Exemple de configuration de SIP TLS dans Unified Border Element

Contenu

Introduction

Le Logiciel Cisco Unified Border Element (CUBE) prend en charge le Protocole SIP (Session Initiation Protocol) POUR SIROTER des appels avec le Transport Layer Security (TLS). Le TLS fournit l'intimité et l'intégrité des données des messages de signalisation de SIP entre deux applications qui communiquent. Le TLS est posé sur un protocole de transport fiable tel que le TCP.

Le TLS sur le CUBE peut être configuré sur une base de par-tronçon afin de permettre le TLS à l'appel de SIP de non-TLS. De même, le CUBE emploie IPSec afin de sécuriser des appels de signalisation et de support de H.323 POUR SIROTER avec H.323 le tronçon, alors que le tronçon de SIP utilise le TLS.

Conditions préalables

Conditions requises

Assurez-vous que vous répondez à ces exigences avant d'essayer cette configuration :

- La connaissance de base de la façon configurer et utiliser le Cisco IOS expriment (comme des cadran-pairs)
- Connaissance de base de la façon configurer et utiliser le CUBE

- Connaissance des concepts de Sécurité de base tels que le cryptage, la certification, les autorités de certification, le PKI (clés), et l'authentification

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- CUBEZ la version sur un ISR qui utilise la Cisco IOS version 12.4T
- Routeur Cisco IOS configuré comme Autorité de certification (CA)

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à Conventions relatives aux conseils techniques Cisco.

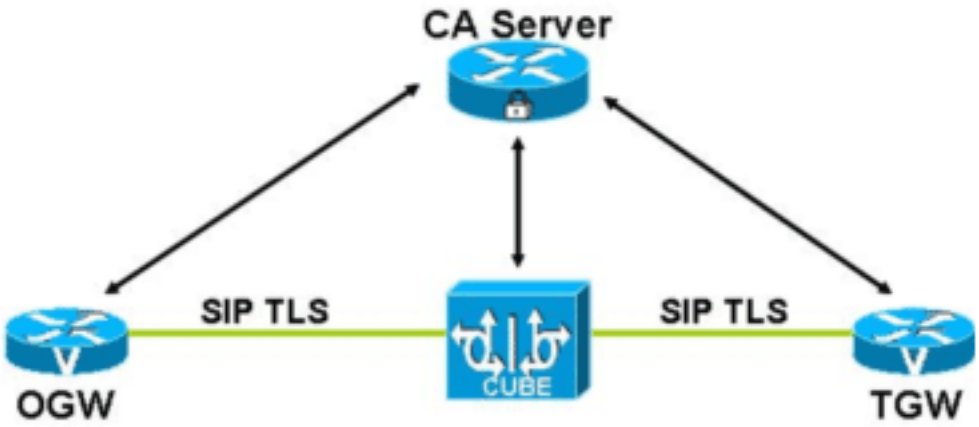
Configurez

Cette section vous fournit des informations pour configurer les fonctionnalités décrites dans ce document.

Remarque: Utilisez l'outil Command Lookup Tool (clients enregistrés seulement) pour obtenir plus d'informations sur les commandes utilisées dans cette section.

Diagramme du réseau

Cette figure affiche un exemple de CUBE avec des connexions de TLS de SIP.



- La passerelle d'origine (OGW), la dernière passerelle (TGW), et les périphériques de CUBE authentifient et s'inscrivent avec un serveur CA. Les Certificats sont signés par le serveur CA.
- Quand un appel est fait, une prise de contact de TLS est initiée entre les périphériques (par exemple, OGW et CUBE) et l'infrastructure de PKI IOS est utilisée pour permuter des

Certificats signés par un CA de confiance par terrain communal pendant la prise de contact.
- Pendant la prise de contact de TLS, des algorithmes symétriques dynamiquement générés principaux et de chiffrement sont négociés entre les périphériques.
- Après que la prise de contact de TLS soit réussie, les périphériques établissent une session de SIP entre eux. Des clés permutées pendant le processus de prise de contact de TLS sont utilisées pour chiffrer ou déchiffrer tous les messages de signalisation de SIP. Sip du schéma d'URI « : » est utilisé pour des messages de TLS de SIP.

