Migrez les téléphones entre les batteries sécurisées

Contenu

[Introduction](#)
[Conditions préalables](#)
[Conditions requises](#)
[Composants utilisés](#)
[Fond](#)
[Configurer](#)
[Vérifier](#)
[Dépanner](#)

Introduction

Ce document décrit comment migrer des téléphones entre deux batteries sécurisées de Cisco Unified Communications Manager (CUCM).

Contribué par le Normand de David, ingénieur TAC Cisco.

Conditions préalables

Exigences

Cisco recommande que vous ayez la connaissance de CUCM.

Composants utilisés

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

Batterie de source : Version 10.5.2.11900-3 CUCM
Cluster de destination : Version 11.0.1.10000-10 CUCM
téléphone 8861 utilisant le micrologiciel sip88xx.10-3-1-20

Des fichiers de la liste de CertificateTrust (CTL) sont signés avec le certificat de CallManager (pas le jeton USB)

Fond

Pendant le procédé de transfert, le téléphone tente d'installer une connexion sécurisée au service de vérification de confiance de Cisco de batteries de source (TV) pour vérifier le certificat de CallManager de clusters de destination. Si le fichier de la liste de la liste de confiance du certificat du téléphone (CTL) et de la confiance d'identité (ITL) sont non valide, le téléphone ne peut pas se terminer la prise de contact sécurisée avec les TV et le transfert au cluster de destination ne

réussira pas. Avant que vous commenciez le procédé de transfert de téléphone, confirmez que les téléphones ont le fichier correct CTL/ITL installé. Également sur la batterie de source, confirmez que la fonctionnalité de l'édition Enterprise « préparent la batterie pour le repositionnement pré à 8.0" est placée à faux.

Configurer

Importez le certificat de CallManager de clusters de destination dans les batteries CallManager-confiance de source et la mémoire de Téléphone-SAST-confiance. Il y a deux méthodes pour faire ceci.

Méthode 1.

Utilisez l'outil en vrac de certificat et terminez-vous ces étapes sur la source et des clusters de destination.

Étape 1. Naviguez vers la page de **gestion de SYSTÈME D'EXPLOITATION de Cisco Unified** > la **Gestion de Sécurité** > de **certificat en vrac** sur la source et les clusters de destination.

Étape 2. Écrivez les détails pour le serveur de Protocole SFTP (Secure File Transfer Protocol) et sélectionnez la **sauvegarde**.

Étape 3. Sélectionnez l'**exportation** et exportez le certificat de Protocole TFTP (Trivial File Transfer Protocol).

Étape 4. Cliquez sur en fonction le bouton de **consolidation** pour exécuter la fusion de certificat. Ceci crée un fichier PKCS12 qui inclut la source et le certificat de CallManager de destination.

Étape 5. Importez les Certificats consolidés de nouveau dans chaque batterie.

Pendant le processus de fusion (étape 5), le certificat de CallManager de batteries de source est téléchargée au cluster de destination en CallManager-confiance et mémoire de Téléphone-SAST-confiance. Ceci permet aux téléphones pour migrer de nouveau à la batterie de source. Si la méthode manuelle est suivie, le certificat de CallManager de batteries de source ne sera pas téléchargé au cluster de destination. Ceci signifie que vous ne pouvez pas migrer les téléphones de nouveau à la batterie de source. Si vous voulez l'option de migrer les téléphones de nouveau à la source groupent, vous devez télécharger le certificat de CallManager de batteries de source aux clusters de destination CallManager-confiance et à la mémoire de Téléphone-SAST-confiance.

> **Note**: Les deux batteries doivent exporter le certificat TFTP au même serveur de SFTP et au même répertoire de SFTP.

> **Note**: Étape 4 est seulement exigée sur une batterie. Si vous des migratephones entre la version 8.x ou 9.x CUCM à la version 10.5.2.13900-12 ou plus récentes CUCM, notez cet ID de bogue Cisco CSCuy43181 avant que vous consolidiez les Certificats.

Méthode 2.

Importez manuellement les Certificats. Terminez-vous ces étapes sur le cluster de destination.

Étape 1. Naviguez vers la page de **gestion de SYSTÈME D'EXPLOITATION de Cisco Unified** > la **Gestion de Sécurité** > de **certificat**.

Étape 2. Le certificat choisi CallManager.pem et la téléchargent.

