en el CUBO soporta solamente las habitaciones obligatorias de la cifra de RFC2246.

El protocolo del SORBO utiliza un modelo entre iguales. Por lo tanto, el CUBO puede ser el servidor o cliente de una conexión TLS y implementa a los ambos lados. El CUBO realiza siempre la autenticación recíproca cuando es el lado del servidor.

Pasos de configuración

Configure el servidor de CA

Usted puede utilizar este comando en el modo de configuración global para configurar a un router del Cisco IOS cargado con una imagen de criptografía:

```
router(config)#crypto pki server <ca-server-name> router(cs-server)#no shutdown
```

Notas:

- Utilice el **comando ip http server** en el modo de configuración global para asegurarse de que un servidor HTTP se ejecuta en el router configurado como el servidor de CA. Esto se requiere puesto que el uso HTTP del trustpoints del cliente (CUBE/OGW/TGW) para recibir los Certificados del servidor de CA.
- Los relojes en el servidor y el trustpoints del cliente (CUBE/OGW/TGW) de CA deben ser sincronizados. Si no, pudo haber problemas con la validez de los Certificados publicados por el servidor de CA. Usted puede utilizar el **reloj** y los **comandos clock set de la demostración** para sincronizar los relojes en el Routers del Cisco IOS. Alternativamente, usted puede desplegar a un servidor NTP para sincronizar los relojes.

Configuración básica para el CUBO

Utilice estos comandos para habilitar el CUBO IP-to-IP funcionalidad de la gateway. Esto permite el terminatation de la llamada VoIP y del reoriginatiion entrantes de la llamada con un VoIP dial-peer de salida.

```
voice service voip
  allow-connections h323 to sip
  allow-connections sip to h323
  allow-connections sip to sip
  allow-connections h323 to h323
```

Configuración de TLS

Complete estos pasos para configurar TLS en el CUBO (y los otros dispositivos como el OGW y el TGW):