Soutien RFC de TLS en CUBE

Les suites de chiffrement exigées pour le TLS selon RFC 3261 de SIP incluent :

- TLS_RSA_WITH_AES_128_CBC_SHA (obligatoire)
- TLS_RSA_WITH_3DES_EDE_CBC_SHA (facultatif) — Requis pour des serveurs de réseau (tels que des proxys et réorientez les serveurs pour la compatibilité ascendante)

Seulement la suite TLS_RSA_WITH_AES_128_CBC_SHA s'applique POUR CUBER et est prise en charge. De même, l'implémentation de TLS en CUBE prend en charge seulement les suites obligatoires de chiffrement de RFC 2246.

Le protocole de SIP utilise un modèle peer-to-peer. Par conséquent, le CUBE peut être le serveur ou client d'une connexion de TLS et implémente les deux côtés. Le CUBE exécute toujours l'authentification mutuelle quand c'est le côté serveur.

Étapes de configuration

Configurez le serveur CA

Vous pouvez employer cette commande en mode de configuration globale afin de configurer un routeur Cisco IOS chargé avec une image chiffrée :

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

Remarques :

- Employez la commande d'**ip http server** en mode de configuration globale afin de s'assurer qu'un serveur HTTP fonctionne sur le routeur configuré en tant que serveur CA. Ceci est exigé puisque les points de confiance de client (CUBE/OGW/TGW) emploient le HTTP afin de recevoir les Certificats du serveur CA.
- Les horloges dans le serveur CA et les points de confiance de client (CUBE/OGW/TGW) doivent être synchronisées. Autrement, il pourrait y avoir des questions avec la validité des Certificats délivrés par le serveur CA. Vous pouvez employer les commandes de **show clock** et de **clock set** afin de synchroniser les horloges sur des routeurs Cisco IOS. Alternativement, vous pouvez déployer un serveur de NTP afin de synchroniser les horloges.

Configuration de base pour le CUBE

Employez ces commandes afin d'activer la fonctionnalité de passerelle IP-à-IP du cube. Ceci permet le terminatation de l'appel VoIP et du reoriginatiion entrants de l'appel avec un homologue de numérotation VoIP sortant.

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

```
allow-connections sip to h323
allow-connections sip to sip
allow-connections h323 to h323
```

Configuration de TLS

Terminez-vous ces étapes afin de configurer le TLS sur le CUBE (et d'autres périphériques comme OGW et TGW) :

