

CUCM-beveiliging door standaardinstelling en werking en probleemoplossing in ITL

Inhoud

[Inleiding](#)

[Achtergrondinformatie](#)

[SBD-Overzicht](#)

[TFTP-downverificatie](#)

[TFTP-configuratie van bestandsencryptie](#)

[Vertrouwelijkheidsverificatiedienst \(controle van certificaten en handtekeningen op afstand\)](#)

[Informatie over SBD-details en probleemoplossing](#)

[ITL-bestanden en -certificaten die op CUCM aanwezig zijn](#)

[Telefonische downloads - ITL- en configuratiebestand](#)

[Telefonisch verifieert ITL- en configuratiebestand](#)

[Telefonische contactgegevens TVS voor Onbekend certificaat](#)

[Controleer handmatig dat de telefoon/ITL overeenkomt met CUCM ITL](#)

[Beperkingen en interactie](#)

[Certificaten opnieuw genereren / een cluster opnieuw bouwen / verlopen volgens certificaat](#)

[Telefoons tussen clusters verplaatsen](#)

[Terug en herstellen](#)

[Host Names of Domain Names wijzigen](#)

[Gecentraliseerde TFTP](#)

[Veelgestelde vragen](#)

[Kan ik SBD uitzetten?](#)

[Kan ik het ITL bestand vanaf alle telefoons makkelijk wissen als CallManager.pem verloren is?](#)

Inleiding

Dit document beschrijft de Security By Default (SBD) optie van Cisco Unified Communications Manager (CUCM) versies 8.0 en hoger. Dit document vormt een aanvulling op de officiële [Security by Default-documenten](#) en biedt operationele informatie en tips voor het oplossen van problemen om beheerders te helpen en het proces voor het oplossen van problemen te vergemakkelijken.

Achtergrondinformatie

CUCM versie 8.0 en voegt later de SBD-functie toe, die bestaat uit bestanden van Identity Trust List (ITL) en de Trust Verification Service (TVS). Elk CUCM-cluster gebruikt nu automatisch op ITL gebaseerde beveiliging. Er is een wisselwerking tussen veiligheid en gebruiksgemak/beheergemak waar beheerders zich van bewust moeten zijn voordat ze bepaalde

wijzigingen aanbrengen in een CUCM-cluster van versie 8.0.

Het is een goed idee om kennis te maken met deze basisconcepten van SBD: [Asymmetric Key Cryptografie Wikipedia-artikel](#) en [Wikipedia-artikel over openbare infrastructuur](#).

SBD-Overzicht

Deze sectie verschaft een snel overzicht van precies wat SBD biedt. Zie voor alle technische details van elke functie het gedeelte INFORMATIE OVER Detail en probleemoplossing van de SBD.

SBD biedt deze drie functies voor ondersteunde IP-telefoons:

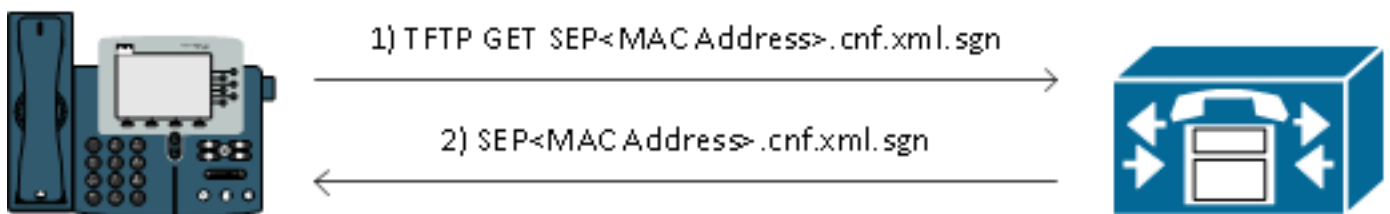
- Standaard authenticatie van TFTP gedownload bestanden (configuratie, locale, ringlist) die een gebarentekst gebruiken
- Optionele encryptie van TFTP-configuratiebestanden die gebruik maken van een signaalsleutel
- Verificatie van certificaten voor HTTPS-verbindingen die gebruik maken van een externe opslag van certificaten op CUCM (TVS)

Dit document geeft een overzicht van elk van deze functies.

TFTP-downverificatie

Wanneer een certificaatlijst (CTL) of ITL-bestand aanwezig is, verzoekt de IP-telefoon om een ondertekend TFTP-configuratiebestand van de CUCM TFTP-server. Met dit bestand kan de telefoon controleren of het configuratiebestand van een vertrouwde bron afkomstig is. Wanneer CTL/ITL-bestanden op telefoons aanwezig zijn, moeten de configuratiebestanden door een vertrouwde TFTP-server worden getekend. Het bestand is onbewerkte tekst op het netwerk terwijl het wordt verzonden, maar heeft een speciale verificatiehandtekening.

De telefoon vraagt om **SEP<MAC-adres>.cnf.xml.sgn** om het configuratiebestand met de speciale handtekening te ontvangen. Dit configuratiebestand is ondertekend door de TFTP-privé-toets die overeenkomt met CallManager.pem op de pagina Besturingssysteem (OS) Management.