1. **Genere un par de claves RSA**Utilice este comando en el modo de configuración global para generar un par de claves RSA:`router(config)#`**`crypto key generate rsa general-keys label <`***`label`***`> modulus 1024`**
2. **Cree un trustpoint PKI (el CUBO)**Utilice este comando en el modo de configuración global para crear un trustpoint PKI (CUBO):`router(config)#`**`crypto pki trustpoint <`***`ca-server-name`***`>` `router(ca-trustpoint)#`**`enrollment url <`***`http://ca-server-ip`***`>` `router(ca-trustpoint)#`**`rsakeypair <`***`rsa keypair label`***`>`**
3. **Autentique un trustpoint PKI (CUBO) con el servidor de CA**Utilice este comando en el modo de configuración global para autenticar un trustpoint PKI (CUBO) con el servidor de CA:`router(config)#`**`crypto pki authenticate <`***`ca-server-name`***`>** Este paso acciona el servidor de CA para enviar su certificado al trustpoint (CUBO), que deben ser validados.
4. **Aliste un trustpoint PKI (CUBO) con el servidor de CA**Utilice este comando en el modo de configuración global:`router(config)#`**`crypto pki enroll <`***`ca-server-name`***`>** Para este paso, usted debe ingresar una contraseña de impugnación. Los problemas de servidor de CA dos Certificados al trustpoint (CUBO): uno para certificar el servidor de CA y el otro para certificar el trustpoint (CUBO). Usted puede marcar los Certificados con el **comando show run**.
5. **Configuración TLS como transporte de la sesión**El transporte de la sesión se puede configurar a TLS con los **tls tcp del transporte de la sesión** ordena en cualquiera el nivel global bajo el “voip del servicio de voz” o en los voip dial peer apropiados.Si el transporte de la sesión se configura para un voip dial peer (entrante o saliente o ambos), después el transporte de TLS se utiliza solamente para la pierna configurada. El transporte de TLS se soporta sobre una base de la pierna-a-pierna.
6. **Configure un trustpoint predeterminado para el SORBO UA**Utilice este comando en el modo “sorbo-UA” para configurar un trustpoint predeterminado para el SORBO UA:`router(config-sip-ua)#`**`[`***`no`***`] crypto signaling [(`***`remote-addr subnet mask`***`) | default] trustpoint <`***`label`***`> [`***`strict-cipher`***`]`** La escritura de la etiqueta del trustpoint refiere al certificado del CUBO que se genera con los comandos del Cisco IOS PKI como parte de los proces de la inscripción. *la estricto-cifra* significa que el proceso de TLS del SORBO utiliza solamente esas habitaciones de la cifra que sean asignadas por mandato por el SORBO RFC.Actualmente, el RFC 3261 especifica el TLS_RSA_WITH_AES_128_CBC_SHA y TLS_RSA_WITH_3DES_EDE_CBC_SHA las habitaciones. Cuando usted utiliza el argumento del comando de la estricto-*cifra* evita los cambios a la configuración si el SORBO asigna más nuevas cifras por mandato.La capa SSL en el Cisco IOS no soporta TLS_RSA_WITH_3DES_EDE_CBC_SHA. Por lo tanto, el CUBO utiliza activamente solamente la habitación TLS_RSA_WITH_AES_128_CBC_SHA en el modo estricto. Cuando la estricto-*cifra* no se especifica, el proceso de TLS del SORBO utiliza un conjunto más grande de las cifras dependiendo del soporte en la capa SSL.*Ejemplo 1*El comando debajo del CUBO de las configuraciones de utilizar su **mylabel de la** escritura de la etiqueta del trustpoint cuando establece o valida una conexión TLS con un dispositivo remoto dentro de la subred de 1.2.3.0. La habitación de la cifra en este caso es el conjunto total que es soportado por la capa SSL en el CUBO.
   `crypto signaling remote-addr 1.2.3.0 255.255.255.0 trustpoint mylabel` *‘Ejemplo 2’*El comando debajo del CUBO de las configuraciones de utilizar a su **cocinero de la** escritura de la etiqueta del trustpoint cuando establece o valida una conexión TLS con cualquier

dispositivo remoto a menos que se corresponda con una configuración individual de la escritura de la etiqueta de la subred.

crypto signaling default trustpoint chef *Ejemplo 3*El comando debajo del CUBO de las configuraciones de utilizar su **mylabel de la** escritura de la etiqueta del trustpoint cuando establece o valida una conexión TLS con un dispositivo remoto dentro de la subred de 1.2.3.0. La habitación de la cifra usada durante la entrada en contacto TLS se limita a la habitación TLS_RSA_WITH_AES_128_CBC_SHA.

```
crypto signaling remote-addr 1.2.3.0 255.255.255.0 trustpoint mylabel strict-cipher
```

7. **Habilitar el puerto del módulo de escucha de TLS**Publique este comando en el modo "sorbo-UA" para permitir al puerto de TLS en TCP 5061 para escuchar:

```
transport tcp tls
```

8. **Configurar el esquema de los SORBOS URL**Los "sorbos: El" esquema URL se puede configurar bajo nivel del VoIP dial-peer o en el nivel global. Este comando se utiliza para configurar los "sorbos: " en un voip dial peer:

voice-class sip url sips Para configurar los "sorbos: El" esquema URL bajo nivel global, utiliza este comando en "modo del sorbo el" del voip" del servicio de voz ":

voice service voip sip url sips El uso de los SORBOS URL requiere todos los saltos en el recorrido de la señal utilizar TLS y los SORBOS. Esto llega a ser importante para el SRTP mientras que las claves están en el SDP y para una conexión segura que la información no se debe enviar en el texto claro. Si un proxy recibe una INVITACIÓN con los SORBOS (por ejemplo, INVITE AL SORBO /2.0 sips:123@proxy) el proxy debe utilizar los SORBOS para el salto siguiente. Cuando TLS se utiliza con un URL llano del SORBO, no hay garantía que todos los saltos utilizarán TLS, potencialmente comprometiendo la seguridad de extremo a extremo de la llamada.Si "sorbe" se configura el URL, el transporte será automáticamente TLS.