1. **Générez une RSA Keypair**Employez cette commande en mode de configuration globale afin de générer un keypair RSA :`router(config)#`**`crypto key generate rsa general-keys label <`**`label`**`> modulus 1024`**
2. **Créez un point de confiance de PKI (le CUBE)**Employez cette commande en mode de configuration globale afin de créer un point de confiance de PKI (CUBE) :`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. **Authentifiez un point de confiance de PKI (CUBE) avec le serveur CA**Employez cette commande en mode de configuration globale afin d'authentifier un point de confiance de PKI (CUBE) avec le serveur CA :`router(config)#`**`crypto pki authenticate <`**`ca-server-name`**`>`** Cette étape déclenche le serveur CA pour envoyer son certificat au point de confiance (CUBE), qui devrait être reçu.
4. **Inscrivez-vous un point de confiance de PKI (CUBE) avec le serveur CA**Utilisez cette commande en mode de configuration globale :`router(config)#`**`crypto pki enroll <`**`ca-server-name`**`>`** Pour cette étape, vous devez entrer un mot de passe de défi. Le serveur CA fournit deux Certificats au point de confiance (CUBE) : un pour certifier le serveur CA et l'autre pour certifier le point de confiance (CUBE). Vous pouvez vérifier les Certificats avec la commande de **passage d'exposition**.
5. **Configurez le TLS comme session transport**La session transport peut être configurée au TLS avec le **tls de TCP de session transport** commandent au niveau global sous le « voip de service vocal » ou dans les pairs de cadran appropriés de VOIP.Si la session transport est configurée pour un pair de cadran de VOIP (entrant ou sortant ou chacun des deux), alors le transport de TLS est utilisé seulement pour le tronçon configuré. Le TLS transporte est pris en charge sur une base de tronçon-à-tronçon.
6. **Configurez un point de confiance par défaut pour le SIP uA**Employez cette commande en mode de « sip-ua » afin de configurer un point de confiance par défaut pour le SIP uA :`router(config-sip-ua)#`**`[`**`no`**`] crypto signaling [(`**`remote-addr subnet mask`**`) | default] trustpoint <`**`label`**`> [`**`strict-cipher`**`]`** L'étiquette de point de confiance se rapporte au certificat du cube qui est généré avec les commandes de PKI de Cisco IOS en tant qu'élément des proces d'inscription. *le strict-chiffrement* signifie que le processus de TLS de SIP utilise seulement ces suites de chiffrement qui sont exigées par le RFC de SIP.Actuellement, RFC 3261 spécifie les suites TLS_RSA_WITH_AES_128_CBC_SHA et TLS_RSA_WITH_3DES_EDE_CBC_SHA. Quand vous utilisez l'argument de commande de strict-*chiffrement* évite des modifications à la configuration si le SIP exige de plus nouveaux chiffrements.La couche SSL dans le Cisco IOS ne prend en charge pas TLS_RSA_WITH_3DES_EDE_CBC_SHA. Par conséquent, le CUBE utilise activement seulement la suite TLS_RSA_WITH_AES_128_CBC_SHA en mode strict. Quand le strict-*chiffrement* n'est pas spécifié, le processus de TLS de SIP utilise un plus grand ensemble de chiffrements selon le support à la couche SSL.*Exemple 1*La commande ci-dessous configure le CUBE pour utiliser son **mylabel d'**étiquette de point de confiance quand elle établit ou reçoit une connexion de TLS avec un périphérique distant dans le sous-réseau de 1.2.3.0. La

suite de chiffrement est dans ce cas le positionnement global qui est pris en charge par la couche SSL sur le CUBE.

`crypto signaling remote-addr 1.2.3.0 255.255.255.0 trustpoint mylabel` *Exemple 2*La commande ci-dessous configure le CUBE pour utiliser son **chef d'**étiquette de point de confiance quand elle établit ou reçoit une connexion de TLS avec n'importe quel périphérique distant à moins qu'une configuration d'étiquette de sous-réseau individuel soit appariée.

`crypto signaling default trustpoint chef` *Exemple 3*La commande ci-dessous configure le CUBE pour utiliser son **mylabel d'**étiquette de point de confiance quand elle établit ou reçoit une connexion de TLS avec un périphérique distant dans le sous-réseau de 1.2.3.0. La suite de chiffrement utilisée pendant la prise de contact de TLS est limitée à la suite TLS_RSA_WITH_AES_128_CBC_SHA.

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

7. **Activation du port d'auditeur de TLS**Émettez cette commande en mode de « sip-ua » afin de permettre au port de TLS sur le TCP 5061 d'écouter :
   `transport tcp tls`
8. **Configurer le schéma URL de SIP**Les « sip : Le » schéma URL peut être configuré sous le niveau d'homologue de numérotation VoIP ou au niveau global. Cette commande est utilisée pour configurer des « sip : » dans un pair de cadran de VOIP :
   `voice-class sip url sips` Afin de configurer les « sip : Le » schéma URL sous le niveau global, utilisent cette commande en « mode de sip » de voip » de service vocal « :
   `voice service voip` `sip url sips` L'utilisation de l'URL de SIP exige de tous les sauts dans le circuit d'utiliser le TLS et les SIP. Ceci devient important pour SRTP pendant que les clés sont dans le SDP et pour une connexion sécurisée que les informations ne devraient pas être introduites le texte clair. Si un proxy reçoit une INVITATION avec des SIP (par exemple, INVITEZ LE SIP /2.0 sips:123@proxy) le proxy doit utiliser des SIP pour le prochain saut. Quand le TLS est utilisé avec un URL ordinaire de SIP, il n'y a aucune garantie que tous les sauts utiliseront le TLS, compromettant potentiellement la sécurité de bout en bout de l'appel.Si « sirote » l'URL est configuré, le transport sera automatiquement TLS.

