Sécurité CUCM par défaut et exécution et dépannage ITL

Contenu

Introduction

Ce document décrit la Sécurité par la caractéristique (SBD) par défaut des versions 8.0 et ultérieures de Cisco Unified Communications Manager (CUCM). Ce document sert de supplément à la Sécurité officielle par les documents par défaut, et fournit les conseils opérationnels de l'information et de dépannage pour aider des administrateurs et pour soulager le processus de dépannage.

Informations générales

La version 8.0 et ultérieures CUCM introduit la caractéristique SBD, qui se compose des fichiers de la liste de confiance d'identité (ITL) et du service de vérification de confiance (TV). Chaque

batterie CUCM utilise maintenant la Sécurité basée sur ITL automatiquement. Il y a un compromis entre la Sécurité et la facilité d'utilisation/facilité de la gestion dont les administrateurs doivent se rendre compte avant qu'ils apportent certaines modifications à une batterie de la version 8.0 CUCM.

C'est une bonne idée de se familiariser avec ces principaux concepts de SBD : Article principal asymétrique de Wikipedia d'article et d'infrastructure de clé publique de Wikipedia de chiffrement.

Aperçu SBD

Cette section fournit une présentation rapide de exactement ce que le SBD fournit. Pour de pleins détails techniques de chaque fonction, voyez le SBD section détailler et d'information de dépannage.

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

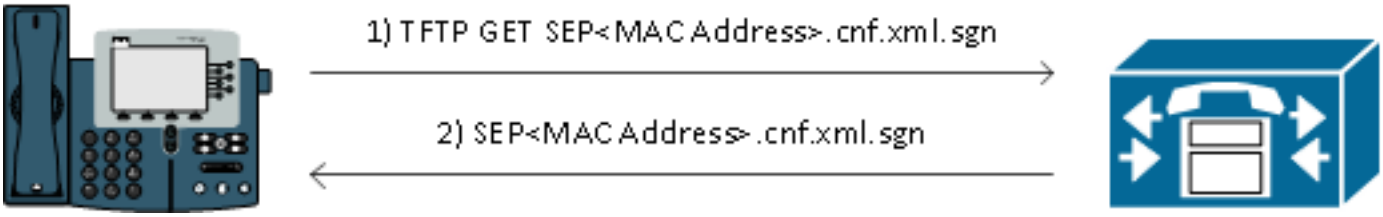
- Authentification par défaut des fichiers téléchargés TFTP (configuration, paramètre régional, ringlist) cette utilisation une clé de signature
- Cryptage facultatif des fichiers de configuration TFTP qui utilisent une clé de signature
- Délivrez un certificat la vérification pour les connexions téléphone-initiées HTTPS qui utilisent une mémoire distante de confiance de certificat sur CUCM (les TV)

Ce document fournit un aperçu de chacune de ces fonctions.

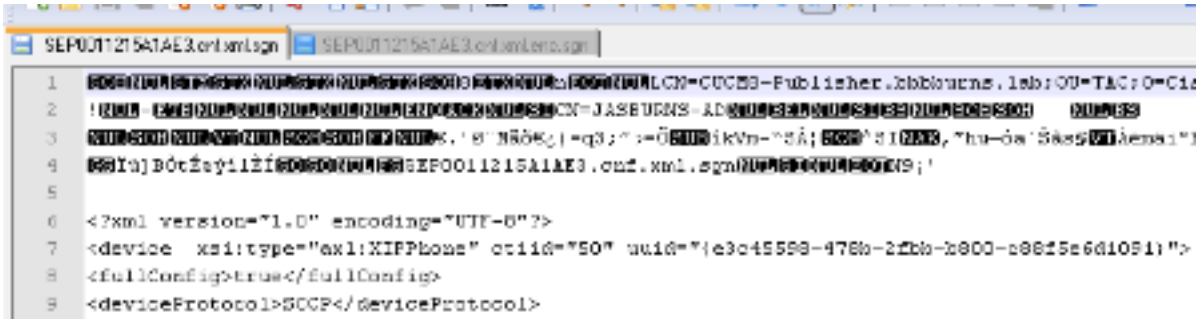
Authentification de téléchargement TFTP

Quand une liste de confiance de certificat (CTL) ou le fichier ITL est présent, le téléphone IP demande un fichier de configuration signé TFTP du serveur CUCM TFTP. Ce fichier permet au téléphone pour vérifier que le fichier de configuration est provenu une source sûre. Avec des fichiers CTL/ITL actuels aux téléphones, des fichiers de configuration doivent être signés par un serveur de confiance TFTP. Le fichier est texte brut sur le réseau tandis qu'il est transmis, mais est livré avec une signature spéciale de vérification.

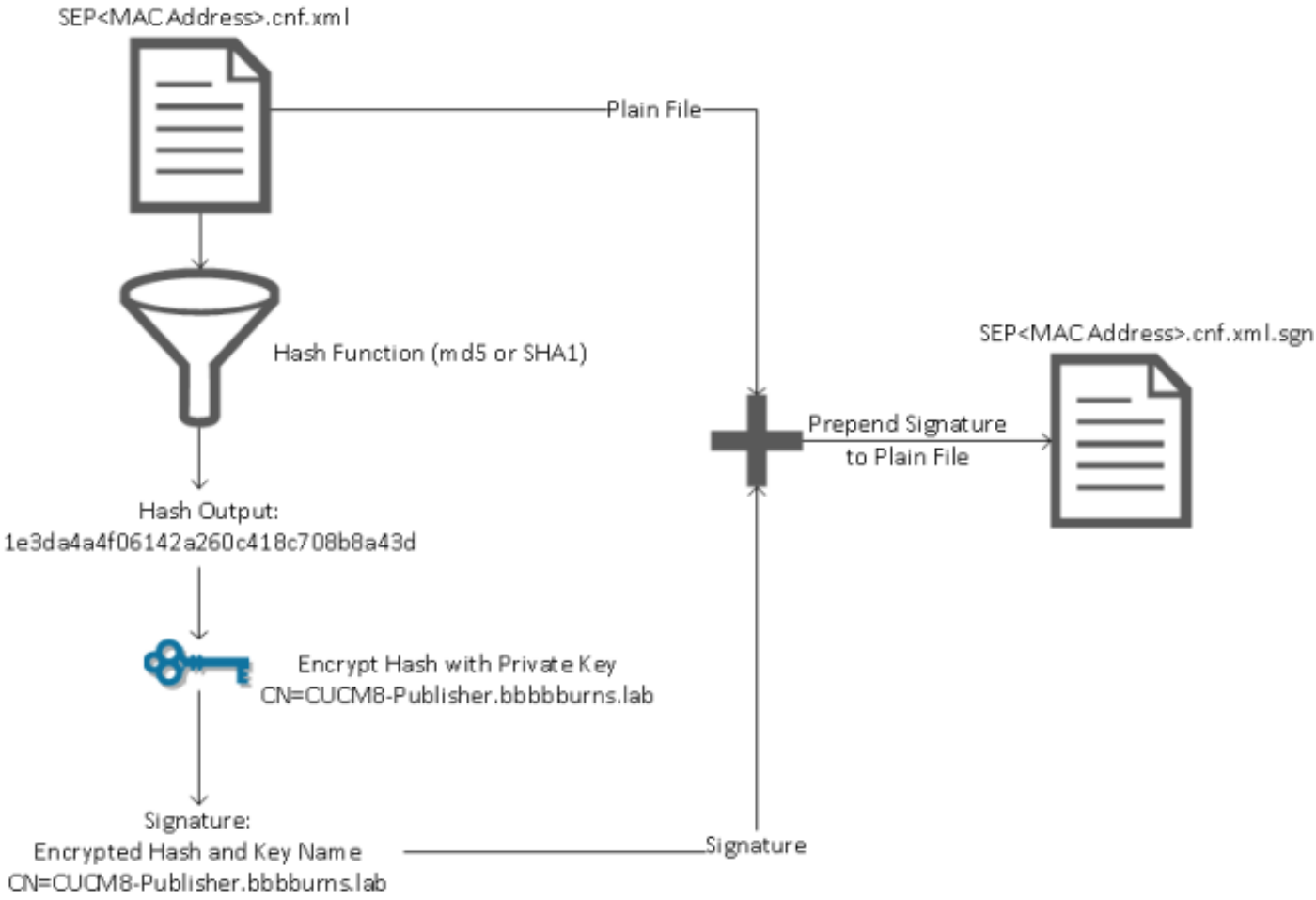
Le téléphone demande **SEPT *< adresse MAC >.cnf.xml.sgn* afin de** recevoir le fichier de configuration avec la signature spéciale. Ce fichier de configuration est signé par la clé privée TFTP qui correspond à CallManager.pem à la page du système d'exploitation de Gestion de certificat de gestion (de SYSTÈME D'EXPLOITATION).



Le fichier signé a une signature au supérieur afin d'authentifier le fichier, mais est autrement en texte brut XML. L'image ci-dessous prouve que le signataire du fichier de configuration est **CN=CUCM8-Publisher.bbbburns.lab** ce qui consécutivement est signé par **CN=JASBURNS-AD**. Ceci signifie que les besoins de téléphone de vérifier la signature de **CUCM8-Publisher.bbbburns.lab** contre le fichier ITL avant ce fichier de configuration est reçus.