Notas de la implementación de TLS

- La operación actual del CUBO requiere el uso de TLS como el transporte cuando se configura el media seguro (SRTP). Una mejora futura puede levantar este requisito.
- Cuando el SRTP se configura para asegurar la conexión de medios, TLS o el IPSec *se debe* también configurar para asegurar los mensajes de señalización del SORBO. Las claves usadas para el cifrado SRTP se intercambian vía los mensajes de señalización – la sujeción de los resultados del canal de señalización en las claves SRTP intercambiadas en el texto claro y éste niega la Seguridad del SRTP para la conexión de medios.
- La operación actual del CUBO requiere el uso de los "sorbos: " Esquema de URI para una llamada de TLS. Una mejora futura puede levantar este requisito.
- La operación actual del CUBO se ha verificado con un solo servidor de CA solamente.

Configuraciones de Ejemplo

CUBO

ipipgw

```
ipipgw#show run Building configuration... Current
configuration : 5096 bytes ! version 12.4 service
timestamps debug datetime msec service timestamps log
datetime msec no service password-encryption ! hostname
ipipgw ! boot-start-marker boot system flash c3845-
```

adventerprisek9_ivs-mz.124-3.9.PI3a boot-end-marker ! logging buffered 10000000 debugging no logging console ! no aaa new-model ! resource policy ! ip subnet-zero ip cef ! no ip domain lookup ! voice-card 0 no dspfarm ! voice service voip **allow-connections sip to sip sip url sips ! crypto pki trustpoint ca-server enrollment url http://9.13.46.14:80 serial-number revocation-check crl rsakeypair kkp ! crypto pki certificate chain ca-server certificate 04 3082020D 30820176 A0030201 02020104 300D0609 2A864886 F70D0101 04050030 14311230 10060355 04031309 63612D73 65727665 72301E17 0D303530 39323231 37333435 315A170D 30363039 32323137 33343531 5A303431 32300F06 03550405 13084337 33323231 3333301F 06092A86 4886F70D 01090216 1270696E 612D3338 34352D69 70697067 77312E30 819F300D 06092A86 4886F70D 01010105 0003818D 00308189 02818100 BBCC2977 637E8E42 17EB7C26 FB2BA0A3 6E1ECECB E01A64F8 8F18200F 9837E4FA 7D908B3C 1297A4DE A403D315 C7BB96C6 50D95291 0433FA7B CB8FFFFD 8FC1C211 CCC7BCA9 140FF942 C3ACF4BC 3EDCE2DC 28FCEA87 AA83629F D217F833 A727940A 0BBB8624 3EA9D1EC 1F69228F E1DFC113 243246B7 BF57696C 2278F5C3 674EE0E1 02030100 01A34F30 4D300B06 03551D0F 04040302 05A0301F 0603551D 23041830 16801486 7414D5D6 9B8299C1 787211AB 1B265B06 D2B62D30 1D060355 1D0E0416 0414FED1 97051946 D2F870D8 0DE819C3 AA1F3830 AD35300D 06092A86 4886F70D 01010405 00038181 00845AB8 F6589AED 17D0BB10 2AEA48AA 9299C130 4B358EA1 96632C84 0387D2DE 4774C776 6A14F25B 5D062E12 45EF730D 27D45795 62C17F55 A0428259 B13669BC 022201C7 EB6B7ACF 4C7143FA 8A038301 CEA17A0B D0662887 26BA8F0E C44410BB 4F982706 11F0D248 77D8A0E5 4417F0F4 3F993CE3 F62F6BDE BA2DD6BB B843391D 6D quit certificate ca 01 30820201 3082016A A0030201 02020101 300D0609 2A864886 F70D0101 04050030 