Comprendre la sécurité CUCM par défaut et le fonctionnement et le dépannage ITL

Table des matières

- Introduction
- Informations générales
- Aperçu du SMD
  - Authentification de téléchargement TFTP
  - Chiffrement du fichier de configuration TFTP
  - Service de vérification de la confiance (vérification à distance des certificats et des signatures)
- Informations détaillées et de dépannage de SBD
  - Fichiers et certificats ITL présents sur CUCM
  - Téléchargements téléphoniques ITL et fichier de configuration
  - Le téléphone vérifie ITL et le fichier de configuration
  - Contacts téléphoniques TVS pour certificat inconnu
  - Vérifier manuellement que le téléphone ITL correspond au CUCM ITL
- Restrictions et interactions
  - Régénération des certificats / Reconstruction d'un cluster / Expiration du certificat
  - Déplacement des téléphones entre les clusters
  - Sauvegarde Et Restauration
  - Modifier les noms d'hôte ou de domaine
  - TFTP centralisé
- Forum aux questions
  - Puis-je désactiver SBD ?
  - Puis-je facilement supprimer le fichier ITL de tous les téléphones une fois que CallManager.pem est perdu ?

Introduction

Ce document décrit la fonctionnalité Sécurité par défaut (SBD) de Cisco Unified Communications Manager (CUCM) versions 8.0 et ultérieures.

Informations générales

CUCM version 8.0 et ultérieures introduit la fonctionnalité SBD, qui se compose de fichiers ITL (Identity Trust List) et du service de vérification de la confiance (Trust Verification Service, TVS).

Chaque cluster CUCM utilise désormais automatiquement la sécurité ITL. Il existe un compromis entre la sécurité et la facilité d'utilisation/d'administration que les administrateurs doivent connaître avant d'apporter certaines modifications à un cluster CUCM version 8.0.

Ce document est un complément aux documents officiels Sécurité par défaut, et fournit des informations opérationnelles et des conseils de dépannage pour aider les administrateurs et faciliter la procédure de dépannage.

Il est bon de se familiariser avec les concepts de base suivants du SMD : Article Wikipédia sur le chiffrement à clé asymétrique et article Wikipédia sur l'infrastructure à clé publique.

Aperçu du SMD

Cette section fournit un aperçu rapide de ce que SBD fournit exactement. Pour obtenir des détails techniques complets sur chaque fonction, reportez-vous à la section Informations détaillées et de dépannage de SBD.

SBD fournit ces trois fonctions pour les téléphones IP pris en charge :

- Authentification par défaut des fichiers téléchargés par TFTP (configuration, paramètres régionaux, liste d'appel) qui utilisent une clé de signature
- Cryptage facultatif des fichiers de configuration TFTP qui utilisent une clé de signature
- Vérification de certificat pour les connexions HTTPS initiées par téléphone qui utilisent un magasin de certificats de confiance distant sur CUCM (TVS)

Ce document fournit une vue d'ensemble de chacune de ces fonctions.

Authentification de téléchargement TFTP

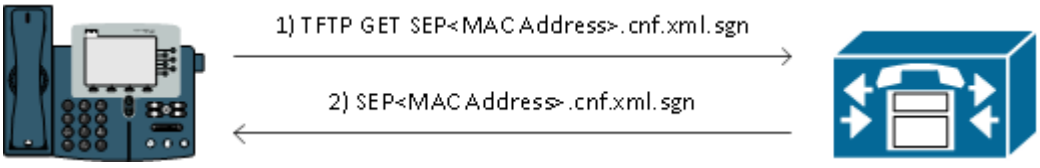
Lorsqu'un fichier CTL (Certificate Trust List) ou ITL est présent, le téléphone IP demande un fichier de configuration TFTP signé au serveur TFTP CUCM.

Ce fichier permet au téléphone de vérifier que le fichier de configuration provient d'une source fiable. Lorsque des fichiers CTL/ITL sont présents sur les téléphones, les fichiers de configuration doivent être signés par un serveur TFTP approuvé.

Le fichier est du texte brut sur le réseau lors de sa transmission, mais il est fourni avec une signature de vérification spéciale.

Le téléphone demande SEP<MAC Address>.cnf.xml.sgn afin de recevoir le fichier de configuration avec la signature spéciale.

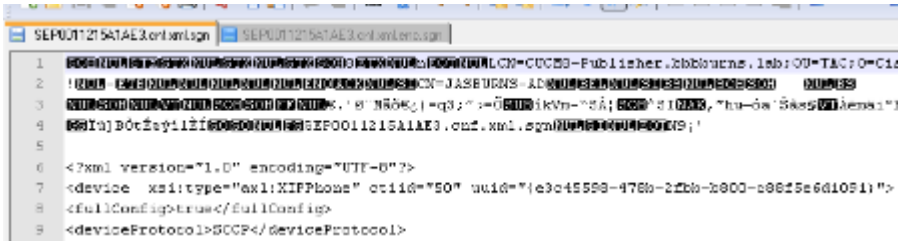
Ce fichier de configuration est signé par la clé privée TFTP qui correspond à CallManager.pem sur la page Administration du système d'exploitation - Gestion des certificats.



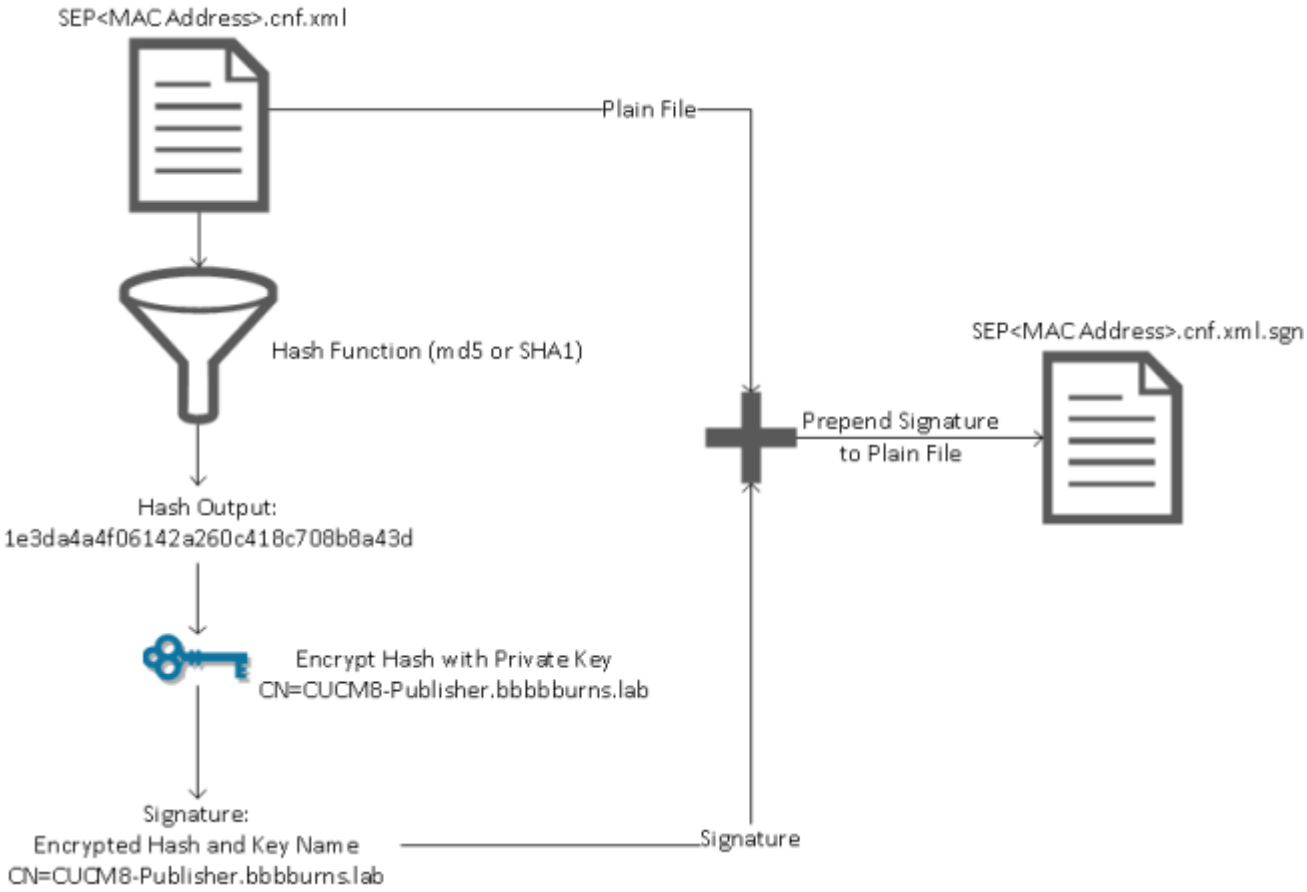
Le fichier signé comporte une signature en haut afin d'authentifier le fichier, mais il est autrement en texte brut XML.

L'image ci-dessous montre que le signataire du fichier de configuration est CN=CUCM8-Publisher.bbbburns.lab qui est à son tour signé par CN=JASBURNS-AD.