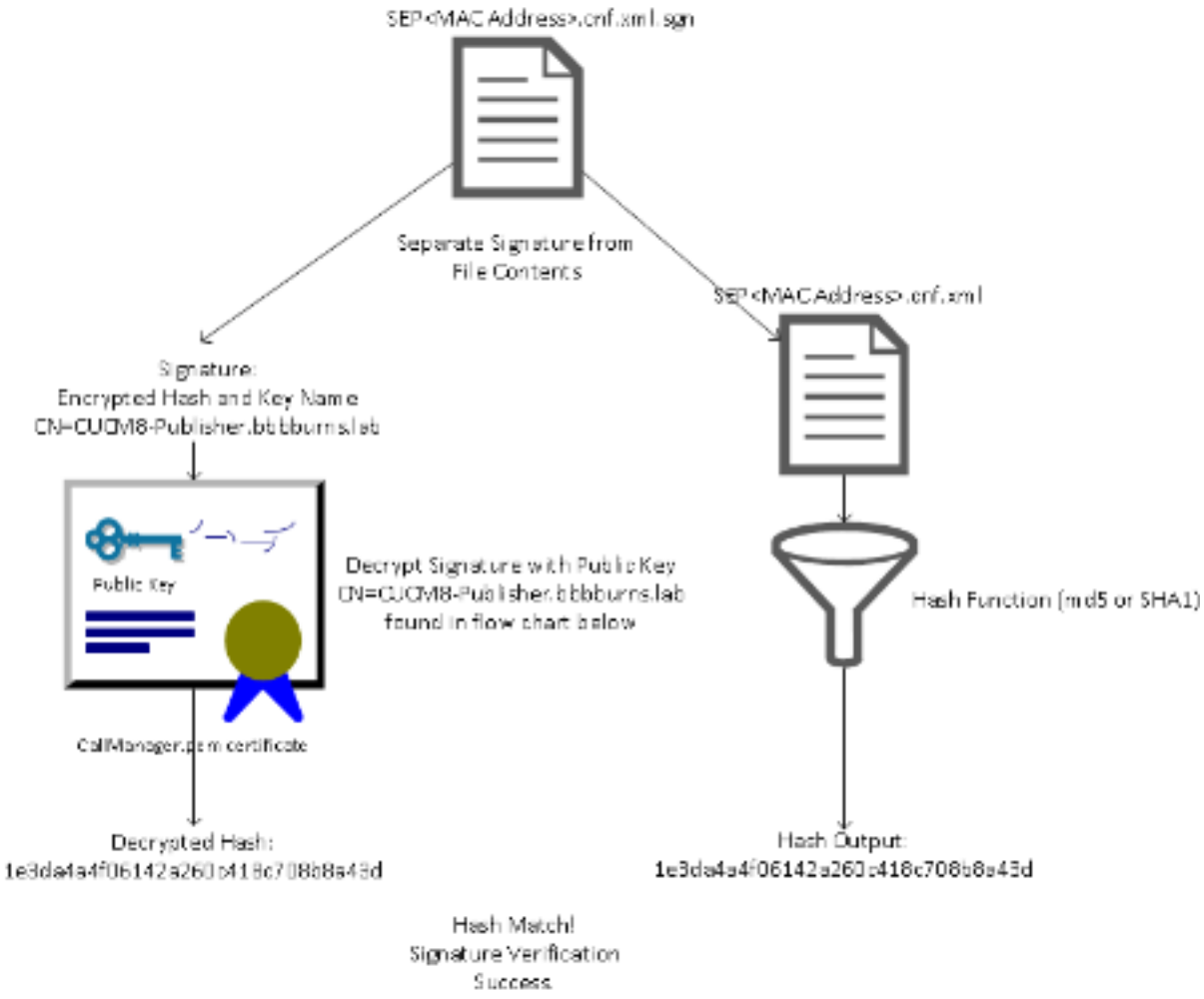


Voici un diagramme qui affiche comment la clé privée est utilisée avec un algorithme de condensé de message (MD)5 ou Secure Hash Algorithm (fonction d'informations parasites SHA)1 afin de créer le fichier signé.



La vérification de signature renverse ce processus par l'utilisation de la clé publique cette des correspondances afin de déchiffrer les informations parasites. Si hache la correspondance, elle affiche :

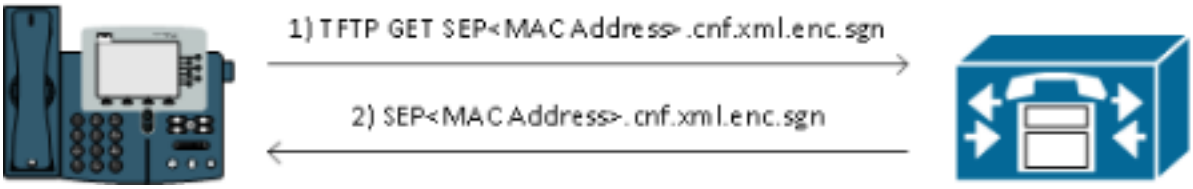
- Ce fichier n'a pas été modifié en transit.
- Ce fichier provient l'interlocuteur répertorié dans la signature, puisque quelque chose déchiffré avec succès avec la clé publique doit avoir été chiffré avec la clé privée.



Cryptage de fichier de configuration TFTP

Si le cryptage facultatif de configuration TFTP est activé dans le profil associé de degré de sécurité de téléphone, le téléphone demande un fichier de configuration chiffré. Ce fichier est signé avec la clé privée TFTP et chiffré avec une clé symétrique permutée entre le téléphone et le CUCM (référez-vous au guide de Sécurité de Cisco Unified Communications Manager, libèrent 8.5(1) pour les détails complets) de sorte que son contenu ne puisse pas être lu avec un analyseur réseau à moins que l'observateur ait les clés nécessaires.

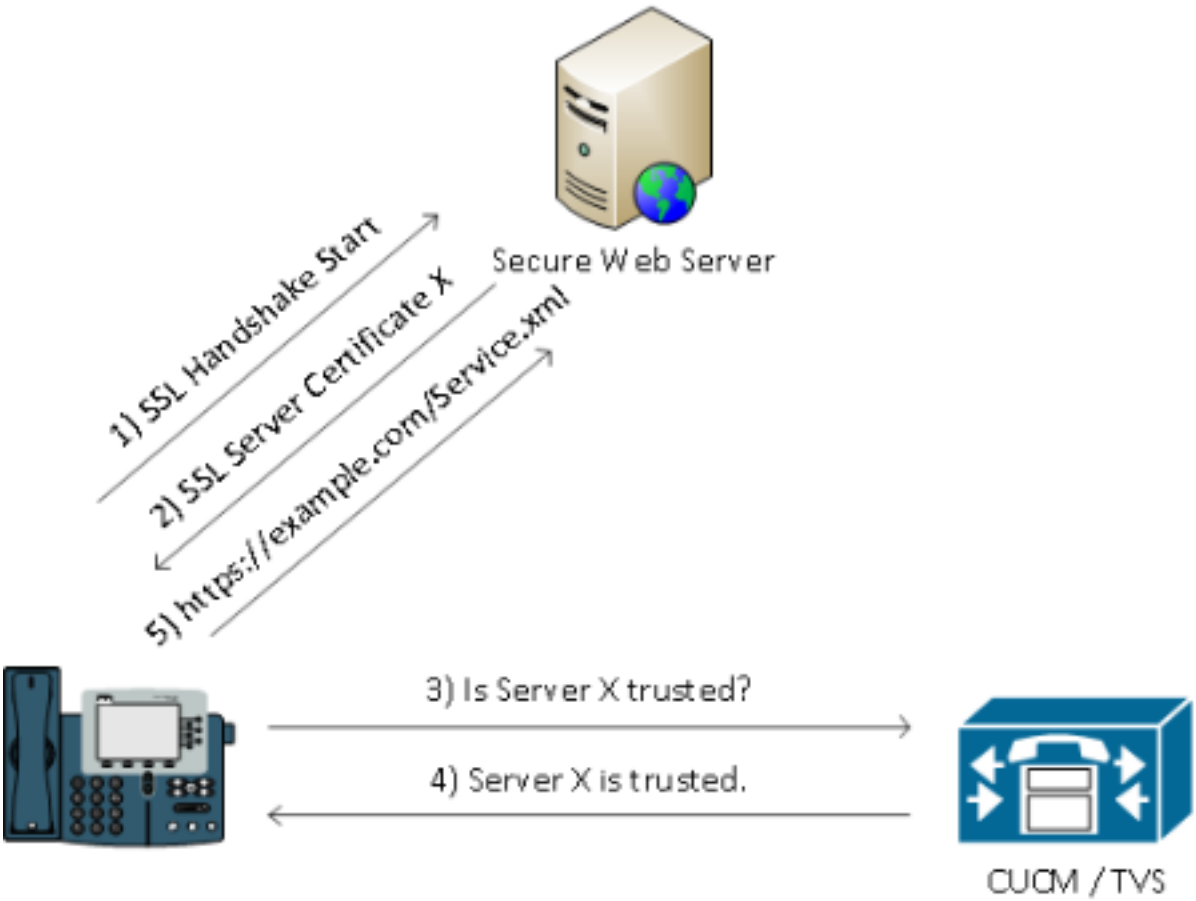
Le téléphone demande **SEPT** ***< adresse MAC >.cnf.xml.enc.sgn*** **afin d'**obtenir le fichier crypté signé.