14311230 10060355 04031309 63612D73 65727665 72301E17 0D303530 39323031 37303335 375A170D 30383039 31393137 30333537 5A301431 12301006 03550403 13096361 2D736572 76657230 819F300D 06092A86 4886F70D 01010105 0003818D 00308189 02818100 BE7F0760 70D3B5C3 923D59FB C10AED17 71C6F477 7580851A 282FFAEB 43B918A1 2D867C1B 63963B36 F779FE18 D5DFFDB6 5E436276 459FC5EA A729C386 CDDD922B 2A0439AE 68A5F4C4 3B05F168 5BB93EF2 DF737F11 0BA3F5EB 3E62F423 CB5364D3 C39CCA09 8ADECBFF 4C0515A6 0750A283 ABA39ED2 F5866B98 D3361C1A B88AA62B 02030100 01A36330 61300F06 03551D13 0101FF04 05300301 01FF300E 0603551D 0F0101FF 04040302 0186301F 0603551D 23041830 16801486 7414D5D6 9B8299C1 787211AB 1B265B06 D2B62D30 1D060355 1D0E0416 04148674 14D5D69B 8299C178 7211AB1B 265B06D2 B62D300D 06092A86 4886F70D 01010405 00038181 00AC7DAF 0DF589CA C6175EC0 8F976C5F E08C3C91 85282FFA 94EE6F30 02EEE5B9 E60198ED 643151E0 CCE192FA A352BA3D 8BC5C006 EF89CFCF 59DA9B12 D729102C 3D6ADC3C 09931B96 3F1FB48C C0A85FDB 4F9A7C16 028673C3 91786D57 9D7C1016 62F9D4E9 78FED276 0C404815 B1FE3A11 4D215FCF 573536B4 477ECDB7 7060E221 31 quit** ! interface GigabitEthernet0/0 ip address 9.13.46.12 255.255.255.0 duplex auto speed auto media-type rj45 negotiation auto ! interface GigabitEthernet0/1 no ip address shutdown duplex auto speed auto media-type rj45 negotiation auto ! ip classless ip route 0.0.0.0 0.0.0.0 9.13.46.1 ! ip http server no ip http secure-server ! no cdp log mismatch duplex ! control-plane ! call treatment on ! dial-peer voice 1 voip session protocol sipv2 incoming called-number 9000 codec g711ulaw ! dial-peer voice 2 voip destination-pattern 9000 session protocol sipv2 session target ipv4:9.13.46.200 codec g711ulaw ! dial-peer voice

```
3 voip session protocol sipv2 incoming called-number
4000 codec g711ulaw ! dial-peer voice 4 voip
destination-pattern 4000 session protocol sipv2 session
target ipv4:9.13.32.75 codec g711ulaw ! dial-peer voice
5 voip destination-pattern 5000 session protocol sipv2
session target ipv4:9.13.0.10 codec g711alaw ! dial-peer
voice 7 voip destination-pattern 9999 session protocol
sipv2 session target ipv4:9.13.2.36 codec g711alaw !
dial-peer voice 12 pots destination-pattern 8400 ! dial-
peer voice 10 voip destination-pattern 50000 session
protocol sipv2 session target ipv4:9.13.2.150 codec
g711alaw ! dial-peer voice 11 voip session protocol
sipv2 session transport tcp tls incoming called-number
8004 codec g711ulaw ! dial-peer voice 13 voip
destination-pattern 8004 session protocol sipv2 session
target ipv4:9.13.2.70 codec g711ulaw ! dial-peer voice
20 voip destination-pattern 4444 session target
ipv4:9.13.46.111 codec g711ulaw ! dial-peer voice 21
voip incoming called-number 4444 codec g711ulaw ! sip-ua
retry invite 10 crypto signaling default trustpoint ca-
server ! gatekeeper shutdown ! line con 0 stopbits 1
line aux 0 stopbits 1 line vty 0 4 login ! scheduler
allocate 20000 1000 ! end
```

