CUCM安全默认情况下和ITL操作和故障排除

目录

简介

默认情况下本文描述Cisco Unified Communications Manager (CUCM)版本8.0和以上安全(SBD)功能。默认情况下本文担当补充对于正式安全文档，并且提供操作信息和故障排除提示帮助管理员和缓和故障排除流程。

背景信息

CUCM版本8.0和以上介绍SBD功能，包括标识信任列表(ITL)文件和信任验证服务(TV)。每CUCM集群自动地当前使用基于ITL的安全。有在安全和管理员一定知道管理之间的易用/方便的一个折衷方案，在他们做对版本8.0 CUCM的某些变动集群前。

它是一个好想法熟悉SBD的这些核心概念：不对称关键加密算法维基百科条款和公共钥匙结构维基百科条款。

SBD概述

此部分提供概述正确地什么SBD提供。关于每个功能技术详细资料，请参阅SBD详细信息和故障排除信息信息部分。

SBD为支持的IP电话提供这三个功能：

- 默认验证TFTP下载的文件(配置、现场， ringlist)该使用一签署的密钥
- 的TFTP配置文件可选加密使用一签署的密钥
- 使用CUCM的电话启动的HTTPS连接的证书验证(TV)远程证书信任存储

本文提供这些功能中的每一的概述个。

TFTP下载验证

当证书信任列表(CTL)或ITL文件存在时， IP电话请求从CUCM TFTP server的一个签字的TFTP配置文件。此文件允许电话验证配置文件来自可信的源。使用CTL/ITL文件在电话，必须由委托TFTP server签字配置文件。文件是在网络的纯文本，当传送时，但是附有一个特殊验证签名。

电话请求**SEP *<MAC地址>.cnf.xml.sgn*为了**接收有特殊签名的配置文件。此配置文件由对应于在操作系统(OS)管理证书管理页的CallManager.pem的TFTP专用密钥签字。

签字的文件有一个签名在顶部为了验证文件，但是否则在纯文本XML。下面的镜像显示配置文件的签署人是**CN=CUCM8-Publisher.bbbburns.lab**哪些反过来由**CN=JASBURNS-AD**签字。这意味着电话需要验证**CUCM8-Publisher.bbbburns.lab**签名ITL文件在此配置文件前接受。

这是显示的图表专用密钥如何与消息分类算法一起使用(MD)5或安全散列算法(SHA)1散列函数为了创建签字的文件。

签名验证通过配比为了解密哈希的使用公共密钥倒转此进程。如果切细匹配，显示：

- 在运送中未修改此文件。
- 因为一定加密用公共密钥顺利地解密的任何与专用密钥，此文件来自在签名列出的当事人。

TFTP配置文件加密

如果可选TFTP配置加密在相关的电话安全配置文件启用，电话请求一个已加密配置文件。此文件签字与TFTP专用密钥并且用对称密钥加密被交换在电话和CUCM之间(参考Cisco Unified Communications Manager安全指南，请发布8.5(1)关于全面的详细信息)，以便其内容不可能读用网络嗅探器，除非观察员有必要的密钥。

电话请求**SEP *<MAC地址>.cnf.xml.enc.sgn*为了**获得签字的已加密文件。

已加密配置文件首先有签名，但是没有纯文本数据以后，只有已加密数据(在此文本编辑的被错误的二进制字符)。镜像显示签署人是同一样在前一个示例，因此此签署人一定是存在ITL文件，在电话接受文件前。进一步，在电话能读文件前的内容解密密钥一定正确。

托拉斯验证服务(远程证书和签名验证)

IP电话包含有限的内存，并且可以也有管理的很大数量的电话在网络。CUCM作为远程信任存储通过TV，以便全双工证书信任存储在每个IP电话不必须被放置。电话不能任何时候验证签名或证书通过CTL或ITL文件，要求验证的TV服务器。如果信任存储是存在所有IP电话，此中央信任存储是更加容易管理比。

SBD详细信息和故障排除信息

此部分选派SBD进程。

ITL文件和证书在CUCM提交

首先，有一定是存在CUCM服务器的一定数量的文件。最重要的片段是TFTP证书和TFTP专用密钥。TFTP证书查找在**OS管理> Security > Certificate Management > CallManager.pem下。**