Le fichier de configuration chiffré a la signature au début aussi bien, mais il n'y a aucune donnée de texte brut après, seulement les données cryptées (caractères binaires déformés dans cet éditeur de texte). L'image prouve que le signataire est le même que dans l'exemple précédent, ainsi ce signataire doit être présent dans le fichier ITL avant que le téléphone reçoive le fichier. De plus, les clés de déchiffrement doivent être correctes avant que le téléphone puisse indiquer le contenu du fichier.

Service de vérification de confiance (certificat distant et vérification de signature)

Les Téléphones IP contiennent une quantité limitée de mémoire, et il peut également y avoir un grand nombre de téléphones à gérer dans un réseau. CUCM agit en tant que mémoire distante de confiance par l'intermédiaire des TV de sorte qu'une pleine mémoire de confiance de certificat ne doive pas être placée sur chaque téléphone IP. N'importe quand le téléphone ne peut pas vérifier une signature ou certificat par l'intermédiaire des fichiers CTL ou ITL, il demande au serveur TV la vérification. Il est plus facile gérer cette mémoire centrale de confiance que si la mémoire de confiance était présente sur tous les Téléphones IP.



Détail et information de dépannage SBD

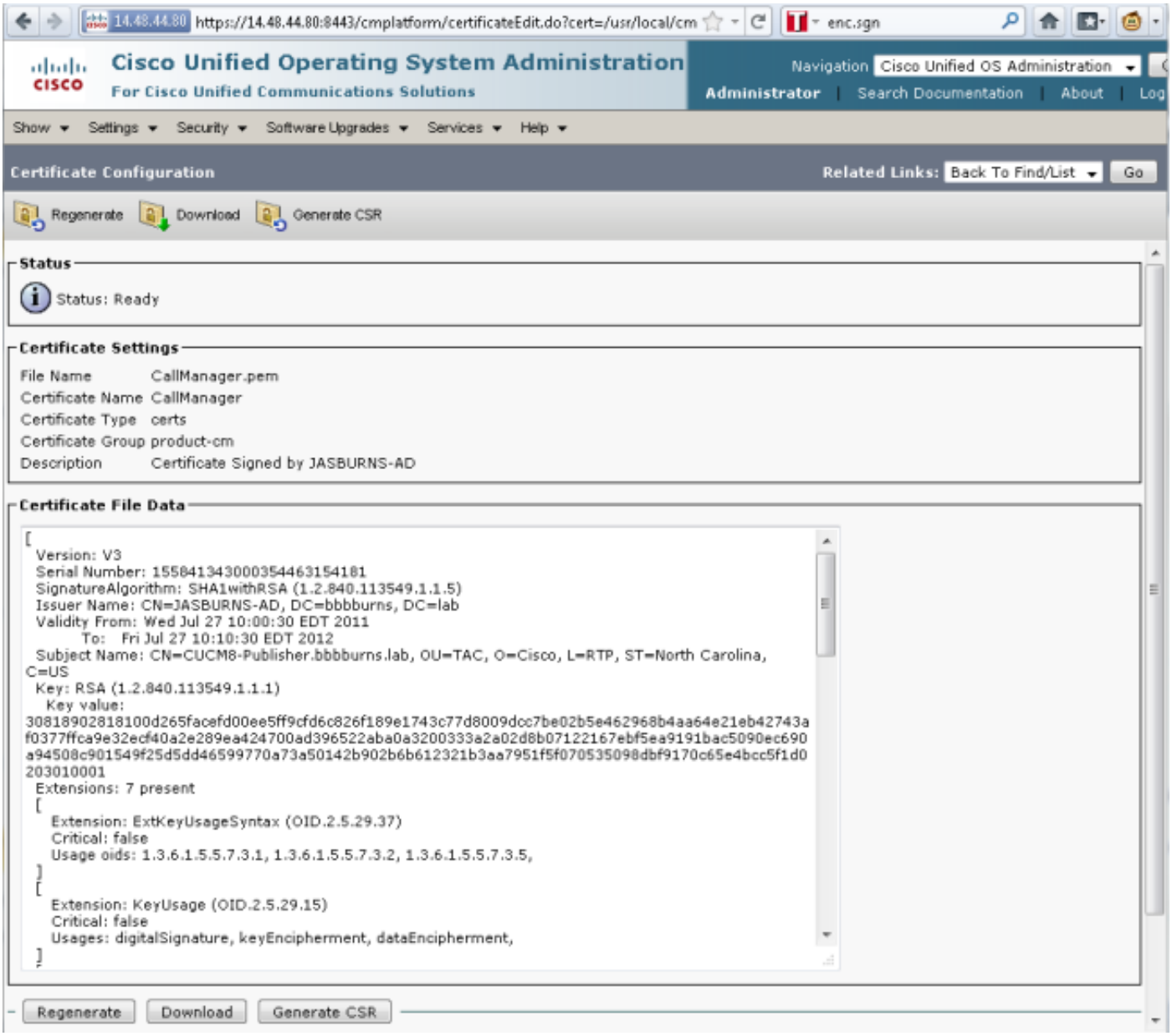
Cette section détaille le processus SBD.

Fichiers et Certificats ITL actuels sur CUCM

D'abord, il y a un certain nombre de fichiers qui doivent être présents sur le serveur CUCM lui-même. La partie la plus importante est le certificat TFTP et la clé privée TFTP. Le certificat TFTP se trouve sous la **gestion de SYSTÈME D'EXPLOITATION > la Gestion de Sécurité > de certificat > le CallManager.pem.**

Le serveur CUCM utilise les clés privées et publiques du certificat CallManager.pem pour le service TFTP (aussi bien que pour le service de Cisco Call manager (CCM)). L'image prouve que le certificat CallManager.pem est fourni à CUCM8-publisher.bbbburns.laband **signé** par JASBURNS-AD**. Tous les** fichiers de configuration TFTP sont signés par la clé privée ci-dessous.

Tous les téléphones peuvent employer la clé publique TFTP dans le certificat CallManager.pem afin de déchiffrer n'importe quel fichier chiffré avec la clé privée TFTP, aussi bien que vérifier n'importe quel fichier signé avec la clé privée TFTP.



En plus de la clé privée de certificat CallManager.pem, le serveur CUCM enregistre également un fichier ITL qui est présenté aux téléphones. **La commande de** showitl affiche le plein contenu de ce fichier ITL par l'intermédiaire de l'accès de Protocole Secure Shell (SSH) au SYSTÈME D'EXPLOITATION CLI de serveur CUCM.

Cette section décompose le fichier ITL pièce par pièce, parce qu'elle a un certain nombre

d'importants composants que le téléphone utilise.

La première partie est les informations de signature. Même le fichier ITL est un fichier signé. Cette sortie prouve qu'elle est signée par la clé privée TFTP qui est associée avec le certificat précédent CallManager.pem.

```
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 chacune contiennent leur but à l'intérieur d'un paramètre de **fonction** spéciale. La première fonction est le jeton de Sécurité d'administrateur système. C'est 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 prochaine fonction est CCM+TFTP. C'est de nouveau la clé publique TFTP qui sert à authentifier et déchiffrer les fichiers de configuration téléchargés TFTP.

