Telefonmigration zwischen sicheren Clustern

Inhalt

Einführung

In diesem Dokument wird beschrieben, wie Telefone zwischen zwei sicheren Cisco Unified Communications Manager (CUCM)-Clustern migriert werden.

Mitarbeiter: David Norman, Cisco TAC Engineer.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über CUCM-Kenntnisse zu verfügen.

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Softwareversionen:

Quell-Cluster: CUCM-Version 10.5.2.11900-3
Ziel-Cluster: CUCM-Version 11.0.1.1000-10
Telefon 8861 mit Firmware-sip88xx.10-3-1-20

CTL-Dateien (CertificateTrust List) werden mit dem CallManager-Zertifikat signiert (nicht mit USB-Token)

Hintergrund

Während des Migrationsprozesses versucht das Telefon, eine sichere Verbindung mit dem Cisco Trust Verification Service (TVS) Quell-Cluster herzustellen, um das CallManager-Zertifikat der Zielcluster zu überprüfen. Wenn die CTL- und Identity Trust List (ITL)-Datei des Telefons ungültig sind, kann das Telefon den sicheren Handshake mit dem TVS nicht abschließen, und die Migration zum Ziel-Cluster ist nicht erfolgreich. Bevor Sie mit der Telefonmigration beginnen, überprüfen Sie, ob auf den Telefonen die richtige CTL/ITL-Datei installiert ist. Stellen Sie auch im Quell-Cluster sicher, dass die Enterprise-Funktion "Prepare Cluster for Rollback to Pre 8.0" auf

False festgelegt ist.

Konfigurieren

Importieren Sie das CallManager-Zielclusterzertifikat in die Quell-Cluster CallManager-trust und Phone-SAST-trust-Store. Dafür gibt es zwei Methoden.

Methode 1.

Verwenden Sie das Bulk Certificate Tool, und führen Sie diese Schritte für die Quell- und Ziel-Cluster aus.

Schritt 1: Navigieren Sie zu **Cisco Unified OS Administration** Seite > **Security** > **Bulk Certificate Management** auf Quell- und Zielclustern.

Schritt 2: Geben Sie die Details für den SFTP-Server (Secure File Transfer Protocol) ein, und wählen Sie **Speichern aus**.

Schritt 3: Wählen Sie **Export** und exportieren Sie das TFTP-Zertifikat (Trivial File Transfer Protocol).

Schritt 4: Klicken Sie auf die Schaltfläche **Konsolidierung**, um die Zertifikatskonsolidierung durchzuführen. Dadurch wird eine PKCS12-Datei erstellt, die sowohl das Quell- als auch das Ziel-CallManager-Zertifikat enthält.

Schritt 5: Importieren Sie die konsolidierten Zertifikate zurück in jeden Cluster.

Während des Konsolidierungsprozesses (Schritt 5) Das CallManager-Zertifikat der Quellcluster wird im CallManager-Vertrauens- und Phone-SAST-Trust-Speicher in das Ziel-Cluster hochgeladen. Dadurch können die Telefone zurück zum Quell-Cluster migriert werden. Wenn die manuelle Methode befolgt wird, wird das CallManager-Zertifikat des Quellclusters aufgerufen nicht in das Ziel-Cluster hochgeladen werden. Das bedeutet, dass Sie die Telefone nicht zurück zum Quell-Cluster migrieren können. Wenn Sie die Möglichkeit haben möchten, die Telefone zurück zum Quell-Cluster zu migrieren, Sie müssen das CallManager-Zertifikat der Quellcluster-Cluster in die Zielcluster CallManager-trust und Phone-SAST-trust-Store hochladen.

> **Hinweis**: Beide Cluster müssen das TFTP-Zertifikat auf denselben SFTP-Server und dasselbe SFTP-Verzeichnis exportieren.

> **Hinweis**: Schritt 4 ist nur für einen Cluster erforderlich. Wenn Sie Telefone zwischen CUCM 8.x oder 9.x und CUCM 10.5.2.13900-12 oder neuer migrieren, notieren Sie sich die Cisco Bug-ID CSCuy43181, bevor Sie die Zertifikate konsolidieren.

Methode 2.

Importieren Sie die Zertifikate manuell. Führen Sie diese Schritte für das Ziel-Cluster aus.

Schritt 1: Navigieren Sie zu **Cisco Unified OS Administration** Seite > **Security** > **Certificate Management**.

Schritt 2: Wählen Sie das Zertifikat CallManager.pem aus, und laden Sie es herunter.

Schritt 3: Wählen Sie das ITLrecovery.pem-Zertifikat aus, und laden Sie es herunter

Schritt 4: Laden Sie das CallManager-Zertifikat als CallManager-Vertrauenszertifikat und Telefon-SAST-Vertrauenszertifikat in den Herausgeber des Quellclusters hoch.

