Migrar teléfonos entre clústeres seguros

Contenido

Introducción

Este documento describe cómo migrar teléfonos entre dos clústeres seguros de Cisco Unified Communications Manager (CUCM).

Colaborado por David Norman, Ingeniero del TAC de Cisco.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento de CUCM.

Componentes Utilizados

La información que contiene este documento se basa en estas versiones de software:

Clúster de origen: CUCM, versión 10.5.2.11900-3
Clúster de destino: CUCM versión 11.0.1.10000-10
teléfono 8861 con firmware sip88xx.10-3-1-20

Los archivos CertificateTrust List (CTL) se firman con el certificado de CallManager (no con el token USB)

Background

Durante el proceso de migración, el teléfono intenta configurar una conexión segura a los clústeres de origen Cisco Trust Verification Service (TVS) para verificar el certificado CallManager de los clústeres de destino. Si el archivo de lista de confianza de certificados (CTL) y lista de confianza de identidad (ITL) no es válido, el teléfono no puede completar el intercambio de señales seguro con la TVS y la migración al clúster de destino no se realizará correctamente. Antes de iniciar el proceso de migración del teléfono, confirme que los teléfonos tienen el archivo CTL/ITL correcto instalado. También en el clúster de origen, confirme que la función empresarial

"Preparar clúster para la devolución a la versión anterior 8.0" esté establecida en False.

Configurar

Importe el certificado CallManager de los clústeres de destino en los clústeres de origen CallManager-trust y Phone-SAST-trust store. Hay dos métodos para hacerlo.

Método 1.

Utilice la herramienta Bulk Certificate Tool y complete estos pasos tanto en los clústeres de origen como de destino.

Paso 1. Vaya a la página **Cisco Unified OS Administration > Security > Bulk Certificate Management** en clústeres de origen y de destino.

Paso 2. Introduzca los detalles del servidor de protocolo de transferencia segura de archivos (SFTP) y seleccione **Guardar**.

Paso 3. Seleccione **Exportar** y exporte el certificado del protocolo de transferencia de archivos trivial (TFTP).

Paso 4. Haga clic en el botón **Consolidar** para realizar la consolidación de certificados. Esto crea un archivo PKCS12 que incluye el certificado de CallManager de origen y de destino.

Paso 5. Vuelva a importar los certificados consolidados a cada clúster.

Durante el proceso de consolidación (Paso 5), el el certificado de CallManager de los clústeres de origen se carga en el clúster de destino en el almacén de confianza de CallManager y Phone-SAST. Esto permite a los teléfonos volver al clúster de origen. Si se sigue el método manual, el origen agrupa el certificado de CallManager no se cargarán en el clúster de destino. Esto significa que no puede volver a migrar los teléfonos al clúster de origen. Si desea la opción de volver a migrar los teléfonos al clúster de origen, necesita cargar el certificado CallManager de los clústeres de origen en los clústeres de destino CallManager-trust y Phone-SAST-trust store.

> **Nota:** Ambos clústeres deben exportar el certificado TFTP al mismo servidor SFTP y al mismo directorio SFTP.

> **Nota:** El paso 4 sólo se requiere en un clúster. Si migra los teléfonos entre CUCM versión 8.x o 9.x a CUCM versión 10.5.2.13900-12 o posterior, tome nota de este Id. de error de Cisco CSCuy43181 antes de consolidar los certificados.

Método 2.

Importar manualmente los certificados. Complete estos pasos en el clúster de destino.

Paso 1. Navegue hasta la página **Cisco Unified OS Administration > Security > Certificate Management**.

Paso 2. Seleccione el certificado CallManager.pem y descárguelo.

Paso 3. Seleccione el certificado ITLrecovery.pem y descárguelo

Paso 4. Cargue el certificado de CallManager al editor del clúster de origen como un certificado de confianza de CallManager y de confianza de Phone-SAST.

Paso 5. Cargue el certificado ITLrecovery en el clúster de origen como Phone-SAST-Trust

Paso 6. Reinicie TVS en todos los nodos del clúster de origen.

A continuación, los certificados se replican a los otros nodos del clúster.

Los pasos 3, 5 y 6 se aplicarán a los escenarios de migración de teléfonos de 8.x a 12.x

> **Nota:** El certificado de CallManager debe descargarse de todos los nodos que ejecutan el servicio TFTP en el clúster de destino.

Una vez cargados los certificados con uno de los métodos anteriores, cambie la opción 150 del protocolo de configuración dinámica de host (DHCP) de los teléfonos para señalar la dirección TFTP de los clústeres de destino.

> **Precaución:** Un método para migrar teléfonos entre clústeres no seguros es establecer "Preparar clúster para la devolución a la versión anterior a 8.0" en True en el clúster de origen y reiniciar los teléfonos. Esta no es una opción cuando migra teléfonos entre clústeres seguros. Esto se debe a que la reversión a la función anterior a la 8.0 sólo vacía el archivo ITL (no deja en blanco el archivo CTL). Esto significa que cuando el teléfono se migra y descarga el archivo CTL del clúster de destino, necesita verificar el nuevo CTL con el TVS de los clústeres de origen. Dado que el archivo ITL del teléfono no contiene el certificado TVS del clúster de origen, el intercambio de señales falla cuando el teléfono intenta establecer una conexión segura al servicio TVS.

Verificación

Este es un extracto de los registros de la consola del teléfono y los registros de TVS (configurados en detallados) del clúster de origen. Los fragmentos muestran el proceso de registro de los teléfonos al clúster de destino.

1. El teléfono inicia y descarga el archivo CTL del clúster de destino.

```
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. El archivo CTL está firmado por el certificado de Call Manager de los clústeres de destino que no está en el archivo CTL o ITL existente de los teléfonos. Esto significa que el teléfono debe ponerse en contacto con su servicio TVS para verificar el certificado. En este momento, el teléfono aún tiene su configuración antigua que contiene la dirección IP del servicio TVS del clúster de origen (la TVS especificada en la configuración de teléfonos es la misma que el grupo de Call Manager de los teléfonos). El teléfono configura una conexión SSL al servicio TVS.

Cuando el servicio TVS presenta su certificado al teléfono, el teléfono verifica el certificado en su archivo ITL. Si son iguales, el intercambio de señales se completa correctamente.

```
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. Los registros de TVS muestran la conexión entrante desde el teléfono y el intercambio de señales se realizó correctamente.

```
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. Los registros de la consola del teléfono muestran que el teléfono envía una solicitud al servicio TVS para verificar el certificado del administrador de llamadas desde el clúster de destino.

```
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. Los registros de TVS muestran que se ha recibido la solicitud.

```
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. Los registros de TVS muestran el certificado en su almacén y TVS envía una respuesta al teléfono.

```
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. Los registros de la consola del teléfono muestran que el certificado se verifica correctamente y que el archivo CTL se actualiza.

```
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. Los registros de la consola del teléfono se muestran cuando el teléfono descarga su archivo 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. El archivo ITL se verifica con el archivo CTL. El archivo CTL contiene el certificado CallManager de los clústeres de destino. Esto significa que el teléfono puede verificar el certificado sin ponerse en contacto con el servicio TVS de clústeres de origen.

```
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.
```

Troubleshoot

Antes del proceso de migración, verifique el CTL/ITL en los teléfonos. Puede encontrar más información sobre cómo verificar el CTL/ITL
aquí: https://www.cisco.com/c/en/us/support/docs/voice-unified-communications/unified-communications-manager-callmanager/116232-technote-sbd-00.html#anc9