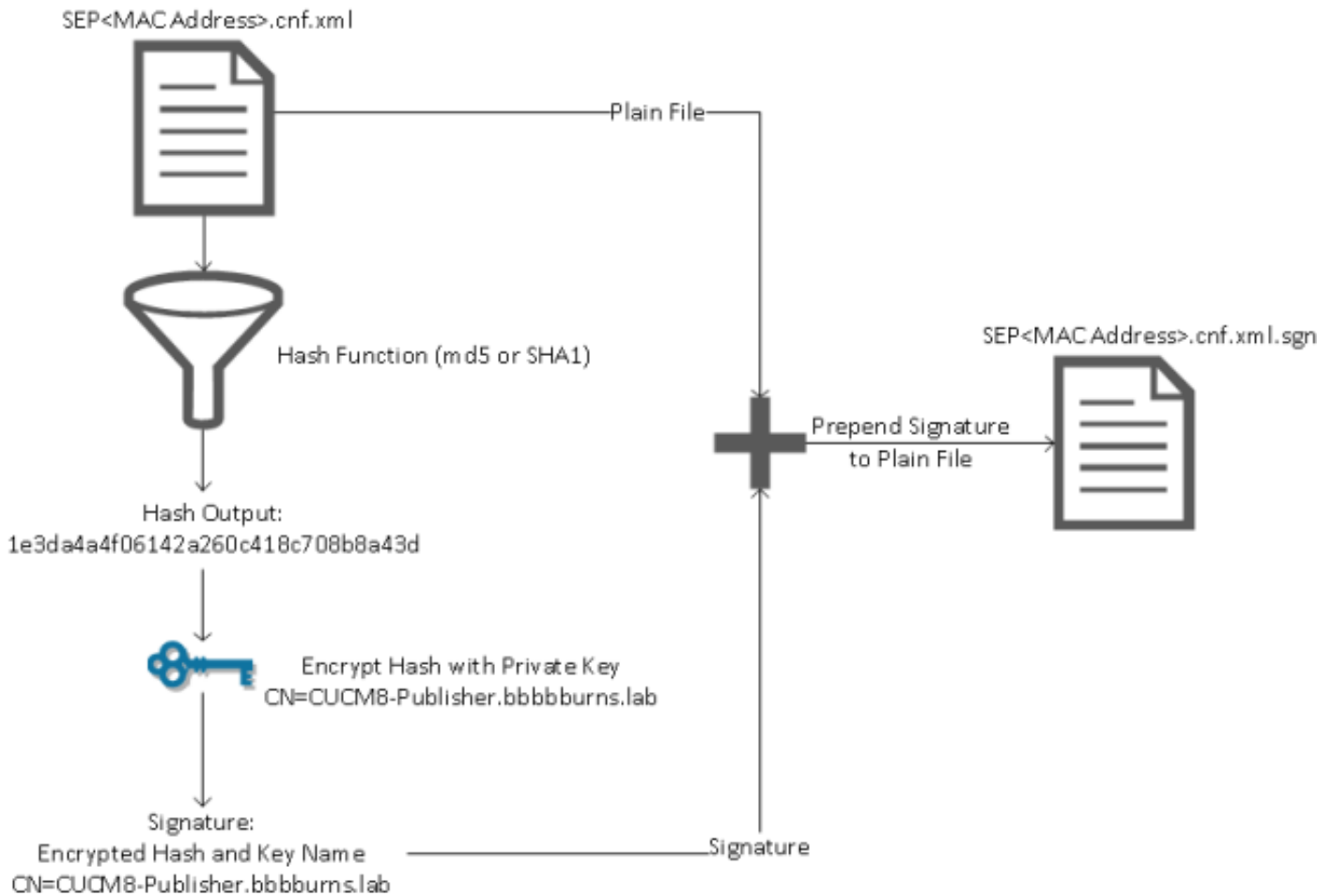
Het getekende bestand heeft een handtekening bovenaan om het bestand te authenticeren, maar is anders in onbewerkte tekst XML. De afbeelding hieronder toont dat de naam van het configuratiebestand **CN=CUCM8-Publisher.bbbburns.lab** is, dat op zijn beurt ondertekend door **CN=JASBURNS-AD**. Dit betekent dat de telefoon de handtekening van **CUCM8-Publisher.bbbburns.lab** moet controleren aan de hand van het ITL-bestand voordat dit configuratiebestand is geaccepteerd.

```

1  [REDACTED]CN=CUCM8-Publisher.bbbburns.lab;OU=TAC;O=Cisco
2  [REDACTED]CN=JASBUDNS-ADMINISTRATOR;OU=NS;O=Cisco
3  [REDACTED]CN=SEP0011215A1AE3;OU=SEP;O=Cisco
4  [REDACTED]SEP0011215A1AE3.cnf.xml.sgn;
5
6  <?xml version="1.0" encoding="UTF-8"?>
7  <device xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="axl:XIPPhone" cn="SEP0011215A1AE3" ou="SEP" oucn="SEP0011215A1AE3" oucn-cn="SEP0011215A1AE3" oucn-ou="SEP" oucn-o="Cisco" fullConfig="true" deviceProtocol="SCCP"/>
8
9

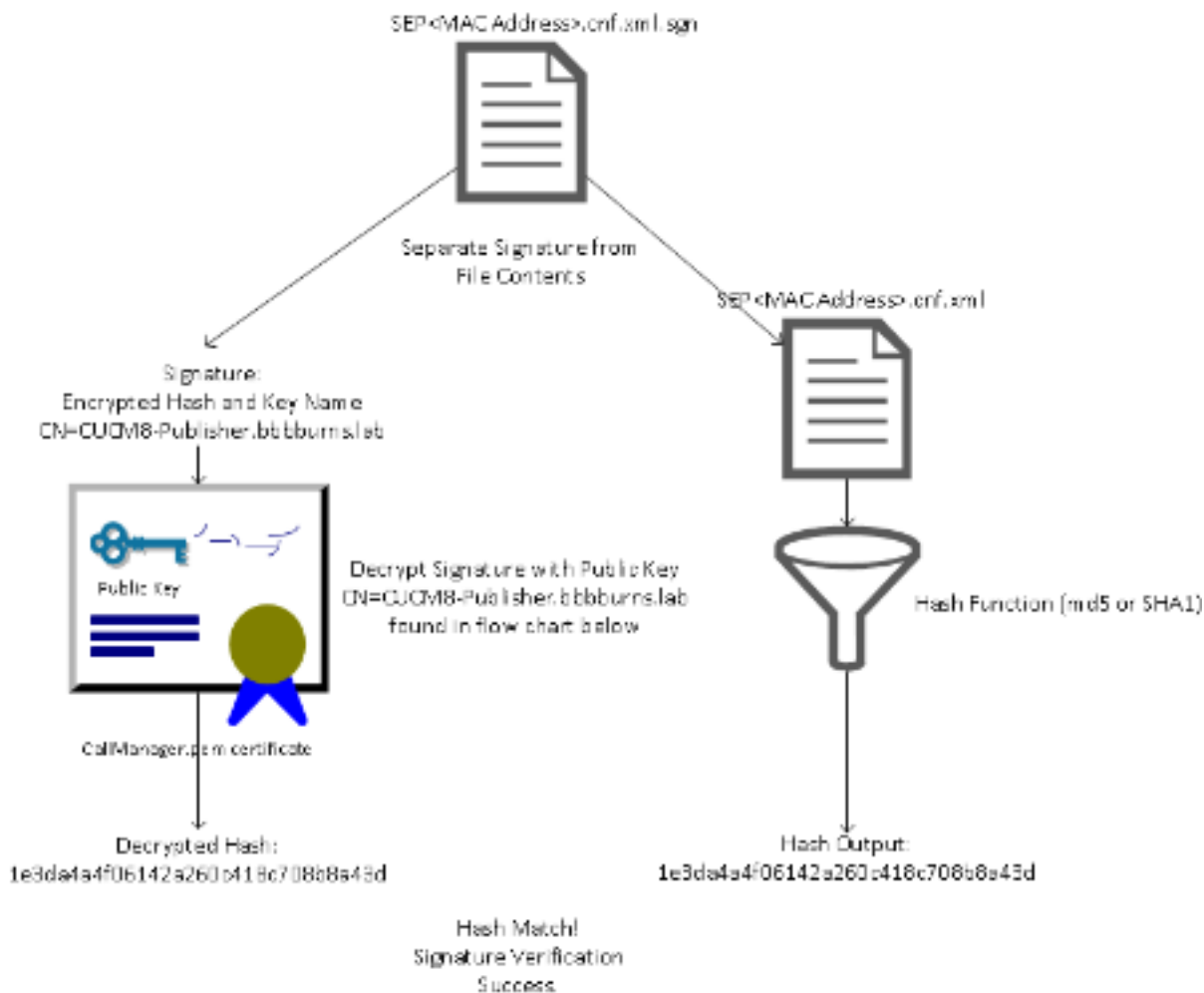
```

Dit is een diagram dat laat zien hoe de privé-toets wordt gebruikt samen met een Message Digest Algorithm (MD) 5 of Secure Hash Algorithm (SHA) 1-hashfunctie om het ondertekende bestand te maken.



De verificatie van de handtekeningen verandert dit proces door het gebruik van de openbare sleutel die overeenkomt met de decryptie van de hash. Als de hashes overeenkomen, toont het:

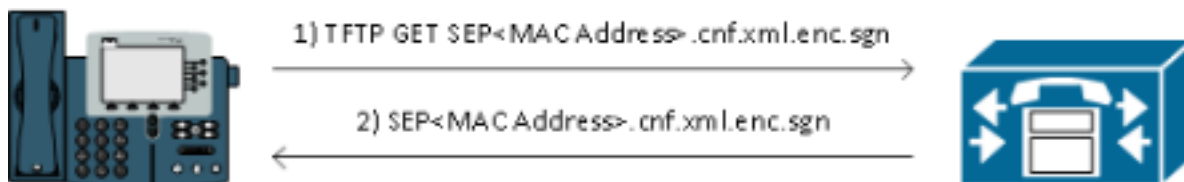
- Dit bestand is niet aangepast tijdens het transport.
- Dit bestand komt van de partij die in de signatuur is opgenomen, aangezien alles dat succesvol is gedecrypteerd met de openbare sleutel, moet zijn versleuteld met de privésleutel.



TFTP-configuratie van bestandsencryptie

Als de optionele TFTP-configuratie is ingeschakeld in het bijbehorende telefoonbeveiligingsprofiel, vraagt de telefoon om een versleuteld configuratiebestand. Dit bestand is getekend met de particuliere TFTP-toets en versleuteld met een symmetrische toets die tussen de telefoon en de CUCM is uitgewisseld (raadpleeg de [Cisco Unified Communications Manager Security Guide, release 8.5\(1\)](#) voor volledige informatie), zodat de inhoud ervan niet met een netwerkzoeker kan worden gelezen tenzij de waarnemer de benodigde toetsen heeft.

De telefoon vraagt om **SEP<MAC-adres>.cnf.xml.enc.sgn** om het ondertekende gecodeerde bestand te krijgen.