CUCM服务器使用私有CallManager.pem的证书的和公共密钥TFTP服务(以及为Cisco Call Manager (CCM)服务)。镜像显示CallManager.pem证书发出对JASBURNS-AD**签字的**CUCM8-publisher.bbbburns.laband**。所有**TFTP配置文件由下面专用密钥签字。

所有电话在CallManager.pem证书能使用TFTP公共密钥为了解密所有文件加密与TFTP专用密钥，以及验证任何文件签字与TFTP专用密钥。

除CallManager.pem证书专用密钥之外， CUCM服务器也存储被提交到电话的ITL文件。showitl**命令**通过对CUCM服务器OS CLI的安全壳SSH访问显示此ITL文件全部内容。

此部分划分ITL文件部分由部分，因为有电话使用的一定数量的重要组件。

第一部分是签名信息。ITL文件是一个签字的文件。此输出显示由关联与上一个CallManager.pem证书的TFTP专用密钥签字。

```
admin:show itl
Length of ITL file: 5438
The ITL File was last modified on Wed Jul 27 10:16:24 EDT 2011

       Parse ITL File
       ----------------

Version:        1.2
HeaderLength:   296 (BYTES)

BYTEPOS TAG             LENGTH  VALUE
------- ---             ------  -----
3       SIGNERID        2       110
4       SIGNERNAME      76      CN=CUCM8-Publisher.bbbburns.lab;
                                OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
5       SERIALNUMBER    10      21:00:2D:17:00:00:00:00:00:05
6       CANAME          15      CN=JASBURNS-AD
*Signature omitted for brevity*
```

以下部分其中每一包含他们的目的在特殊功能参数里面。第一个功能是系统管理员安全标记。这是TFTP公共密钥的签名。

```
        ITL Record #:1
                 ----
BYTEPOS TAG             LENGTH  VALUE
```

```
------- ---              ------  -----
1       RECORDLENGTH     2       1972
2       DNSNAME          2
3       SUBJECTNAME      76      CN=CUCM8-Publisher.bbbburns.lab;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4       FUNCTION         2       System Administrator Security Token
5       ISSUERNAME       15      CN=JASBURNS-AD
6       SERIALNUMBER     10      21:00:2D:17:00:00:00:00:00:05
7       PUBLICKEY        140
8       SIGNATURE        256
9       CERTIFICATE      1442    0E 1E 28 0E 5B 5D CC 7A 20 29 61 F5
                                 8A DE 30 40 51 5B C4 89 (SHA1 Hash HEX)
This etoken was used to sign the ITL file.
```

下个功能是CCM+TFTP。这再是服务验证和解密下载的TFTP配置文件的TFTP公共密钥。

```
        ITL Record #:2
                  ----
BYTEPOS TAG              LENGTH  VALUE
------- ---              ------  -----
1       RECORDLENGTH     2       1972
2       DNSNAME          2
3       SUBJECTNAME      76      CN=CUCM8-Publisher.bbbburns.lab;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4       FUNCTION         2       CCM+TFTP
5       ISSUERNAME       15      CN=JASBURNS-AD
6       SERIALNUMBER     10      21:00:2D:17:00:00:00:00:00:05
7       PUBLICKEY        140
8       SIGNATURE        256
9       CERTIFICATE      1442    0E 1E 28 0E 5B 5D CC 7A 20 29 61 F5
                                 8A DE 30 40 51 5B C4 89 (SHA1 Hash HEX)
```

下个功能是TV。有电话接通每个TV服务器的公共密钥的一个条目。这允许电话建立安全套接字协议层(SSL)会话到TV服务器。

```
        ITL Record #:3
                  ----
BYTEPOS TAG              LENGTH  VALUE
------- ---              ------  -----
1       RECORDLENGTH     2       743
2       DNSNAME          2
3       SUBJECTNAME      76      CN=CUCM8-Publisher.bbbburns.lab;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4       FUNCTION         2       TVS
5       ISSUERNAME       76      CN=CUCM8-Publisher.bbbburns.lab;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
6       SERIALNUMBER     8       2E:3E:1A:7B:DA:A6:4D:84
7       PUBLICKEY        270
8       SIGNATURE        256
11      CERTHASH         20      C7 E1 D9 7A CC B0 2B C2 A8 B2 90 FB
                                 AA FE 66 5B EC 41 42 5D
12      HASH ALGORITHM   1       SHA-1
```