CA-servidor IOS

Ca-servidor

```
ca-server#show run Building configuration... Current
configuration : 2688 bytes ! ! Last configuration change
at 17:11:41 UTC Tue Sep 20 2005 ! NVRAM config last
updated at 16:57:43 UTC Tue Sep 20 2005 ! version 12.4
service timestamps debug datetime msec service
timestamps log datetime msec no service password-
encryption ! hostname ca-server ! boot-start-marker boot
system flash c2800nm-adventerprisek9_ivs-mz.124-3.9.PI3a
boot-end-marker ! no aaa new-model ! resource policy !
ip subnet-zero ! ip cef ! voice-card 0 no dspfarm !
crypto pki server ca-server grant auto ! crypto pki
trustpoint ca-server revocation-check crl rsakeypair ca-
server ! crypto pki certificate chain ca-server
certificate ca 01 30820201 3082016A A0030201 02020101
300D0609 2A864886 F70D0101 04050030 14311230 10060355
04031309 63612D73 65727665 72301E17 0D303530 39323031
37303335 375A170D 30383039 31393137 30333537 5A301431
12301006 03550403 13096361 2D736572 76657230 819F300D
06092A86 4886F70D 01010105 0003818D 00308189 02818100
BE7F0760 70D3B5C3 923D59FB C10AED17 71C6F477 7580851A
282FFAEB 43B918A1 2D867C1B 63963B36 F779FE18 D5DFFDB6
5E436276 459FC5EA A729C386 CDDD922B 2A0439AE 68A5F4C4
3B05F168 5BB93EF2 DF737F11 0BA3F5EB 3E62F423 CB5364D3
C39CCA09 8ADECBFF 4C0515A6 0750A283 ABA39ED2 F5866B98
D3361C1A B88AA62B 02030100 01A36330 61300F06 03551D13
0101FF04 05300301 01FF300E 0603551D 0F0101FF 04040302
0186301F 0603551D 23041830 16801486 7414D5D6 9B8299C1
787211AB 1B265B06 D2B62D30 1D060355 1D0E0416 04148674
14D5D69B 8299C178 7211AB1B 265B06D2 B62D300D 06092A86
4886F70D 01010405 00038181 00AC7DAF 0DF589CA C6175EC0
8F976C5F E08C3C91 85282FFA 94EE6F30 02EEE5B9 E60198ED
643151E0 CCE192FA A352BA3D 8BC5C006 EF89CFCF 59DA9B12
D729102C 3D6ADC3C 09931B96 3F1FB48C C0A85FDB 4F9A7C16
028673C3 91786D57 9D7C1016 62F9D4E9 78FED276 0C404815
B1FE3A11 4D215FCF 573536B4 477ECDB7 7060E221 31 quit !
```

```
interface FastEthernet0/0 ip address 9.13.46.14
255.255.255.0 duplex auto speed auto ! interface
FastEthernet0/1 no ip address shutdown duplex auto speed
auto ! ip classless ip route 0.0.0.0 0.0.0.0 9.13.46.1 !
ip http server no ip http secure-server ! no cdp log
mismatch duplex ! control-plane ! gatekeeper shutdown !
line con 0 line aux 0 line vty 0 4 login ! scheduler
allocate 20000 1000 ! end
```

Verificación

Después de que se haga una llamada, este **comando show** puede ser utilizado para verificar si el transporte usado para la llamada es TLS:

```
router#show sip-ua connections tcp tls ? brief Show summary of connections detail Show detail
connection information
```

La salida de muestra para este comando se muestra en estos ejemplos:

Ejemplo 1: Salida del detalle

```
==========================================================================
router#show sip-ua connections tcp tls detail Total active connections : 1 No. of send failures
: 0 No. of remote closures : 3 No. of conn. failures : 0 No. of inactive conn. ageouts : 0 Max.
tls send msg queue size of 0, recorded for 0.0.0.0:0 TLS client handshake failures : 0 TLS
server handshake failures : 0 ---------Printing Detailed Connection Report--------- Note: **
Tuples with no matching socket entry - Do 'clear sip <tcp[tls]/udp> conn t ipv4:<addr>:<port>'
to overcome this error condition ++ Tuples with mismatched address/port entry - Do 'clear sip
<tcp[tls]/udp> conn t ipv4:<addr>:<port> id <connid>' to overcome this error condition Remote-
Agent:9.13.46.12, Connections-Count:1 Remote-Port Conn-Id Conn-State WriteQ-Size ===========
======= =========== =========== 5061 1 Established 0
==========================================================================
```