Cela signifie que le téléphone doit vérifier la signature de CUCM8-Publisher.bbburns.lab par rapport au fichier ITL avant que ce fichier de configuration ne soit accepté.

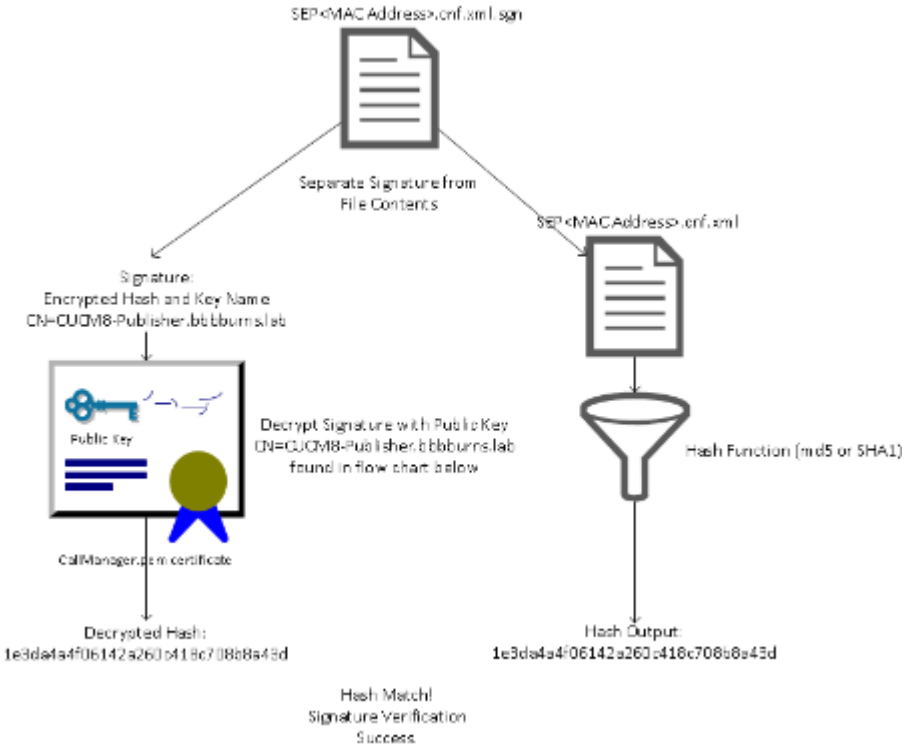


Voici un diagramme qui montre comment la clé privée est utilisée avec une fonction de hachage MD (Message Digest Algorithm)5 ou SHA (Secure Hash Algorithm)1 afin de créer le fichier signé.



La vérification des signatures inverse ce processus en utilisant la clé publique correspondante afin de déchiffrer le hachage. Si les hashs correspondent, il affiche :

- Ce fichier n'a pas été modifié en cours de transfert.
- Ce fichier provient du tiers répertorié dans la signature, car tout élément déchiffré avec succès avec la clé publique doit avoir été chiffré avec la clé privée.



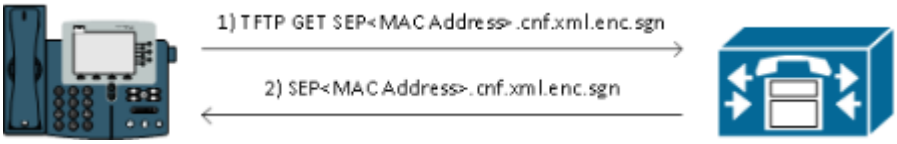
Chiffrement du fichier de configuration TFTP

Si le cryptage de configuration TFTP facultatif est activé dans le profil de sécurité du téléphone associé, le téléphone demande un fichier de configuration crypté.

Ce fichier est signé avec la clé privée TFTP et chiffré avec une clé symétrique échangée entre le téléphone et le CUCM (reportez-vous au Guide de sécurité Cisco Unified Communications Manager, version 8.5(1) pour plus de détails).

Son contenu ne peut pas être lu avec un analyseur réseau à moins que l'observateur ne dispose des clés nécessaires.

Le téléphone demande SEP<MAC Address>.cnf.xml.enc.sgn afin d'obtenir le fichier chiffré signé.



Le fichier de configuration chiffré a également la signature au début, mais il n'y a pas de données en texte clair après, seulement des données chiffrées (caractères binaires brouillés dans cet éditeur de texte).

L'image montre que le signataire est le même que dans l'exemple précédent, de sorte que ce signataire doit être présent dans le fichier ITL avant que le téléphone n'accepte le fichier.

En outre, les clés de déchiffrement doivent être correctes avant que le téléphone puisse lire le contenu du fichier.



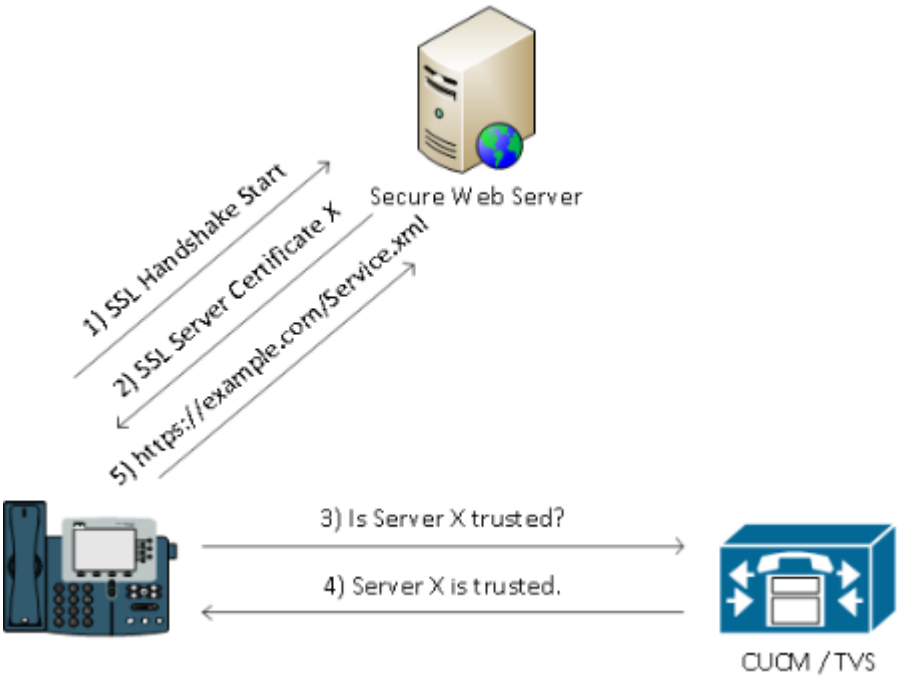
Service de vérification de la confiance (vérification à distance des certificats et des signatures)

Les téléphones IP contiennent une quantité limitée de mémoire et il peut également y avoir un grand nombre de téléphones à gérer sur un réseau.

CUCM agit comme un magasin de confiance distant via le TVS, de sorte qu'un magasin de confiance de certificat complet n'a pas besoin d'être placé sur chaque téléphone IP.

Chaque fois que le téléphone ne peut pas vérifier une signature ou un certificat via les fichiers CTL ou ITL, il demande au serveur TVS de procéder à la vérification.

Ce magasin de confiance central est plus facile à gérer que s'il était présent sur tous les téléphones IP.



Informations détaillées et de dépannage de SBD

Cette section décrit en détail le processus du SMD.