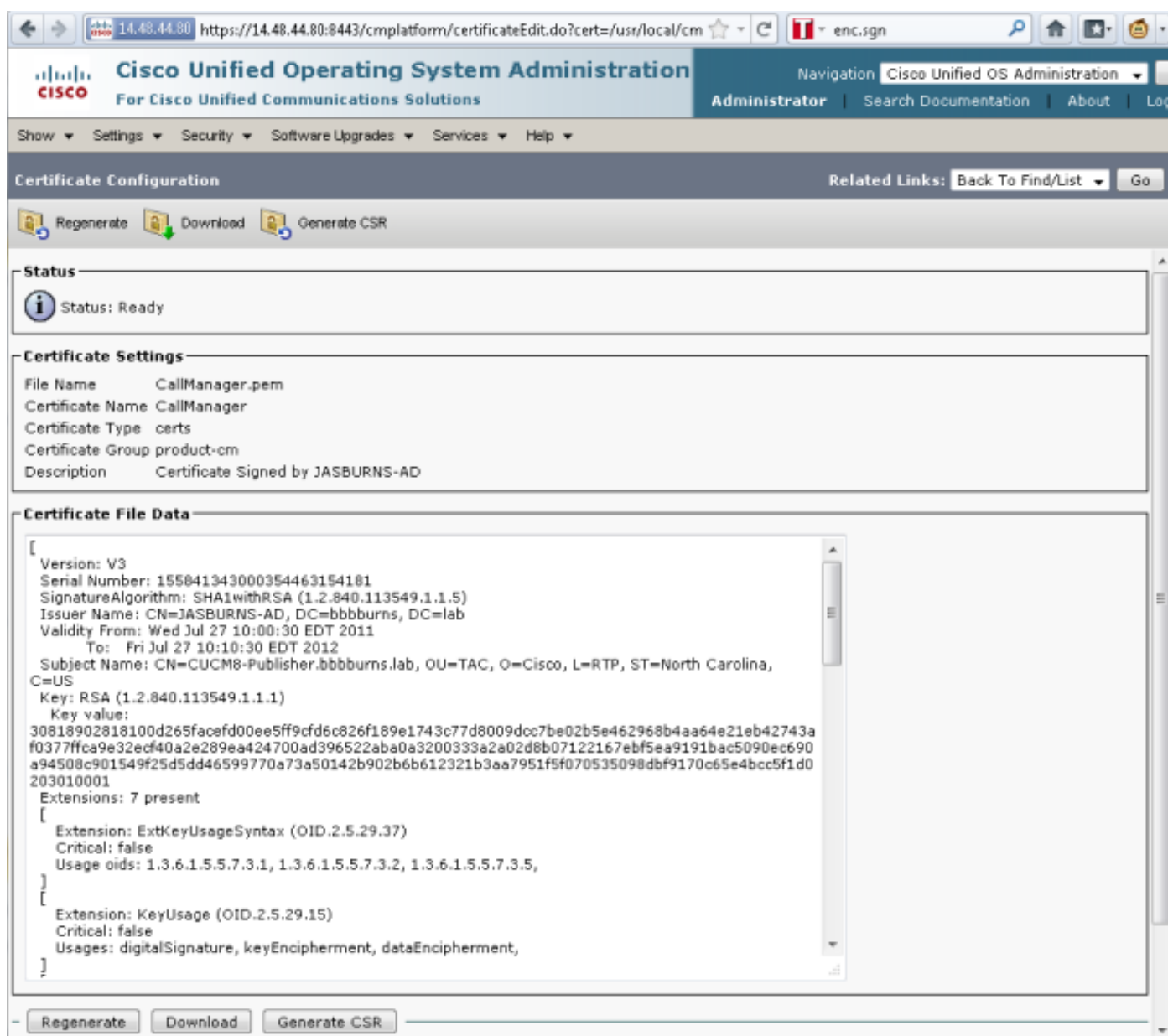


Het gecodeerde configuratiebestand heeft ook de signatuur aan het begin, maar er is geen onduidelijke tekstgegevens daarna, slechts gecodeerde gegevens (verdubbelde binaire tekens in deze teksteditor). De afbeelding toont dat de ondertekenaar hetzelfde is als in het vorige voorbeeld, dus deze tekenaar moet aanwezig zijn in het ITL-bestand voordat de telefoon het bestand accepteert. Bovendien moeten de decryptie toetsen correct zijn voordat de telefoon de inhoud van het bestand kan lezen.

Eerst is er een aantal bestanden die op de CUCM-server zelf aanwezig moeten zijn. Het belangrijkste onderdeel is het TFTP-certificaat en de TFTP-private sleutel. Het TFTP-certificaat bevindt zich onder **OS-beheer > Beveiliging > certificaatbeheer > CallManager.pem**.

De CUCM-server gebruikt de privé- en openbare toetsen van het CallManager.pem-certificaat voor de TFTP-service (en voor de Cisco Call Manager (CCM)-service). De afbeelding laat zien dat het CallManager.pem certificaat is afgegeven aan **CUCM8-uitgever.bbburns.lab** en ondertekend door **JASBURNS-AD**. Alle TFTP-configuratiebestanden zijn door de onderstaande privé-toets getekend.

Alle telefoons kunnen de TFTP openbare sleutel in het certificaat CallManager.pem gebruiken om een bestand te decrypteren dat versleutelen met de privésleutel van TFTP, en om elk bestand te verifiëren dat met de private sleutel van TFTP werd getekend.



The screenshot displays the Cisco Unified Operating System Administration web interface. The page title is "Certificate Configuration" and the user is logged in as "Administrator". The interface shows the following details for the "CallManager.pem" certificate:

- Status:** Ready
- Certificate Settings:**
 - File Name: CallManager.pem
 - Certificate Name: CallManager
 - Certificate Type: certs
 - Certificate Group: product-cm
 - Description: Certificate Signed by JASBURNS-AD
- Certificate File Data:**

```
[
  Version: V3
  Serial Number: 155041343000354463154181
  Signature Algorithm: SHA1withRSA (1.2.840.113549.1.1.5)
  Issuer Name: CN=JASBURNS-AD, DC=bbburns, DC=lab
  Validity From: Wed Jul 27 10:00:30 EDT 2011
  To: Fri Jul 27 10:10:30 EDT 2012
  Subject Name: CN=CUCM8-Publisher.bbburns.lab, OU=TAC, O=Cisco, L=RTP, ST=North Carolina, C=US
  Key: RSA (1.2.840.113549.1.1.1)
  Key value:
  30818902818100d265facefd00ee5ff9cfd6c826f189e1743c77d8009doc7be02b5e462968b4aa64e21eb42743a
  f0377ffca9e32ecf40a2e289ea424700ad396522aba0a3200333a2a02d8b07122167ebf5ea9191bac5090ec690
  a94508c901549f25d5dd46599770a73a50142b902b6b612321b3aa7951f5f070535098dbf9170c65e4bcc5f1d0
  203010001
  Extensions: 7 present
  [
    Extension: ExtKeyUsageSyntax (OID.2.5.29.37)
    Critical: false
    Usage oids: 1.3.6.1.5.5.7.3.1, 1.3.6.1.5.5.7.3.2, 1.3.6.1.5.5.7.3.5,
  ]
  [
    Extension: KeyUsage (OID.2.5.29.15)
    Critical: false
    Usages: digitalSignature, keyEncipherment, dataEncipherment,
  ]
]
```

Naast de privé sleutel van het certificaat CallManager.pem, slaat de CUCM server ook een ITL bestand op dat aan telefoons wordt aangeboden. De opdracht **tonen** toont de volledige inhoud van dit ITL-bestand via Secure Shell (SSH)-toegang tot de CUCM-server OS CLI.

Deze sectie breekt het ITL bestand stuk voor stuk uit, omdat het een aantal belangrijke componenten heeft die de telefoon gebruikt.

Het eerste deel is de signatuurinformatie. Zelfs het ITL-bestand is een ondertekend bestand. Deze uitvoer toont aan dat het door de TFTP privé sleutel wordt ondertekend die met het vorige certificaat CallManager.pem wordt geassocieerd.

```
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:17:00:00:00:00:05
6	CANAME	15	CN=JASBURNS-AD

```
*Signature omitted for brevity*
```

De volgende secties bevatten elk hun doel binnen een speciale **Funcie** parameter. De eerste functie is het Beveiligingstoken voor het systeem. Dit is de ondertekening van de TFTP-publieke sleutel.

```
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	ISSUENAME	15	CN=JASBURNS-AD
6	SERIALNUMBER	10	21:00:2D:17:00:00:00:00:05
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.
```

De volgende functie is CCM+TFTP. Dit is opnieuw de openbare sleutel van TFTP die gebruikt om gedownload TFTP configuratiebestanden voor de authenticatie en decryptie te verklaren.

```
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	ISSUENAME	15	CN=JASBURNS-AD
6	SERIALNUMBER	10	21:00:2D:17:00:00:00:00:05
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)
```

De volgende functie is TVS. Er is een ingang voor de openbare sleutel van elke TVS server waaraan de telefoon verbindt. Hiermee kan de telefoon een Secure Socket Layer (SSL) sessie naar de TVS server maken.

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 ISSUERNAM         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
```

De laatste functie in het ITL-bestand is de Proxy-functie (CAPF) van de certificaatinstantie. Met dit certificaat kunnen de telefoons een beveiligde verbinding naar de CAPF-service maken op de CUCM-server, zodat de telefoon een lokaal belangrijk certificaat (LSC) kan installeren of bijwerken. Dit proces zal worden behandeld in een ander document dat nog moet worden vrijgegeven.

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 ISSUERNAM         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.