```
        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 prochaine fonction est des TV. Il y a une entrée pour la clé publique de chaque serveur TV à laquelle le téléphone se connecte. Ceci permet au téléphone pour établir une session de Secure Sockets Layer (SSL) au serveur TV.

```
        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 fonction finale incluse dans le fichier ITL est la fonction de proxy d'autorité de certification (CAPF). Ce certificat permet aux téléphones pour établir une connexion sécurisée au service CAPF sur le serveur CUCM de sorte que le téléphone puisse installer ou mettre à jour a localement - le certificat significatif (LSC). Ce processus sera couvert dans un autre document qui doit être libéré encore.

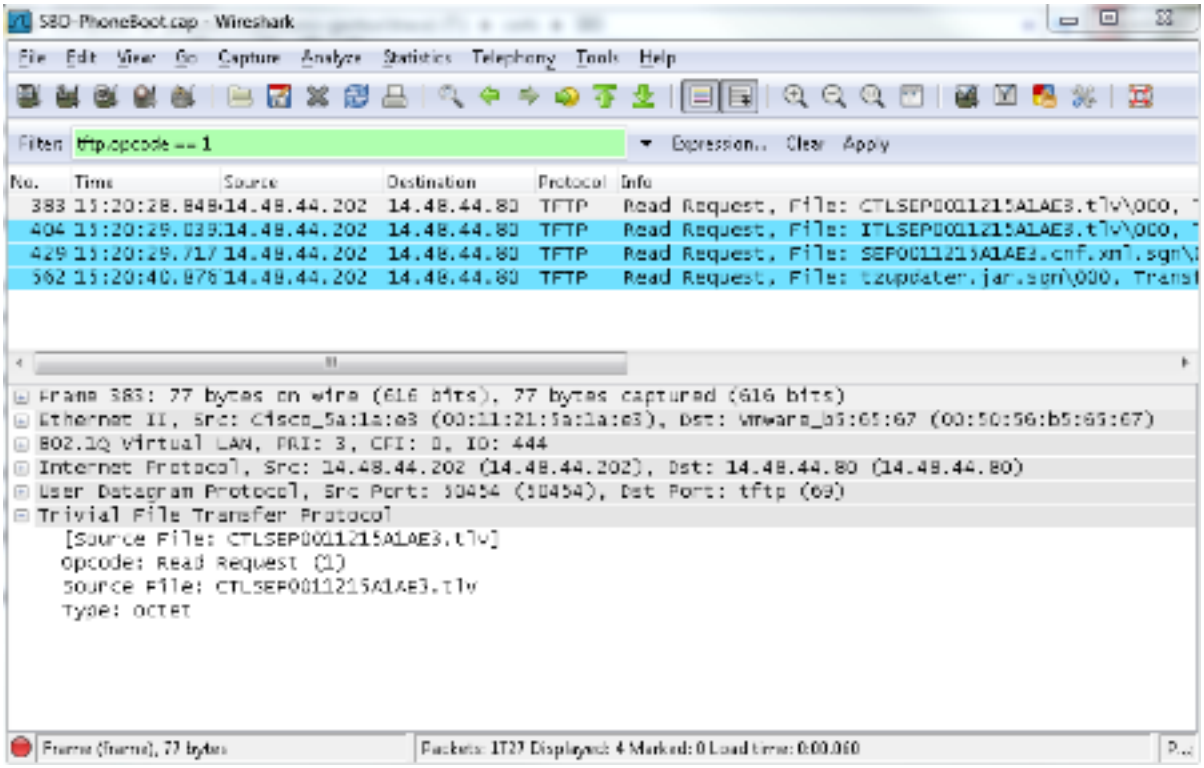
```
        ITL Record #:4
                  ----
BYTEPOS TAG             LENGTH  VALUE
------- ---             ------  -----
1       RECORDLENGTH    2       455
2       DNSNAME         2
3       SUBJECTNAME     61      CN=CAPF-9c4cba7d;
                                OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
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.
```

Les couvertures de section suivante exactement ce qui se produit quand un téléphone démarre.

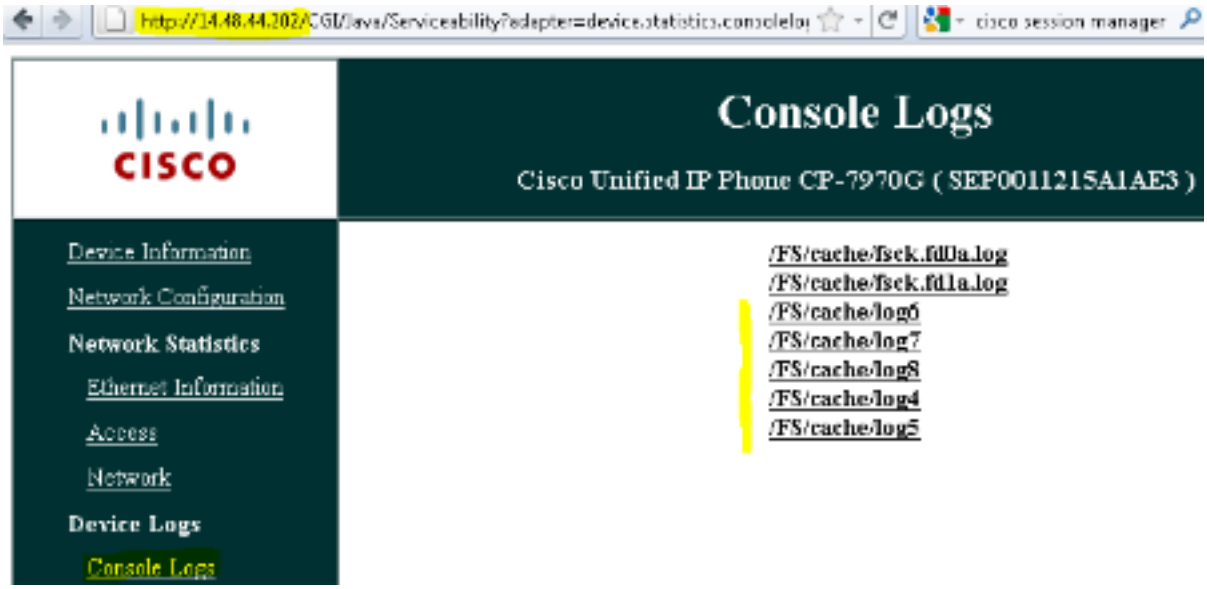
Le téléphone télécharge l'ITL et le fichier de configuration

Après que le téléphone démarre et obtienne une adresse IP aussi bien que l'adresse d'un serveur TFTP, il demande le CTL et les fichiers ITL d'abord.

Cette capture de paquet affiche une demande de téléphone du fichier ITL. Si vous filtrez sur le **== 1 tftp.opcode, vous** voyez chaque TFTP lire la demande du téléphone :



Puisque le téléphone a reçu des fichiers CTL et ITL du TFTP avec succès, le téléphone demande un fichier de configuration signé. Les logs de console de téléphone qui affichent ce comportement sont fournis par l'interface web du téléphone :



D'abord le téléphone demande un fichier CTL, 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 :

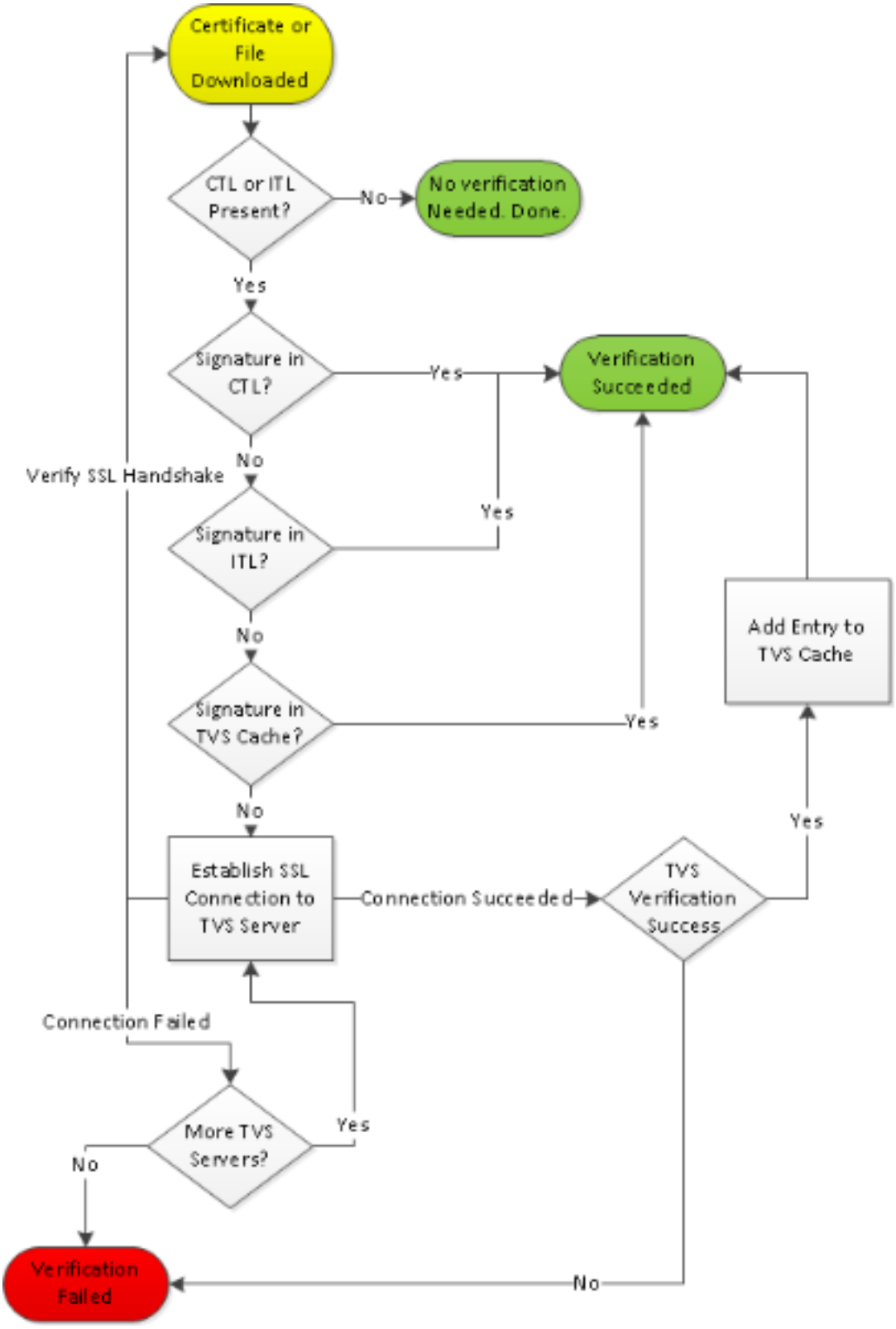
```
868: NOT 09:13:17.860613 TFTP: [28]:Requesting ITLSEP0011215A1AE3.tlv from
14.48.44.80
869: NOT 09:13:17.875059 TFTP: [28]:Finished --> rcvd 5438 bytes
```

Le téléphone vérifie l'ITL et le fichier de configuration

Après que le fichier ITL soit téléchargé, il doit être vérifié. Il y a un certain nombre de déclarer qu'un téléphone peut être dedans en ce moment, ainsi ce document les couvre tous.

- Le téléphone présent n'a aucun fichier CTL ou ITL ou l'ITL est vide en raison de la **batterie de préparation pour le repositionnement au** paramètre **pré 8.0**. dans cet état, le téléphone fait confiance aveugle au prochain fichier CTL ou ITL téléchargé et utilise cette signature dorénavant.
- Le téléphone n'a déjà un CTL mais aucune ITL. Dans cet état, le téléphone fait confiance seulement à une ITL s'il peut être vérifié par la fonction CCM+TFTP dans le fichier CTL.
- Le téléphone a déjà un CTL et un fichier ITL. Dans cet état, le téléphone vérifie que récemment les fichiers téléchargés apparient la signature dans le serveur CTL, ITL, ou TV.

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 l'ITL et les fichiers CTL. Le téléphone a déjà un CTL et l'ITL ainsi il a simplement vérifié contre eux et fonde la signature correcte.

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

Puisque le téléphone a téléchargé les fichiers CTL et ITL, à partir de là il demande SEULEMENT les fichiers de configuration signés. Ceci illustre que la logique du téléphone est de déterminer que le serveur TFTP est sécurisé, basé sur la présence de CTL et d'ITL, et puis de 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 que le fichier de configuration signé est téléchargé, le téléphone doit l'authentifier contre la fonction pour CCM+TFTP à l'intérieur de l'ITL :

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

Le téléphone entre en contact avec des TV pour le certificat inconnu

Le fichier ITL fournit une fonction TV qui contient le certificat du service TV qui fonctionne sur le port TCP 2445 de serveur CUCM. Les TV fonctionne sur tous les serveurs où le service de CallManager est lancé. Le service TFTP CUCM utilise le groupe configuré de CallManager afin d'établir une liste de serveurs TV que le téléphone devrait contacter sur le fichier de configuration de téléphone.

Quelques laboratoires utilisent seulement un serveur simple CUCM. Dans une batterie du multi-noeud CUCM, il peut y avoir jusqu'à trois entrées TV pour un téléphone, un pour chaque CUCM dans le groupe CUCM du téléphone.

Cet exemple affiche ce qui se produit quand le **bouton répertoires** sur le téléphone IP est appuyé sur. L'URL de répertoires est configuré pour HTTPS, ainsi le téléphone est présenté avec le certificat de Web de Tomcat du serveur de répertoires. Ce certificat de Web de Tomcat (tomcat.pem dans la gestion de SYSTÈME D'EXPLOITATION) n'est pas chargé dans le téléphone, ainsi le téléphone doit entrer en contact avec des TV afin d'authentifier le certificat.

Référez-vous au diagramme d'aperçu précédent TV pour une description de l'interaction. Voici le point de vue de log de console de téléphone :

D'abord vous trouvez l'URL de 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
```

