

# CUCM Mix-modus met Tokenloze CTL

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[Van niet-beveiligde modus naar gemengde modus \(Token less CTL\)](#)

[Van hardwareTokens tot kenloze oplossing](#)

[Van Tokenless Solutions to Hardware Tokens](#)

[certificaatregeneratie voor kenloze CTL-oplossing](#)

## Inleiding

Dit document beschrijft het verschil tussen Cisco Unified Communications Manager (CUCM)-beveiliging met en zonder het gebruik van hardware-USB-penningen. In dit document worden ook de basisuitvoeringsscenario's beschreven die Tokenless certificaatlijst (CTL) omvatten en het proces dat wordt gebruikt om ervoor te zorgen dat het systeem na de wijzigingen naar behoren functioneert.

## Voorwaarden

### Vereisten

Cisco raadt u aan om kennis te hebben van CUCM versie 10.0(1) of hoger. Zorg er bovendien voor dat:

- Uw licentieserver voor CUCM versie 11.5.1SU3 en hoger moet Cisco Prime License Manager (PLM) 11.5.1SU2 of hoger zijn. Dit komt doordat CUCM versie 11.5.1SU3 de encryptie-licentie vereist om gemengde modus mogelijk te maken en PLM de encryptie-licentie niet ondersteunt tot 11.5.1SU2. Raadpleeg voor meer informatie de [Releaseopmerkingen van Cisco Prime License Manager, release 11.5\(1\)SU2](#).
- U hebt toegang tot de Opdracht Line Interface (CLI) van het CUCM Publisher-knooppunt.
- U hebt toegang tot de hardware-USB-penningen en u kunt zien dat de CTL-clientstekker op uw PC is geïnstalleerd voor scenario's die u ertoe nopen terug te migreren naar het gebruik van hardwareTokens. Voor meer helderheid is deze eis slechts vereist als u op elk moment een scenario hebt waar de USB-penningen nodig zijn. De kans is klein dat USB-penningen voor de meeste mensen nodig zijn.
- Er is volledige connectiviteit tussen alle CUCM-knooppunten in de cluster. Dit is zeer belangrijk omdat het CTL-bestand wordt gekopieerd naar alle knooppunten in het cluster via

SSH File Transfer Protocol (SFTP).

- De DB-replicatie van de database in de cluster werkt goed en de servers repliceren de gegevens in real-time.
- De apparaten in uw plaatsing steunen Security door Standaard (TVS). U kunt de *functiekaart voor Unified CM-telefoon* gebruiken vanuit de Cisco Unified Reporting website (<https://<CUCM IP of FQDN>/ucreports/>) om de apparaten te bepalen die Beveiliging door standaard ondersteunen.

Opmerking: Cisco Jabber en veel Cisco TelePresence of Cisco 7940/7960 Series IP-telefoons ondersteunen momenteel geen beveiliging door standaard. Als u Tokenless CTL met apparaten implementeert die Beveiliging door Standaard niet ondersteunen, zal elke update aan uw systeem die het certificaat CallManager op de uitgever verandert de normale functionaliteit van die apparaten verhinderen tot het CTL handmatig wordt verwijderd. Apparaten die Security ondersteunen door Standaard, zoals 7945 en 7965 telefoons of nieuwer, kunnen CTL bestanden installeren wanneer het CallManager-certificaat op de uitgever wordt bijgewerkt omdat ze de TVS (Trust Verification Service) kunnen gebruiken.

## Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- CUCM versie 10.5.1.1000-7 (cluster van twee knooppunten)
- Cisco 7975 Series IP-telefoons die zijn geregistreerd via Sony Client Control Protocol (SCCP) met firmware versie SCCP75.9-3-1SR4-1S
- Twee Cisco Security Tokens die worden gebruikt om het cluster in te stellen op Gemengde modus met het gebruik van CTL-clientsoftware

## Achtergrondinformatie

Tokenless CTL is een nieuwe functie in CUCM versies 10.0(1) en later die de encryptie van callsignalering en media voor IP-telefoons mogelijk maakt zonder dat hardware-USB-penningen en de CTL-clientstekker hoeven te worden gebruikt, wat de vereiste was in eerdere CUCM-releases.

Wanneer het cluster in Gemengde modus wordt gezet met het gebruik van de CLI-opdracht, wordt het CTL-bestand getekend met het CCM+TFTP-certificaat (server) van het Uitgevers-knooppunt en zijn er geen eToken-certificaten aanwezig in het CTL-bestand.

Opmerking: Wanneer u het CallManager (CCM+TFTP)-certificaat op de uitgever regeneert, verandert dit de naam van het bestand. De telefoons en apparaten die geen Beveiliging door Standaard ondersteunen zullen het nieuwe CTL-bestand niet accepteren tenzij de CTL-bestanden handmatig van elk apparaat worden verwijderd. Raadpleeg het laatste voorschrift dat in het gedeelte [Eisen](#) van dit document is opgenomen voor meer informatie.

# Van niet-beveiligde modus naar gemengde modus (Token less CTL)

In deze sectie wordt het proces beschreven dat wordt gebruikt om de CUCM-clusterbeveiliging naar gemengde modus via de CLI te verplaatsen.

Voor dit scenario was CUCM in Non-Secure modus, wat betekent dat er geen CTL-bestand aanwezig was op een van de knooppunten en dat de geregistreerde IP-telefoons alleen een ITL-bestand (Identity Trust List) hadden geïnstalleerd, zoals in deze output wordt getoond:

```
admin:show ctl
Length of CTL file: 0
CTL File not found. Please run CTLClient plugin or run the CLI - utils ctl.. to
generate the CTL file. Error parsing the CTL File. admin:
```

Opmerking: Als er een CTL-bestand op de server is gevonden terwijl de cluster niet in gemengde modus is, betekent dit dat de cluster eenmaal in gemengde modus was en vervolgens naar niet-gemengde modus werd teruggebracht en het CTL-bestand niet uit het cluster werd verwijderd.

Het opdrachtbestand **verwijderd activelog cm/ftpdata/CTLFile.tlv** verwijderd het CTL-bestand uit knooppunten in de CUCM-cluster. de opdracht moet echter op elk knooppunt worden ingevoerd. Gebruik deze opdracht alleen als uw servers een CTL-bestand hebben en de cluster niet in gemengde modus is.