Étape 3. Le certificat choisi ITLrecovery.pem et la téléchargent

Étape 4. Téléchargez le certificat de CallManager à l'éditeur de batterie de source comme CallManger-confiance et certificat de Téléphone-SAST-confiance.

Étape 5. Téléchargez le certificat d'ITLrecovery à la batterie de source comme Téléphone-SAST-confiance

Étape 6. Redémarrez les TV dans tous les Noeuds de la batterie de source.

Puis la réplique de Certificats aux autres Noeuds dans la batterie.

Étapes 3, 5, 6 s'appliqueront aux scénarios de migrer le téléphone à partir de 8.x vers 12.x

> **Note**: Le certificat de CallManager doit être téléchargé de tous les Noeuds exécutant le service TFTP sur le cluster de destination.

Une fois que les Certificats ont été téléchargés avec une des méthodes ci-dessus, changez l'option 150 du protocole DHCP de téléphones (DHCP) d'indiquer l'adresse des clusters de destination TFTP.

> **Attention** : Une méthode pour migrer des téléphones les batteries non-sécurisées qu'intermédiaires est de placer « préparent la batterie pour le repositionnement pré à 8.0" rectifier sur la batterie de source et redémarrer les téléphones. Ce n'est pas une option quand vous migrez des téléphones entre les batteries sécurisées. C'est parce que le repositionnement à la caractéristique pré 8.0 masque seulement le fichier ITL (il ne masque pas le fichier CTL). Ceci signifie que quand le téléphone est migré et il télécharge le fichier CTL du cluster de destination, il doit vérifier le nouveau CTL avec les batteries TV de source. Puisque le fichier ITL du téléphone ne contient pas la source groupe le certificat TV, la prise de contact quand les essais de téléphone n'établit pas une connexion sécurisée au service TV.

Vérifiez

C'est un extrait des logs de console de téléphone et des logs TV (placez à détaillé) de la batterie de source. Les extraits affichent le processus de l'enregistrement de téléphones au cluster de destination.

1. Le téléphone démarre et télécharge le fichier CTL du cluster de destination.

```
3232 NOT Nov 29 06:33:59.011270 downd-DDFORK - execing [/usr/sbin/dgetfile][-L620][ ]
3233 NOT Nov 29 06:33:59.033132 dgetfile(870)-GETXXTP
[GT870][src=CTLSEPB000B4BA0AEE.tlv][dest=/tmp/CTLFile.tlv][serv=][serv6=][sec=0]
```