Fichiers et certificats ITL présents sur CUCM

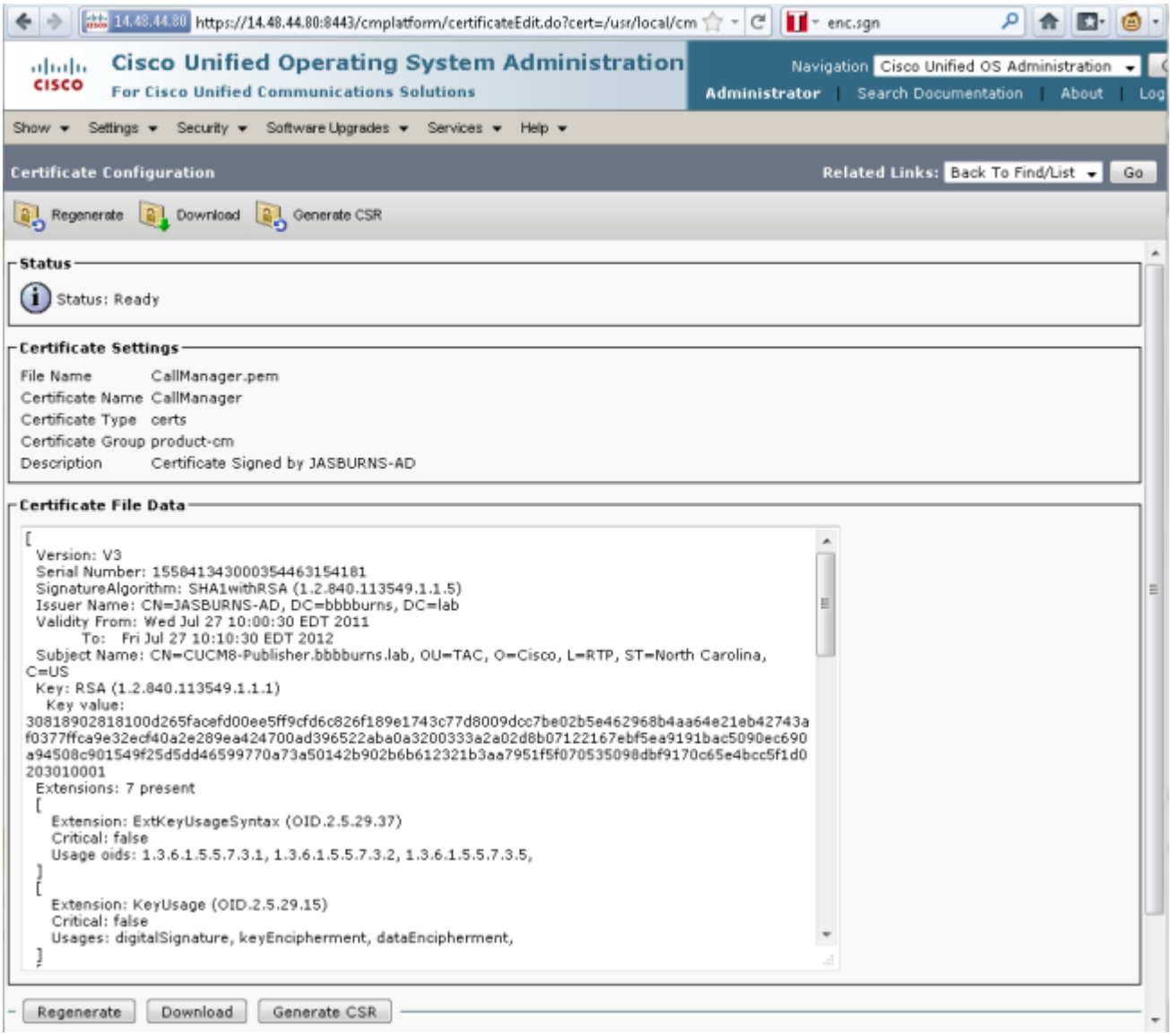
Tout d'abord, un certain nombre de fichiers doivent être présents sur le serveur CUCM lui-même. L'élément le plus important est le certificat TFTP et la clé privée TFTP.

Le certificat TFTP se trouve sous OS Administration > Security > Certificate Management > CallManager.pem.

Le serveur CUCM utilise les clés privée et publique du certificat CallManager.pem pour le service TFTP (ainsi que pour le service Cisco Call Manager (CCM)).

L'image montre que le certificat CallManager.pem est émis vers CUCM8-publisher.bbburns.lab et signé par JASBURNS-AD. Tous les fichiers de configuration TFTP sont signés par la clé privée ci-dessous.

Tous les téléphones peuvent utiliser la clé publique TFTP dans le certificat CallManager.pem afin de déchiffrer tout fichier chiffré avec la clé privée TFTP, ainsi que pour vérifier tout fichier signé avec la clé privée TFTP.



En plus de la clé privée de certificat CallManager.pem, le serveur CUCM stocke également un fichier ITL qui est présenté aux téléphones.

La commande show itl affiche le contenu complet de ce fichier ITL via l'accès SSH (Secure Shell) à l'interface de ligne de commande du système d'exploitation du serveur CUCM.

Cette section décompose le fichier ITL pièce par pièce, car il contient un certain nombre de composants importants que le téléphone utilise.

La première partie correspond aux informations de signature. Même le fichier ITL est signé. Ce résultat montre qu'il est signé par la clé privée TFTP associée au certificat CallManager.pem précédent.

<#root>

```
admin:

show itl


Length of ITL file: 5438
The ITL File was last modified on Wed Jul 27 10:16:24 EDT 2011

        Parse ITL File
        ----------------

Version:        1.2
HeaderLength:   296 (BYTES)

BYTEPOS TAG             LENGTH  VALUE
------- ---             ------  -----
3       SIGNERID        2       110
4       SIGNERNAME      76      CN=CUCM8-Publisher.bbbburns.lab;
                                OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
5       SERIALNUMBER    10      21:00:2D:17:00:00:00:00:00:05
6       CANAME          15      CN=JASBURNS-AD
*Signature omitted for brevity*
```

Les sections suivantes contiennent chacune leur fonction à l'intérieur d'un paramètre Function spécial. La première fonction est le jeton de sécurité de l'administrateur système. Il s’agit de la signature de la clé publique TFTP.

```
        ITL Record #:1
                  ----
BYTEPOS TAG             LENGTH  VALUE
------- ---             ------  -----
1       RECORDLENGTH    2       1972
2       DNSNAME         2
3       SUBJECTNAME     76      CN=CUCM8-Publisher.bbbburns.lab;
                                OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4       FUNCTION        2       System Administrator Security Token
5       ISSUERNAME      15      CN=JASBURNS-AD
6       SERIALNUMBER    10      21:00:2D:17:00:00:00:00:00:05
7       PUBLICKEY       140
8       SIGNATURE       256
9       CERTIFICATE     1442    0E 1E 28 0E 5B 5D CC 7A 20 29 61 F5
                                8A DE 30 40 51 5B C4 89 (SHA1 Hash HEX)
This etoken was used to sign the ITL file.
```

La fonction suivante est CCM+TFTP. Il s'agit à nouveau de la clé publique TFTP qui sert à authentifier et à déchiffrer les fichiers de configuration TFTP téléchargés.

```
        ITL Record #:2
                  ----
BYTEPOS TAG             LENGTH  VALUE
------- ---             ------  -----
1       RECORDLENGTH    2       1972
2       DNSNAME         2
3       SUBJECTNAME     76      CN=CUCM8-Publisher.bbbburns.lab;
                                OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4       FUNCTION        2       CCM+TFTP
5       ISSUERNAME      15      CN=JASBURNS-AD
6       SERIALNUMBER    10      21:00:2D:17:00:00:00:00:00:05
7       PUBLICKEY       140
8       SIGNATURE       256
9       CERTIFICATE     1442    0E 1E 28 0E 5B 5D CC 7A 20 29 61 F5
                                8A DE 30 40 51 5B C4 89 (SHA1 Hash HEX)
```

La fonction suivante est TVS. Il existe une entrée pour la clé publique de chaque serveur TVS auquel le téléphone se connecte.

Cela permet au téléphone d'établir une session SSL (Secure Sockets Layer) vers le serveur TVS.

```
        ITL Record #:3
                  ----
BYTEPOS TAG             LENGTH  VALUE
------- ---             ------  -----
1       RECORDLENGTH    2       743
2       DNSNAME         2
3       SUBJECTNAME     76      CN=CUCM8-Publisher.bbbburns.lab;
                                OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4       FUNCTION        2       TVS
5       ISSUERNAME      76      CN=CUCM8-Publisher.bbbburns.lab;
                                OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
6       SERIALNUMBER    8       2E:3E:1A:7B:DA:A6:4D:84
7       PUBLICKEY       270
8       SIGNATURE       256
11      CERTHASH        20      C7 E1 D9 7A CC B0 2B C2 A8 B2 90 FB
                                AA FE 66 5B EC 41 42 5D
12      HASH ALGORITHM  1       SHA-1
```

La dernière fonction incluse dans le fichier ITL est la fonction proxy de l'autorité de certification (CAPF).

Ce certificat permet aux téléphones d'établir une connexion sécurisée au service CAPF sur le serveur CUCM afin que le téléphone puisse installer ou mettre à jour un certificat LSC (Locally Significative Certificate).

```
        ITL Record #:4
                  ----
BYTEPOS TAG              LENGTH  VALUE
------- ---              ------  -----
1       RECORDLENGTH     2       455
2       DNSNAME          2
3       SUBJECTNAME      61      CN=CAPF-9c4cba7d;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4       FUNCTION         2       CAPF
5       ISSUERNAME       61      CN=CAPF-9c4cba7d;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
6       SERIALNUMBER     8       0A:DC:6E:77:42:91:4A:53
7       PUBLICKEY        140
8       SIGNATURE        128
11      CERTHASH         20      C7 3D EA 77 94 5E 06 14 D2 90 B1
                                 A1 43 7B 69 84 1D 2D 85 2E
12      HASH ALGORITHM   1       SHA-1

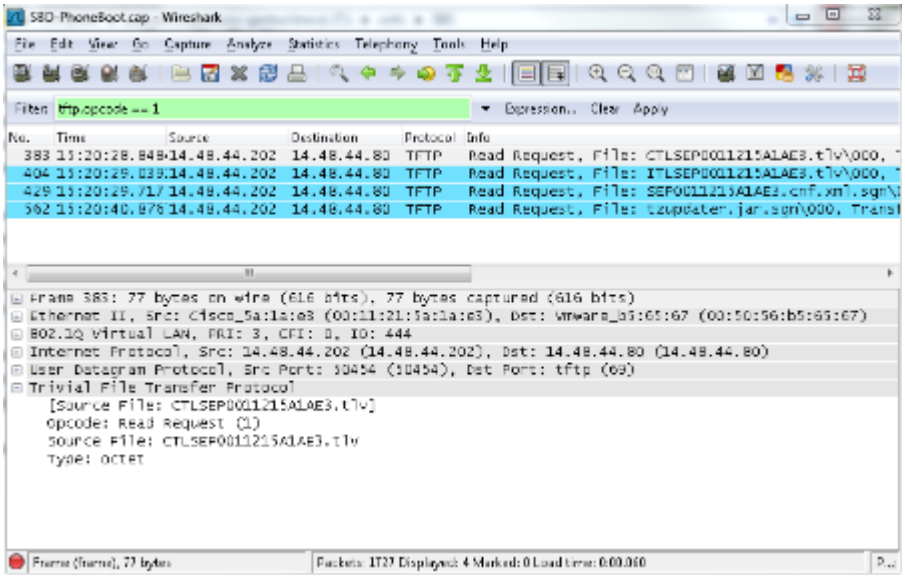
The ITL file was verified successfully.
```

La section suivante décrit exactement ce qui se passe lorsqu'un téléphone démarre.