在ITL文件包括的最终功能是认证机关代理功能(CAPF)。此证书允许电话建立对CAPF服务的一个安全连接在CUCM服务器，以便电话能安装或更新一局部重要的证书(LSC)。此进程在将发布的另一个文档将报道。

```
        ITL Record #:4
                  ----
BYTEPOS TAG              LENGTH  VALUE
------- ---              ------  -----
1       RECORDLENGTH     2       455
2       DNSNAME          2
3       SUBJECTNAME      61      CN=CAPF-9c4cba7d;
```

```
                                OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4       FUNCTION        2       CAPF
5       ISSUERNAME      61      CN=CAPF-9c4cba7d;
                                OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
6       SERIALNUMBER    8       0A:DC:6E:77:42:91:4A:53
7       PUBLICKEY       140
8       SIGNATURE       128
11      CERTHASH        20      C7 3D EA 77 94 5E 06 14 D2 90 B1
                                A1 43 7B 69 84 1D 2D 85 2E
12      HASH ALGORITHM  1       SHA-1

The ITL file was verified successfully.
```

确切下一部分盖板什么发生，当电话启动。

电话下载ITL和配置文件

在电话启动并且得到IP地址以及TFTP server的地址后，首先请求CTL和ITL文件。

此数据包捕获显示一个电话要求ITL文件。如果在**tftp.opcode == 1**过滤，您看到每TFTP读从电话的请求：

因为电话顺利地接收从TFTP的CTL和ITL文件，电话请求一个签字的配置文件。显示此行为的电话控制台日志从电话的Web接口是可得到：

首先电话请求CTL文件，成功：

```
837: NOT 09:13:17.561856 SECD: tlRequestFile: Request CTLSEP0011215A1AE3.tlv
846: NOT 09:13:17.670439 TFTP: [27]:Requesting CTLSEP0011215A1AE3.tlv from
14.48.44.80
847: NOT 09:13:17.685264 TFTP: [27]:Finished --> rcvd 4762 bytes
```

其次电话也请求ITL文件：

```
868: NOT 09:13:17.860613 TFTP: [28]:Requesting ITLSEP0011215A1AE3.tlv from
14.48.44.80
869: NOT 09:13:17.875059 TFTP: [28]:Finished --> rcvd 5438 bytes
```

电话验证ITL和配置文件

在ITL文件下载后，必须验证。有很多位阐明，电话可以这时，因此本文包括他们全部。

- 电话没有CTL或ITL文件存在或ITL是空白的由于**准备团星回退对前8.0**参数。在此状态下，电话盲人委托下载的下个CTL或ITL文件并且从此使用此签名。
- 电话已经没有一个CTL，但是ITL。在此状态下，如果可以由在CTL文件的CCM+TFTP功能验证电话只委托ITL。
- 电话已经有一个CTL和一个ITL文件。在此状态下，电话验证下载的文件最近匹配在CTL、ITL或者TV服务器的签名。

这是描述的流程图电话如何验证签字的文件和HTTPS证书：

在这种情况下，电话能验证在ITL和CTL文件的签名。电话已经有一个CTL和ITL，因此检查他们并且查找正确签名。

```
877: NOT 09:13:17.925249 SECD: validate_file_envelope:
File sign verify SUCCESS; header length <296>
```

因为电话下载CTL和ITL文件，只从这时起请求签字的配置文件。这说明电话的逻辑是确定TFTP server根据CTL和ITL出现是安全，然后请求一个签字的文件：

```
917: NOT 09:13:18.433411 tftpClient: tftp request rcv'd from /usr/tmp/tftp,
srcFile = SEP0011215A1AE3.cnf.xml, dstFile = /usr/ram/SEP0011215A1AE3.cnf.xml
max size = 550001
918: NOT 09:13:18.457949 tftpClient: auth server - tftpList[0] = ::ffff:
14.48.44.80
919: NOT 09:13:18.458937 tftpClient: look up server - 0
920: NOT 09:13:18.462479 SECD: lookupCTL: TFTP SRVR secure
921: NOT 09:13:18.466658 tftpClient: secVal = 0x9 922: NOT 09:13:18.467762
tftpClient: ::ffff:14.48.44.80 is a secure server
923: NOT 09:13:18.468614 tftpClient: retval = SRVR_SECURE
924: NOT 09:13:18.469485 tftpClient: Secure file requested
925: NOT 09:13:18.471217 tftpClient: authenticated file approved - add .sgn
-- SEP0011215A1AE3.cnf.xml.sgn
926: NOT 09:13:18.540562 TFTP: [10]:Requesting SEP0011215A1AE3.cnf.xml.sgn
from 14.48.44.80 with size limit of 550001
927: NOT 09:13:18.559326 TFTP: [10]:Finished --> rcvd 7652 bytes
```