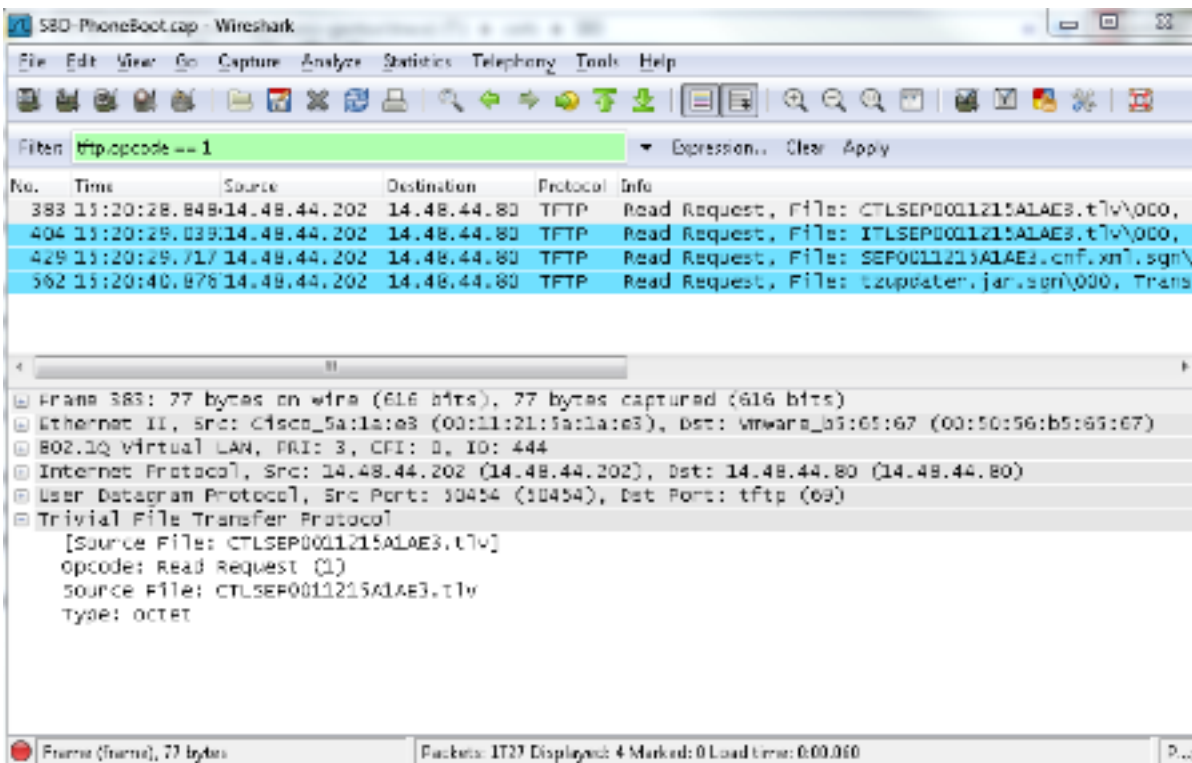
De volgende sectie bespreekt precies wat er gebeurt wanneer een telefoon start.

Telefonische downloads - ITL- en configuratiebestand

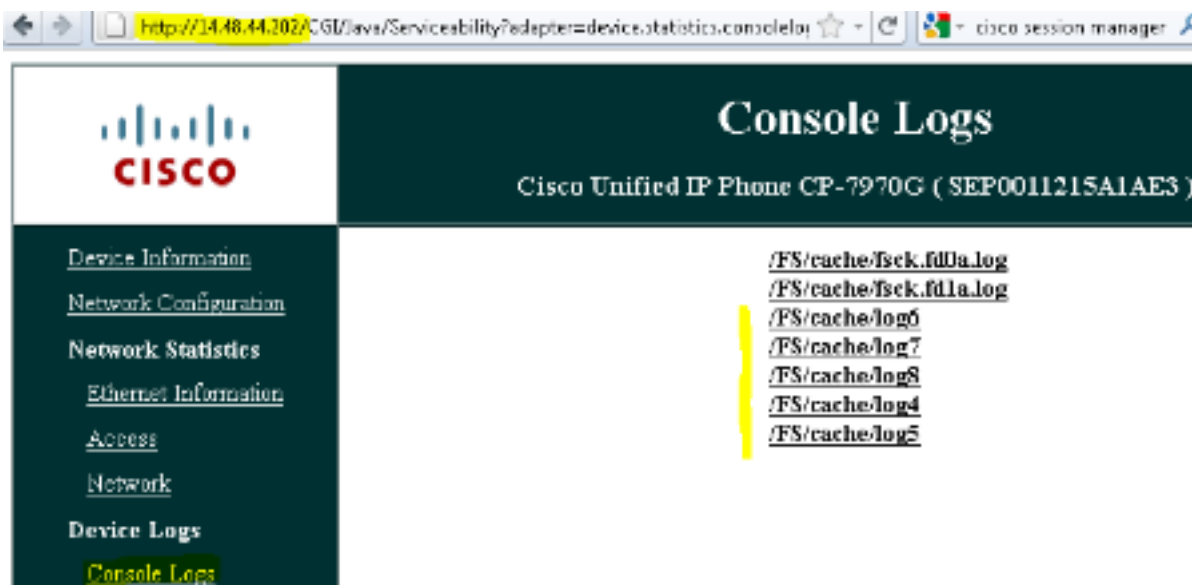
Nadat de telefoon start en een IP adres evenals het adres van een TFTP server verkrijgt vraagt het eerst om de CTL en de ITL bestanden.

Deze pakketvastlegging toont een telefoonverzoek voor het ITL-bestand. Als u op **tftp.opcode = 1**

filtert, ziet u elke TFTP Read Application vanuit de telefoon:



Aangezien de telefoon CTL en ITL bestanden van TFTP met succes heeft ontvangen, vraagt de telefoon om een ondertekend configuratiebestand. De logboeken van de telefoonconsole die dit gedrag laten zien zijn beschikbaar van de web interface van de telefoon:



Eerst vraagt de telefoon om een CTL bestand, dat slaagt:

```
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
```

Vervolgens vraagt de telefoon ook om een ITL-bestand:

```
868: NOT 09:13:17.860613 TFTP: [28]:Requesting ITLSEP0011215A1AE3.tlv from
```

14.48.44.80

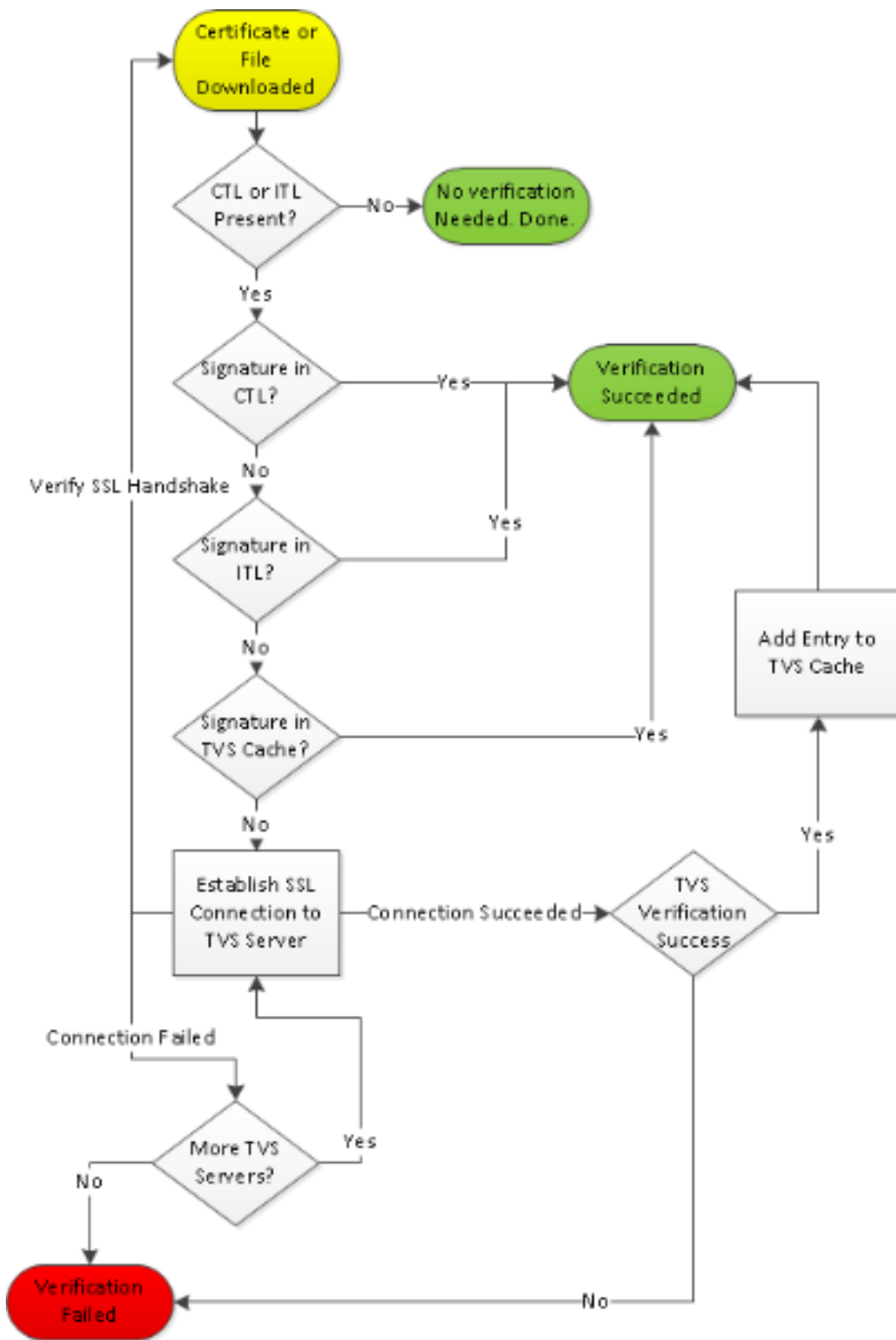
869: NOT 09:13:17.875059 TFTP: [28]:Finished --> rcvd 5438 bytes

Telefonisch verifieert ITL- en configuratiebestand

Nadat het ITL-bestand is gedownload, moet het worden geverifieerd. Er zijn een aantal staten die op dit moment een telefoon kunnen hebben, dus dit document behandelt ze allemaal.