C'est une session de HTTP sécurisé SSL/Transport Layer Security (TLS) qui exige la 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 le CTL. Alors le téléphone regarde les fonctions dans le fichier ITL afin de voir s'il trouve une correspondance. Ce message d'erreur indique le « CERT HTTPS pas dans CTL, » qui signifie « que la certification ne peut pas être trouvée dans le 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>
```

Après que le contenu direct du fichier CTL et ITL soit vérifié le certificat, la prochaine chose les contrôles de téléphone est le cache TV. Ceci est fait afin de réduire le trafic réseau si le téléphone a récemment demandé au serveur TV le même certificat. Si le certificat HTTPS n'est pas trouvé dans le cache de téléphone, vous pouvez établir une connexion TCP au serveur TV elle-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.
```

Souvenez-vous que la connexion aux TV elle-même est SSL/TLS (HTTP sécurisé, ou HTTPS), ainsi c'est également un certificat qui doit être authentifié contre l'ITL d'ot CTL. Si tout va correctement, le certificat de serveur TV devrait être trouvé dans la fonction TV du fichier ITL. Voir l'ITL #3 record dans le fichier ITL d'exemple 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>
```

Succès ! Le téléphone a maintenant une connexion sécurisée au serveur TV. L'étape suivante est de demander au serveur TV « bonjour, font confiance I ce certificat de serveur de répertoires ? »

Cet exemple affiche la réponse à cette question - une réponse de 0 ce qui signifie le succès (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
```

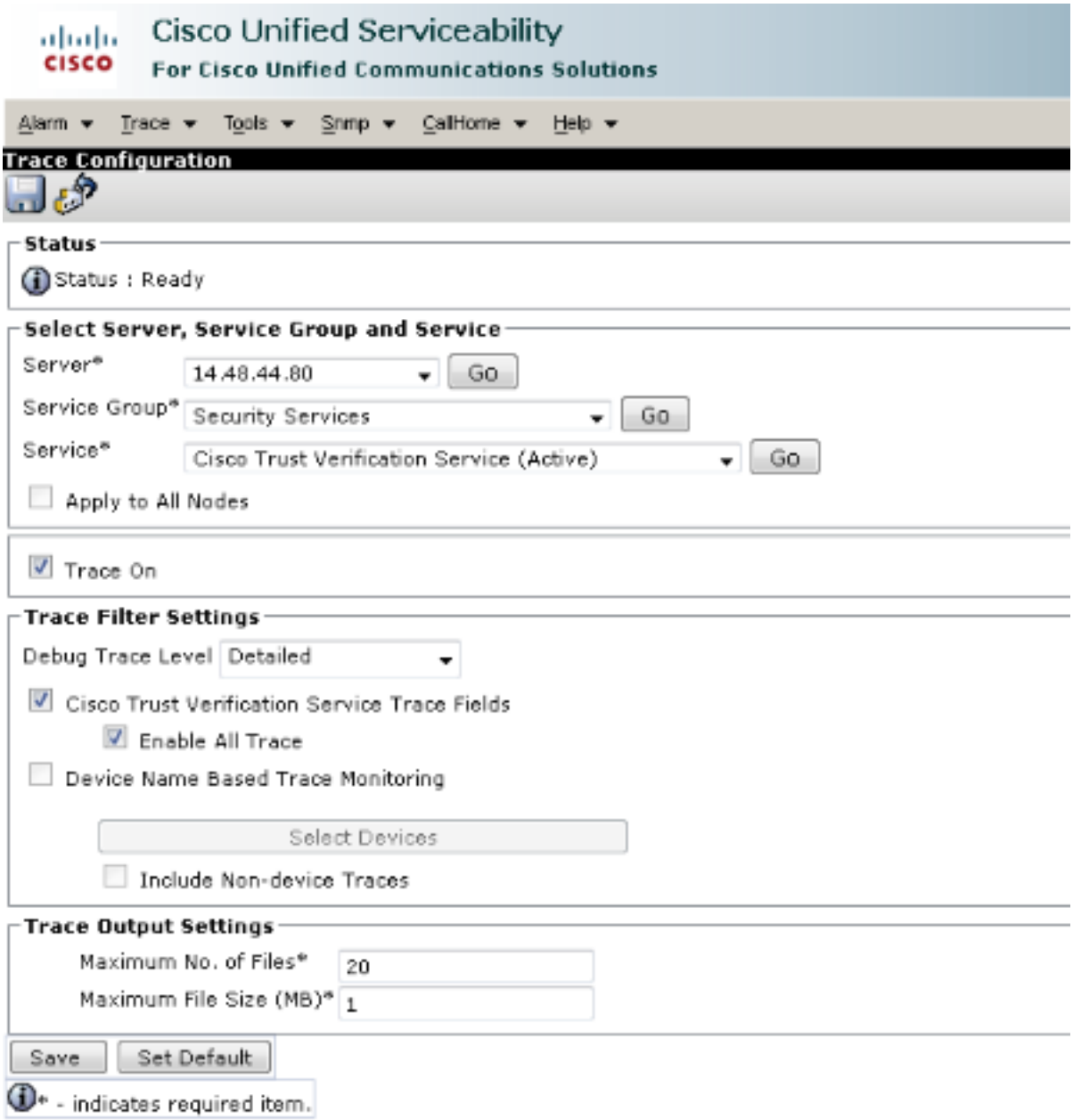
Puisqu'il y a une réponse réussie des TV, les résultats pour ce certificat sont enregistrés dans le cache. Ceci signifie que, si vous appuyez sur le **bouton répertoires** de nouveau dans les 86,400 secondes suivantes, vous n'avez pas besoin de contacter le serveur TV afin de vérifier le certificat. Vous pouvez simplement accéder au cache local.