2. Le fichier CTL est signé par le certificat de gestionnaire d'appel de clusters de destination qui n'est pas dans les téléphones fichier existant CTL ou ITL. Ceci signifie que le téléphone doit atteindre à son service TV pour vérifier le certificat. En ce moment le téléphone a toujours sa configuration ancienne qui contient l'adresse IP du service de la batterie TV de source (les TV spécifiées dans la configuration de téléphones est identiques comme le groupe de gestionnaire d'appel téléphonique). Le téléphone a installé une connexion SSL au service TV. Quand le service TV présente son certificat au téléphone, le téléphone vérifie le certificat contre le certificat dans son fichier ITL. S'ils sont identiques, la prise de contact se termine avec succès.

```
3287 INF Nov 29 06:33:59.395199 SECUREAPP-Attempting connect to TVS server addr [192.168.11.32],
mode [IPv4]
3288 INF Nov 29 06:33:59.395294 SECUREAPP-TOS set to [96] on sock, [192.168.11.32][11]
3289 INF Nov 29 06:33:59.396011 SECUREAPP-TCP connect() successful, [192.168.11.32] [11]
3290 DEB Nov 29 06:33:59.396111 SECUREAPP-BIO created with: addr:192.168.11.32, port:2445,
mode:IPv4
3291 INF Nov 29 06:33:59.396231 SECUREAPP-Sec SSL Connection - TVS.
3292 INF Nov 29 06:33:59.396379 SECUREAPP-SSL session setup - Requesting Cert
3293 DEB Nov 29 06:33:59.396402 SECUREAPP-Obtaining certificate.
3294 INF Nov 29 06:33:59.396444 SECUREAPP-SSL session setup - Get Active cert ok
3295 DEB Nov 29 06:33:59.396464 SECUREAPP-SSL session setup - cert len=785, type=LSC
3296 DEB Nov 29 06:33:59.396854 SECUREAPP-Certificate subject name = /serialNumber=PID:CP-8861
SN:FCH18198CNQ/C=AU/O=stormin/OU=IST/CN=CP-8861-SEPB000B4BA0AEE
3297 DEB Nov 29 06:33:59.396917 SECUREAPP-SSL session setup - Certificate issuer name =
/C=AU/O=stormin/OU=IST/CN=CAPF-a7fb32bf/ST=NSQ/L=Sydney
3298 INF Nov 29 06:33:59.396947 SECUREAPP-SSL session setup - Requesting Pkey
3299 INF Nov 29 06:33:59.397024 SECUREAPP-SSL session setup - Get private key ok
3300 DEB Nov 29 06:33:59.397045 SECUREAPP-SSL session setup - key len=1191
3301 INF Nov 29 06:33:59.399181 SECUREAPP-Setup SSL session - SSL use certificate okay
3302 INF Nov 29 06:33:59.399477 SECUREAPP-Setup SSL session - SSL use private key okay
3303 DEB Nov 29 06:33:59.399974 SECUREAPP-Sec SSL Connection - Added SSL connection handle
0x40e01270, connDesc 11 to table.
3304 DEB Nov 29 06:33:59.400225 SECUREAPP-Sec SSL Connection - check status & perform handshake.
3305 DEB Nov 29 06:33:59.401086 SECUREAPP-Blocked TVS Secure Connection - Waiting (0) ....
3306 DEB Nov 29 06:33:59.401796 SECUREAPP-Sec SSL Connection - check status & perform handshake.
3307 DEB Nov 29 06:33:59.403321 SECUREAPP-SSL session setup Cert Verification - Role is = 21
3308 INF Nov 29 06:33:59.403412 SECUREAPP-SSL session setup Cert Verification - Invoking
certificate validation helper plugin.
3309 INF Nov 29 06:33:59.403662 SECUREAPP-SSL session setup Cert Verification - Certificate
validation helper plugin returned.
3310 INF Nov 29 06:33:59.403731 SECUREAPP-SSL session setup Cert Verification - Certificate is
valid.
3311 DEB Nov 29 06:33:59.403784 SECUREAPP-SSL session setup Cert Verification - returning
validation result = 1
3312 ERR Nov 29 06:33:59.428892 downd-SOCKET accept errno=4 "Interrupted system call"
3313 DEB Nov 29 06:33:59.907337 SECUREAPP-Blocked TVS Secure Connection - Waiting (1) ....
3314 DEB Nov 29 06:33:59.907393 SECUREAPP-Sec SSL Connection - check status & perform handshake.
3315 NOT Nov 29 06:33:59.908586 SECUREAPP-Sec SSL Connection - Handshake successful.
3316 INF Nov 29 06:33:59.908696 SECUREAPP-Sec SSL Connection - caching disabled, session not
saved
3317 DEB Nov 29 06:33:59.908752 SECUREAPP-Connection to server succeeded
```

3. Les logs TV affichent que la connexion entrante du téléphone et de la prise de contact était réussie.

```
18:01:05.333 | debug Accepted TCP connection from socket 0x00000012, fd = 8
18:01:05.333 | debug Total Session attempted = 7 accepted = 7
```

```
18:01:05.334 | debug tvsGetNextThread
18:01:05.334 | debug Recd event
18:01:05.334 | debug new ph on fd 8
18:01:05.334 | debug 7:UNKNOWN:Got a new SCB from RBTree
18:01:05.334 | debug ipAddrStr (Phone) 192.168.11.100
18:01:05.334 | debug 8:UNKNOWN:Got a new ph conn 192.168.11.100 on 8, Total Acc = 7..
18:01:05.334 | debug addded 8 to readset
18:01:05.338 | debug after select, 8 was set
18:01:05.338 | debug ipAddrStr (Phone) 192.168.11.100
18:01:05.855 | debug tvsSSLHandShakeNotify
18:01:05.855 | debug 192.168.11.100: tvsSSLHandShake Session ciphers - AES256-SHA
18:01:05.855 | debug addded 8 to readset
18:01:05.855 | debug Recd event
18:01:05.855 | debug TLS HS Done for ph_conn
```