Téléchargements téléphoniques ITL et fichier de configuration

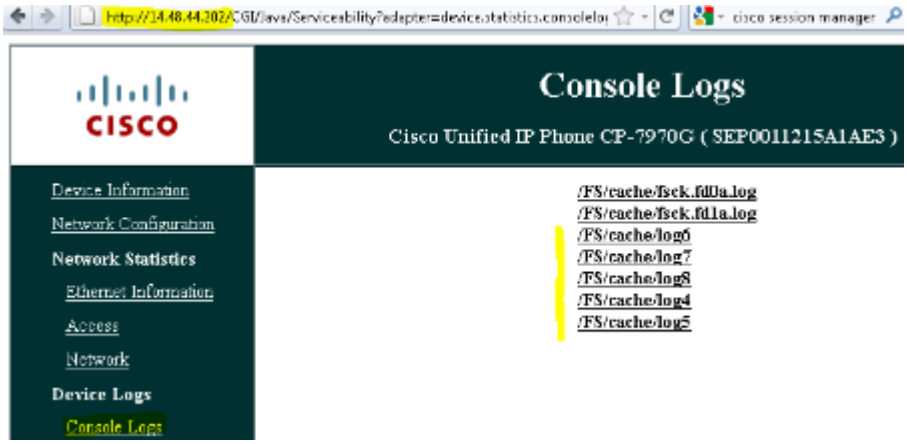
Une fois que le téléphone a démarré et qu'il a obtenu une adresse IP ainsi que l'adresse d'un serveur TFTP, il demande d'abord les fichiers CTL et ITL.

Cette capture de paquets montre une demande de téléphone pour le fichier ITL. Si vous filtrez sur tftp.opcode == 1, vous voyez chaque requête de lecture TFTP du téléphone :



Comme le téléphone a reçu des fichiers CTL et ITL de TFTP, il demande un fichier de configuration signé.

Les journaux de la console téléphonique qui affichent ce comportement sont disponibles à partir de l'interface Web du téléphone :



Tout d'abord, le téléphone demande un fichier CTL, ce qui réussit :

```
837: NOT 09:13:17.561856 SECD: tlRequestFile: Request CTLSEP0011215A1AE3.tlv
846: NOT 09:13:17.670439 TFTP: [27]:Requesting CTLSEP0011215A1AE3.tlv from
14 . 48 . 44 . 80
847: NOT 09:13:17.685264 TFTP: [27]:Finished --> rcvd 4762 bytes
```

Ensuite, le téléphone demande également un fichier ITL :

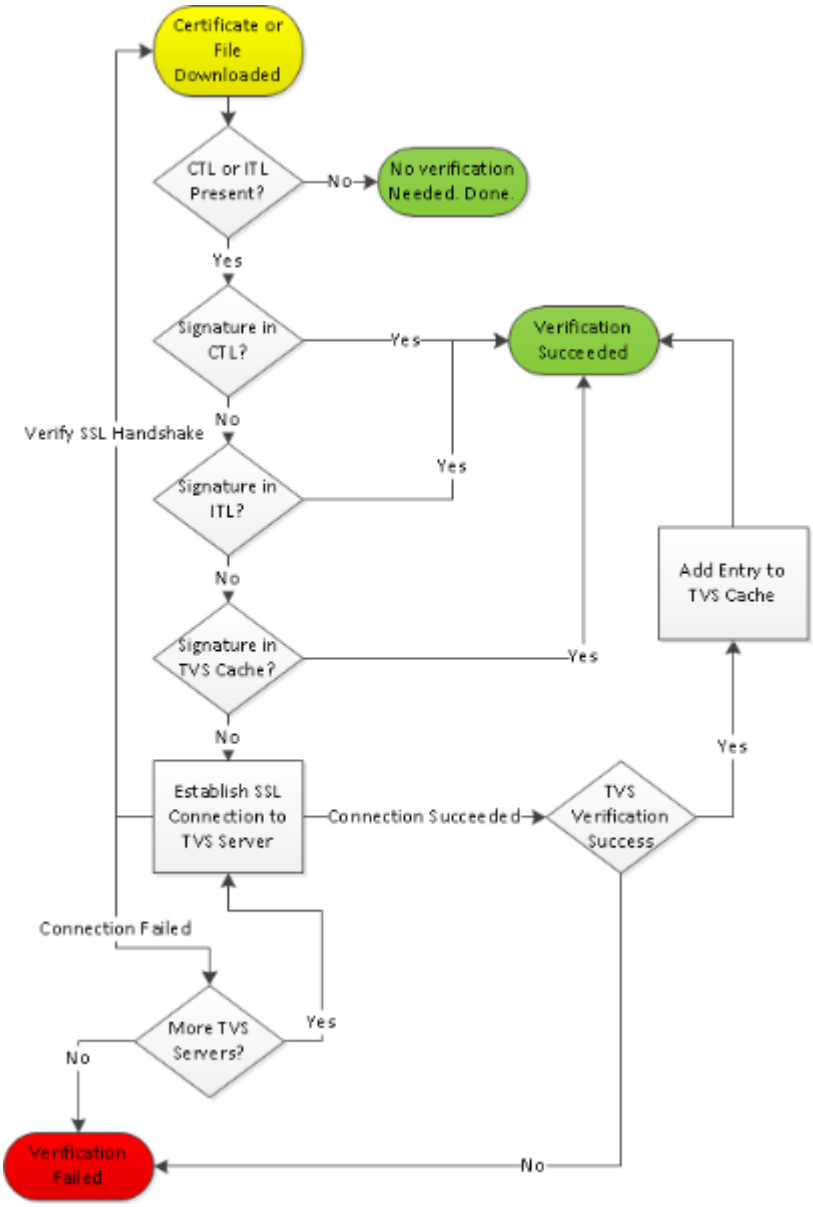
```
868: NOT 09:13:17.860613 TFTP: [28]:Requesting ITLSEP0011215A1AE3.tlv from
14 . 48 . 44 . 80
869: NOT 09:13:17.875059 TFTP: [28]:Finished --> rcvd 5438 bytes
```

Le téléphone vérifie ITL et le fichier de configuration

Une fois le fichier ITL téléchargé, il doit être vérifié. Il existe plusieurs états dans lesquels un téléphone peut se trouver à ce stade. Ce document couvre donc tous ces états.

- Aucun fichier CTL ou ITL n'est présent sur le téléphone ou ITL est vide en raison du paramètre Prepare Cluster for Rollback to Pre 8.0. dans cet état, le téléphone fait confiance aveuglément au prochain fichier CTL ou ITL téléchargé et utilise désormais cette signature.
- Le téléphone possède déjà une CTL, mais pas d'ITL. Dans cet état, le téléphone n'approuve un ITL que s'il peut être vérifié par la fonction CCM+TFTP dans le fichier CTL.
- Le téléphone dispose déjà d'une CTL et d'un fichier ITL. Dans cet état, le téléphone vérifie que les fichiers récemment téléchargés correspondent à la signature sur le serveur CTL, ITL ou TVS.

Voici un organigramme qui décrit comment le téléphone vérifie les fichiers signés et les certificats HTTPS :



Dans ce cas, le téléphone peut vérifier la signature dans les fichiers ITL et CTL. Le téléphone dispose déjà d'une CTL et d'une ITL. Il a donc simplement vérifié leur présence et trouvé la signature correcte.

```
877: NOT 09:13:17.925249 SECD: validate_file_envelope:
File sign verify SUCCESS; header length <296>
```

Depuis que le téléphone a téléchargé les fichiers CTL et ITL, il ne demande plus que les fichiers de configuration signés.