Een makkelijke manier om te bevestigen als een cluster in gemengde modus is om de opdracht **run sql** te gebruiken **selecteert paramname, paramvalue van procesconfiguratie waar paramname='ClusterSecurityMode'**. Als de paramwaarde 0 is, dan is het cluster niet in gemengde modus.

```
run sql select paramname,paramvalue from processconfig where paramname='ClusterSecurityMode'
paramname          paramvalue
=====
ClusterSecurityMode 0
```



Voltooi de volgende stappen om de beveiliging van het CUCM-cluster in gemengde modus te verplaatsen met behulp van de nieuwe functie Tokenless CTL:

1. Beheerstoegang verkrijgen tot het CUCM Publisher-knooppunt CLI.
2. Voer de `utils ctl set-cluster gemengde-mode` opdracht in in de CLI:

```
admin:utils ctl set-cluster mixed-mode
This operation will set the cluster to Mixed mode. Do you want to continue? (y/n):y

Moving Cluster to Mixed Mode
Cluster set to Mixed Mode
Please Restart the TFTP and Cisco CallManager services on all nodes in the cluster
that run these services
admin:
```

3. Navigeer naar **CUCM Admin Pagina > Systeem > Enterprise-parameters** en controleer of het cluster op Gemengde modus is ingesteld (een waarde van 1 geeft Gemengde modus aan):

Security Parameters	
<a href="#">Cluster Security Mode</a> *	1
<a href="#">LBM Security Mode</a> *	Insecure ▼
<a href="#">CAPF Phone Port</a> *	3804
<a href="#">CAPF Operation Expires in (days)</a> *	10
<a href="#">Enable Caching</a> *	True ▼

4. Start de TFTP- en Cisco CallManager-services opnieuw op alle knooppunten in de cluster die deze services uitvoeren.
5. Start alle IP-telefoons opnieuw zodat ze het CTL-bestand kunnen verkrijgen van de CUCM TFTP-service.
6. Om de inhoud van het CTL-bestand te verifiëren, voer de `show ctl`-opdracht in in de CLI. In

het CTL-bestand kunt u zien dat het CCM+TFTP-certificaat (server) voor het CUCM Publisher-knooppunt wordt gebruikt om het CTL-bestand te tekenen (dit bestand is hetzelfde op alle servers in het cluster). Hier wordt een voorbeelduitvoer weergegeven:

```
admin:show ctl
The checksum value of the CTL file:
0c05655de63fe2a042cf252d96c6d609 (MD5)
8c92d1a569f7263cf4485812366e66e3b503a2f5 (SHA1)

Length of CTL file: 4947
The CTL File was last modified on Fri Mar 06 19:45:13 CET 2015
```

[...]

```
CTL Record #:1
----
BYTEPOS TAG LENGTH VALUE
-----
1 RECORDLENGTH 2 1156
2 DNSNAME 16 cucm-1051-a-pub
3 SUBJECTNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
4 FUNCTION 2 System Administrator Security Token
5 ISSUERNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
6 SERIALNUMBER 16 70:CA:F6:4E:09:07:51:B9:DF:22:F4:9F:75:4F:C5:BB
7 PUBLICKEY 140
8 SIGNATURE 128
9 CERTIFICATE 694 E9 D4 33 64 5B C8 8C ED 51 4D 8F E5 EA 5B 6D 21
A5 A3 8C 9C (SHA1 Hash HEX)
10 IPADDRESS 4 This etoken was used to sign the CTL file.
CTL Record #:2
```

```
----
BYTEPOS TAG LENGTH VALUE
-----
1 RECORDLENGTH 2 1156
2 DNSNAME 16 cucm-1051-a-pub
3 SUBJECTNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
4 FUNCTION 2 CCM+TFTP
5 ISSUERNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
6 SERIALNUMBER 16 70:CA:F6:4E:09:07:51:B9:DF:22:F4:9F:75:4F:C5:BB
7 PUBLICKEY 140
8 SIGNATURE 128
9 CERTIFICATE 694 E9 D4 33 64 5B C8 8C ED 51 4D 8F E5 EA 5B 6D 21
A5 A3 8C 9C (SHA1 Hash HEX)
10 IPADDRESS 4
```

[...]

The CTL file was verified successfully.

7. Aan de kant van de IP-telefoon kunt u verifiëren dat nadat de service opnieuw is gestart het CTL-bestand wordt gedownload, dat nu op de TFTP-server aanwezig is (de MD5-checksum komt overeen met de uitvoer van CUCM):

**Opmerking:** wanneer u de checksum aan de telefoon controleert, ziet u **MD5** of **SHA1**, afhankelijk van het type telefoon.



## Van hardwareTokens tot kenloze oplossing

In dit deel wordt beschreven hoe de CUCM-clusterbeveiliging van hardware-penningen naar het gebruik van de nieuwe Tokenless-oplossing kan worden gemigreerd.

In sommige situaties is Gemengde modus al ingesteld op CUCM met behulp van de CTL-client en gebruiken de IP-telefoons CTL-bestanden die de certificaten van de hardware-USB-penningen bevatten. Bij dit scenario wordt het CTL-bestand getekend door een certificaat van een van de USB-Tokens en is het geïnstalleerd op de IP-telefoons. Hier in een voorbeeld:

```
admin:show ctl
The checksum value of the CTL file:
256a661f4630cd86ef460db5aad4e91c (MD5)
3d56cc01476000686f007aac6c278ed9059fc124 (SHA1)

Length of CTL file: 5728
The CTL File was last modified on Fri Mar 06 21:48:48 CET 2015

[...]
CTL Record #:5
-----
BYTEPOS TAG LENGTH VALUE
-----
1 RECORDLENGTH 2 1186
2 DNSNAME 1
3 SUBJECTNAME 56 cn="SAST-ADN008580ef ";ou=IPCBU;o="Cisco Systems
4 FUNCTION 2 System Administrator Security Token
5 ISSUERNAME 42 cn=Cisco Manufacturing CA;o=Cisco Systems
6 SERIALNUMBER 10 83:E9:08:00:00:00:55:45:AF:31
7 PUBLICKEY 140
9 CERTIFICATE 902 85 CD 5D AD EA FC 34 B8 3E 2F F2 CB 9C 76 B0 93
3E 8B 3A 4F (SHA1 Hash HEX)
10 IPADDRESS 4
This etoken was used to sign the CTL file.

The CTL file was verified successfully.
```



Voltooi deze stappen om de CUCM-clusterbeveiliging te verplaatsen naar het gebruik van Tokenless CTL's:

1. Beheerstoegang verkrijgen tot het CUCM Publisher-knooppunt CLI.
2. Typ de opdracht **utils ctl update CTLFile** CLI:

```
admin:utils ctl update CTLFile
This operation will update the CTLFile. Do you want to continue? (y/n):y

Updating CTL file
CTL file Updated
Please Restart the TFTP and Cisco CallManager services on all nodes in
the cluster that run these services
```

3. Start de TFTP- en CallManager-services opnieuw op alle knooppunten in de cluster die deze services uitvoeren.
4. Start alle IP-telefoons opnieuw zodat ze het CTL-bestand kunnen verkrijgen van de CUCM TFTP-service.
5. Voer de opdracht **show ctl** in in de CLI om de inhoud van het CTL-bestand te controleren. In het CTL-bestand kunt u zien dat het CCM+TFTP-certificaat (server) van het CUCM Publisher-knooppunt wordt gebruikt om het CTL-bestand te tekenen in plaats van het certificaat van de hardware-USB-penningen. Een belangrijker verschil in dit geval is dat de certificaten van alle hardware USB eTokens uit het CTL-bestand worden verwijderd. Hier wordt een voorbeelduitvoer weergegeven:

```
admin:show ctl
The checksum value of the CTL file:
1d97d9089dd558a062cccfcb1dc4c57f (MD5)
3b452f9ec9d6543df80e50f8b850cddc92fcf847 (SHA1)

Length of CTL file: 4947
The CTL File was last modified on Fri Mar 06 21:56:07 CET 2015
```

[...]