Notes en implémentation de TLS

- L'exécution en cours de CUBE exige l'utilisation du TLS comme transport quand le support sécurisé est configuré (SRTP). Une future amélioration peut soulever cette condition requise.
- Quand SRTP est configuré pour sécuriser la connexion de medias, le TLS ou l'IPSec *doit* également être configuré pour sécuriser les messages de signalisation de SIP. Les clés utilisées pour le cryptage SRTP sont permutées par l'intermédiaire des messages de signalisation – ne pas sécuriser les résultats de canal de signalisation dans les clés SRTP permutées dans le libellé et ceci réalise une inversion la Sécurité de SRTP pour la connexion de medias.
- L'exécution en cours de CUBE exige l'utilisation des « sip : » Schéma d'URI pour un appel de TLS. Une future amélioration peut soulever cette condition requise.
- L'exécution en cours de CUBE a été vérifiée avec un serveur simple CA seulement.

Exemples de configuration

CUBE

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-serveur IOS

Ca-serveur

```
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
```

Vérifiez

Après qu'un appel soit fait, cette **commande show** peut être utilisée afin de vérifier si le transport utilisé pour l'appel est TLS :

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

La sortie témoin pour cette commande est affichée dans ces exemples :

Exemple 1 : Sortie de détail

```
==========================================================================
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
==========================================================================
```

Exemple 2 : Brève sortie

```
==========================================================================
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
==========================================================================
```

Alternativement, la commande de **debug ccsip messages** peut être utilisée pour vérifier « par l'intermédiaire de : la » en-tête pour le TLS est incluse. Cette sortie est un échantillon INVITENT la demande d'un appel qui utilise le TLS de SIP et les « sip : » Schéma d'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
```

Dépannez

Quelques conseils de dépannage pour des appels de TLS incluent :

- Afin de permettre au serveur CA pour fournir des Certificats aux points de confiance, assurez-vous que le routeur IOS qui est configuré car un serveur CA fait activer le HTTP (**ip http server de** commande).
- L'horloge sur le serveur CA et les points de confiance doit être synchronisée.
- Si la prise de contact de TLS échoue entre deux périphériques (par exemple, OGW et CUBE), vérifiez la validité des Certificats sur les périphériques. La commande de **PKI de debug crypto** peut être utilisée pour dépanner des questions pendant la prise de contact de TLS.
- Parfois quand les périphériques (par exemple, OGW et CUBE) sont sur des différents sous-réseaux, là peut une question de la négociation de taille de la fenêtre de TCP qui entraîne ces erreurs : *L'E/S envoient l'erreur* et l'*erreur de lecture E/S*. Cette question peut être résolue avec la commande d'**ip tcp path-mtu-discovery** sur les deux périphériques. Cette question pourrait se produire après une prise de contact réussie de TLS.
- « Les connexions claires de sip-ua » commandent en mode de sip-ua peuvent être utilisées pour effacer des connexions de TLS.`Router#`**`clear sip-ua tcp [tls] connections <id <`***`conn id`***`> | target <`***`ipv4:ip address:port`***`>`** L'option de **tls** apparaît après **TCP** puisque le TLS monte sur le TCP. Cette commande fonctionne comme les commandes claires existantes pour le TCP et UDP.

Informations connexes

- Assistance technique concernant la technologie vocale
- Assistance concernant les produits vocaux et de communications unifiées
- Dépannage des problèmes de téléphonie IP Cisco
- Support et documentation techniques - Cisco Systems