Ceci montre que la logique téléphonique consiste à déterminer que le serveur TFTP est sécurisé, en fonction de la présence de CTL et ITL, puis à demander un fichier signé :

```
917: NOT 09:13:18.433411 tftpClient: tftp request rcv'd from /usr/tmp/tftp,
srcFile = SEP0011215A1AE3.cnf.xml, dstFile = /usr/ram/SEP0011215A1AE3.cnf.xml
```

```
max size = 550001
918: NOT 09:13:18.457949 tftpClient: auth server - tftpList[0] = ::ffff:
14 . 48 . 44 . 80
919: NOT 09:13:18.458937 tftpClient: look up server - 0
920: NOT 09:13:18.462479 SECD: lookupCTL: TFTP SRVR secure
921: NOT 09:13:18.466658 tftpClient: secVal = 0x9 922: NOT 09:13:18.467762
tftpClient: ::ffff:14 . 48 . 44 . 80 is a secure server
923: NOT 09:13:18.468614 tftpClient: retval = SRVR_SECURE
924: NOT 09:13:18.469485 tftpClient: Secure file requested
925: NOT 09:13:18.471217 tftpClient: authenticated file approved - add .sgn
-- SEP0011215A1AE3.cnf.xml.sgn
926: NOT 09:13:18.540562 TFTP: [10]:Requesting SEP0011215A1AE3.cnf.xml.sgn
from 14 . 48 . 44 . 80 with size limit of 550001
927: NOT 09:13:18.559326 TFTP: [10]:Finished --> rcvd 7652 bytes
```

Une fois le fichier de configuration signé téléchargé, le téléphone doit l'authentifier par rapport à la fonction CCM+TFTP dans l'ITL :

```
937: NOT 09:13:18.656906 SECD: verifyFile: verify SUCCESS
</usr/ram/SEP0011215A1AE3.cnf.xml>
```

Contacts téléphoniques TVS pour certificat inconnu

Le fichier ITL fournit une fonction TVS qui contient le certificat du service TVS qui s'exécute sur le port TCP 2445 du serveur CUCM.

TVS s'exécute sur tous les serveurs sur lesquels le service CallManager est activé. Le service TFTP CUCM utilise le groupe CallManager configuré afin de créer une liste de serveurs TVS que le téléphone doit contacter dans le fichier de configuration du téléphone.

Certains TP utilisent un seul serveur CUCM. Dans un cluster CUCM à plusieurs noeuds, il peut y avoir jusqu'à trois entrées TVS pour un téléphone, une pour chaque CUCM du groupe CUCM du téléphone.

Cet exemple montre ce qui se passe lorsque le bouton Directories du téléphone IP est enfoncé. L'URL des répertoires étant configurée pour HTTPS, le téléphone reçoit le certificat Web Tomcat du serveur de répertoires.

Ce certificat Web Tomcat (tomcat.pem dans Administration du système d'exploitation) n'est pas chargé dans le téléphone. Le téléphone doit donc contacter TVS pour authentifier le certificat.

Reportez-vous au schéma de présentation TVS précédent pour une description de l'interaction. Voici la perspective du journal de la console téléphonique :

Vous trouverez d'abord l'URL du répertoire :

```
1184: NOT 15:20:55.219275 JVM: Startup Module Loader|cip.dir.TandunDirectories:
```

```
? - Directory url https://14 . 48 . 44 . 80:8443/ccmcip/xmldirectory.jsp
```

Il s'agit d'une session HTTP sécurisée SSL/TLS (Transport Layer Security) qui nécessite une vérification.

```
1205: NOT 15:20:59.404971 SECD: clpSetupSsl: Trying to connect to IPV4, IP:
14 . 48 . 44 . 80, Port : 8443
1206: NOT 15:20:59.406896 SECD: clpSetupSsl: TCP connect() waiting,
<14 . 48 . 44 . 80> c:8 s:9 port: 8443
1207: NOT 15:20:59.408136 SECD: clpSetupSsl: TCP connected,
<14 . 48 . 44 . 80> c:8 s:9
1208: NOT 15:20:59.409393 SECD: clpSetupSsl: start SSL/TLS handshake,
<14 . 48 . 44 . 80> c:8 s:9
1209: NOT 15:20:59.423386 SECD: srvr_cert_vfy: Server Certificate
Validation needs to be done
```

Le téléphone vérifie d'abord que le certificat présenté par le serveur SSL/TLS est présent dans la CTL. Le téléphone examine ensuite les fonctions du fichier ITL afin de voir s'il trouve une correspondance.

Ce message d'erreur indique « HTTPS cert not in CTL », ce qui signifie « que la certification est introuvable dans la CTL ou l'ITL ».

```
1213: NOT 15:20:59.429176 SECD: findByCertAndRoleInTL: Searching TL from CTL file
1214: NOT 15:20:59.430315 SECD: findByCertAndRoleInTL: Searching TL from ITL file
1215: ERR 15:20:59.431314 SECD: EROR:https_cert_vfy: HTTPS cert not in CTL,
<14 . 48 . 44 . 80>
```

Une fois que le contenu direct des fichiers CTL et ITL est vérifié pour le certificat, la prochaine chose que le téléphone vérifie est le cache TVS.

Ceci est fait afin de réduire le trafic réseau si le téléphone a récemment demandé le même certificat au serveur TVS.

Si le certificat HTTPS est introuvable dans le cache du téléphone, vous pouvez établir une connexion TCP au serveur TVS lui-même.

```
1220: NOT 15:20:59.444517 SECD: processTvsClntReq: TVS Certificate
Authentication request
1221: NOT 15:20:59.445507 SECD: lookupAuthCertTvsCacheEntry: No matching
entry found at cache
1222: NOT 15:20:59.446518 SECD: processTvsClntReq: No server sock exists,
must be created
1223: NOT 15:20:59.451378 SECD: secReq_initClient: clnt sock fd 11 bound
to </tmp/secClnt_secd>
```

```
1224: NOT 15:20:59.457643 SECD: getTvsServerInfo: Phone in IPv4 only mode
1225: NOT 15:20:59.458706 SECD: getTvsServerInfo: Retreiving IPv4 address
1230: NOT 15:20:59.472628 SECD: connectToTvsServer: Successfully started
a TLS connection establishment to the TVS server: IP:14 . 48 . 44 . 80, port:2445
(default); Waiting for it to get connected.
```

N'oubliez pas que la connexion à TVS est SSL/TLS (HTTP sécurisé ou HTTPS). Il s'agit donc également d'un certificat qui doit être authentifié par rapport à la CTL vers ITL.

Si tout se passe correctement, le certificat du serveur TVS se trouve dans la fonction TVS du fichier ITL. Voir Enregistrement ITL #3 dans l'exemple de fichier ITL précédent.

```
1244: NOT 15:20:59.529938 SECD: srvr_cert_vfy: Server Certificate Validation
needs to be done
1245: NOT 15:20:59.533412 SECD: findByIssuerAndSerialAndRoleInTL:
Searching TL from CTL file
1246: NOT 15:20:59.534936 SECD: findByIssuerAndSerialAndRoleInTL:
Searching TL from ITL file
1247: NOT 15:20:59.537359 SECD: verifyCertWithHashFromTL: cert hash and
hash in TL MATCH
1248: NOT 15:20:59.538726 SECD: tvs_cert_vfy: TVS cert verified with hash
from TL, <14 . 48 . 44 . 80>
```

Réussite ! Le téléphone dispose désormais d'une connexion sécurisée au serveur TVS. L'étape suivante consiste à demander au serveur TVS : « Bonjour, puis-je faire confiance à ce certificat de serveur de répertoires ? »

Cet exemple montre la réponse à cette question - une réponse de 0 qui signifie réussite (aucune erreur).

```
1264: NOT 15:20:59.789738 SECD: sendTvsClientReqToSrvr: Authenticate
Certificate : request sent to TVS server - waiting for response
1273: NOT 15:20:59.825648 SECD: processTvsSrvrResponse: Authentication Response
received, status : 0
```

Étant donné que TVS a répondu avec succès, les résultats de ce certificat sont enregistrés dans le cache.

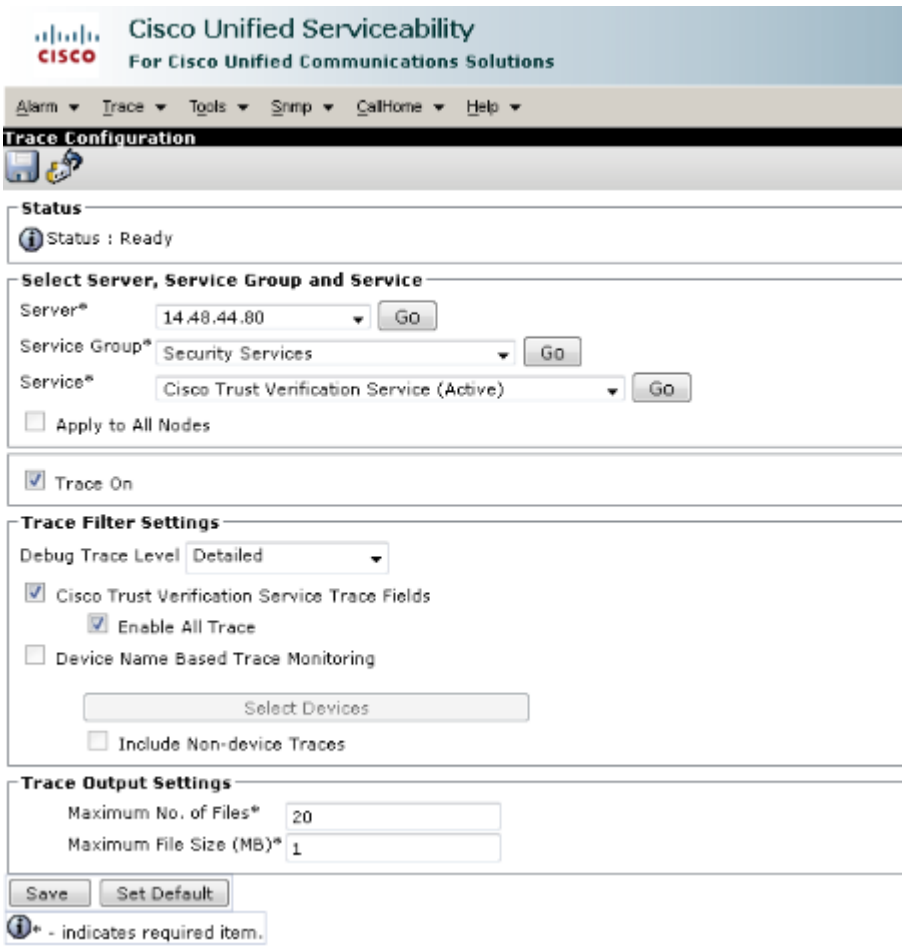
Cela signifie que, si vous appuyez à nouveau sur le bouton Directories dans les 86,400 secondes suivantes, vous n'avez pas besoin de contacter le serveur TVS afin de vérifier le certificat. Vous pouvez simplement accéder au cache local.