CTL Record #:1

----

BYTEPOS TAG LENGTH VALUE

-----

```
1 RECORDLENGTH 2 1156
2 DNSNAME 16 cucm-1051-a-pub
3 SUBJECTNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
4 FUNCTION 2 System Administrator Security Token
5 ISSUERNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
6 SERIALNUMBER 16 70:CA:F6:4E:09:07:51:B9:DF:22:F4:9F:75:4F:C5:BB
7 PUBLICKEY 140
8 SIGNATURE 128
9 CERTIFICATE 694 E9 D4 33 64 5B C8 8C ED 51 4D 8F E5 EA 5B 6D
21 A5 A3 8C 9C (SHA1 Hash HEX)
10 IPADDRESS 4
```

**This etoken was used to sign the CTL file.**

CTL Record #:2

----

BYTEPOS TAG LENGTH VALUE

-----

```
1 RECORDLENGTH 2 1156
2 DNSNAME 16 cucm-1051-a-pub
3 SUBJECTNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
4 FUNCTION 2 CCM+TFTP
5 ISSUERNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
6 SERIALNUMBER 16 70:CA:F6:4E:09:07:51:B9:DF:22:F4:9F:75:4F:C5:BB
7 PUBLICKEY 140
8 SIGNATURE 128
9 CERTIFICATE 694 E9 D4 33 64 5B C8 8C ED 51 4D 8F E5 EA 5B 6D
21 A5 A3 8C 9C (SHA1 Hash HEX)
10 IPADDRESS 4
```

[...]

The CTL file was verified successfully.

**Opmerking:** In de bovenstaande uitvoer, als CCM+TFTP (server) certificaat van de CUCM Publisher niet is getekend, verplaats dan terug naar de op hardware gebaseerde clusterbeveiligingsmodus en herhaal de wijzigingen opnieuw voor een ondoorzichtige oplossing.

6. Aan de kant van de IP-telefoon kunt u controleren dat nadat de IP-telefoons opnieuw zijn opgestart, ze de bijgewerkte versie van het CTL-bestand hebben gedownload (de MD5-checksum komt overeen met de uitvoer van CUCM):





## Van Tokenless Solutions to Hardware Tokens

In dit deel wordt beschreven hoe de CUCM-clusterbeveiliging van de nieuwe Tokenless-oplossing moet worden weggesluisd en kan worden teruggebracht naar het gebruik van hardware-eTokens.

Wanneer de CUCM-clusterbeveiliging is ingesteld op Gemengde modus met het gebruik van de CLI-opdrachten, en het CTL-bestand is getekend met het CCM+TFTP-certificaat (server) voor het CUCM Publisher-knooppunt, bestaan er geen certificaten van de hardware-USB-penningen in het CTL-bestand. Om deze reden, wanneer u de CTL client runt om het CTL bestand bij te werken (verplaatst naar het gebruik van hardware Tokens), verschijnt deze foutmelding:

```
The Security Token you have inserted does not exist in the CTL File
Please remove any Security Tokens already inserted and insert another
Security Token. Click Ok when done.
```

Dit is vooral belangrijk in scenario's die een downgrade (wanneer de versie wordt teruggezet) van het systeem omvatten naar een pre-10.x versie die de `utils ctl`-opdrachten niet bevat. Het vorige CTL-bestand wordt gemigreerd (zonder wijzigingen in de inhoud) tijdens het proces van een vernieuwing of een Linux-upgrade (L2), en bevat niet de eerder genoemde eToken-certificaten. Hier wordt een voorbeelduitvoer weergegeven:

```
admin:show ctl
The checksum value of the CTL file:
1d97d9089dd558a062cccfcbldc4c57f (MD5)
3b452f9ec9d6543df80e50f8b850cddc92fcf847 (SHA1)

Length of CTL file: 4947
The CTL File was last modified on Fri Mar 06 21:56:07 CET 2015

Parse CTL File
-----

Version: 1.2
HeaderLength: 336 (BYTES)

BYTEPOS TAG LENGTH VALUE
-----
```

3 SIGNERID 2 149  
4 SIGNERNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;  
ST=Malopolska;C=PL  
5 SERIALNUMBER 16 70:CA:F6:4E:09:07:51:B9:DF:22:F4:9F:75:4F:C5:BB  
6 CANAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;  
ST=Malopolska;C=PL  
7 SIGNATUREINFO 2 15  
8 DIGESTALGORTITHM 1  
9 SIGNATUREALGOINFO 2 8  
10 SIGNATUREALGORTITHM 1  
11 SIGNATUREMODULUS 1  
12 SIGNATURE 128  
65 ba 26 b4 ba de 2b 13  
b8 18 2 4a 2b 6c 2d 20  
7d e7 2f bd 6d b3 84 c5  
bf 5 f2 74 cb f2 59 bc  
b5 c1 9f cd 4d 97 3a dd  
6e 7c 75 19 a2 59 66 49  
b7 64 e8 9a 25 7f 5a c8  
56 bb ed 6f 96 95 c3 b3  
72 7 91 10 6b f1 12 f4  
d5 72 e 8f 30 21 fa 80  
bc 5d f6 c5 fb 6a 82 ec  
f1 6d 40 17 1b 7d 63 7b  
52 f7 7a 39 67 e1 1d 45  
b6 fe 82 0 62 e3 db 57  
8c 31 2 56 66 c8 91 c8  
d8 10 cb 5e c3 1f ef a  
14 FILENAME 12  
15 TIMESTAMP 4

CTL Record #:1

----

BYTEPOS TAG LENGTH VALUE

-----

1 RECORDLENGTH 2 1156  
2 DNSNAME 16 cucm-1051-a-pub  
3 SUBJECTNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;  
ST=Malopolska;C=PL  
4 FUNCTION 2 System Administrator Security Token  
5 ISSUERNAM 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;  
ST=Malopolska;C=PL  
6 SERIALNUMBER 16 **70:CA:F6:4E:09:07:51:B9:DF:22:F4:9F:75:4F:C5:BB**  
7 PUBLICKEY 140  
8 SIGNATURE 128  
9 CERTIFICATE 694 E9 D4 33 64 5B C8 8C ED 51 4D 8F E5 EA 5B 6D  
21 A5 A3 8C 9C (SHA1 Hash HEX)  
10 IPADDRESS 4

**This etoken was used to sign the CTL file.**

CTL Record #:2

----

BYTEPOS TAG LENGTH VALUE

-----

1 RECORDLENGTH 2 1156  
2 DNSNAME 16 cucm-1051-a-pub  
3 SUBJECTNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;  
ST=Malopolska;C=PL  
4 FUNCTION 2 **CCM+TFTP**  
5 ISSUERNAM 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;  
ST=Malopolska;C=PL  
6 SERIALNUMBER 16 **70:CA:F6:4E:09:07:51:B9:DF:22:F4:9F:75:4F:C5:BB**  
7 PUBLICKEY 140  
8 SIGNATURE 128

```
9 CERTIFICATE 694 E9 D4 33 64 5B C8 8C ED 51 4D 8F E5 EA 5B 6D
21 A5 A3 8C 9C (SHA1 Hash HEX)
10 IPADDRESS 4
```

CTL Record #:3

----

BYTEPOS TAG LENGTH VALUE

-----

```
1 RECORDLENGTH 2 1138
2 DNSNAME 16 cucm-1051-a-pub
3 SUBJECTNAME 60 CN=CAPF-e41e7d87;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
4 FUNCTION 2 CAPF
5 ISSUERNAME 60 CN=CAPF-e41e7d87;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
6 SERIALNUMBER 16 74:4B:49:99:77:04:96:E7:99:E9:1E:81:D3:C8:10:9B
7 PUBLICKEY 140
8 SIGNATURE 128
9 CERTIFICATE 680 46 EE 5A 97 24 65 B0 17 7E 5F 7E 44 F7 6C 0A
F3 63 35 4F A7 (SHA1 Hash HEX)
10 IPADDRESS 4
```

CTL Record #:4

----

BYTEPOS TAG LENGTH VALUE

-----

```
1 RECORDLENGTH 2 1161
2 DNSNAME 17 cucm-1051-a-sub1
3 SUBJECTNAME 63 CN=cucm-1051-a-sub1;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
4 FUNCTION 2 CCM+TFTP
5 ISSUERNAME 63 CN=cucm-1051-a-sub1;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
6 SERIALNUMBER 16 6B:EB:FD:CD:CD:8C:A2:77:CB:2F:D1:D1:83:A6:0E:72
7 PUBLICKEY 140
8 SIGNATURE 128
9 CERTIFICATE 696 21 7F 23 DE AF FF 04 85 76 72 70 BF B1 BA 44
DB 5E 90 ED 66 (SHA1 Hash HEX)
10 IPADDRESS 4
```

The CTL file was verified successfully.