- De telefoon heeft geen CTL of ITL bestand aanwezig of ITL is leeg wegens de **Bereid Cluster voor Terug naar Pre 8.0** parameter. In deze staat vertrouwt de telefoon blind op het volgende CTL of ITL bestand gedownload en gebruikt deze handtekening vanaf nu.
- De telefoon heeft al een CTL maar geen ITL. In deze staat, vertrouwt de telefoon slechts op een ITL als het door de CCM+TFTP functie in het CTL dossier kan worden geverifieerd.
- De telefoon heeft al een CTL en een ITL bestand. In deze staat, verifieert de telefoon dat de recent gedownload bestanden overeenkomen met de signatuur in of de CTL, ITL, of TVS server.

Hier is een stroomschema dat beschrijft hoe de telefoon ondertekende bestanden en HTTPS certificaten verifieert:



In dit geval kan de telefoon de handtekening in de ITL en CTL bestanden verifiëren. De telefoon heeft al zowel een CTL als ITL, dus het werd gewoon tegen hen gecontroleerd en vond de juiste handtekening.

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

Sinds de telefoon de CTL en ITL bestanden heeft gedownload, vraagt vanuit dit punt ALLEEN om ondertekende configuratiebestanden. Dit illustreert dat de logica van de telefoon is om te bepalen dat de TFTP server veilig is, gebaseerd op de aanwezigheid van CTL en ITL, en dan om een ondertekend bestand te vragen:

```
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
```

Nadat het getekende configuratiebestand is gedownload, moet de telefoon het authenticeren in vergelijking met de functie voor CCM+TFTP in de ITL:

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

Telefonische contactgegevens TVS voor Onbekend certificaat

Het ITL-bestand biedt een TVS-functie die het certificaat van de TVS-service bevat dat op de CUCM-server TCP poort 2445 draait. TVS werkt op alle servers waar de CallManager-service is geactiveerd. De CUCM TFTP-service gebruikt de geconfigureerde CallManager-groep om een lijst met TVS-servers te maken die de telefoon moet benaderen in het telefoonconfiguratiebestand.

Sommige labs gebruiken slechts één CUCM server. In een CUCM-cluster met meerdere knooppunten kunnen er maximaal drie TVS-items zijn voor een telefoon, één voor elke CUCM in de CUCM-groep van de telefoon.

Dit voorbeeld toont wat er gebeurt wanneer de knop **Mappen** op de IP-telefoon wordt ingedrukt. De URL van de directoraten is ingesteld voor HTTPS, dus de telefoon wordt voorgesteld met het Tomcat-webcertificaat van de server van de Mappen. Dit Tomcat-webcertificaat (tomcat.pem in OS-beheer) is niet geladen in de telefoon, dus de telefoon moet contact opnemen met TVS om het certificaat te echt te maken.

Raadpleeg het vorige TVS - Overzicht voor een beschrijving van de interactie. Dit is het logperspectief van de telefoonconsole:

Eerst vindt u de URL van de map:

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

Dit is een SSL/Transport Layer Security (TLS) veilige HTTP-sessie die verificatie vereist.

```
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
```

De telefoon verifieert eerst dat het certificaat dat door de SSL/TLS server wordt aangeboden aanwezig is in de CTL. Vervolgens kijkt de telefoon naar de Functies in het ITL-bestand om te zien of er een match is gevonden. Deze foutmelding zegt "HTTPS cert niet in CTL" wat betekent dat "dat certificatie niet gevonden kan worden in het CTL of het ITL."

```
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>
```

Nadat de directe inhoud van het CTL- en ITL-bestand op het certificaat is gecontroleerd, is het volgende wat de telefoon controleert het TVS cache. Dit gebeurt om op netwerkverkeer te snijden als de telefoon onlangs de TVS server om het zelfde certificaat heeft gevraagd. Als het HTTPS certificaat niet in het telefoongeheugen wordt gevonden, kunt u een TCP verbinding maken met de TVS server zelf.

```
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_sec>
1224: NOT 15:20:59.457643 SECD: getTvsServerInfo: Phone in IPv4 only mode
1225: NOT 15:20:59.458706 SECD: getTvsServerInfo: Retrieving 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.
```

Onthoud dat de verbinding met TVS zelf SSL/TLS is (veilig HTTP, of HTTPS), dus het is ook een certificaat dat tegen het CTL aan ITL geauthentificeerd moet worden. Als alles correct gaat, dient het certificaat van de TVS server gevonden te worden in de TVS functie van het ITL bestand. Zie ITL Record #3 in het vorige voorbeeld ITL-bestand.

```
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>
```

Succes! De telefoon heeft nu een veilige verbinding met de TVS server. De volgende stap is om de TVS server "Hallo, vertrouw ik op dit Diversecertificaat?" te vragen.

Dit voorbeeld laat het antwoord op die vraag zien - een antwoord van 0, wat succes betekent (geen fout).

```
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
```

Aangezien TVS met succes heeft gereageerd, worden de resultaten van dat certificaat in de cache opgeslagen. Dit betekent dat, als u binnen de volgende 86.400 seconden op de knop **Mappen** drukt, u geen contact hoeft op de TVS server op te nemen om het certificaat te controleren. Je kunt simpelweg toegang krijgen tot de lokale cache.

```
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
```

Ten slotte, controleer je of je verbinding met de Mappen server is gelukt.

```
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
```

Hier is een voorbeeld van wat er gebeurt op de CUCM server waar TVS draait. U kunt TVS-bestanden verzamelen met het Cisco Unified Real-Time Monitoring Tool (RTMT).