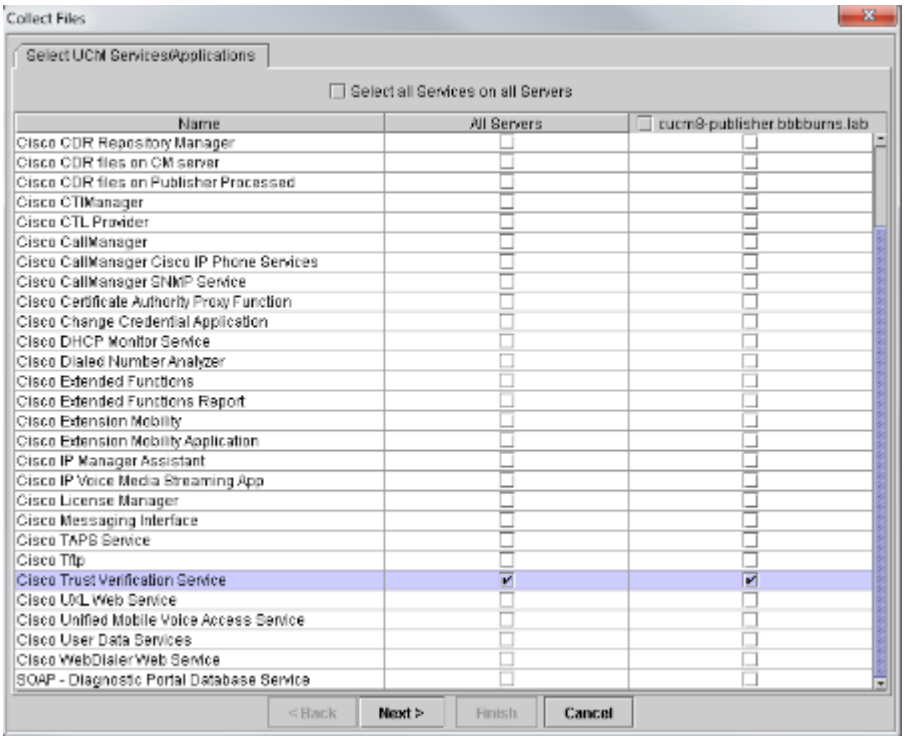
```
1279: NOT 15:20:59.837086 SECD: saveCertToTvsCache: Saving certificate
in TVS cache with default time-to-live value: 86400 seconds
1287: ERR 15:20:59.859993 SECD: Authenticated the HTTPS conn via TVS
```

Enfin, vous devez vérifier que votre connexion au serveur Répertoires a réussi.

```
1302: ERR 15:21:01.959700 JVM: Startup Module Loader|cip.http.ae:?
- listener.httpSucceed:  https://14 . 48 . 44 . 80:8443/ccmcip/
xmldirectoryinput.jsp?name=SEP0011215A1AE3
```

Voici un exemple de ce qui se passe sur le serveur CUCM où TVS s'exécute. Vous pouvez collecter les journaux TVS à l'aide de l'outil Cisco Unified Real-Time Monitoring Tool (RTMT).





Les journaux CUCM TVS indiquent que vous avez établi une connexion SSL avec le téléphone, le téléphone demande à TVS le certificat Tomcat, puis TVS répond pour indiquer que le certificat correspond dans le magasin de certificats TVS.

```
15:21:01.954 |   debug 14 . 48 . 44 . 202: tvsSSLHandShake Session ciphers - AES256-SHA
15:21:01.954 |   debug TLS HS Done for ph_conn .
15:21:02.010 |   debug     MsgType               : TVS_MSG_CERT_VERIFICATION_REQ
15:21:02.011 |   debug tvsGetIssuerNameFromX509 - issuerName : CN=CUCM8-
Publisher.bbbburns.lab;OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US and Length: 75

15:21:02.011 |   debug CertificateDBCache::getCertificateInformation -
Certificate compare return =0
15:21:02.011 |   debug CertificateDBCache::getCertificateInformation -
Certificate found and equal
15:21:02.011 |   debug     MsgType               : TVS_MSG_CERT_VERIFICATION_RES
```

Le magasin de certificats TVS est une liste de tous les certificats contenus sur la page Web Administration du système d'exploitation > Certificate Management.

Vérifier manuellement que le téléphone ITL correspond au CUCM ITL

Une idée fausse courante observée lors du dépannage concerne la tendance à supprimer le fichier ITL dans l'espoir qu'il résout un problème de vérification de fichier.

Parfois, la suppression du fichier ITL est nécessaire, mais le fichier ITL ne doit être supprimé que lorsque TOUTES ces conditions sont remplies.

- La signature du fichier ITL sur le téléphone ne correspond pas à la signature du fichier ITL sur le serveur TFTP CM.

- La signature TVS dans le fichier ITL ne correspond pas au certificat présenté par TVS.
- Le téléphone affiche « Échec de la vérification » lorsqu'il tente de télécharger le fichier ITL ou les fichiers de configuration.
- Il n'existe aucune sauvegarde de l'ancienne clé privée TFTP.

Voici comment vous vérifiez les deux premières de ces conditions.

Tout d'abord, vous pouvez comparer la somme de contrôle du fichier ITL présent sur CUCM avec le fichier ITL de somme de contrôle du téléphone.

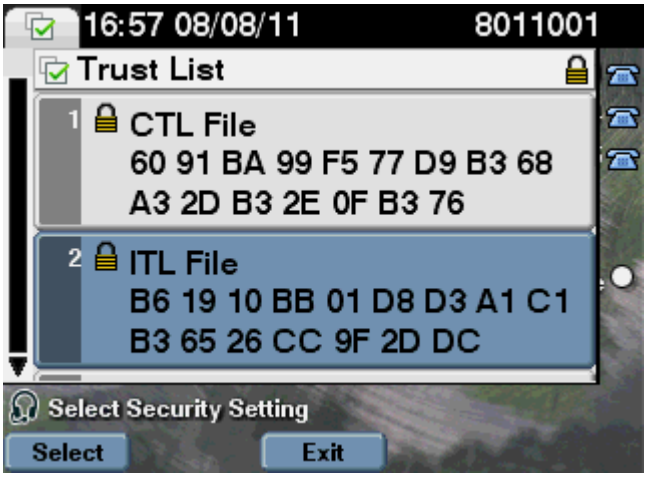
Il n'y a actuellement aucun moyen de regarder la somme MD5sum du fichier ITL sur CUCM à partir de CUCM lui-même jusqu'à ce que vous exécutiez une version avec le correctif pour ce bogue Cisco ID CSCto60209.

Entre-temps, exécutez ceci avec votre interface graphique utilisateur ou vos programmes CLI préférés :

```
jasburns@jasburns-gentoo /data/trace/jasburns/certs/SBD $ tftp 14 . 48 . 44 . 80
tftp> get ITLSEP0011215A1AE3.tlv
Received 5438 bytes in 0.0 seconds
tftp> quit
jasburns@jasburns-gentoo /data/trace/jasburns/certs/SBD $ md5sum
ITLSEP0011215A1AE3.tlv
b61910bb01d8d3a1c1b36526cc9f2ddc  ITLSEP0011215A1AE3.tlv
```

Cela montre que la somme MD5sum du fichier ITL dans CUCM est b61910bb01d8d3a1c1b36526cc9f2ddc.

Maintenant, vous pouvez regarder le téléphone lui-même afin de déterminer le hachage du fichier ITL chargé là : Paramètres > Configuration de la sécurité > Liste de confiance.