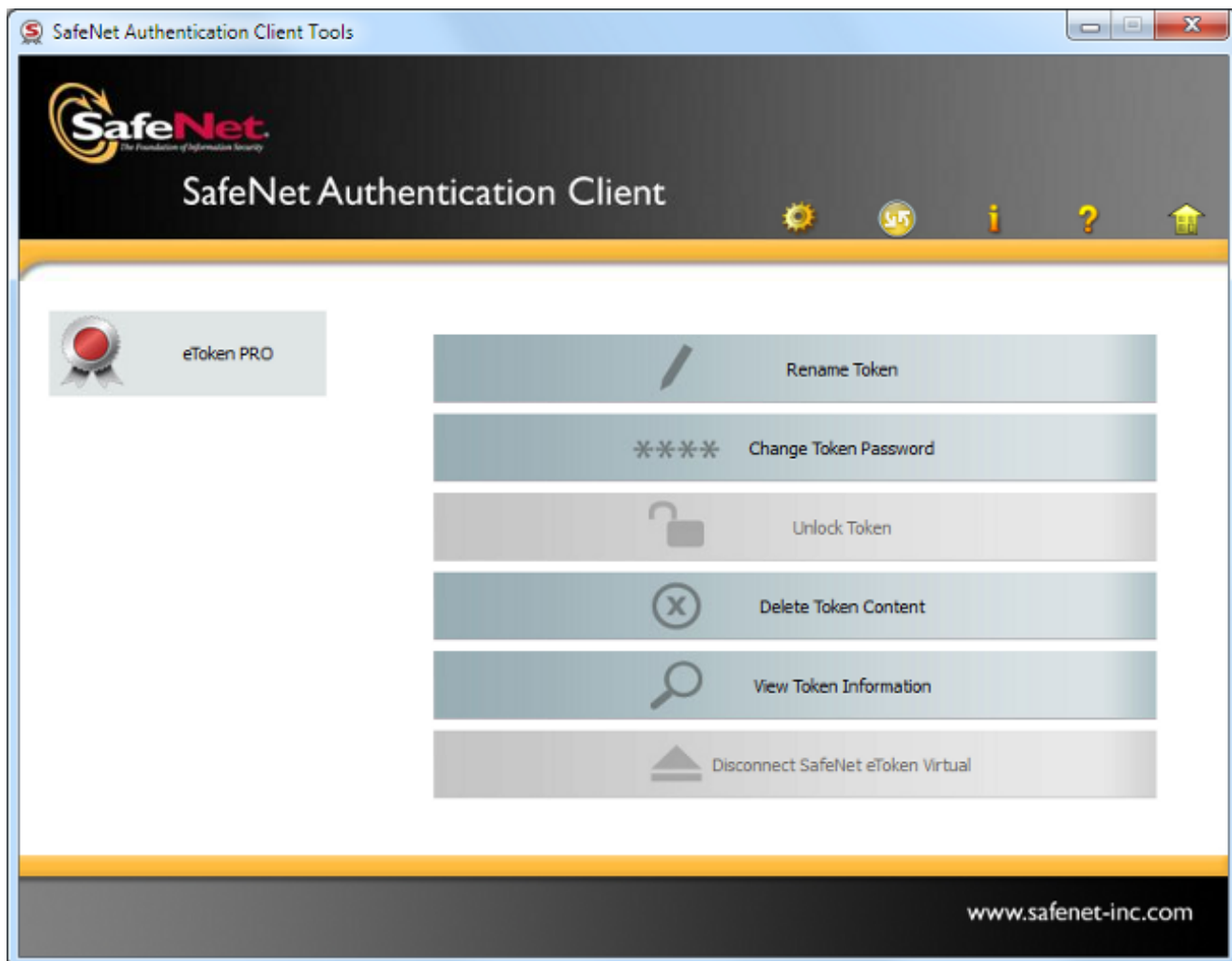
admin:

Voor dit scenario, voltooi deze stappen om de CTL bestanden veilig bij te werken zonder de procedure te gebruiken voor verloren eTokens, wat in handmatige verwijdering van het CTL-bestand uit alle IP-telefoons eindigt:

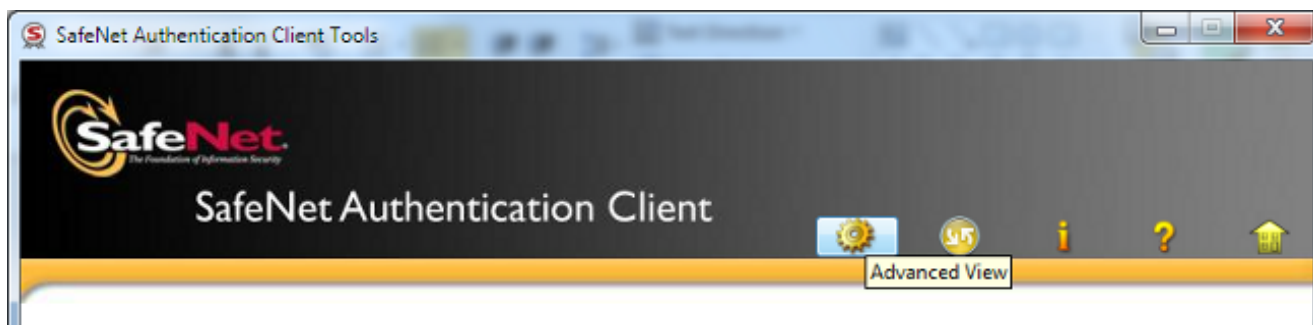
1. Beheerstoegang verkrijgen tot het CUCM Publisher-knooppunt CLI.
2. Voer de opdracht **voor het verwijderen van tftp CTLFile.tlv** in in het knooppunt CLI van de Uitgever om het CTL-bestand te verwijderen:

```
admin:file delete tftp CTLFile.tlv
Delete the File CTLFile.tlv?
Enter "y" followed by return to continue: y
files: found = 1, deleted = 1
```

3. Open **SafeNet-verificatie-client** op de Microsoft Windows-machine waar de CTL-client is geïnstalleerd (deze wordt automatisch met CTL-client geïnstalleerd):