一旦签字的配置文件下载，电话必须利用CCM+TFTP的功能验证它在ITL里面：

```
937: NOT 09:13:18.656906 SECD: verifyFile: verify SUCCESS
</usr/ram/SEP0011215A1AE3.cnf.xml>
```

电话与未知证书的TV联系

ITL文件提供包含TV服务证书在CUCM服务器TCP端口2445运作的一个TV功能。TV在CallManager服务被启动的所有服务器运行。CUCM TFTP服务使用已配置的Callmanager组为了创建的TV服务器列表电话在电话配置文件应该联系。

一些实验室使用仅单个CUCM服务器。在多节点CUCM集群，在电话的CUCM组中可以有电话的三个TV条目，一个每CUCM的。

此示例显示发生了什么，当在IP电话的**目录按键**按时。目录URL为HTTPS配置，因此电话提交与从目录服务器的Tomcat Web证书。此Tomcat Web证书(tomcat.pem在OS管理中)在电话，因此电话没有装载必须与TV联系为了验证证书。

参考交互作用的说明的上一个TV概述图表。这是电话console log前景：

首先您查找目录URL ：

```
1184: NOT 15:20:55.219275 JVM: Startup Module Loader|cip.dir.TandunDirectories:
? - Directory url https://14.48.44.80:8443/ccmcip/xmldirectory.jsp
```

这是需要验证的SSL/Transport层安全(TLS)安全HTTP会话。

```
1205: NOT 15:20:59.404971 SECD: clpSetupSsl: Trying to connect to IPV4, IP:
14.48.44.80, Port : 8443
1206: NOT 15:20:59.406896 SECD: clpSetupSsl: TCP connect() waiting,
<14.48.44.80> c:8 s:9 port: 8443
1207: NOT 15:20:59.408136 SECD: clpSetupSsl: TCP connected,
<14.48.44.80> c:8 s:9
1208: NOT 15:20:59.409393 SECD: clpSetupSsl: start SSL/TLS handshake,
<14.48.44.80> c:8 s:9
1209: NOT 15:20:59.423386 SECD: srvr_cert_vfy: Server Certificate
Validation needs to be done
```

电话首先验证SSL/TLS服务器提交的证书是存在CTL。然后电话查看在ITL文件的功能为了发现是否查找一匹配。此错误消息说“在CTL的HTTPS cert不”，含义“证明不可能在CTL或ITL找到”。

```
1213: NOT 15:20:59.429176 SECD: findByCertAndRoleInTL: Searching TL from CTL file
1214: NOT 15:20:59.430315 SECD: findByCertAndRoleInTL: Searching TL from ITL file
1215: ERR 15:20:59.431314 SECD: EROR:https_cert_vfy: HTTPS cert not in CTL,
```

```
<14.48.44.80>
```
在CTL和ITL文件的直接内容被检查证书后，下件事电话检查是TV缓存。如果电话最近询问同一证书的， TV服务器这执行为了减少网络流量。如果HTTPS证书没有在电话缓存被找到，您能建立对TV服务器的TCP联系。

```
1220: NOT 15:20:59.444517 SECD: processTvsClntReq: TVS Certificate
Authentication request
1221: NOT 15:20:59.445507 SECD: lookupAuthCertTvsCacheEntry: No matching
entry found at cache
1222: NOT 15:20:59.446518 SECD: processTvsClntReq: No server sock exists,
must be created
1223: NOT 15:20:59.451378 SECD: secReq_initClient: clnt sock fd 11 bound
to </tmp/secClnt_secd>
1224: NOT 15:20:59.457643 SECD: getTvsServerInfo: Phone in IPv4 only mode
1225: NOT 15:20:59.458706 SECD: getTvsServerInfo: Retreiving IPv4 address
1230: NOT 15:20:59.472628 SECD: connectToTvsServer: Successfully started
a TLS connection establishment to the TVS server: IP:14.48.44.80, port:2445
(default); Waiting for it to get connected.
```
切记对TV的连接是SSL/TLS (安全HTTP或者HTTPS)，因此它也是需要验证CTL ot ITL的证书。如果一切正确地去，在ITL文件的TV功能应该找到TV服务器证明。请参阅ITL在前一个示例ITL文件的记录#3。

```
1244: NOT 15:20:59.529938 SECD: srvr_cert_vfy: Server Certificate Validation
needs to be done
1245: NOT 15:20:59.533412 SECD: findByIssuerAndSerialAndRoleInTL:
Searching TL from CTL file
1246: NOT 15:20:59.534936 SECD: findByIssuerAndSerialAndRoleInTL:
Searching TL from ITL file
1247: NOT 15:20:59.537359 SECD: verifyCertWithHashFromTL: cert hash and
hash in TL MATCH
1248: NOT 15:20:59.538726 SECD: tvs_cert_vfy: TVS cert verified with hash
from TL, <14.48.44.80>
```
成功!电话当前有对TV服务器的一个安全连接。下一步是要求TV服务器“Hello，我委托此目录服务器证书？”