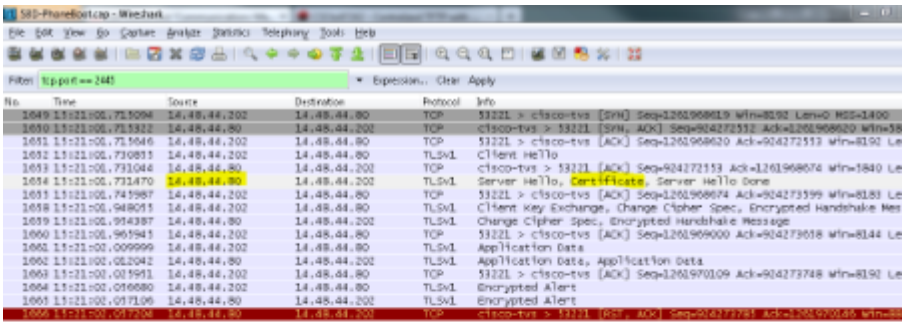
Cela montre que les sommes MD5 correspondent. Cela signifie que le fichier ITL sur le téléphone correspond au fichier sur le CUCM, de sorte qu'il n'a pas besoin d'être supprimé.

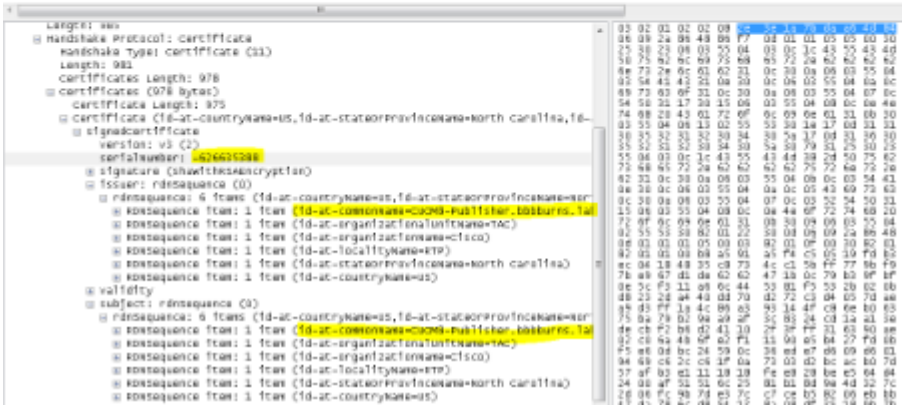
S'il CORRESPOND, vous devez passer à l'opération suivante : déterminer si le certificat TVS de

l'ITL correspond ou non au certificat présenté par TVS. Cette opération est un peu plus compliquée.

Commencez par observer la capture de paquets du téléphone qui se connecte au serveur TVS sur le port TCP 2445.

Cliquez avec le bouton droit sur un paquet de ce flux dans Wireshark, cliquez sur Decode As, et sélectionnez SSL. Recherchez le certificat de serveur qui ressemble à ceci :





Examinez le certificat TVS contenu dans le fichier ITL précédent. Vous voyez alors une entrée avec le numéro de série 2E3E1A7BDAA64D84.

<#root>

admin:

show itl

```
          ITL Record #:3
                    ----
BYTEPOS TAG              LENGTH  VALUE
------- ---              ------  -----
1       RECORDLENGTH     2       743
2       DNSNAME          2
3       SUBJECTNAME      76      CN=CUCM8-Publisher.bbbburns.lab;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4       FUNCTION         2       TVS
5       ISSUERNAME       76      CN=CUCM8-Publisher.bbbburns.lab;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
6       SERIALNUMBER     8       2E:3E:1A:7B:DA:A6:4D:84
```

En cas de succès, le TVS.pem à l'intérieur du fichier ITL correspond au certificat TVS présenté sur le réseau. Vous n'avez pas besoin de supprimer l'ITL et TVS présente le certificat correct.

Si l'authentification du fichier échoue toujours, vérifiez le reste de l'organigramme précédent.

Restrictions et interactions

Régénération des certificats / Reconstruction d'un cluster / Expiration du certificat

Le certificat le plus important est désormais le certificat CallManager.pem. Cette clé privée de certificat est utilisée afin de signer tous les fichiers de configuration TFTP, qui incluent le fichier ITL.

Si le fichier CallManager.pem est régénéré, un nouveau certificat CCM+TFTP est généré avec une nouvelle clé privée. En outre, le fichier ITL est désormais signé par cette nouvelle clé CCM+TFTP.

Une fois que vous avez régénéré CallManager.pem et redémarré le service TVS et TFTP, cela se produit lorsqu'un téléphone démarre.

1. Le téléphone tente de télécharger le nouveau fichier ITL signé par le nouveau CCM+TFTP à partir du serveur TFTP. À ce stade, le téléphone ne dispose que de l'ancien fichier ITL et les nouvelles clés ne figurent pas dans le fichier ITL présent sur le téléphone.
2. Comme le téléphone n'a pas pu trouver la nouvelle signature CCM+TFTP dans l'ancien ITL, il tente de contacter le service TVS.

   Remarque : Cette partie est extrêmement importante. Le certificat TVS de l'ancien fichier ITL doit toujours correspondre. Si CallManager.pem et TVS.pem sont régénérés en même temps, les téléphones ne peuvent pas télécharger de nouveaux fichiers sans supprimer manuellement l'ITL du téléphone.

   ---

3. Lorsque le téléphone contacte TVS, le serveur CUCM qui exécute TVS dispose du nouveau certificat CallManager.pem dans le magasin de certificats du système d'exploitation.
4. Le serveur TVS renvoie une réponse positive et le téléphone charge le nouveau fichier ITL dans la mémoire.
5. Le téléphone tente à présent de télécharger un fichier de configuration, qui a été signé par la nouvelle clé CallManager.pem.
6. Comme le nouveau ITL a été chargé, le fichier de configuration nouvellement signé est vérifié par le ITL en mémoire.

Principaux points :

- Ne régénérez jamais les certificats CallManager.pem et TVS.pem en même temps.
- Si TVS.pem ou CallManager.pem est régénéré, TVS et TFTP doivent être redémarrés et les téléphones réinitialisés afin d'obtenir les nouveaux fichiers ITL.
- Les versions plus récentes de CUCM gèrent automatiquement cette réinitialisation du téléphone et avertissent l'utilisateur lors de la régénération du certificat.

- S'il existe plusieurs serveurs TVS (plusieurs serveurs dans le groupe CallManager), les serveurs supplémentaires peuvent authentifier le nouveau certificat CallManager.pem.

Déplacement des téléphones entre les clusters

Lorsque vous déplacez des téléphones d'un cluster à un autre avec des ITL en place, la clé privée ITL et TFTP doit être prise en compte.

Tout nouveau fichier de configuration présenté au téléphone DOIT correspondre à une signature dans CTL, ITL ou une signature dans le service TVS actuel du téléphone.

Ce document explique comment s'assurer que le nouveau fichier ITL de cluster et les fichiers de configuration peuvent être approuvés par le fichier ITL actuel sur le téléphone. https://supportforums.cisco.com/docs/DOC-15799.

Sauvegarde Et Restauration

Le certificat CallManager.pem et la clé privée sont sauvegardés via le système de récupération après sinistre (DRS). Si un serveur TFTP est reconstruit, il DOIT être restauré à partir d'une sauvegarde afin que la clé privée puisse être restaurée.

Sans la clé privée CallManager.pem sur le serveur, les téléphones avec des ITL actuels qui utilisent l'ancienne clé ne font pas confiance aux fichiers de configuration signés.

Si un cluster est reconstruit et non restauré à partir d'une sauvegarde, il est exactement comme le document "Déplacement des téléphones entre les clusters". En effet, un cluster avec une nouvelle clé est un cluster différent en ce qui concerne les téléphones.

La sauvegarde et la restauration présentent un défaut grave. Si un cluster est sensible à l'ID de bogue Cisco CSCtn50405, les sauvegardes DRS ne contiennent pas le certificat CallManager.pem.

Tout serveur restauré à partir de cette sauvegarde génère des fichiers ITL endommagés jusqu'à ce qu'un nouveau fichier CallManager.pem soit généré.

Si aucun autre serveur TFTP fonctionnel n'a effectué l'opération de sauvegarde et de restauration, cela signifie peut-être que tous les fichiers ITL doivent être supprimés des téléphones.

Afin de vérifier si votre fichier CallManager.pem doit être régénéré, entrez la commande show itl suivie de :

```
run sql select c.subjectname, c.serialnumber, c.ipv4address, t.name from
certificate as c, certificatetrustrolemap as r, typetrustrole as t where  c.pkid =
r.fkcertificate and t.enum = r.tktrustrole
```

Dans la sortie ITL, les principales erreurs à rechercher sont les suivantes :

```
This etoken was not used to sign the ITL file.
```

et

```
Verification of the ITL file failed.
Error parsing the ITL file!!
```

La requête SQL (Structured Query Language) précédente recherche les certificats ayant le rôle « Authentification et autorisation ».

Le certificat CallManager.pem dans la requête de base de données précédente qui a le rôle d'authentification et d'autorisation doit également être présent dans la page Web Gestion des certificats d'administration du système d'exploitation.

Si le défaut précédent est rencontré, il existe une incompatibilité entre les certificats CallManager.pem dans la requête et dans la page Web du système d'exploitation.