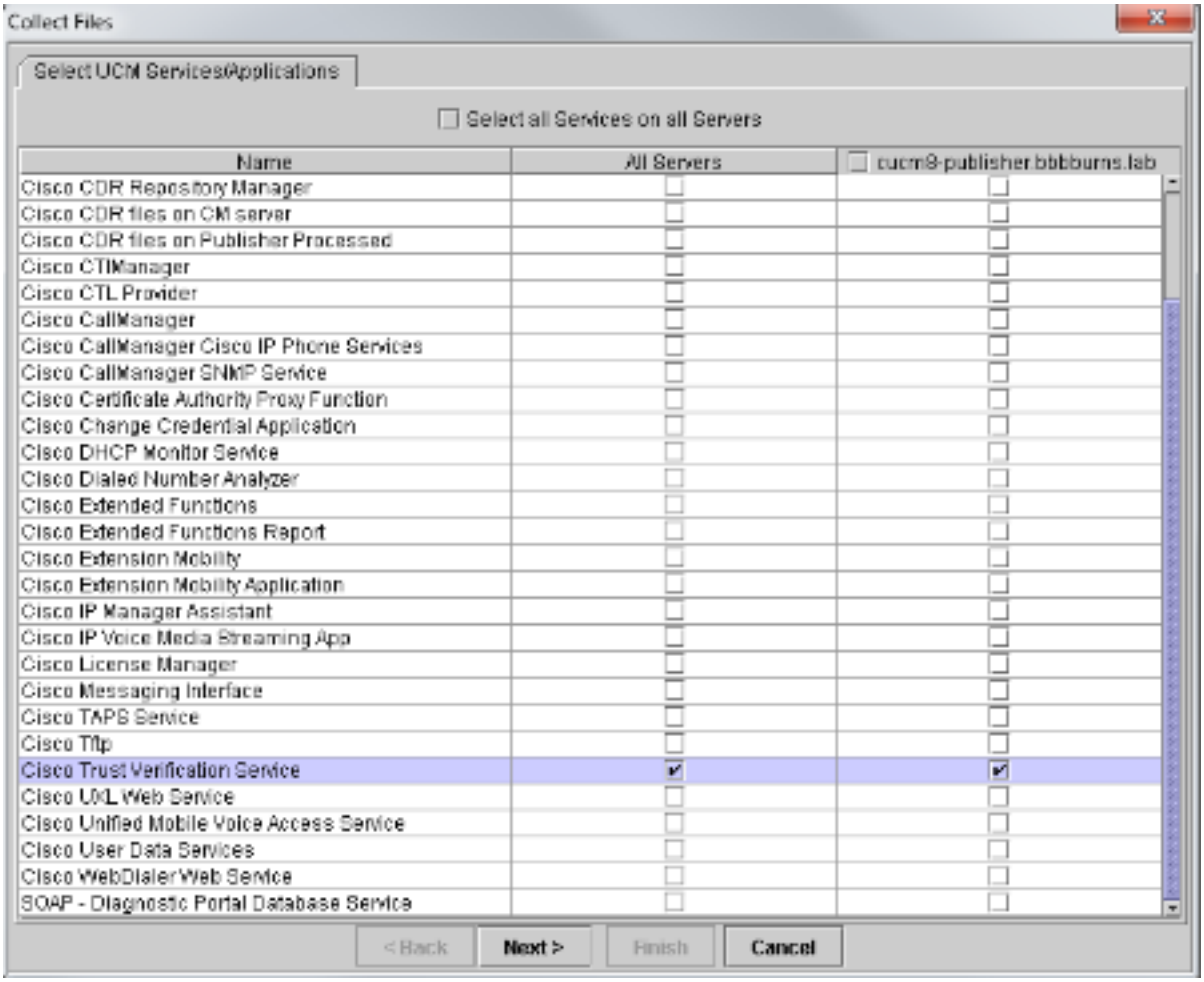
```
1279: NOT 15:20:59.837086 SECD: saveCertToTvsCache: Saving certificate
in TVS cache with default time-to-live value: 86400 seconds
1287: ERR 15:20:59.859993 SECD: Authenticated the HTTPS conn via TVS
```

En conclusion, vous vérifiez que votre connexion au serveur de 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 produit sur le serveur CUCM où les TV fonctionne. Vous pouvez collecter des logs TV avec l'outil de suivi en temps réel de Cisco Unified (RTMT).





Les logs CUCM TV prouvent que vous prise de contact SSL avec le téléphone, le téléphone demande des TV sur le certificat de Tomcat, puis des TV répond pour indiquer que le certificat est apparié dans la mémoire de certificat TV.

```
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
```

La mémoire de certificat TV est une liste de tous les Certificats contenus sur la page Web d'**Administration > Certificate Management de SYSTÈME D'EXPLOITATION**.

Vérifiez manuellement que l'ITL de téléphone apparie l'ITL CUCM

Une fausse idée commune vue tandis que le dépannage concerne la tendance de supprimer le fichier ITL avec l'espoir qu'il résoudra un problème de vérification de fichier. Parfois la suppression de fichier ITL est exigée, mais il pourrait y a une meilleure manière.

Le fichier ITL doit seulement être supprimé quand TOUTES ces conditions sont remplies.

- La signature du fichier ITL au téléphone n'apparie pas la signature du fichier ITL sur le serveur

cm TFTP.
- La signature TV dans le fichier ITL n'apparie pas le certificat présenté par des TV.
- Le téléphone affiche que la « vérification » quand il des attemps n'a pas téléchargé le fichier ou des fichiers de configuration ITL.
- Aucune sauvegarde n'existe de la vieille clé privée TFTP.

Voici comment vous vérifiez les deux premiers de ces conditions.

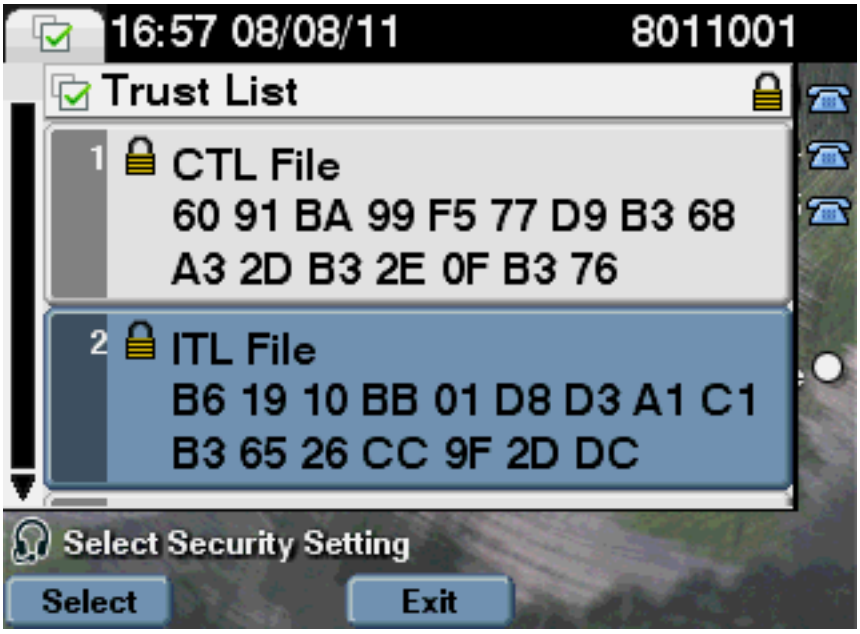
D'abord, vous pouvez comparer la somme de contrôle du fichier ITL actuel sur CUCM au fichier ITL de somme de contrôle du téléphone. Il n'y a actuellement aucune manière de regarder le MD5sum du fichier ITL sur CUCM de CUCM lui-même jusqu'à ce que vous exécutiez une version avec la difficulté pour cet ID de bogue Cisco CSCto60209.

Dans l'intervalle, exécutez ceci avec vos programmes GUI ou CLI de favori :

```
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
```

Ceci prouve que le MD5sum du fichier ITL dans CUCM est **b61910bb01d8d3a1c1b36526cc9f2ddc**.

Maintenant vous pouvez regarder le téléphone lui-même afin de déterminer les informations parasites du fichier ITL chargé là : **Configurations > liste de configuration de sécurité > de confiance**.