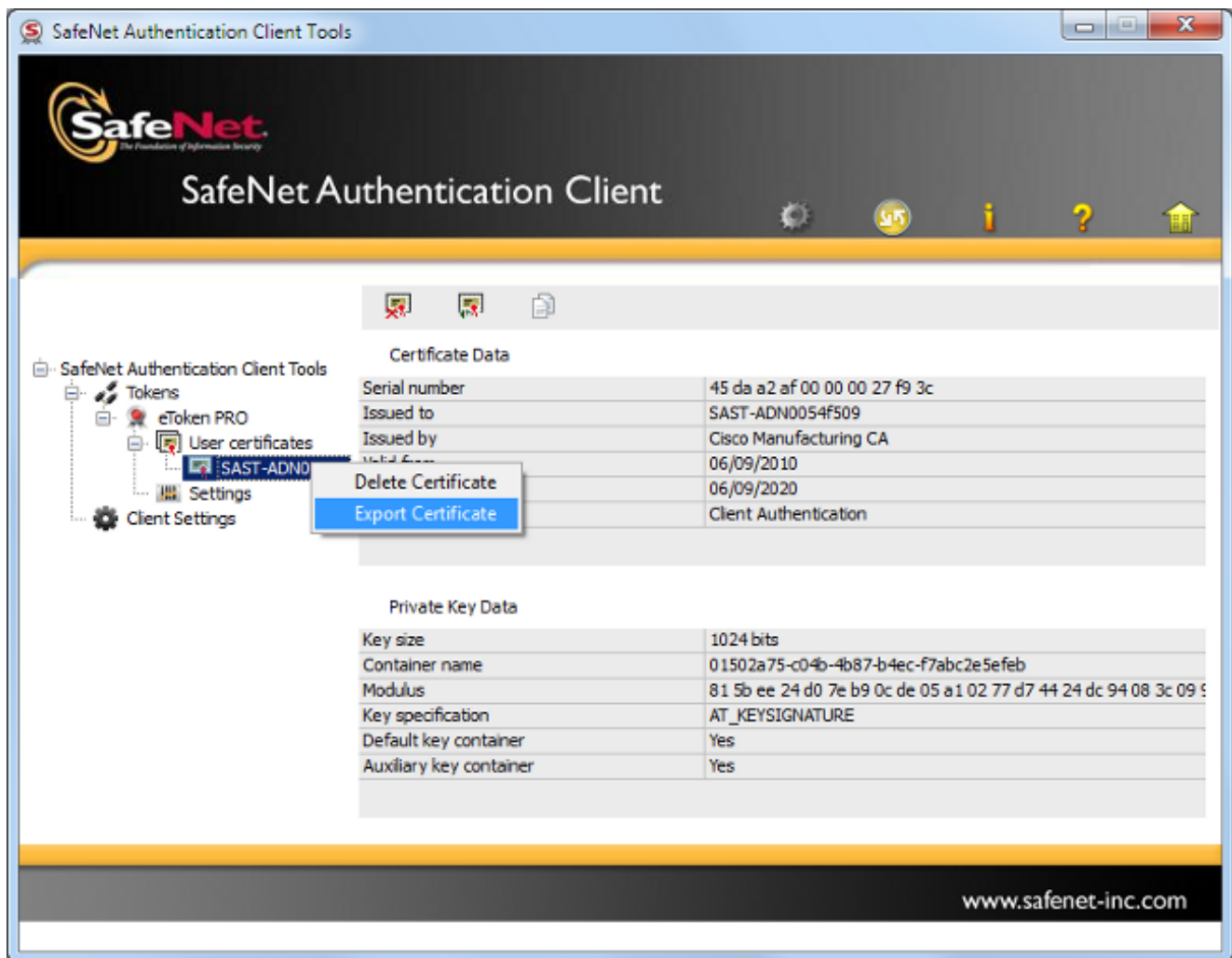


4. In SafeNet-verificatie-client navigeren naar de *geavanceerde weergave*:



5. Plaats de eerste USB-hardware.

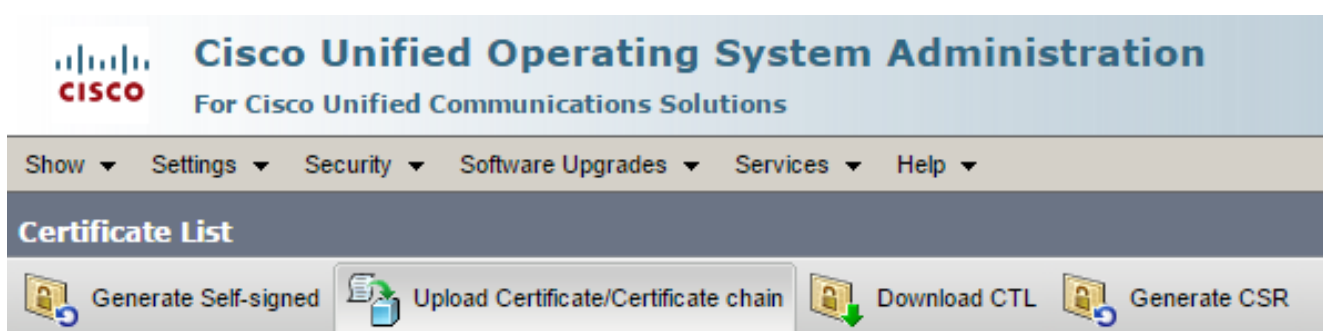
6. Selecteer het certificaat onder de map *Gebruikerscertificaten* en voer het naar de map op de pc. Wanneer u om een wachtwoord wordt gevraagd, gebruikt u het standaardwachtwoord van **Cisco123**:



7. Herhaal deze stappen voor de tweede hardware-USB-Token zodat beide certificaten naar de PC worden geëxporteerd:

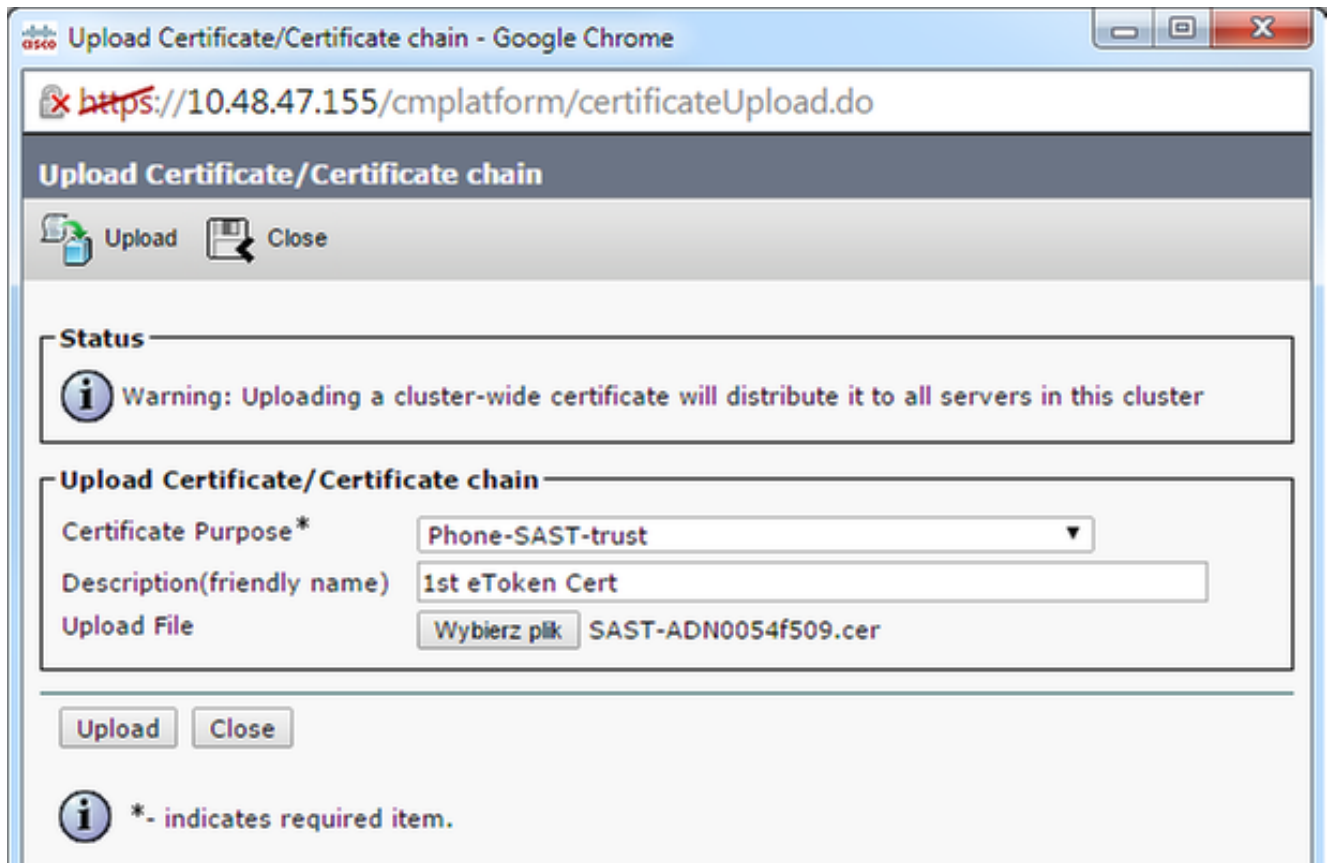
Name	Date modified	Type	Size
SAST-ADN0054f509	06-03-2015 22:32	Security Certificate	1 KB
SAST-ADN008580ef	06-03-2015 22:33	Security Certificate	1 KB

8. Meld u aan bij het Cisco Unified Operating System (OS) Management en navigeer naar **Security > certificaatbeheer > Upload Certificate**:

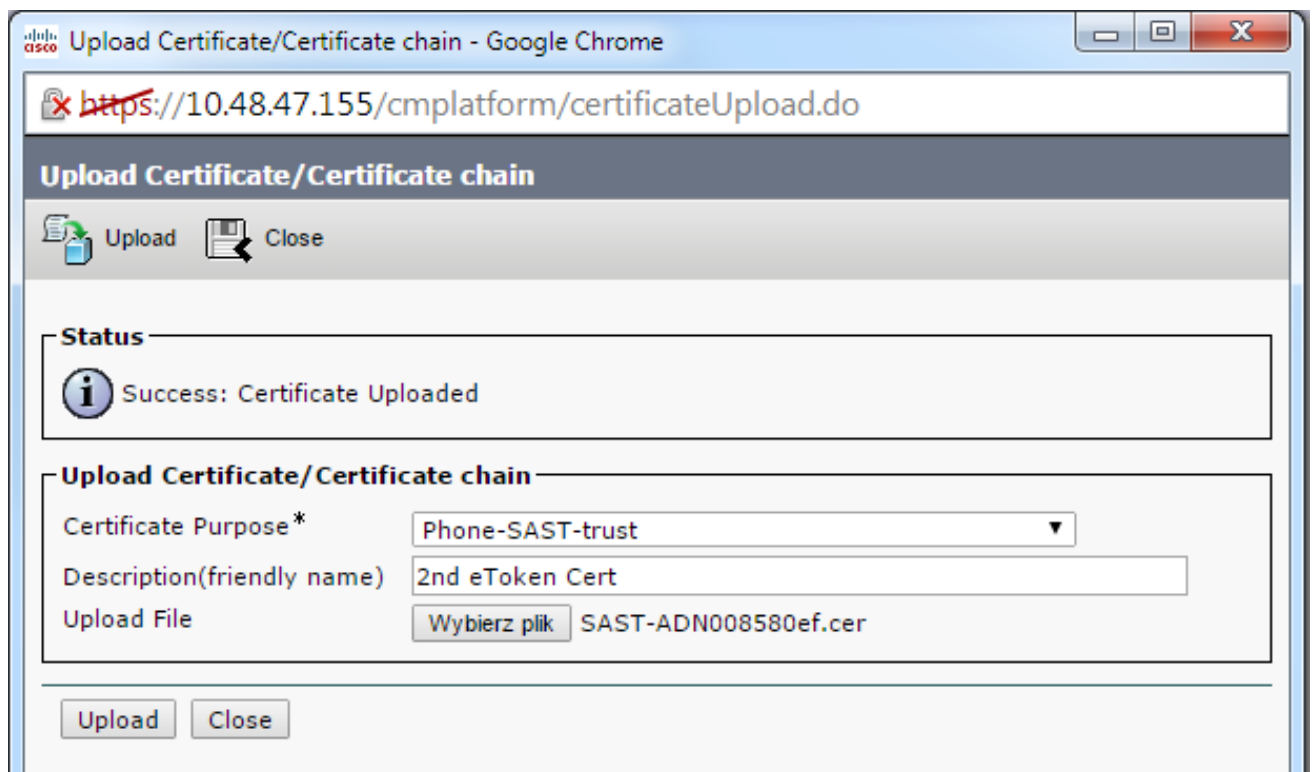


9. De pagina met uploadcertificaat wordt vervolgens weergegeven. Kies **telefoon-SAST-trust** in het uitrolmenu certificaatdoel en selecteer het certificaat dat u vanuit het eerste Token hebt