Modifier les noms d'hôte ou de domaine

Si vous modifiez le nom d'hôte ou de domaine d'un serveur CUCM, il régénère tous les certificats à la fois sur ce serveur. La section de régénération de certificat a expliqué que la régénération de TVS.pem et CallManager.pem est une « mauvaise chose ».

Dans certains cas, la modification d'un nom d'hôte échoue et dans d'autres, elle fonctionne sans problème. Cette section les couvre tous et les relie à ce que vous savez déjà sur TVS et ITL de ce document.

Cluster à noeud unique avec uniquement ITL (attention, cette opération se brise sans préparation)

- Avec un déploiement de serveur Business Edition ou de serveur de publication uniquement, CallManager.pem et TVS.pem sont régénérés en même temps lorsque vous modifiez les noms d'hôte.
- Si le nom d'hôte est modifié sur un cluster à noeud unique sans utiliser au préalable le paramètre Rollback Enterprise traité ici, les téléphones ne peuvent pas vérifier le nouveau fichier ITL ou les fichiers de configuration par rapport à leur fichier ITL actuel.
- Les téléphones ne peuvent pas se connecter à TVS, car le certificat TVS n'est plus approuvé.
- Les téléphones affichent une erreur relative à « La vérification de la liste de confiance a échoué », aucune nouvelle modification de configuration n'est prise en compte et les URL de service sécurisé échouent.
- La seule solution si la précaution de l'étape 2 n'est pas prise en premier est de supprimer

manuellement l'ITL de chaque téléphone.

Cluster à noeud unique avec CTL et ITL (ce problème peut être temporairement interrompu, mais facilement résolu)

- Après avoir renommé les serveurs, réexécutez le client CTL. Le nouveau certificat CallManager.pem est alors placé dans le fichier CTL téléchargé par le téléphone.
- Les nouveaux fichiers de configuration, qui incluent les nouveaux fichiers ITL, peuvent être approuvés en fonction de la fonction CCM+TFTP dans le fichier CTL.
- Cela fonctionne car le fichier CTL mis à jour est approuvé sur la base d'une clé privée eToken USB qui reste la même.

Cluster multinoeud avec uniquement ITL (cela fonctionne généralement, mais peut être définitivement interrompu si cela est fait à la hâte)

- Étant donné qu'un cluster multinoeud comporte plusieurs serveurs TVS, chaque serveur peut voir ses certificats régénérés sans problème. Lorsque le téléphone reçoit cette nouvelle signature inhabituelle, il demande à un autre serveur TVS de vérifier le nouveau certificat de serveur.
- Deux problèmes principaux peuvent entraîner l'échec de cette opération :
  - Si tous les serveurs sont renommés et redémarrés en même temps, aucun des serveurs TVS n'est accessible avec des certificats connus lorsque les serveurs et les téléphones redémarrent.
  - Si un téléphone ne comporte qu'un seul serveur dans le groupe CallManager, les serveurs TVS supplémentaires ne font aucune différence. Reportez-vous au scénario « Cluster à noeud unique » pour résoudre ce problème ou ajoutez un autre serveur au groupe CallManager du téléphone.

Cluster multi-noeuds avec CTL et ITL (ne peut pas être définitivement rompu)

- Une fois que vous avez modifié les noms, le service TVS authentifie les nouveaux certificats.
- Même si tous les serveurs TVS sont indisponibles pour une raison quelconque, le client CTL peut toujours être utilisé afin de mettre à jour les téléphones avec les nouveaux certificats CallManager.pem CCM+TFTP.

TFTP centralisé

Lorsqu'un téléphone équipé d'un ITL démarre, il demande les fichiers suivants : CTLSEP<MAC Address>.tlv, ITLSEP<MAC Address>.tlv, et SEP<MAC Address>.cnf.xml.sgn.

Si le téléphone ne trouve pas ces fichiers, il demande ITLFile.tlv et CTLFile.tlv, qu'un serveur TFTP centralisé fournit à tout téléphone qui le demande.

Avec le protocole TFTP centralisé, il existe un cluster TFTP unique qui pointe vers un certain nombre d'autres sous-clusters.

Cette opération est souvent effectuée parce que les téléphones sur plusieurs clusters CUCM partagent la même étendue DHCP et doivent donc avoir le même serveur TFTP DHCP Option

150.

Tous les téléphones IP pointent vers le cluster TFTP central, même s'ils s'enregistrent vers d'autres clusters. Ce serveur TFTP central interroge les serveurs TFTP distants chaque fois qu'il reçoit une demande pour un fichier qu'il ne trouve pas.

En raison de cette opération, le protocole TFTP centralisé ne fonctionne que dans un environnement homogène ITL.

Tous les serveurs doivent exécuter CUCM version 8.x ou ultérieure, ou tous les serveurs doivent exécuter des versions antérieures à la version 8.x.

Si un fichier ITLFile.tlv est présenté à partir du serveur TFTP centralisé, les téléphones ne font confiance à aucun fichier du serveur TFTP distant car les signatures ne correspondent pas.

Cela se produit dans un mélange hétérogène. Dans un mélange homogène, le téléphone demande ITLSEP<MAC>.tlv qui est extrait du cluster distant approprié.

Dans un environnement hétérogène avec un mélange de clusters antérieurs à la version 8.x et à la version 8.x, le « Préparer le cluster pour la restauration vers la version 8.0 » doit être activé sur le cluster de la version 8.x comme décrit dans l'ID de bogue Cisco CSCto87262 .

Configurez les paramètres d'URL de téléphone sécurisé avec HTTP au lieu de HTTPS. Cela désactive efficacement les fonctions ITL sur le téléphone.

Forum aux questions

Puis-je désactiver SBD ?

Vous ne pouvez désactiver SBD que si SBD et ITL fonctionnent actuellement.

SBD peut être temporairement désactivé sur les téléphones avec le paramètre d'entreprise « Préparer le cluster pour la restauration à une version antérieure à 8.0 » et en configurant les « Paramètres d'URL de téléphone sécurisés » avec HTTP au lieu de HTTPS.

Lorsque vous définissez le paramètre Rollback, il crée un fichier ITL signé avec des entrées de fonction vides.

Le fichier ITL « vide » est toujours signé, de sorte que le cluster doit être dans un état de sécurité entièrement fonctionnel avant que ce paramètre puisse être activé.

Une fois que ce paramètre est activé et que le nouveau fichier ITL avec des entrées vides est téléchargé et vérifié, les téléphones acceptent tout fichier de configuration, quelle que soit la personne qui l'a signé.

Il n'est pas recommandé de laisser le cluster dans cet état, car aucune des trois fonctions précédemment mentionnées (fichiers de configuration authentifiés, fichiers de configuration chiffrés et URL HTTPS) n'est disponible.

Puis-je facilement supprimer le fichier ITL de tous les téléphones une fois que CallManager.pem est perdu ?

Il n'existe actuellement aucune méthode permettant de supprimer toutes les ITL d'un téléphone fourni à distance par Cisco. C'est pourquoi les procédures et interactions décrites dans ce document sont si importantes à prendre en compte.

Il existe actuellement une amélioration non résolue du bogue Cisco ayant l'ID CSCto47052 qui demande cette fonctionnalité, mais elle n'a pas encore été implémentée.

Dans l'intervalle, une nouvelle fonctionnalité a été ajoutée via l'ID de bogue Cisco CSCts01319 qui peut permettre au centre d'assistance technique de Cisco (TAC) de revenir à l'ITL précédemment approuvé s'il est toujours disponible sur le serveur.

Cela ne fonctionne que dans certains cas où le cluster est sur une version avec ce correctif de défaut, et où l'ITL précédent existe dans une sauvegarde stockée dans un emplacement spécial sur le serveur.

Affichez le défaut pour voir si votre version a le correctif. Contactez le TAC Cisco afin de suivre la procédure de récupération potentielle expliquée dans le défaut.

Si la procédure précédente n'est pas disponible, vous devez appuyer manuellement sur les boutons du téléphone pour supprimer le fichier ITL. C'est le compromis qui est fait entre la sécurité et la facilité d'administration. Pour que le fichier ITL soit réellement sécurisé, il ne doit pas être facile à supprimer à distance.

Même en appuyant sur des boutons scriptés avec des objets XML SOAP (Simple Object Access Protocol), l'ITL ne peut pas être supprimé à distance.

En effet, à ce stade, l'accès TVS (et donc l'accès à l'URL d'authentification sécurisée pour valider les objets push de bouton XML SOAP entrants) n'est pas fonctionnel.

Si l'URL d'authentification n'est pas configurée comme sécurisée, il est possible d'exécuter un script à l'aide des touches pour supprimer une ITL, mais ce script n'est pas disponible auprès de Cisco.

D'autres méthodes permettant d'écrire des scripts à distance sans utiliser l'URL d'authentification sont éventuellement disponibles auprès d'un tiers, mais ces applications ne sont pas fournies par Cisco.

La méthode la plus fréquemment utilisée pour supprimer l'ITL est une diffusion par e-mail à tous les utilisateurs du téléphone qui leur indique la séquence de touches.

Si l'accès aux paramètres est défini sur Restreint ou Désactivé, le téléphone doit être réinitialisé en usine, car les utilisateurs n'ont pas accès au menu Paramètres du téléphone.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.