4. L'exposition de logs de console de téléphone le téléphone envoient une demande au service TV de vérifier le certificat de gestionnaire d'appel du cluster de destination.

```
3318 DEB Nov 29 06:33:59.908800 SECUREAPP-TVS provider Init - connect returned TVS srvr sock: 11
3319 DEB Nov 29 06:33:59.908848 SECUREAPP-TVS process request - processing TVS Query Certificate
request.
3320 NOT Nov 29 06:33:59.909322 SECUREAPP-TVS process request - Successfully sent the TVS
request to TVS server, bytes written : 153
3321 DEB Nov 29 06:33:59.909364 SECUREAPP-==== TVS process request - request byte dump ====, len
= 153
3322 DEB Nov 29 06:33:59.913075 SECUREAPP-TVS Service receives 1480 bytes of data
3323 DEB Nov 29 06:33:59.913270 SECUREAPP-==== TVS process response - response byte dump ====,
len = 1480
3324 DEB Nov 29 06:33:59.914466 SECUREAPP-Found the work order from pending req list element at
index 0
```

5. L'exposition de logs TV la demande est reçue.

```
18:01:06.345 | debug 8:UNKNOWN:Incoming Phone Msg:
HEX_DUMP: Len = 153:
18:01:06.345 | debug 57 01 03 00 00 00 03 e9
18:01:06.345 | debug 00 8f 01 00 18 01 43 50
18:01:06.345 | debug 2d 38 38 36 31 2d 53 45
18:01:06.345 | debug 50 42 30 30 30 42 34 42
18:01:06.345 | debug 41 30 41 45 45 03 00 42
18:01:06.345 | debug 43 4e 3d 75 63 6d 31 31
18:01:06.345 | debug 70 75
18:01:06.345 | debug tvsPhoneDecodeMsg -
Decoded Phone Msg:
18:01:06.345 | debug Protocol Discriminator: 57
18:01:06.345 | debug MsgType : TVS_MSG_QUERY_CERT_REQ
18:01:06.345 | debug Session Id : 0
18:01:06.345 | debug Length : 143
18:01:06.345 | debug 8:UNKNOWN:TVS CORE: Rcvd Event: TVS_EV_QUERY_CERT_REQ in State:
TVS_STATE_AWAIT_REQ
18:01:06.345 | debug tvsHandleQueryCertReq
18:01:06.345 | debug tvsHandleQueryCertReq : Subject Name is:
CN=ucm11pub.stormin.local;OU=IST;O=Stormin;L=Brisbane;ST=QLD;C=AU
18:01:06.345 | debug tvsHandleQueryCertReq : Issuer Name is: CN=stormin-WIN2012-CA
18:01:06.345 | debug tvsHandleQueryCertReq : Serial Number is:
24000000179479B8F124AC3F3B000000000017
```

```
18:01:06.345 | debug CertificateDBCache::getCertificateInformation - Looking up the certificate
cache using Unique MAP ID : 24000000179479B8F124AC3F3B000000000017CN=stormin-WIN2012-CA
18:01:06.345 | debug CertificateDBCache::getCertificateInformation - Found entry {rolecount : 2}
18:01:06.345 | debug CertificateDBCache::getCertificateInformation - {role : 0}
18:01:06.346 | debug CertificateDBCache::getCertificateInformation - {role : 3}
18:01:06.346 | debug convertX509ToDER -x509cert : 0xbb696e0
```