此示例显示对回答该问题-答复0哪些含义成功(没有错误)。

```
1264: NOT 15:20:59.789738 SECD: sendTvsClientReqToSrvr: Authenticate
Certificate : request sent to TVS server - waiting for response
1273: NOT 15:20:59.825648 SECD: processTvsSrvrResponse: Authentication Response
received, status : 0
```
因为有从TV的一成功的答复，该证书的结果保存到缓存。这意味着，如果再按**目录按键**在以后86,400秒以内，您不需要联系TV服务器为了验证证书。您能访问本地缓存。

```
1279: NOT 15:20:59.837086 SECD: saveCertToTvsCache: Saving certificate
in TVS cache with default time-to-live value: 86400 seconds
1287: ERR 15:20:59.859993 SECD: Authenticated the HTTPS conn via TVS
```
最后，您验证您的对目录服务器的连接成功。

```
1302: ERR 15:21:01.959700 JVM: Startup Module Loader|cip.http.ae:?
- listener.httpSucceed:  https://14.48.44.80:8443/ccmcip/
xmldirectoryinput.jsp?name=SEP0011215A1AE3
```
这是什么的示例在TV运行的CUCM服务器发生。您能收集TV日志用Cisco Unified实时监控工具(RTMT)。

CUCM TV日志显示您SSL握手用电话，电话询问TV Tomcat证书，然后TV响应表明证书在TV证书存储匹配。

```
15:21:01.954 |   debug 14.48.44.202: tvsSSLHandShake Session ciphers - AES256-SHA
15:21:01.954 |   debug TLS HS Done for ph_conn .
15:21:02.010 |   debug     MsgType               : TVS_MSG_CERT_VERIFICATION_REQ
15:21:02.011 |   debug tvsGetIssuerNameFromX509 - issuerName : CN=CUCM8-
Publisher.bbbburns.lab;OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US and Length: 75

15:21:02.011 |   debug CertificateDBCache::getCertificateInformation -
Certificate compare return =0
15:21:02.011 |   debug CertificateDBCache::getCertificateInformation -
Certificate found and equal
15:21:02.011 |   debug     MsgType               : TVS_MSG_CERT_VERIFICATION_RES
```

TV证书存储是在**OS Administration > Certificate Management**网页包含的所有证书列表。

请手工验证电话ITL匹配CUCM ITL

被看到的一种常见的误解，当排除故障关系到倾向删除ITL文件以希望时将解决一文件验证问题。有时ITL文件删除要求，但是也许有一个更加好的方式。

当所有这些情况符合时， ITL文件只需要删除。

- ITL文件的签名在电话的不匹配ITL文件的签名在CM TFTP server的。
- 在ITL文件的TV签名不匹配TV提交的证书。
- 电话显示“失败的验证”，当它尝试下载ITL文件或配置文件。
- 备份不存在旧有TFTP专用密钥。