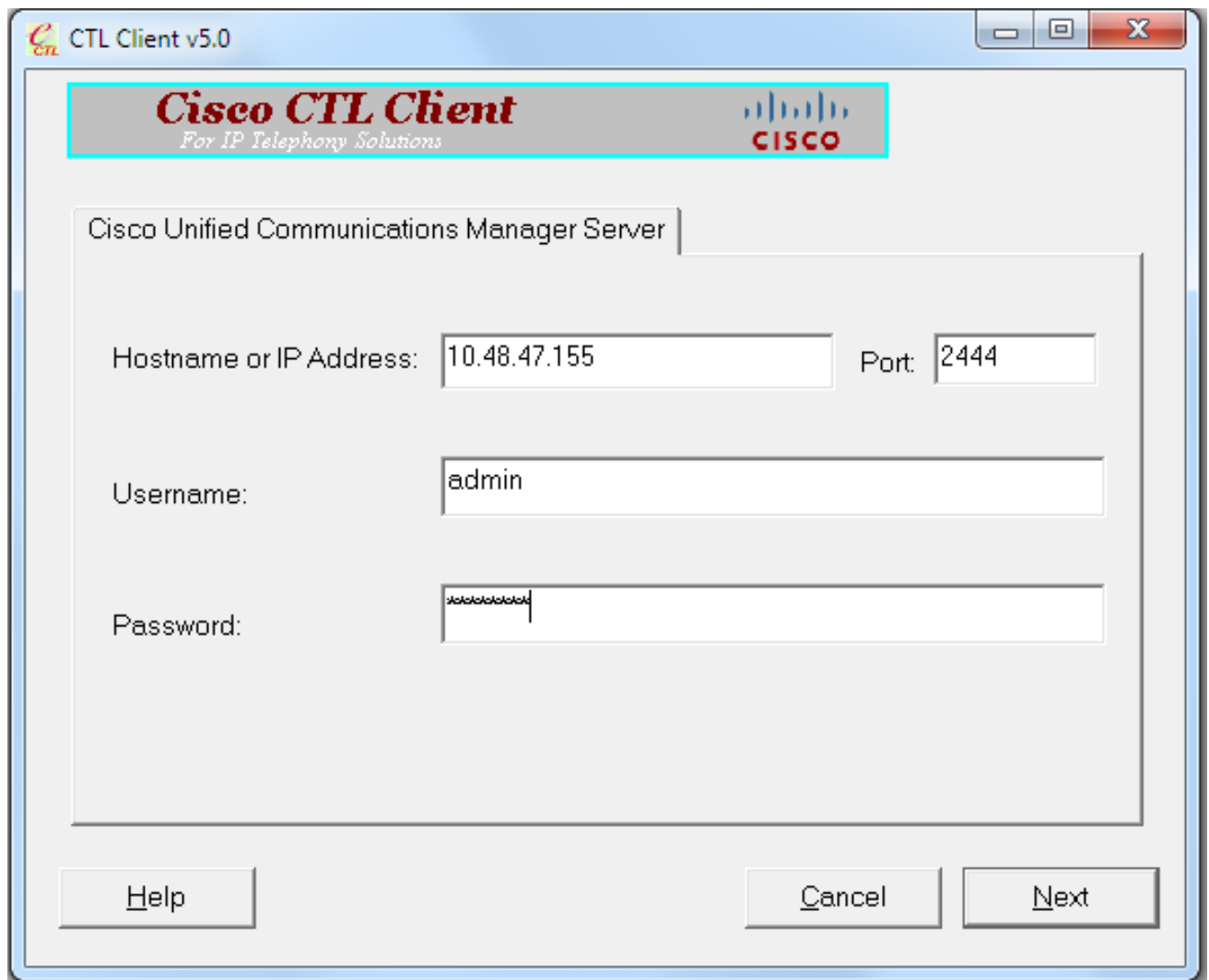
geëxporteerd:



10. Voltooi de vorige stappen om het certificaat te uploaden dat u vanuit het tweede Token hebt geëxporteerd:



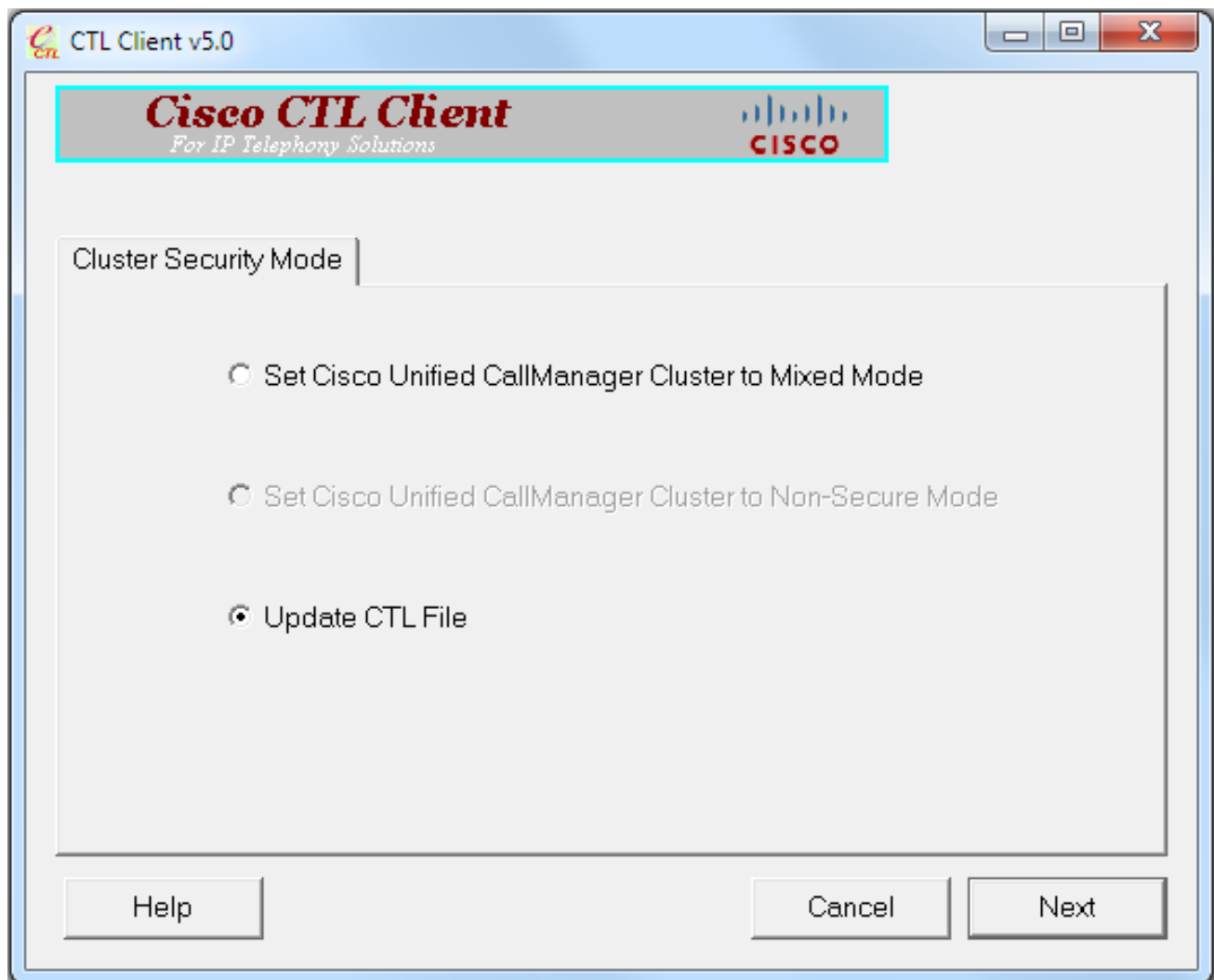
11. Start de CTL-client, voer het IP-adres/hostname van het CUCM-knooppunt in en voer de CCM-Administrator-referenties in:



12. Aangezien het cluster al in Gemengde modus is, maar er geen CTL-bestand op het knooppunt van de Uitgever bestaat, verschijnt dit waarschuwingsbericht (klik op **OK** om het te negeren):

No CTL File exists on the server but the Call Manager Cluster Security Mode is in Secure Mode.  
For the system to function, you must create the CTL File and set Call Manager Cluster the Secure Mode.

