セキュアクラスタ間での電話機の移行

内容

概要

このドキュメントでは、2つのセキュアなCisco Unified Communications Manager(CUCM)クラスタ間で電話機を移行する方法について説明します。

著者：Cisco TACエンジニア、David Norman

前提条件

要件

CUCM について十分に理解しておくことをお勧めします。

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアのバージョンに基づいています。

ソースクラスタ：CUCM バージョン 10.5.2.11900-3
宛先クラスタ： CUCM バージョン 11.0.1.10000-10
ファームウェアsip88xx.10-3-1-20を使用する8861電話機

CertificateTrust List(CTL)ファイルは、CallManager証明書で署名されています（USBトークンではありません）

背景

移行プロセス中に、電話機はソースクラスタCisco Trust Verification Service(TVS)へのセキュアな接続をセットアップして、宛先クラスタのCallManager証明書を検証します。電話機の証明書信頼リスト(CTL)およびID信頼リスト(ITL)ファイルが無効な場合、電話機はTVSとのセキュアハンドシェイクを完了できず、宛先クラスタへの移行は成功しません。電話機の移行プロセスを開始する前に、電話機に正しいCTL/ITLファイルがインストールされていることを確認します。また、ソースクラスタで、エンタープライズ機能[Prepare Cluster for Rollback to Pre 8.0]が[False]に設定されていることを確認します。

設定

宛先クラスタのCallManager証明書をソースクラスタのCallManager-trustおよびPhone-SAST-trustストアにインポートします。これを行うには2つの方法があります。

方式 1.

証明書の一括処理ツールを使用して、送信元クラスタと宛先クラスタの両方で次の手順を実行します。

ステップ1:[**Cisco Unified OS Administration**]ページ> [**Security**] > [**Bulk Certificate Management**]の順に移動し**ます。**

ステップ2:Secure File Transfer Protocol(SFTP)サーバの詳細を入力し、[**Save**]を選択します。

ステップ3:[**Export**]を選択し、トリビアルファイル転送プロトコル(TFTP)証明書をエクスポートします。

ステップ4:[Consolidate]ボタンをクリックし**て証明書**の統合を実行します。 これにより、送信元と宛先の両方のCallManager証明書を含むPKCS12ファイルが作成されます。

ステップ5：統合証明書を各クラスタにインポートし直します。

統合プロセス（ステップ5）では、 ソースクラスタCallManager証明書は、CallManager-trustおよびPhone-SAST-trustストアの宛先クラスタにアップロードされます。これにより、電話機をソースクラスタに戻すことができます。手動の方法に従うと、ソースクラスタのCallManager証明書が 勝った 宛先クラスタにアップロードされます。つまり、電話機をソースクラスタに戻すことはできません。電話機をソースクラスタに戻すオプションを使用する場合は、 ソースクラスタCallManager証明書を宛先クラスタCallManager-trustおよびPhone-SAST-trust storeにアップロードする必要があります。

> **注**：両方のクラスタが、同じSFTPサーバと同じSFTPディレクトリにTFTP証明書を工クスポートする必要があります。

> **注**：ステップ4は1つのクラスタでのみ必要です。CUCMバージョン8.xまたは9.xからCUCMバージョン10.5.2.13900-12以降に電話機を移行する場合は、証明書を統合する前に、このCisco Bug ID CSCuy43181をメモしてください。

方式 2.

証明書を手動でインポートします。宛先クラスタで次の手順を実行します。

ステップ1:[**Cisco Unified OS Administration**]ページ> [**Security**] > [**Certificate Management**]に移動し**ます。**

ステップ2:CallManager.pem証明書を選択し、ダウンロードします。

ステップ3:ITLrecovery.pem証明書を選択し、ダウンロードします

ステップ4:CallManager証明書をCallManager-trust証明書およびPhone-SAST-trust証明書としてソ

ースクラスタパブリッシャにアップロードします。

ステップ5:ITLrecovery証明書をPhone-SAST-Trustとしてソースクラスタにアップロードします

ステップ6：ソースクラスタのすべてのノードでTVSを再起動します。

次に、証明書はクラスタ内の他のノードに複製されます。

手順3、5、6は、電話機を8.xから12.xに移行するシナリオに適用されます

> **注**：CallManager証明書は、宛先クラスタでTFTPサービスを実行しているすべてのノードからダウンロードする必要があります。

上記のいずれかの方法で証明書がアップロードされたら、電話機のDynamic Host Configuration Protocol(DHCP)オプション150を、宛先クラスタのTFTPアドレスをポイントするように変更します。

> **注意**：非セキュアクラスタ間で電話機を移行する1つの方法は、ソースクラスタで[Prepare Cluster for Rollback to pre 8.0]を[True]に設定し、電話機を再起動することです。これは、セキュアなクラスタ間で電話機を移行する場合のオプションではありません。これは、8.0より前へのロールバック機能ではITLファイルが空白になるだけであるためです（CTLファイルは空白になりません）。 つまり、電話機が移行され、宛先クラスタからCTLファイルをダウンロードする際には、送信元クラスタTVSで新しいCTLを確認する必要があります。電話機のITLファイルにはソースクラスタのTVS証明書が含まれていないため、電話機がTVSサービスへのセキュアな接続を確立すると、ハンドシェイクが失敗します。

確認

これは、電話コンソールログとソースクラスタのTVSログ（詳細に設定）からの抜粋です。スニペットは、宛先クラスタへの電話登録プロセスを示します。

1.電話機が起動し、宛先クラスタからCTLファイルをダウンロードします。

```
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. CTLファイルは、電話機の既存のCTLファイルまたはITLファイルにない宛先クラスタcall manager証明書によって署名されます。これは、証明書を確認するために、電話機がTVSサービスにアクセスする必要があることを意味します。この時点で、電話機には引き続き元のクラスタTVSサービスのIPアドレスを含む古い設定があります（電話機の設定で指定されたTVSは、電話機のコールマネージャグループと同じです）。 電話機は、TVSサービスへのSSL接続をセットアップします。TVSサービスがその証明書を電話機に提示すると、電話機はITLファイルの証明書と照合して証明書を検証します。同じ場合、ハンドシェイクは正常に完了します。

```
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. TVSログには、電話機からの着信接続とハンドシェイクが成功したことが示されます。

```
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.電話コンソールログには、電話機が宛先クラスタからのCall Manager証明書を確認する要求を

TVSサービスに送信していることが示されます。

```
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. TVSログには、要求が受信されたことが示されます。

```
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. TVSログにストア内の証明書が表示され、TVSは電話に応答を送信します。

```
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.電話コンソールのログには、証明書が正常に検証され、CTLファイルが更新されたことが示されます。

```
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.電話がITLファイルをダウンロードすると、電話コンソールのログが表示されます。

```
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. ITLファイルがCTLファイルと照合されます。CTLファイルには、宛先クラスタのCallManager証明書が含まれます。つまり、電話機はソースクラスタTVSサービスに接続せずに証明書を確認できます。

```
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.
```

トラブルシュート

移行プロセスの前に、電話機のCTL/ITLを確認します。CTL/ITLの検証方法の詳細については、次のサイトを参照してください。 https://www.cisco.com/c/en/us/support/docs/voice-unified-communications/unified-communications-manager-callmanager/116232-technote-sbd-00.html#anc9