这是您如何检查前两个这些情况。

首先，您能比较ITL文件的校验和在CUCM用电话的校验和ITL文件。当前没有方式查看ITL文件的MD5sum在CUCM的从CUCM，直到您运行版本以此Cisco Bug ID的CSCto60209修正。

在此期间，请运行此以您的收藏夹GUI或CLI程序：

```
jasburns@jasburns-gentoo /data/trace/jasburns/certs/SBD $ tftp 14.48.44.80
tftp> get ITLSEP0011215A1AE3.tlv
Received 5438 bytes in 0.0 seconds
tftp> quit
jasburns@jasburns-gentoo /data/trace/jasburns/certs/SBD $ md5sum
ITLSEP0011215A1AE3.tlv
b61910bb01d8d3a1c1b36526cc9f2ddc  ITLSEP0011215A1AE3.tlv
```

这显示ITL文件的MD5sum在CUCM的是**b61910bb01d8d3a1c1b36526cc9f2ddc**。

现在您能查看电话为了确定装载的ITL文件的哈希那里：**设置> Security Configuration>托拉斯列表**。

这显示MD5sums匹配。这意味着在电话的ITL文件匹配在CUCM的文件，因此不需要删除。

如果它配比，您需要继续前进向下操作-请确定在ITL的TV证书是否匹配TV提交的证书。此操作是有点更加包含的。

首先，看看连接对TV在TCP端口2445的服务器电话的数据包捕获。

用鼠标右键单击在此数据流的所有数据包在Wireshark，点击**解码和**，并且选择**SSL**。查找如下所示:的服务器证书

查看在上一个ITL文件内包含的TV证书。您应该看到一个条目用序列号**2E3E1A7BDAA64D84**。

```
admin:show itl
       ITL Record #:3
                  ----
BYTEPOS TAG             LENGTH  VALUE
------- ---             ------  -----
1       RECORDLENGTH    2       743
2       DNSNAME         2
3       SUBJECTNAME     76      CN=CUCM8-Publisher.bbbburns.lab;
                                OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4       FUNCTION        2       TVS
5       ISSUERNAME      76      CN=CUCM8-Publisher.bbbburns.lab;
                                OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
6       SERIALNUMBER    8       2E:3E:1A:7B:DA:A6:4D:84
```

成功， **TVS.pem在**ITL文件里面匹配在网络提交的TV证书。您不需要删除ITL，并且TV提交正确证书。

如果文件验证仍然发生故障，请检查上一个流程图的其余。

限制和交互作用

再生证书/重建团星/证书到期

最重要的证书当前是CallManager.pem证书。此证书的专用密钥用于为了签署所有TFTP配置文件，包含ITL文件。

如果CallManager.pem文件被重新生成，一新的CCM+TFTP证书生成与一新的专用密钥。ITL文件由此当前另外签字新建的CCM+TFTP密钥。

在您重新生成CallManager.pem并且重新启动TV和TFTP服务后，这发生，当电话启动。

1. 电话尝试下载从TFTP server的新的CCM+TFTP签字的新的ITL文件。电话这时有仅旧有ITL文件，并且新的密钥不在ITL文件在电话。
2. 因为电话找不到在旧有ITL的新的CCM+TFTP签名，尝试与TV服务联系。
   注意：这部分是非常重要的。TV证书从旧有ITL文件必须仍然配比。如果CallManager.pem和TVS.pem被重新生成在同一个确切的时间，电话不能下载任何新的文件不手工删除从电话的ITL。
3. 当电话与TV联系时，运行TV的CUCM服务器有新的CallManager.pem证书在OS证书存储。
4. TV服务器返回成功，并且电话装载新的ITL文件到内存。
5. 电话当前尝试下载配置文件，由新的CallManager.pem密钥签了字。
6. 因为新的ITL装载，最近签字的配置文件由在内存的ITL顺利地验证。

关键点

- 同时请勿重新生成CallManager.pem和TVS.pem证书。
- 如果TVS.pem或CallManager.pem被重新生成，应该重新启动TV和TFTP并且给重置打电话为了得到新的ITL文件。CUCM新版本处理自动地重置的此电话并且警告用户在证书重新生成时间。
- 如果超过一个TV服务器存在(超过一个服务器在Callmanager组中)，另外的服务器能验证新的CallManager.pem证书。