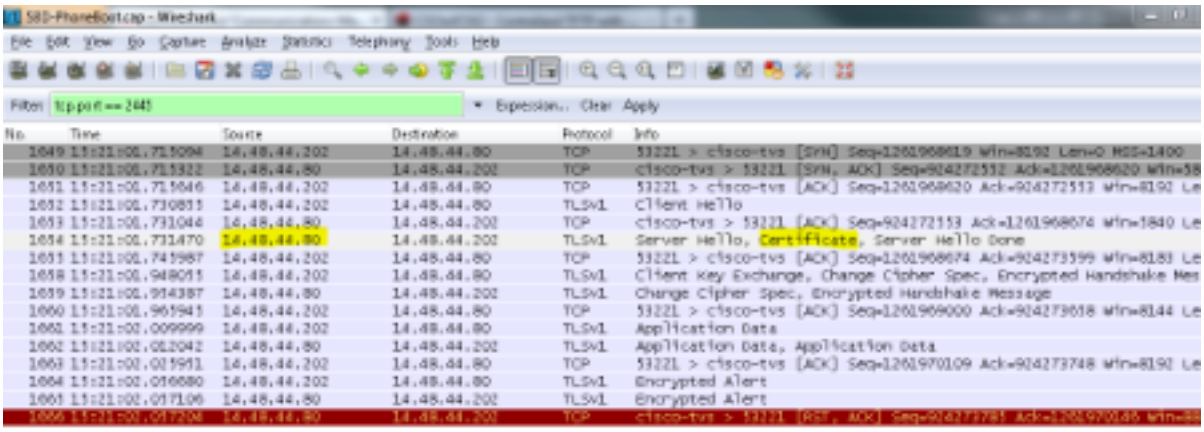


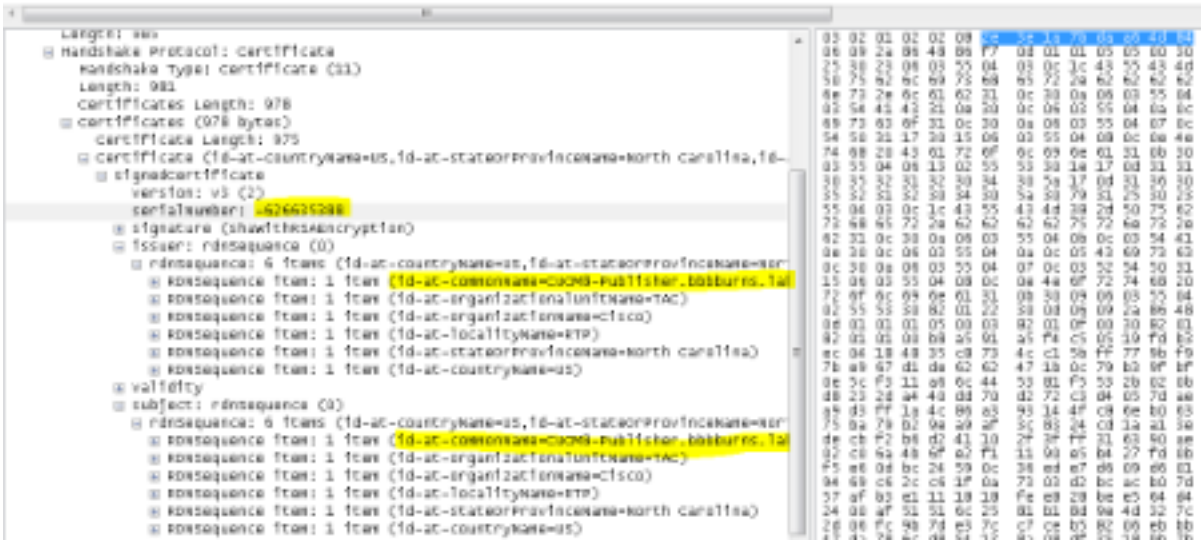
Ceci affiche à cela la correspondance MD5sums. Ceci signifie que le fichier ITL au téléphone sélectionne le fichier sur le CUCM, ainsi il n'a pas besoin d'être supprimé.

S'il s'assortit, vous devez passer à la prochaine exécution - déterminez si le certificat TV dans l'ITL apparie le certificat présenté par des TV. Cette exécution est un peu plus impliquée.

D'abord, regardez la capture de paquet du téléphone qui connecte aux TV le serveur sur le port TCP 2445.

Cliquez avec le bouton droit sur n'importe quel paquet dans ce flot dans Wireshark, le clic **décodent comme**, et sélectionnent le **SSL**. Trouvez le certificat de serveur qui ressemble à ceci :





Regardez le certificat TV contenu dans le fichier précédent ITL. Vous devriez voir une entrée avec le numéro de série **2E3E1A7BDAA64D84**.

```
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
```

Le succès, le **TVS.pem à l'intérieur de du** fichier ITL apparie le certificat TV présenté sur le réseau. Vous n'avez pas besoin de supprimer l'ITL, et les TV présente le certificat correct.

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

Restrictions et interactions

Certificats régénérés/reconstruction une expiration de batterie/certificat

Le certificat le plus important est maintenant le certificat CallManager.pem. La clé privée de ce certificat est utilisée afin de signer tous les fichiers de configuration TFTP, qui inclut le fichier ITL.

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

Après que vous régénériez CallManager.pem et redémarriez les TV et le service TFTP, ceci se produit quand un téléphone démarre.

1. Les tentatives de téléphone de télécharger le nouveau fichier ITL ont signé par le nouveau CCM+TFTP du serveur TFTP. Le téléphone a seulement le vieux fichier ITL en ce moment, et les nouvelles clés ne sont pas dans le fichier ITL actuel au téléphone.
2. Puisque le téléphone ne pourrait pas trouver la nouvelle signature CCM+TFTP dans la vieille ITL, il tente d'entrer en contact avec le service TV.
   Remarque: La présente partie est extrêmement importante. Le certificat TV à partir du vieux fichier ITL doit encore s'assortir. Si les CallManager.pem et TVS.pem sont régénérés à la même heure exacte, les téléphones ne peuvent pas ne télécharger aucun nouveau fichier sans supprimer l'ITL du téléphone manuellement.
3. Quand le téléphone entre en contact avec des TV, le serveur CUCM qui exécute des TV a le nouveau certificat CallManager.pem dans la mémoire de certificat de SYSTÈME D'EXPLOITATION.
4. Le succès de retours de serveur TV et le téléphone charge le nouveau fichier ITL dans la mémoire.
5. De téléphone les tentatives maintenant de télécharger un fichier de configuration, qui a été signé par la nouvelle clé CallManager.pem.
6. Puisque la nouvelle ITL a été chargée, le fichier de configuration nouvellement signé est avec succès vérifié par l'ITL dans la mémoire.

Points clé :

- Ne régénérez en même temps jamais les Certificats CallManager.pem et TVS.pem.
- Si TVS.pem ou CallManager.pem est régénéré, des TV et le TFTP devraient être redémarrés et téléphonent la remise afin d'obtenir les nouveaux fichiers ITL. De plus nouvelles versions de CUCM manipulent ce téléphone remis à l'état initial automatiquement et avertissent l'utilisateur au temps de régénération de certificat.
- Si plus d'un serveur TV existe (plus d'un serveur dans le groupe de CallManager), les serveurs supplémentaires peuvent authentifier le nouveau certificat CallManager.pem.

Déplacez les téléphones entre les batteries

Quand vous déplacez des téléphones d'une batterie à l'autre avec ITLs en place, la clé privée ITL et TFTP doit être prise en considération. N'importe quel nouveau fichier de configuration présenté au téléphone DOIT apparier une signature dans CTL, ITL, ou une signature dans le service en cours TV du téléphone.

Ce document explique comment s'assurer que de la nouvelle le fichier et les fichiers de configuration ITL batterie mettent en boîte sont de confiance par le fichier ITL de courant au téléphone. https://supportforums.cisco.com/docs/DOC-15799.

De sauvegarde et restauration

Le certificat CallManager.pem et la clé privée sont sauvegardés par l'intermédiaire du système de Reprise sur sinistre (jeu rouleau-tambour). Si un serveur TFTP est reconstruit, il DOIT être restauré de la sauvegarde de sorte que la clé privée puisse être restaurée. Sans clé privée CallManager.pem sur le serveur, les téléphones avec ITLs en cours qui utilisent la vieille clé ne font pas confiance aux fichiers de configuration signés.

Si une batterie est reconstruite et pas restaurée de la sauvegarde, elle est exactement comme « les téléphones mobiles le document entre batteries ». C'est parce qu'une batterie avec une nouvelle clé est une batterie différente en ce qui concerne les téléphones.

Il y a un défaut sérieux associé avec de sauvegarde et la restauration. Si une batterie est susceptible de l'ID de bogue Cisco CSCtn50405, les sauvegardes jeu rouleau-tambour ne contiennent pas le certificat CallManager.pem. Ceci entraîne n'importe quel serveur restauré de cette sauvegarde pour générer les fichiers corrompus ITL jusqu'à ce qu'un nouveau CallManager.pem soit généré. S'il n'y a de pas autres serveurs fonctionnels TFTP qui ne sont pas passés par l'exécution de sauvegarde et de restauration, ceci pourrait signifier que tous les fichiers ITL doivent être supprimés des téléphones.

Afin de vérifier si votre fichier CallManager.pem doit être régénéré, sélectionnez la **commande de** showitl 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 erreurs principales à rechercher sont :

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

et

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

La requête précédente du SQL (SQL) recherche les Certificats qui ont un rôle de la « authentification et de l'autorisation. » Le certificat CallManager.pem dans l'interrogation de bases de données précédente qui a le rôle de l'authentification et de l'autorisation devrait ÉGALEMENT être présent dans la page Web de Gestion de certificat de gestion de SYSTÈME D'EXPLOITATION. Si le défaut précédent est produit, il y a une non-concordance entre les Certificats CallManager.pem dans la requête et dans la page Web de SYSTÈME D'EXPLOITATION.

