

在安全的簇之间的迁移电话

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[背景](#)

[Configure](#)

[Verify](#)

[Troubleshoot](#)

Introduction

本文描述如何移植电话在两安全的Cisco Unified通信管理器(CUCM)簇之间。

贡献由大卫诺曼底人， Cisco TAC工程师。

Prerequisites

Requirements

Cisco建议您有CUCM知识。

Components Used

本文档中的信息基于以下软件版本：

来源簇：CUCM版本10.5.2.11900-3

目的地集群：CUCM版本11.0.1.10000-10

8861电话使用固件sip88xx.10-3-1-20

CertificateTrust列表(CTL)文件签字与呼叫管理器认证(不是USB令牌)

背景

在迁移进程中，电话尝试设置对来源簇Cisco信任验证服务(TV)的一个安全连接验证目的地集群呼叫管理器认证。如果电话的证书信任列表(CTL)和身份信任列表(ITL)文件无效，电话不能完成与TV的安全的握手，并且对目的地集群的迁移不会成功。在您开始电话迁移进程前，请确认电话有安装的正确CTL/ITL文件。并且在来源簇，请确认企业功能“准备簇回退到前8.0”设置对错误。

Configure

导入目的地集群呼叫管理器认证到来源簇呼叫管理器信任和电话SAST信任存储。有要执行此的两

个方法。

方法1。

请使用大批认证工具并且完成在来源和目的地集群的这些步骤。

步骤1.连接对Cisco Unified OS管理页面> Security >在来源和目的地集群的容量证书管理。

步骤2.输入安全文件传输协议(SFTP)服务器的详细资料并且选择保存。

步骤3.选择导出并且导出简单文件传输协议(TFTP)认证。

步骤4.点击统一按钮执行认证合并。这创建包括来源和目的地呼叫管理器认证的一个PKCS12文件。

步骤5.导入统一的证书回到每簇。

在合并流程中(第5)步，来源簇呼叫管理器认证被加载到目的地集群在呼叫管理器信任和电话SAST信任存储。这允许电话移植回到来源簇。如果手工方法被按照，来源簇呼叫管理器认证不会被加载到目的地集群。这意味着您不能移植电话回到来源簇。如果希望选项移植回到来源的电话集群，您需要加载来源簇呼叫管理器认证到目的地集群呼叫管理器信任和电话SAST信任存储。

Note:两簇必须对同一个SFTP服务器和同一个SFTP目录导出TFTP认证。

Note:第4步在一簇只需要。如果在CUCM版本8.x或9.x之间的migratephones对CUCM版本10.5.2.13900-12或以上，注意到此Cisco Bug ID [CSCuy43181](#)，在您统一证书前。

方法2。

请手工导入证书。完成在目的地集群的这些步骤。

步骤1.连接对Cisco Unified OS管理页面> Security > Certificate Management。

步骤2.选择CallManager.pem认证并且下载它。

步骤3.选择ITLrecovery.pem认证并且下载它

步骤4.加载呼叫管理器认证到来源簇发布人作为CallManger信任和电话SAST信任认证。

步骤5.加载ITLrecovery认证到来源簇作为电话SAST信任

步骤6.重新启动在所有节点的TV从来源簇。

然后对其他节点的证书复制品在簇。

第3步，5，6将适用于移植电话方案从8.x到12.x

Note:呼叫管理器认证需要从运行在目的地集群的所有节点下载TFTP服务。

一旦证书加载了与其中一个上述方法，请更改电话动态主机配置协议(DHCP)选项150指向目的地集

群TFTP地址。

警告：移植介于中间的不安全的簇是设置“的电话的一个方法准备簇回退到前8.0”对真在来源簇并且重新启动电话。当您移植电话在安全的簇之间时，这不是选项。这是因为对前8.0功能的回退只成为空白ITL文件(不成为空白CTL文件)。这意味着，当电话被移植时，并且从目的地集群下载CTL文件，需要验证与来源簇TV的新的CTL。因为电话的ITL文件不包含来源集群TV认证，握手发生故障，当电话设法建立对TV服务时的一个安全连接。

Verify

这是摘自电话控制台日志和TV日志的一个部分(请设置对详细)来源簇。片断显示电话注册的进程目的地集群。

1. 电话从目的地集群引导并且下载CTL文件。

```
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文件由不在电话现有的CTL或ITL文件的目的地集群呼叫管理器认证签字。这意味着电话需要提供援助到其TV服务验证认证。这时电话仍然有包含来源簇TV服务的IP地址的其旧的配置(在电话配置指定的TV是作为电话管理器组)的相同的。电话设置与TV服务的SSL连接。当TV服务提交其认证到电话时，电话验证认证在其ITL文件的认证。如果他们是相同的，握手成功地完成。

```
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. TV日志显示从电话和握手的流入的连接是成功的。

```
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 added 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 added 8 to readset
18:01:05.855 | debug Recd event
18:01:05.855 | debug TLS HS Done for ph_conn
```

4. 电话控制台日志显示电话发送一个请求到TV服务验证从目的地集群的呼叫管理器认证。

```
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. TV日志显示请求收到。

```
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. TV日志显示在其存储和TV的认证发送对电话的一种回应。

```

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. 电话控制台日志表示，certifcate顺利地验证，并且CTL文件是更新的。

```
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. 当电话下载其ITL文件，电话控制台日志显示。

```
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文件被验证CTL文件。CTL文件包含目的地集群呼叫管理器认证。这意味着电话能验证认证，无需与来源簇TV服务联系。

```
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

在迁移进程前，请验证在电话的CTL/ITL。可以找到关于如何的更多信息验证CTL/ITL这里：
<https://www.cisco.com/c/en/us/support/docs/voice-unified-communications/unified-communications-manager-callmanager/116232-technote-sbd-00.html#anc9>