在集群之间的移动电话

当您移动从一集群的电话到另一个与到位时ITLs，必须考虑到ITL和TFTP专用密钥。所有新配置文件被提交对电话必须匹配在CTL、ITL或者一个签名的一个签名在电话的当前TV服务中。

本文解释如何确保新的集群的ITL文件，并且配置文件可以由在电话的当前ITL文件委托。https://supportforums.cisco.com/docs/DOC-15799。

备份和恢复

CallManager.pem证书和专用密钥通过灾难恢复系统(DR)备份。如果TFTP server重建，必须从备份恢复，以便专用密钥可以恢复。没有在服务器的CallManager.pem专用密钥，使用旧有密钥有当前ITLs的电话不委托签字的配置文件。

如果集群从备份重建和没有恢复，就象“移动电话在集群之间”文档。这是因为与新密钥的一集群是不同的集群，就电话而言。

有一个严重的缺陷关联与备份和恢复。如果集群是易受Cisco Bug ID CSCtn50405，DR备份不包含CallManager.pem证书。这引起从此备份恢复的所有服务器生成损坏的ITL文件，直到新的CallManager.pem生成。如果没有没有通过备份和恢复操作的其他功能TFTP服务器，这也许含义所有ITL文件需要从电话删除。

为了验证，如果您的CallManager.pem文件需要被重新生成，请输入showitl**命令**遵从：

```
run sql select c.subjectname, c.serialnumber, c.ipv4address, t.name from
certificate as c, certificatetrustrolemap as r, typetrustrole as t where  c.pkid =
r.fkcertificate and t.enum = r.tktrustrole
```

在ITL输出中，寻找的关键错误是：

```
This etoken was not used to sign the ITL file.
```

并且

```
Verification of the ITL file failed.
Error parsing the ITL file!!
```

有“认证和授权角色的证书的上一个结构化查询语言(SQL)查询搜索”。在有认证和授权角色的上一个数据库查询的CallManager.pem证书应该也是存在OS管理证书管理网页。如果上一个缺陷遇到，有CallManager.pem证书之间的一不匹配在查询和在OS网页。

崔凡吉莱主机名或域名

如果更改CUCM服务器的主机名或域名，立即重新生成所有证书在该服务器。证书重新生成部分解释TVS.pem和CallManager.pem的重新生成是“坏事”。

有主机名更改发生故障的一些个方案和一些运作不出问题的地方。此部分包括所有并且连接他们回到什么您已经了解TV和ITL从此文档。

与仅ITL的单个节点团星(请当心，这中断，不用准备)

- 使用企业版本服务器或发行商部署，当您更改主机名时， CallManager.pem和TVS.pem同时被重新生成。
- 如果主机名在单个节点集群更改，不用首先使用报道的回退企业参数此处，电话不能验证新的ITL文件或配置文件他们的当前ITL文件。另外，因为TV证书是不再也委托，他们不能连接到

TV。
- 电话显示关于“托拉斯失败的列表验证的一个错误”，新的配置更改不生效，并且获取服务URL失败。
- 唯一的解决方案，如果在步骤2的注意事项不是被采取的第一将手工删除从每个电话的ITL。

与两CTL和ITL (可以临时地中断这的单个节点团星，但是容易地修复)

- 在您通过服务器后重命名运作，请重新运行CTL客户端。这在电话下载的CTL文件安置新的CallManager.pem证书。
- 新的配置文件，包含新的ITL文件，可以委托根据在CTL文件的CCM+TFTP功能。
- 这工作，因为更新CTL文件委托根据依然是同样的USB eToken专用密钥。

与仅ITL的多节点团星(这通常运作，但是可以永久被中断，如果仓促地执行)

- 由于多节点集群有多个TV服务器，所有单个服务器能有其被重新生成的证书不出问题。当电话提交与新建时的此，不熟悉的签名，请求别的TV服务器验证新的服务器证书。
- 有能造成此发生故障的两主要问题：
  如果所有服务器同时重命名并且重新启动， TV服务器都不是可及的与已知证书，当服务器和电话恢复时。如果电话有仅单个服务器在Callmanager组中，其他TV服务器不产生变化。请参阅“单个节点团星”方案为了解决此或者添加另一个服务器到电话的Callmanager组。