6. Les logs TV affichent que le certificat dans sa mémoire et TV envoie une réponse au téléphone.

```
18:01:06.346 | debug 8:UNKNOWN:Sending QUERY_CERT_RES msg
18:01:06.346 | debug tvsPhoneDecodeMsg -
Decoded Phone Msg:
18:01:06.346 | debug Protocol Discriminator: 57
18:01:06.346 | debug MsgType : TVS_MSG_QUERY_CERT_RES
18:01:06.346 | debug Session Id : 0
18:01:06.346 | debug Length : 1470
18:01:06.346 | debug ReasonInfo : 00$
18:01:06.346 | debug Number of Certs : 1
18:01:06.346 | debug Cert[0] :
18:01:06.346 | debug Cert Type : 0
HEX_DUMP: Len = 1451:
18:01:06.346 | debug 30 82 05 a7 30 82 04 8f
18:01:06.346 | debug a0 03 02 01 02 02 13 24
18:01:06.346 | debug 00 00 00 17 94 79 b8 f1
18:01:06.346 | debug 24 ac 3f 3b 00 00 00 00
18:01:06.346 | debug 00 17 30 0d 06 09 2a 86
18:01:06.346 | debug 48 86 f7 0d 01 01 0b 05
18:01:06.346 | debug 00 30
18:01:06.346 | debug Version : 0
18:01:06.346 | debug PublicKey :
HEX_DUMP: Len = 4:
18:01:06.347 | debug 00 01 51 80
18:01:06.347 | debug Sending TLS Msg ..
HEX_DUMP: Len = 1480:
18:01:06.347 | debug 57 01 04 f7 00 00 03 e9
18:01:06.347 | debug 05 be 07 00 01 00 02 05
18:01:06.347 | debug ab 30 82 05 a7 30 82 04
18:01:06.347 | debug 8f a0 03 02 01 02 02 13
18:01:06.347 | debug 24 00 00 00 17 94 79 b8
18:01:06.347 | debug f1 24 ac 3f 3b 00 00 00
18:01:06.347 | debug 00 00
18:01:06.347 | debug ipAddrStr (Phone) 192.168.11.100
```

7. Les logs de console de téléphone prouvent que le certifcate est vérifié avec succès et le fichier CTL est mis à jour.

```
3325 INF Nov 29 06:33:59.915121 SECUREAPP-TVS added cert to TVS cache - expires in 24 hours
3333 NOT Nov 29 06:34:00.411671 SECUREAPP-Hashes match... authentication successful.
3334 WRN Nov 29 06:34:00.412849 SECUREAPP-AUTH: early exit from parser loop; old version header?
3335 WRN Nov 29 06:34:00.412945 SECUREAPP-AUTH: hdr ver 1.2 (knows only upto 1.1)
3336 NOT Nov 29 06:34:00.413031 SECUREAPP-updateFromFile: TL parse to table: CTL_SUCCESS
3337 NOT Nov 29 06:34:00.413088 SECUREAPP-updateFromFile: Updating master TL table
3338 DEB Nov 29 06:34:00.413442 SECUREAPP-TL file verified successfully.
3339 INF Nov 29 06:34:00.413512 SECUREAPP-TL file updated.
```

8. Les logs de console de téléphone affichent quand le téléphone télécharge son fichier ITL.

```
3344 NOT Nov 29 06:34:00.458890 dgetfile(877)-GETXXTP
[GT877][src=ITLSEPB000B4BA0AEE.tlv][dest=/tmp/ITLFile.tlv][serv=][serv6=][sec=0]
3345 NOT Nov 29 06:34:00.459122 dgetfile(877)-In normal mode, call - > makeXXTPrequest (V6...)

3281 NOT Dec 14 06:34:00.488697 dgetfile(851)-XXTP complete - status = 100
3282 NOT Dec 14 06:34:00.488984 dgetfile(851)-XXTP actualserver [192.168.11.51]
```

9. Le fichier ITL est vérifié contre le fichier CTL. Le fichier CTL contient le certificat de CallManager de clusters de destination. Ceci signifie que le téléphone peut vérifier le certificat sans contacter le service des batteries TV de source.

```
3287 NOT Nov 29  06:34:00.499372 SECUREAPP-Hashes match... authentication successful.
3288 WRN Nov 29  06:34:00.500821 SECUREAPP-AUTH: early exit from parser loop; old version
header?
3289 WRN Nov 29 06:34:00.500987 SECUREAPP-AUTH: hdr ver 1.2 (knows only upto 1.1)
3290 NOT Nov 29  06:34:00.501083 SECUREAPP-updateFromFile: TL parse to table: CTL_SUCCESS
3291 NOT Nov 29  06:34:00.501147 SECUREAPP-updateFromFile: Updating master TL table
3292 DEB Nov 29  06:34:00.501584 SECUREAPP-TL file verified successfully.
3293 INF Nov 29  06:34:00.501699 SECUREAPP-TL file updated.
```

Dépanner

Avant le procédé de transfert, vérifiez le CTL/ITL aux téléphones. Plus d'informations sur la façon dont vérifier le CTL/ITL peuvent être trouvées ici : https://www.cisco.com/c/en/us/support/docs/voice-unified-communications/unified-communications-manager-callmanager/116232-technote-sbd-00.html#anc9