Trace Configuration



Status

Status : Ready

Select Server, Service Group and Service

Server*

Service Group*

Service*

Apply to All Nodes

Trace On

Trace Filter Settings

Debug Trace Level

Cisco Trust Verification Service Trace Fields

Enable All Trace

Device Name Based Trace Monitoring

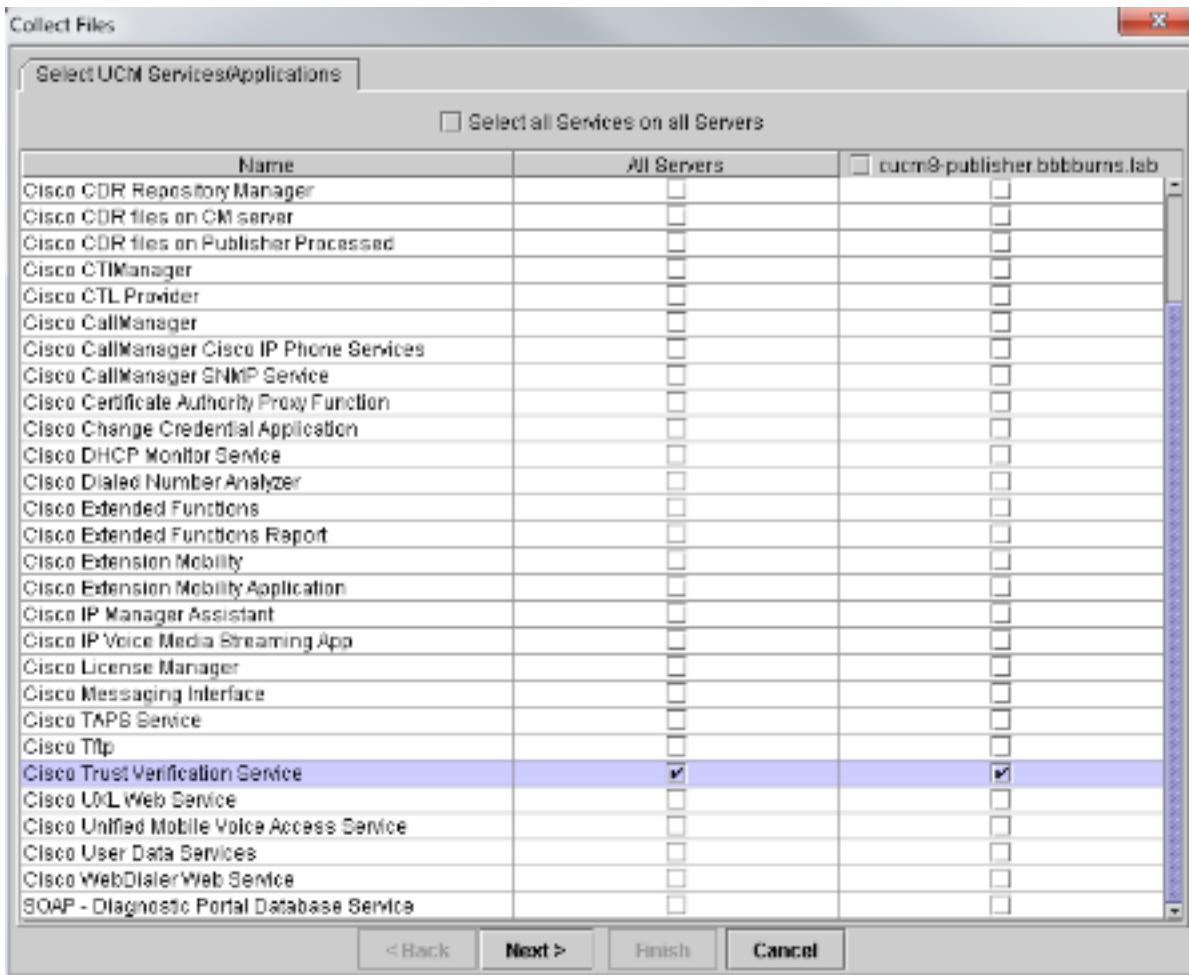
Include Non-device Traces

Trace Output Settings

Maximum No. of Files*

Maximum File Size (MB)*

* - indicates required item.



In de CUCM TVS-documenten is te zien dat u SSL-handdruk met de telefoon hebt, de telefoon naar TVS vraagt om het Tomcat-certificaat en vervolgens reageert TVS op een indicatie dat het certificaat in de TVS-certificaatwinkel is gevonden.

```
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.bbburns.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
```

Het TVS-certificaatopslaan is een lijst van alle certificaten die in de webpagina **OS-beheer > certificaatbeheer** zijn opgenomen.

Controleer handmatig dat de telefoon/ITL overeenkomt met CUCM ITL

Eén veel voorkomende misvattingen die worden gezien tijdens het oplossen van problemen betreft de neiging om het ITL-bestand te verwijderen in de hoop dat er een probleem met de bestandsverificatie zal worden opgelost. Soms is het wissen van ITL-bestanden vereist, maar er kan een betere manier zijn.

Het ITL-bestand hoeft alleen te worden verwijderd wanneer aan al deze voorwaarden is voldaan.

- De handtekening van het ITL-bestand op de telefoon komt niet overeen met de handtekening van het ITL-bestand op de CM TFTP-server.
- De TVS-handtekening in het ITL-bestand komt niet overeen met het door TVS ingediende certificaat.
- De telefoon toont "Verificatie mislukt" wanneer het probeert het ITL-bestand of de configuratie bestanden te downloaden.
- Er is geen back-up van de oude TFTP-privé-toets.

Zo controleer je de eerste twee van deze voorwaarden.

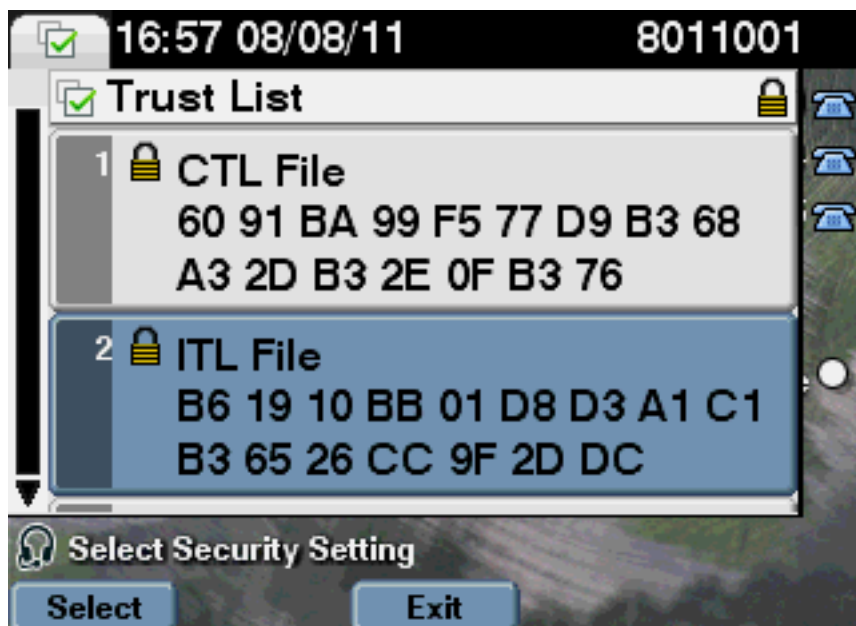
Eerst kunt u de som van het ITL-bestand dat op CUCM aanwezig is vergelijken met het ITL-bestand van de checksum aan de telefoon. Er is momenteel geen manier om de MD5sum van het ITL-bestand op CUCM zelf te bekijken totdat u een versie met de oplossing voor dit [Cisco bug-ID CSCto60209](#) uitvoert.

Start vervolgens uw favoriete GUI- of CLI-programma's:

```
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
```

Dit toont aan dat de MD5sum van het ITL-bestand in CUCM **b61910bb01d8d3a1c1b36526cc9f2ddc**.

U kunt nu naar de telefoon zelf kijken om de hash van het ITL bestand te bepalen dat daar geladen is: **Instellingen > Beveiligingsconfiguratie > Trustlijst**.

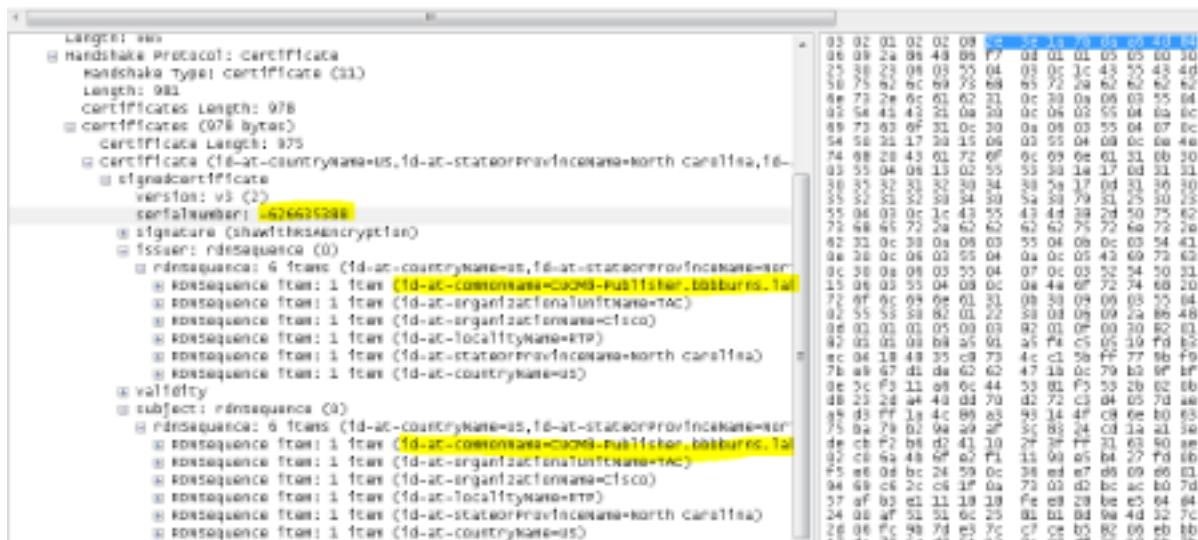
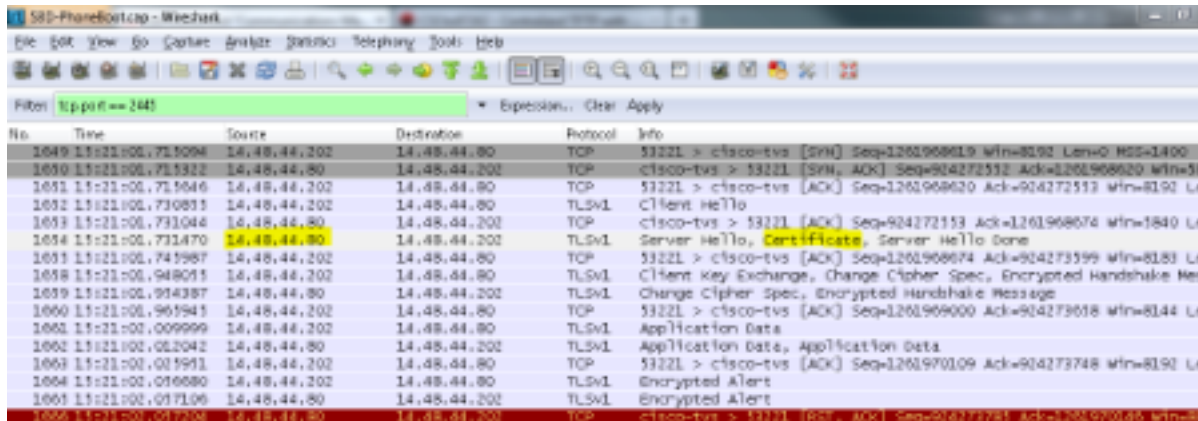