Schritt 5: Laden Sie das ITL-Wiederherstellungszertifikat als Phone-SAST-Trust in den Quellcluster hoch.

Schritt 6: Starten Sie TVS in allen Knoten vom Quell-Cluster neu.

Anschließend replizieren die Zertifikate an die anderen Knoten im Cluster.

Die Schritte 3, 5 und 6 gelten für Szenarien der Migration des Telefons von 8.x auf 12.x.

> **Hinweis**: Das CallManager-Zertifikat muss von allen Knoten heruntergeladen werden, auf denen der TFTP-Dienst auf dem Ziel-Cluster ausgeführt wird.

Nachdem die Zertifikate mit einer der oben genannten Methoden hochgeladen wurden, ändern Sie die DHCP-Option 150 (Dynamic Host Configuration Protocol) der Telefone, um auf die TFTP-Adresse des Ziel-Clusters zu zeigen.

> **Vorsicht**: Eine Möglichkeit, Telefone zwischen nicht sicheren Clustern zu migrieren, besteht darin, "Prepare Cluster for Rollback to pre 8.0" (Cluster für Rollback auf Version 8.0 vorbereiten) auf True im Quell-Cluster festzulegen und die Telefone neu zu starten. Dies ist keine Option, wenn Sie Telefone zwischen sicheren Clustern migrieren. Dies liegt daran, dass beim Rollback zur Funktion vor 8.0 nur die ITL-Datei gelöscht wird (die CTL-Datei wird nicht leer angezeigt). Das bedeutet, dass das Telefon bei der Migration und beim Herunterladen der CTL-Datei aus dem Ziel-Cluster die neue CTL mit den Quell-Clustern TVS verifizieren muss. Da die ITL-Datei des Telefons nicht das TVS-Zertifikat der Quellcluster enthält, schlägt der Handshake fehl, wenn das Telefon versucht, eine sichere Verbindung zum TVS-Dienst herzustellen.

Überprüfen

Dies ist ein Auszug aus den Telefonkonsolenprotokollen und den TVS-Protokollen (auf "detailliert" eingestellt) des Quell-Clusters. Die Ausschnitte zeigen den Prozess der Telefonregistrierung für das Ziel-Cluster an.

1. Das Telefon startet und lädt die CTL-Datei vom Ziel-Cluster herunter.

```
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. Die CTL-Datei wird vom Anrufverwaltungs-Zertifikat der Zielcluster signiert, das sich nicht in der vorhandenen CTL- oder ITL-Datei der Telefone befindet. Das bedeutet, dass das Telefon sich an

seinen TVS-Dienst wenden muss, um das Zertifikat zu überprüfen. An diesem Punkt hat das Telefon noch seine alte Konfiguration, die die IP-Adresse des Quell-Cluster-TVS-Dienstes enthält (der in der Telefonkonfiguration angegebene TVS entspricht der Telefonanrufverwaltungs-Gruppe). Das Telefon richtet eine SSL-Verbindung zum TVS-Dienst ein. Wenn der TVS-Dienst dem Telefon sein Zertifikat vorlegt, verifiziert das Telefon das Zertifikat anhand des Zertifikats in seiner ITL-Datei. Wenn sie identisch sind, wird der Handshake erfolgreich abgeschlossen.

```
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. Die TVS-Protokolle zeigen die eingehende Verbindung vom Telefon an, und der Handshake war erfolgreich.

```
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. Die Telefonkonsolenprotokolle zeigen an, dass das Telefon eine Anforderung an den TVS-Dienst sendet, um das Anrufmanager-Zertifikat vom Ziel-Cluster zu überprüfen.

```
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. Die TVS-Protokolle zeigen an, dass die Anfrage empfangen wurde.

```
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. Die TVS-Protokolle zeigen das Zertifikat in seinem Geschäft an, und der TVS sendet eine Antwort an das Telefon.

```
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. Die Telefonkonsolenprotokolle zeigen an, dass das Zertifikat erfolgreich verifiziert und die CTL-Datei aktualisiert wurde.

```
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. Die Telefonkonsolenprotokolle werden angezeigt, wenn das Telefon seine ITL-Datei herunterlädt.

```
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. Die ITL-Datei wird mit der CTL-Datei verifiziert. Die CTL-Datei enthält das CallManager-Zertifikat der Zielcluster. Das bedeutet, dass das Telefon das Zertifikat überprüfen kann, ohne sich an den TVS-Dienst des Quellclusters zu wenden.

```
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.
```

Fehlerbehebung

Überprüfen Sie vor dem Migrationsprozess die CTL/ITL auf den Telefonen. Weitere Informationen zur Überprüfung von CTL/ITL finden Sie hier: https://www.cisco.com/c/en/us/support/docs/voice-unified-communications/unified-communications-manager-callmanager/116232-technote-sbd-00.html#anc9