Migration des téléphones entre des clusters sécurisés

Contenu

Introduction

Ce document décrit comment migrer des téléphones entre deux clusters sécurisés Cisco Unified Communications Manager (CUCM).

Avec la collaboration de David Norman, ingénieur TAC Cisco.

Conditions préalables

Conditions requises

Cisco vous recommande de connaître CUCM.

Components Used

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

Cluster source : CUCM version 10.5.2.11900-3
Cluster de destination : CUCM version 11.0.1.10000-10
Téléphone 8861 avec microprogramme sip88xx.10-3-1-20

Les fichiers CTL (CertificateTrust List) sont signés avec le certificat CallManager (pas le jeton USB)

Fond

Au cours du processus de migration, le téléphone tente de configurer une connexion sécurisée aux clusters source Cisco Trust Verification Service (TVS) pour vérifier le certificat CallManager des clusters de destination. Si le fichier Certificate Trust List (CTL) et Identity Trust List (ITL) du téléphone n'est pas valide, le téléphone ne peut pas terminer la connexion sécurisée avec TVS et la migration vers le cluster de destination échouera. Avant de démarrer le processus de migration

du téléphone, vérifiez que le fichier CTL/ITL approprié est installé sur les téléphones. Également sur le cluster source, vérifiez que la fonction Enterprise « Prepare Cluster for Rollback to Pre 8.0 » a la valeur False.

Configuration

Importez le certificat CallManager du cluster de destination dans le magasin CallManager-trust et Phone-SAST-trust du cluster source. Il y a deux méthodes pour le faire.

Méthode 1.

Utilisez l'outil Bulk Certificate Tool et complétez ces étapes sur les clusters source et de destination.

Étape 1. Accédez à la page **Administration de Cisco Unified OS > Sécurité > Gestion des certificats en bloc** sur les clusters source et de destination.

Étape 2. Entrez les détails du serveur SFTP (Secure File Transfer Protocol) et sélectionnez **Enregistrer**.

Étape 3. Sélectionnez **Exporter** et exporter le certificat TFTP (Trivial File Transfer Protocol).

Étape 4. Cliquez sur le bouton **Consolider** pour effectuer la consolidation des certificats. Ceci crée un fichier PKCS12 qui inclut le certificat CallManager source et de destination.

Étape 5. Importez les certificats consolidés dans chaque cluster.

Au cours du processus de consolidation (étape 5), le cluster source Le certificat CallManager est téléchargé vers le cluster de destination dans le magasin CallManager-trust et Phone-SAST-trust. Cela permet aux téléphones de revenir au cluster source. Si la méthode manuelle est suivie, le certificat CallManager du cluster source est ne être téléchargé vers le cluster de destination. Cela signifie que vous ne pouvez pas remigrer les téléphones vers le cluster source. Si vous voulez que l'option de migration des téléphones vers le cluster source, vous Vous devez télécharger le certificat CallManager des clusters sources dans le magasin CallManager-trust et Phone-SAST-trust des clusters de destination.

> **Note**: Les deux clusters doivent exporter le certificat TFTP vers le même serveur SFTP et le même répertoire SFTP.

> **Note**: L'étape 4 n'est requise que sur un cluster. Si vous migrez des téléphones entre CUCM version 8.x ou 9.x et CUCM version 10.5.2.13900-12 ou ultérieure, prenez note de cet ID de bogue Cisco CSCuy43181 avant de consolider les certificats.

Méthode 2.

Importez manuellement les certificats. Effectuez ces étapes sur le cluster de destination.

Étape 1. Accédez à la page **Administration de Cisco Unified OS > Sécurité > Gestion des certificats**.

Étape 2. Sélectionnez le certificat CallManager.pem et téléchargez-le.

Étape 3. Sélectionnez le certificat ITLrecovery.pem et téléchargez-le

Étape 4. Téléchargez le certificat CallManager vers l'éditeur de cluster source en tant que certificat CallManager-trust et Phone-SAST-trust.

Étape 5. Télécharger le certificat de récupération ITL vers le cluster source en tant que Phone-SAST-Trust

Étape 6. Redémarrez TVS dans tous les noeuds du cluster source.

Ensuite, les certificats se répliquent aux autres noeuds du cluster.

Les étapes 3, 5 et 6 s'appliquent aux scénarios de migration du téléphone de 8.x vers 12.x

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

Une fois les certificats chargés avec l'une des méthodes ci-dessus, modifiez l'option 150 DHCP (Dynamic Host Configuration Protocol) des téléphones pour qu'elle pointe vers l'adresse TFTP des clusters de destination.

> **Attention** : Une méthode pour migrer les téléphones entre des clusters non sécurisés consiste à définir la valeur True sur le cluster source pour préparer le cluster pour la restauration sur la version antérieure à 8.0 et à redémarrer les téléphones. Ce n'est pas une option lorsque vous migrez des téléphones entre des clusters sécurisés. En effet, la fonction de restauration vers la version antérieure à 8.0 n'occulte que le fichier ITL (il ne vide pas le fichier CTL). Cela signifie que lorsque le téléphone est migré et qu'il télécharge le fichier CTL à partir du cluster de destination, il doit vérifier le nouveau CTL avec le TVS du cluster source. Comme le fichier ITL du téléphone ne contient pas le certificat TVS du cluster source, la connexion échoue lorsque le téléphone tente d'établir une connexion sécurisée au service TVS.

Vérification

Il s'agit d'un extrait des journaux de la console du téléphone et des journaux TVS (définis en détail) du cluster source. Les extraits indiquent le processus d'enregistrement des téléphones vers le cluster de destination.

1. Le téléphone démarre et télécharge le fichier CTL à partir 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 du gestionnaire d'appels du cluster de destination qui ne figure pas dans le fichier CTL ou ITL existant des téléphones. Cela signifie que le téléphone doit

contacter son service TVS pour vérifier le certificat. À ce stade, l'ancienne configuration du téléphone contient toujours l'adresse IP du service TVS du cluster source (le TVS spécifié dans la configuration des téléphones est identique au groupe du gestionnaire d'appels des téléphones). Le téléphone configure une connexion SSL au service TVS. Lorsque le service TVS présente son certificat au téléphone, le téléphone vérifie le certificat par rapport au certificat dans son fichier ITL. S'ils sont identiques, la connexion s'effectue correctement.

```
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 journaux TVS indiquent la connexion entrante à partir du téléphone et la connexion a réussi.

```
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. Les journaux de la console du téléphone indiquent que le téléphone envoie une demande au service TVS pour vérifier le certificat du gestionnaire d'appels à partir 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. Les journaux TVS indiquent que 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 journaux TVS affichent le certificat dans son magasin et TVS 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 journaux de la console du téléphone indiquent que le certificat est vérifié avec succès et que 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 journaux de la console du téléphone s'affichent lorsque 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 par rapport au fichier CTL. Le fichier CTL contient le certificat CallManager du cluster de destination. Cela signifie que le téléphone peut vérifier le certificat sans contacter le service TVS du cluster 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.
```

Dépannage

Avant le processus de migration, vérifiez la CTL/ITL sur les téléphones. Pour plus d'informations sur la vérification de la CTL/ITL, cliquez ici : https://www.cisco.com/c/en/us/support/docs/voice-unified-communications/unified-communications-manager-callmanager/116232-technote-sbd-00.html#anc9