Dit toont aan dat de MD5bedragen overeenkomen. Dit betekent dat het ITL-bestand op de telefoon overeenkomt met het bestand op het UCM, zodat het niet hoeft te worden verwijderd.

Als deze DOES match heeft, moet u naar de volgende handeling gaan - bepalen of het TVS-certificaat in het ITL overeenkomt met het door TVS gepresenteerde certificaat. Deze operatie is een beetje meer betrokken.

Kijk eerst naar de pakketvastlegging van de telefoon die op de TVS server op TCP poort 2445 verbindt.

Klik met de rechtermuisknop op een willekeurig pakket in deze stream in Wireless-shark, klik op **Decode As** en selecteer **SSL**. Vind het servercertificaat dat er als volgt uitziet:



Kijk naar het TVS-certificaat in het vorige ITL-bestand. U dient een artikel te zien met het serienummer **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	ISSUENAME	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

Success, de **TVS.pem** binnen het ITL-bestand komt overeen met het TVS-certificaat dat op het netwerk wordt aangeboden. U hoeft het ITL niet te verwijderen en TVS geeft het juiste certificaat weer.

Als de verificatie van bestanden nog steeds mislukt, controleert u de rest van het vorige stroomschema.

Beperkingen en interactie

Certificaten opnieuw genereren / een cluster opnieuw bouwen / verlopen volgens certificaat

Het belangrijkste certificaat is nu het CallManager.pem certificaat. De privétoets van dit certificaat wordt gebruikt om alle TFTP-configuratiebestanden te ondertekenen, die het ITL-bestand bevatten.

Als het CallManager.pem-bestand opnieuw wordt gegenereerd, wordt een nieuw CCM+TFTP-certificaat gegenereerd met een nieuwe privé-toets. Daarnaast wordt het ITL-bestand nu getekend door deze nieuwe CCM+TFTP-toets.

Nadat u CallManager.pem regeneert en de TVS en TFTP service opnieuw start, gebeurt dit wanneer een telefoon start.

1. De telefoon probeert het nieuwe ITL-bestand te downloaden dat door de nieuwe CCM+TFTP-server is getekend. De telefoon heeft op dit moment alleen het oude ITL-bestand en de nieuwe toetsen zijn niet in het ITL-bestand aanwezig op de telefoon.
2. Aangezien de telefoon de nieuwe handtekening CCM+TFTP in het oude ITL niet kon vinden, probeert de telefoon de TVS-service te contacteren.
Opmerking: Dit is een uiterst belangrijk onderdeel. Het TVS-certificaat uit het oude ITL-bestand moet nog steeds overeenkomen. Als zowel CallManager.pem als TVS.pem op het zelfde ogenblik geregenereerd worden, kunnen de telefoons geen nieuwe bestanden downloaden zonder het ITL van de telefoon handmatig te wissen.
3. Wanneer de telefoon TVS contacteert, heeft de CUCM server die TVS runt het nieuwe certificaat CallManager.pem in de OS certificaatopslag.
4. De TVS server keert succes terug en de telefoon laadt het nieuwe ITL bestand in het geheugen.
5. De telefoon probeert nu een configuratiebestand te downloaden, dat door de nieuwe CallManager.pem-toets is getekend.
6. Aangezien het nieuwe ITL geladen is, wordt het nieuw ondertekende configuratiebestand door de ITL in het geheugen geverifieerd.

Belangrijkste punten:

- Regenereren nooit tegelijkertijd zowel de CallManager.pem- als TVS.pem-certificaten.
- Als ofwel TVS.pem of CallManager.pem wordt geregenereerd, moeten TVS en TFTP opnieuw worden gestart en telefoons worden gereset om de nieuwe ITL bestanden te verkrijgen. Nieuwe versies van CUCM gaan automatisch over op deze telefoon en waarschuwen de gebruiker tijdens de regeneratietijd van het certificaat.
- Als er meer dan één TVS-server bestaat (meer dan één server in de CallManager Group), kunnen de extra servers het nieuwe CallManager.pem-certificaat authenticeren.

Telefoons tussen clusters verplaatsen

Wanneer u telefoons van het ene cluster naar het andere verplaatsen met ITL's op zijn plaats, moet rekening worden gehouden met de ITL en TFTP Private Key. Elk nieuw configuratiebestand dat aan de telefoon wordt aangeboden MOET overeenkomen met een handtekening in CTL, ITL of een handtekening in de huidige TVS-service van de telefoon.

Dit document legt uit hoe u ervoor kunt zorgen dat het ITL-bestand en de configuratiebestanden van de nieuwe groep kunnen worden vertrouwd in het huidige ITL-bestand aan de telefoon.

<https://supportforums.cisco.com/docs/DOC-15799>.

Terug en herstellen

Van het certificaat van CallManager.pem en de privé-sleutel wordt een back-up gemaakt via het noodherstelsysteem (DRS). Als een TFTP-server opnieuw wordt opgebouwd, MOET deze uit een back-up worden hersteld, zodat de privétoets kan worden hersteld. Zonder de privé sleutel CallManager.pem op de server, kunnen telefoons met huidige ITLs die de oude sleutel gebruiken geen ondertekende configuratiebestanden vertrouwen hebben.

Als een cluster opnieuw gebouwd en niet hersteld wordt van steun, is het precies zoals het "[Telefoons tussen clusters](#)" document "[Beweeg](#)". Dit komt doordat een cluster met een nieuwe sleutel een ander cluster is wat de telefoons betreft.

Er is één ernstig defect verbonden aan back-up en herstel. Als een cluster gevoelig is voor [Cisco bug ID CSCtn50405](#), bevatten de back-ups van DRS niet het certificaat CallManager.pem. Dit veroorzaakt dat elke server die uit deze back-up is gerestaureerd, corrupte ITL bestanden kan genereren totdat er een nieuwe CallManager.pem wordt gegenereerd. Als er geen andere functionele TFTP-servers zijn die niet door de back-up- en terugzetbewerking zijn gegaan, kan dit betekenen dat alle ITL-bestanden uit de telefoons moeten worden verwijderd.

Om te verifiëren of uw CallManager.pem-bestand moet worden geregenereerd, voer u de opdracht Show itl in, gevolgd door:

```
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
```

In de ITL-uitvoer zijn de belangrijkste te zoeken fouten:

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

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

De vorige Structured Search Query Language (SQL) zoekopdrachten naar de certificaten die een rol spelen als "Verificatie en autorisatie". Het CallManager.pem certificaat in de vorige database query die de rol van verificatie en autorisatie heeft, zou OOK aanwezig moeten zijn in de webpagina van het OS-beheercertificaat. Als het vorige defect wordt aangetroffen, is er een mismatch tussen de CallManager.pem certificaten in de query en in de OS webpagina.

Host Names of Domain Names wijzigen

Als u de hostname of de domeinnaam van een CUCM server wijzigt, genereert het alle certificaten tegelijkertijd op die server. De regeneratie van het certificaat verklaarde dat regeneratie van zowel TVS.pem als CallManager.pem "slecht" is.

Er zijn een paar scenario's waar een hostname verandering mislukt, en een paar waar het zonder problemen werkt. In deze sectie worden alle besproken en gekoppeld aan de resultaten die u in dit document al over TVS en ITL hebt.