Ejemplo 2: Salida abreviada

```
==========================================================================
router#show sip-ua connections tcp tls brief Total active connections : 2 No. of send failures :
0 No. of remote closures : 0 No. of conn. failures : 0 No. of inactive conn. ageouts : 0 Max.
tls send msg queue size of 0, recorded for 0.0.0.0:0 TLS client handshake failures : 0 TLS
server handshake failures : 0
==========================================================================
```

Alternativamente, el **comando debug ccsip messages** puede ser utilizado para verificar “vía: la” encabezado para TLS es incluida. Esta salida es una muestra INVITA a la petición de una llamada que utilice el SORBO TLS y los “sorbos: ” Esquema de URI:

```
INVITE sips:777@172.18.203.181 SIP/2.0
Via: SIP/2.0/TLS 172.18.201.173:5060;branch=z9hG4bK2C419
From: <sips:333@172.18.201.173>;tag=581BB98-1663
To: <sips:5555555@172.18.197.154>
Date: Wed, 28 Dec 2005 18:31:38 GMT
Call-ID: EB5B1948-770611DA-804F9736-BFA4AC35@172.18.201.173
Remote-Party-ID: "Bob" <sips:+14085559999@1.2.3.4>
Contact: <sips:123@host>
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO
Max-Forwards: 70
Cseq: 104 INVITE
Expires: 60
Timestamp: 730947404
Content-Length: 298
Content-Type: application/sdp
```

```
v=0
o=CiscoSystemsSIP-GW-UserAgent 8437 1929 IN IP4 172.18.201.173
s=SIP Call
c=IN IP4 1.1.1.1
t=0 0
m=audio 18378 RTP/AVP 0 19
c=IN IP4 1.1.1.1
a=rtpmap:0 PCMU/8000
a=rtpmap:19 CN/8000
a=ptime:20
```

Troubleshooting

Algunos consejos de Troubleshooting para las llamadas de TLS incluyen:

- Para permitir que el servidor de CA publique los Certificados al trustpoints, aseegurese que el router IOS se configura que pues un servidor de CA tiene HTTP habilitado (comando ip http server).
- El reloj en el servidor de CA y el trustpoints deben ser sincronizados.
- Si la entrada en contacto TLS falla entre dos dispositivos (por ejemplo, OGW y CUBO), marque la validez de los Certificados en los dispositivos. El comando del **pki del debug crypto** se puede utilizar para resolver problemas los problemas durante la entrada en contacto TLS.
- A veces cuando los dispositivos (por ejemplo, OGW y CUBO) están en diversas subredes, puede una aplicación la negociación del tamaño de la ventana TCP que causa estos errores: *La entrada-salida envía el error* y el *error de lectura entrada-salida*. Este problema se puede resolver con el **comando ip tcp path-mtu-discovery** en ambos dispositivos. Este problema pudo suceder después de una entrada en contacto TLS acertada.
- “Las conexiones claras sorbo-UA” ordenan en el modo sorbo-UA se pueden utilizar para borrar las conexiones TLS.`Router#`**`clear sip-ua tcp [tls] connections <id <`***`conn id`***`> | target <`***`ipv4:ip address:port`***`>`** La opción de los **tls** aparece después del **tcp** puesto que TLS monta encima del TCP. Este comando trabaja como los comandos clear existentes para el TCP y el UDP.

Información Relacionada

- Soporte de tecnología de voz
- Soporte de Productos de Voice and Unified Communications
- Troubleshooting de Cisco IP Telephony
- Soporte Técnico y Documentación - Cisco Systems