与两CTL和ITL (不可能永久中断这的)多节点团星

- 在您通过运作重命名， TV服务验证新的证书后。
- 由于某种原因即使所有TV服务器不可用， CTL客户端可能仍然使用为了更新有新的CallManager.pem CCM+TFTP证书的电话。

集中化TFTP

当有ITL的一个电话启动时，请求这些文件：**CTLSEP *< MAC地址>.tlv***、**ITLSEP *< MAC地址>.tlv***和**SEP *< MAC地址>.cnf.xml.sgn*。**

如果电话找不到这些文件，请求**ITLFile.tlv和**CTLFile.tlv，一集中化TFTP server提供给所有电话请求它。

使用集中化TFTP，有单个TFTP集群对一定数量的其他子集群的该点。通常这执行，因为在多CUCM集群的电话共享同样DHCP范围，并且必须有同样DHCP选项150 TFTP server。对中央TFTP集群的所有IP电话点，即使他们注册对其他集群。此中央TFTP server查询远程TFTP服务器，每当收到找不到的一个要求文件。

因此操作，集中化TFTP在ITL同种环境只运作。所有服务器必须运行CUCM版本8.x或以上，或者所有服务器必须在版本8.x之前运行版本。

如果ITLFile.tlv从集中化TFTP server被提交，电话不委托从远程TFTP服务器的任何文件，因为签名不配比。这在异种混合发生。在同类的混合，电话请求从正确远程集群被拉的ITLSEP **<MAC>.tlv**。

在与PRE版本8.x和版本8.x集群的混合的一个异构环境， “请准备回退的集群到前8.0"必须启用在版本8.x集群正如Cisco Bug ID CSCto87262和“获取的电话URL参数所描述”配置与HTTP而不是HTTPS。这有效禁用在电话的ITL功能。

常见问题

能否关闭SBD ？

如果SBD和ITL当前运作，您能只关闭SBD。

SBD在有准备的团星电话可以临时地禁用回退对前8.0"企业参数和通过配置“获取的电话URL参数”与HTTP而不是HTTPS。当您设置回退参数时，创建有空白的功能条目的一个签字的ITL文件。“空”ITL文件仍然签字，因此集群必须在一功能完备的安全状态，在此参数可以启用前。

在此参数启用后，并且有空白的条目的新的ITL文件下载并且验证，电话接受所有配置文件，不管谁签署了它。

没有在此状态推荐离开集群，因为以前metioned的三个功能都(已验证配置文件、已加密配置文件和HTTPS URL)不是可用的。

一旦CallManager.pem丢失，能否容易地删除从所有电话的ITL文件？

当前没有删除从思科远程提供的电话的所有ITLs的方法。所以在本文和交互作用描述的步骤是很重要考虑到。

当前有一未解决的增强对请求此功能的Cisco Bug ID CSCto47052，但是未实现。

在此期间期限，新特性通过也许允许Cisco技术支持中心(TAC)复原到以前委托ITL的Cisco Bug ID CSCts01319被添加了，如果是可用的在服务器。这在集群在一个版本以此故障修正的某些实例只运作，并且上一个ITL在服务器的地方一个特殊位置存储的备份存在。查看缺陷发现您的版本是否有修正。请与Cisco TAC联系为了通过在缺陷解释的潜在的恢复流程运行。

如果上一个步骤不是可用的，在电话必须手工按电话按钮为了删除ITL文件。这是做在管理之间安全和方便的折衷方案。为了ITL的文件能真安全，它不能远程容易地删除。

用脚本按钮按与简单对象访问协议(SOAP) XML对象， ITL不能远程删除。这是因为，这时， TV访问(和因而验证流入SOAP XML按钮推送对象的安全验证URL访问)是不运行的。如果验证URL没有配置如安全，写脚本密钥压入命令删除ITL也许是可能的，但是此脚本从思科不是可得到。

其他方法为了写脚本远程关键按，无需使用验证URL也许取得到从第三方，但是思科没有提供这些应用程序。

频繁地使用的方法为了删除ITL是提示他们键序列的电子邮件广播给所有电话用户。如果设置访问设置的**限制**或**已禁用**，电话需要是出厂重置，因为用户不访问电话的Settings菜单。