Cluster met één knooppunt met alleen ITL (gebruiks voorzichtigheid, dit breekt zonder voorbereiding)

- Met een server van Business Edition of uitgeverij-only plaatsing, zowel CallManager.pem als TVS.pem worden geregenereerd gelijktijdig wanneer u hostnamen verandert.
- Als de hostname op één knooppunt is gewijzigd zonder eerst de [hier gedekte parameter Rollback Enterprise te](#) gebruiken, kunnen de telefoons het nieuwe ITL-bestand of de configuratie bestanden niet vergelijken met hun huidige ITL-bestand. Bovendien zijn ze niet in staat om verbinding te maken met de TVS omdat het TVS-certificaat niet langer wordt vertrouwd.
- De telefoons geven een fout weer over "Vertrouwde lijst is mislukt," geen nieuwe configuratiewijzigingen worden uitgevoerd en de beveiligde URL's van de service zijn defect.
- De enige oplossing als de voorzorgsmaatregel in stap 2 niet eerst wordt genomen is [het ITL van elke telefoon handmatig te verwijderen](#).

Cluster met één knooppunt met zowel CTL als ITL (dit kan tijdelijk worden verbroken, maar gemakkelijk worden opgelost)

- Nadat u door de nieuwe naam van servers loopt, herhaal de CTL client. Dit plaatst het nieuwe certificaat CallManager.pem in het CTL dossier dat de telefoon downloads.
- Nieuwe configuratiebestanden, die de nieuwe ITL-bestanden omvatten, kunnen worden vertrouwd op basis van de CCM+TFTP-functie in het CTL-bestand.
- Dit werkt omdat het aangepaste CTL-bestand is vertrouwd op basis van een USB-Token privé-toets die hetzelfde blijft.

Cluster met meerdere knooppunten met alleen ITL (dit werkt over het algemeen, maar kan permanent worden onderbroken als u dit snel doet)

- Omdat een cluster met meerdere knooppunten meerdere TVS-servers heeft, kan elke server zijn certificaten zonder problemen laten regenereren. Wanneer de telefoon met deze nieuwe, onbekende handtekening wordt voorgesteld, vraagt het een andere TVS-server om het nieuwe servercertificaat te controleren.
- Er zijn twee belangrijke problemen die dit kunnen doen mislukken:
Als alle servers tegelijkertijd een andere naam krijgen en worden herstart, is geen van de TVS-servers bereikbaar met bekende certificaten wanneer de servers en telefoons opnieuw worden opgestart. Als een telefoon in de CallManager Group maar één server heeft, maken de extra TVS-servers geen verschil. Zie het scenario "Single Node Cluster" om dit op te lossen, of een andere server aan de CallManager-groep van de telefoon toe te voegen.

Cluster met meerdere knooppunten met zowel CTL als ITL (dit kan niet permanent worden verbroken)

- Nadat u door de namen hebt gedraaid, verklaart de TVS-service de nieuwe certificaten echt.
- Zelfs als alle TVS servers om één of andere reden niet beschikbaar zijn, kan de CTL client nog steeds gebruikt worden om de telefoons bij te werken met de nieuwe CallManager.pem CCM+TFTP certificaten.

Gecentraliseerde TFTP

Wanneer een telefoon met een ITL laarst, vraagt het deze bestanden: **CTLSEP<MAC-adres>.tlv**, **ITLSEP<MAC-adres>.tlv** en **SEP<MAC-adres>.cnf.xml.sgn**.

Als de telefoon deze bestanden niet kan vinden, vraagt het om de **ITLFile.tlv** en de **CTLFile.tlv**, die een gecentraliseerde TFTP server aan om het even welke telefoon die het wenst verstrekt.

Met gecentraliseerd TFTP is er één TFTP-cluster dat wijst op een aantal andere subclusters. Dit gebeurt vaak omdat telefoons op meerdere CUCM-clusters dezelfde DHCP-scope hebben en daarom dezelfde DHCP-optie 150 TFTP-server moeten hebben. Alle IP-telefoons wijzen naar de centrale TFTP-cluster, zelfs als deze zich in andere clusters registreren. Deze centrale TFTP-server vraagt de externe TFTP-servers wanneer zij een verzoek voor een bestand ontvangt dat zij niet kan vinden.

Vanwege deze operatie werkt gecentraliseerd TFTP alleen in een ITL homogeen milieu. Alle servers moeten CUCM versie 8.x of hoger uitvoeren, of alle servers moeten versies uitvoeren voordat versie 8.x gestart is.

Als een ITLFile.tlv van de Gecentraliseerde TFTP server wordt aangeboden, vertrouwen de telefoons geen bestanden van de externe TFTP server omdat de handtekeningen niet overeenkomen. Dit gebeurt in een heterogene mix. In een homogene mix vraagt de telefoon om **ITLSEP<MAC>.tlv** die uit de juiste externe cluster wordt getrokken.

In een heterogene omgeving met een combinatie van pre-versie 8.x en Versie 8.x clusters, moet "Voorbereiden van Cluster voor Terug naar Pre 8.0" ingeschakeld zijn op de Versie 8.x-cluster zoals beschreven in [Cisco HTTP ID CSCto87262](#) en de "Beveiligde telefoon URL-parameters" ingesteld met in plaats van HTTPS. Dit schakelt de ITL functies in de telefoon effectief uit.

Veelgestelde vragen

Kan ik SBD uitzetten?

U kunt SBD alleen uitschakelen indien SBD en ITL momenteel werken.

SBD kan tijdelijk op telefoons worden uitgeschakeld met het [preparaat van Cluster voor terugdraaiing naar pre 8.0](#) Enterprise-parameter en door de "Beveiligde URL-parameters van de telefoon" te configureren met HTTP in plaats van met HTTPS. Wanneer u de Rollback-parameter instelt, creëert deze een ondertekend ITL-bestand met lege functionele items. Het "lege" ITL-bestand is nog getekend, dus het cluster moet in een volledig functionele security status zijn voordat deze parameter ingeschakeld kan worden.

Nadat deze parameter is ingeschakeld en het nieuwe ITL-bestand met lege items wordt gedownload en geverifieerd, accepteren de telefoons elk configuratiebestand, ongeacht wie het

heeft ondertekend.

Het wordt niet aanbevolen om het cluster in deze staat te verlaten, omdat geen van de drie eerder genoemde functies (geauthentiseerde configuratiebestanden, gecodeerde configuratiebestanden en HTTPS URLs) beschikbaar is.

Kan ik het ITL bestand vanaf alle telefoons makkelijk wissen als CallManager.pem verloren is?

Er is momenteel geen methode om alle ITL's te verwijderen van een telefoon die van afstand door Cisco wordt geleverd. Daarom zijn de in dit document beschreven procedures en interacties zo belangrijk om rekening te houden met de situatie.

Er is een momenteel onopgeloste verbetering aan [Cisco bug ID CSCto47052](#) die deze functionaliteit vraagt, maar het is nog niet geïmplementeerd.

Gedurende de interimperiode is een nieuwe functie toegevoegd via [Cisco bug-ID CSCts01319](#) waardoor het Cisco Technical Assistance Center (TAC) naar het eerder vertrouwde ITL kan terugkeren als deze nog beschikbaar is op de server. Dit werkt alleen in bepaalde gevallen waar het cluster op een versie is met deze defect fixeren, en waar het vorige ITL bestaat in een back-up die is opgeslagen op een speciale locatie op de server. Bekijk het defect om te zien of de versie van het artikel is aangepast. Neem contact op met Cisco TAC om de in het defect beschreven mogelijke herstelprocedure te doorlopen.

Als de vorige procedure niet beschikbaar is, moeten de telefoonknoppen handmatig op de telefoon worden ingedrukt om het ITL-bestand te verwijderen. Dit is de afweging die gemaakt wordt tussen veiligheid en makkelijk bestuur. Om het ITL-bestand echt veilig te maken, mag het niet op afstand makkelijk verwijderd worden.

Zelfs met scripted knop pressen met Simple Object Access Protocol (SOAP) XML-objecten kan de ITL niet extern worden verwijderd. Dit is omdat, op dit punt, de toegang van TVS (en dus de toegang tot Secure Verificatie URL om inkomende SOAP XML-toets te valideren tegen objecten) niet functioneel is. Als de verificatie-URL niet zo veilig is ingesteld, kan het zijn dat de toetsen op de toets ingedrukt worden om een ITL te verwijderen, maar dit script niet beschikbaar is bij Cisco.

Andere methoden om externe toetsen aan te wijzen zonder de URL van de verificatie te gebruiken kunnen bij een derde beschikbaar zijn, maar deze toepassingen worden niet door Cisco geleverd.

De meest gebruikte methode om het ITL te verwijderen is een e-mail uitzending naar alle telefoongebruikers die hen van de belangrijkste volgorde opdraagt. Als de instellingentoeegang is ingesteld op **Beperkt** of **Uitgeschakeld**, moet de telefoon fabrieksreset zijn omdat de gebruikers geen toegang hebben tot het menu Instellingen van de telefoon.