Noms d'hôte ou noms de domaine de modification

Si vous changez l'adresse Internet ou le nom de domaine d'un serveur CUCM, il régénère tous les Certificats immédiatement sur ce serveur. La section de régénération de certificat a expliqué que la régénération du TVS.pem et de CallManager.pem est une « mauvaise chose. »

Il y a quelques scénarios où une modification d'adresse Internet échoue, et quelques uns où cela fonctionne sans problèmes. Cette section couvre tous et les joint de nouveau au ce que vous connaissez déjà les TV et l'ITL de ce document.

Batterie de noeud simple avec seulement l'ITL (la précaution d'usage, ceci se casse sans préparation)

- Avec un serveur de Business Edition ou un déploiement réservé à la Publisher, les CallManager.pem et TVS.pem sont régénérés en même temps quand vous changez des adresses Internet.
- Si l'adresse Internet est changée sur une batterie de noeud simple sans d'abord utilisant le paramètre d'entreprise de repositionnement couvert ici, les téléphones ne peuvent pas vérifier le nouveau fichier ITL ou les fichiers de configuration contre leur ITL de courant classent. Supplémentaire, ils ne peuvent pas se connecter aux TV parce que le certificat TV n'est également plus fait confiance.
- Les téléphones affichent une erreur au sujet « de vérification de liste de confiance ont manqué, » nouvelle modification de configuration ne la prend pas effet, et échouer sécurisé du service URLs.
- La seule solution si la précaution dans l'étape 2 n'est pas première prise est de supprimer manuellement l'ITL de chaque téléphone.

Batterie de noeud simple avec CTL et ITL (ceci peut être temporairement cassé, mais être facilement réparé)

- Après que vous vous exécutiez par le renommer des serveurs, réexécutez le client CTL. Ceci place le nouveau certificat CallManager.pem dans le fichier CTL que le téléphone télécharge.
- Les nouveaux fichiers de configuration, qui incluent les nouveaux fichiers ITL, peuvent sont de confiance basé sur la fonction CCM+TFTP dans le fichier CTL.
- Ceci fonctionne parce que le fichier CTL mis à jour est de confiance basé sur un USB eToken la clé privée qui demeure la même.

Batterie de Multi-noeud avec seulement l'ITL (ceci fonctionne généralement, mais peut être de manière permanente cassé si fait à la hâte)

- Puisqu'une batterie de multi-noeud a de plusieurs serveurs TV, n'importe quel serveur unique peut faire régénérer ses Certificats sans problème. Quand le téléphone est présenté avec ce nouveau, signature peu familière, elle demande à des autres des serveurs TV de vérifier le nouveau certificat de serveur.
- Il y a deux problèmes principaux qui peuvent faire échouer ceci :
  Si tous les serveurs sont renommés et redémarrés en même temps, aucun des serveurs TV n'est accessible avec les Certificats connus quand les serveurs et les téléphones se réactivent.Si un téléphone a seulement un serveur unique dans le groupe de CallManager, les serveurs supplémentaires TV ne font aucune différence. Voyez le scénario « de batterie de noeud simple » afin de résoudre ceci, ou ajoutez un autre serveur au groupe du CallManager du téléphone.

Batterie de Multi-noeud avec CTL et ITL (ceci ne peut pas être de manière permanente cassé)

- Après que vous vous exécutiez par le renomme, le service TV authentifie les nouveaux Certificats.
- Même si tous les serveurs TV sont indisponibles pour quelque raison, le client CTL peut encore être utilisé afin de mettre à jour les téléphones avec les nouveaux Certificats

CallManager.pem CCM+TFTP.

TFTP centralisé

Quand un téléphone avec une ITL démarre, il demande ces fichiers : **CTLSEP *< adresse MAC >.tlv***, **ITLSEP *< adresse MAC >.tlv***, et **SEPT *< adresse MAC >.cnf.xml.sgn*.**

Si le téléphone ne peut pas trouver ces fichiers, il demande l'**ITLFile.tlv et le** CTLFile.tlv**, qu'un** serveur centralisé TFTP fournit à n'importe quel téléphone qui le demande.

Avec le TFTP centralisé, il y a une batterie simple TFTP ces points à un certain nombre d'autres sous batteries. Souvent ceci est fait parce que les téléphones sur de plusieurs batteries CUCM partagent la même portée de DHCP, et doit donc avoir le même serveur de l'option 150 TFTP DHCP. Tout le point de Téléphones IP à la batterie centrale TFTP, même si ils s'enregistrent à d'autres batteries. Ce serveur central TFTP questionne les serveurs TFTP distants toutes les fois qu'il reçoit une demande d'un fichier qu'elle ne peut pas la trouver.

En raison de cette exécution, le TFTP centralisé fonctionne seulement dans un environnement homogène ITL. Tous les serveurs doivent exécuter la version 8.x ou ultérieures CUCM, ou tous les serveurs doivent exécuter des versions avant la version 8.x.

Si un ITLFile.tlv est présenté du serveur centralisé TFTP, les téléphones ne font confiance à aucun fichier du serveur TFTP distant parce que les signatures ne s'assortissent pas. Ceci se produit dans un mélange hétérogène. Dans un mélange homogène, le téléphone demande ITLSEP **<MAC>.tlv qui** est tiré de la batterie distante correcte.

Dans un environnement hétérogène avec un mélange de batteries de la pré-version 8.x et de la version 8.x, « préparez la batterie pour le repositionnement pré à 8.0" doit être activé sur la batterie de version 8.x comme décrit dans l'ID de bogue Cisco CSCto87262 et « les paramètres sécurisés URL de téléphone » configurés avec le HTTP au lieu de HTTPS. Ceci désactive efficacement les fonctions ITL au téléphone.

Forum aux questions

Est-ce que je peux arrêter le SBD ?

Vous pouvez seulement arrêter le SBD si les SBD et les ITL fonctionnent actuellement.

Le SBD peut être temporairement désactivé aux téléphones avec la batterie de préparation pour le repositionnement au paramètre d'entreprise pré de 8.0" et en configurant « les paramètres sécurisés URL de téléphone » avec le HTTP au lieu de HTTPS. Quand vous placez le paramètre de repositionnement, il crée un fichier signé ITL avec les entrées vides de fonction. Le fichier « vide » ITL est encore signé, ainsi la batterie doit être dans l'état fonctionnel de Sécurité a entièrement - avant que ce paramètre puisse être activé.

Après que ce paramètre soit activé et le nouveau fichier ITL avec les entrées vides est téléchargé et vérifié, les téléphones reçoivent n'importe quel fichier de configuration, n'importe qui l'a signé.

Il n'est pas recommandé pour partir de la batterie dans cet état, parce qu'aucune des trois

fonctions précédemment metioned (les fichiers de configuration authentifiés, les fichiers de configuration chiffrés, et le HTTPS URLs) n'est disponible.

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

Il n'y a actuellement aucune méthode pour supprimer tout l'ITLs d'un téléphone à distance fourni par Cisco. C'est pourquoi il est si importante prendre en considération les procédures et les interactions décrites dans ce document.

Il y a actuellement une amélioration non résolue à l'ID de bogue Cisco CSCto47052 qui demande cette fonctionnalité, mais elle n'a pas été encore mise en application.

Dans l'intervalle période, une nouvelle caractéristique a été ajoutée par l'intermédiaire de l'ID de bogue Cisco CSCts01319 qui pourrait permettre au centre d'assistance technique Cisco (TAC) pour retourner à l'ITL précédemment de confiance si elle est encore disponible sur le serveur. Ceci fonctionne seulement dans certains exemples où la batterie est sur une version avec cette correction de défauts, et où l'ITL précédente existe dans une sauvegarde enregistrée dans un emplacement spécial sur le serveur. Visualisez le défaut pour voir si votre version a la difficulté. Contactez Cisco TAC afin de s'exécuter par la procédure de récupération potentielle expliquée dans le défaut.

Si la procédure précédente n'est pas disponible, les touches du téléphone doivent être poussées manuellement au téléphone afin de supprimer le fichier ITL. C'est le compromis qui est fait entre la Sécurité et la facilité de la gestion. Pour que le fichier ITL soyez vraiment sécurisé, il ne doit pas soit facilement retiré à distance.

Même avec les presses Par script de bouton avec le protocole simple d'Access d'objet (SAVON) le XML objecte, l'ITL ne peut pas être à distance retiré. C'est parce que, en ce moment, l'accès TV (et accès sécurisé d'authentication url pour valider ainsi les objets entrants de pousser de bouton de SAVON XML) est non fonctionnel. Si l'authentication url n'est pas configuré comme sécurisé, il pourrait être possible au script que la clé enfonce la commande pour supprimer une ITL, mais ce script n'est pas fourni par Cisco.

D'autres presses principales distantes de script de méthodes sans utiliser l'authentication url pourraient être fournies par un tiers, mais ces applications ne sont pas fournies par Cisco.

Le plus souvent la méthode utilisée afin de supprimer l'ITL est une émission d'email à tous les utilisateurs du téléphone qui les instruit de l'ordre principal. Si l'accès de configurations est placé **restreint** ou **handicapé**, le téléphone doit être réinitialisation aux paramètres d'usine, car les utilisateurs n'ont pas accès au menu Settings du téléphone.