13. Klik vanuit de CTL-client op de radioknop **Update CTL File** en klik vervolgens op **Next**:

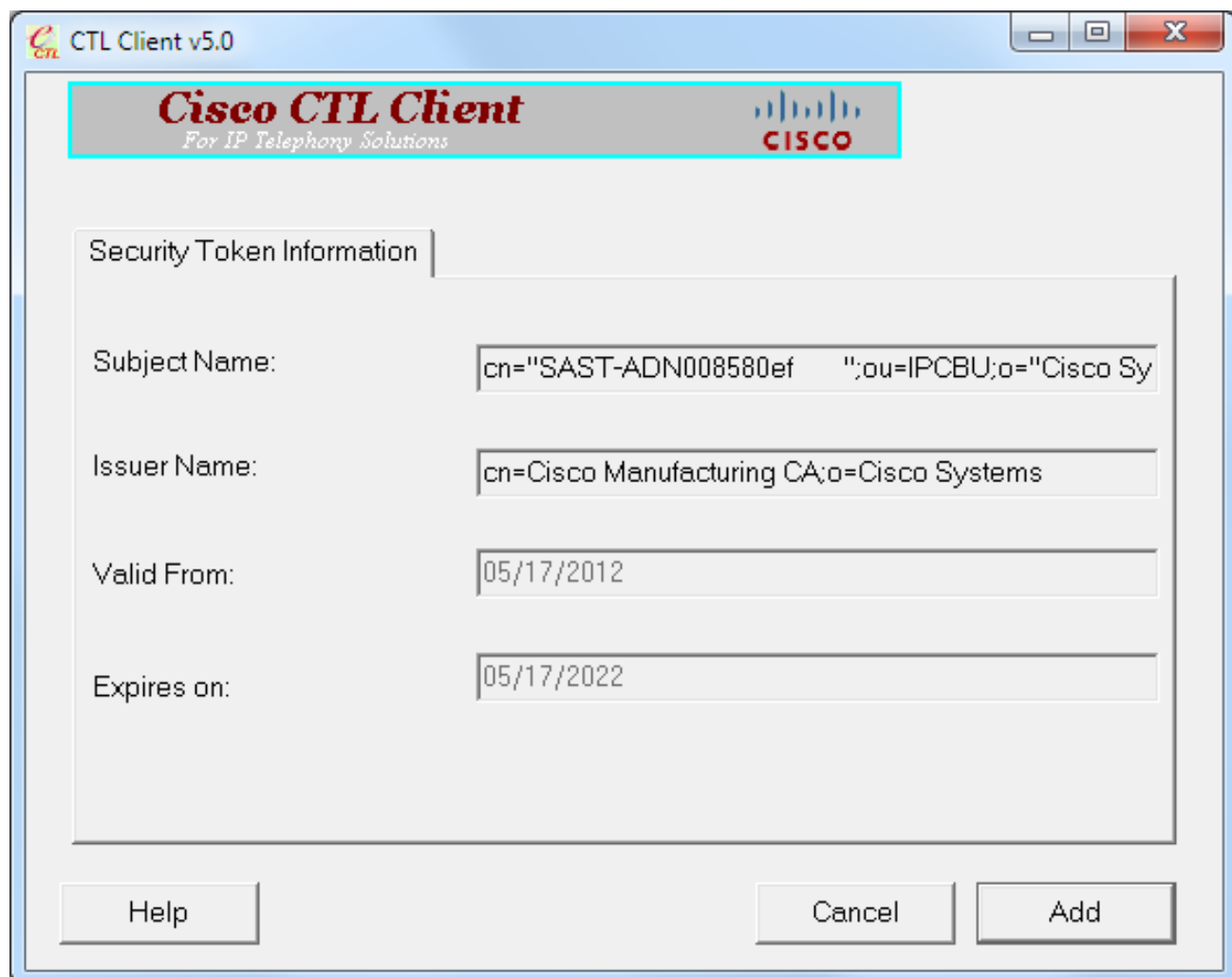


14. Plaats het eerste beveiligingstoken en klik op **OK**:

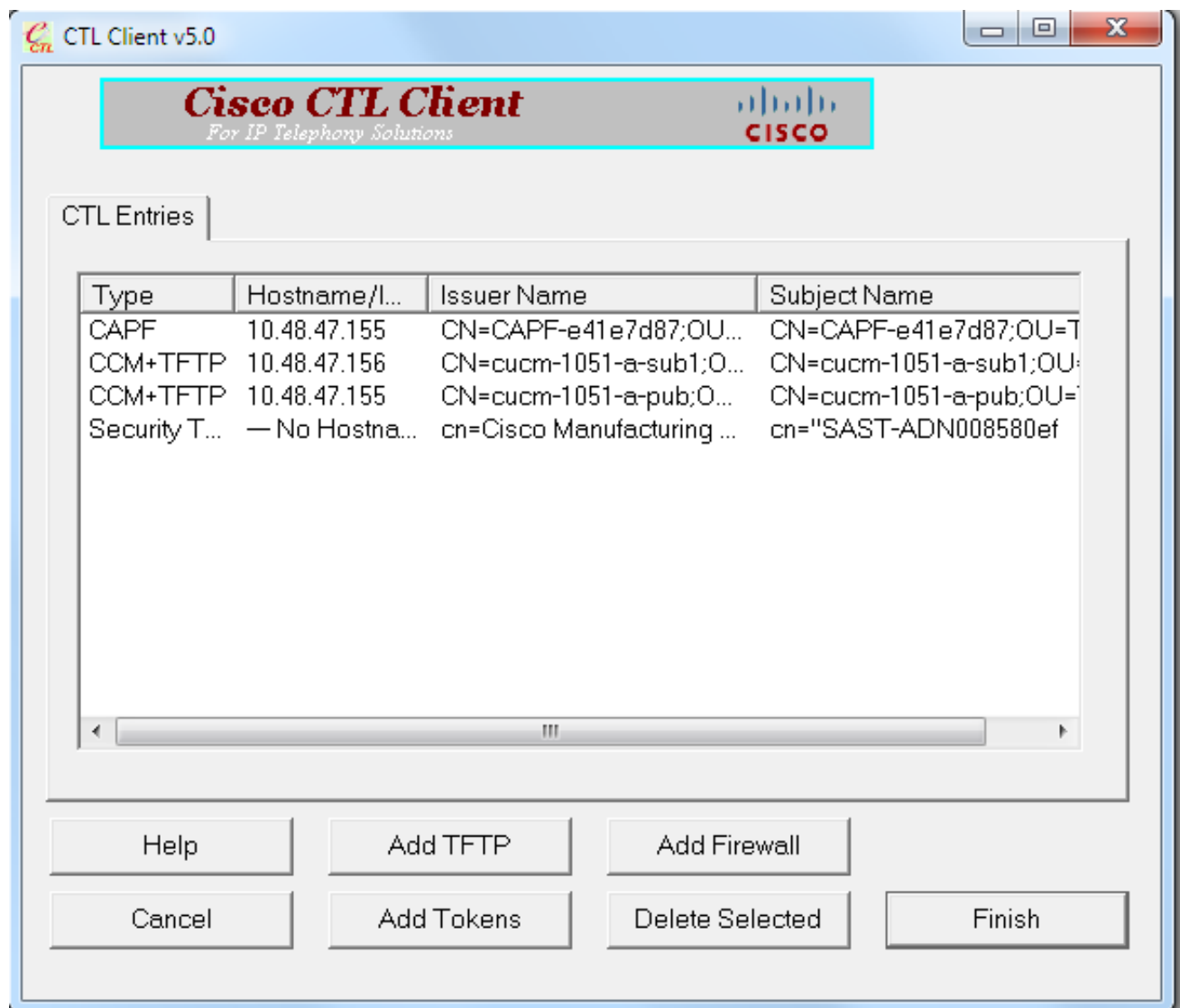


15. Klik nadat de informatie over beveiligingsToken is weergegeven op **Toevoegen**:

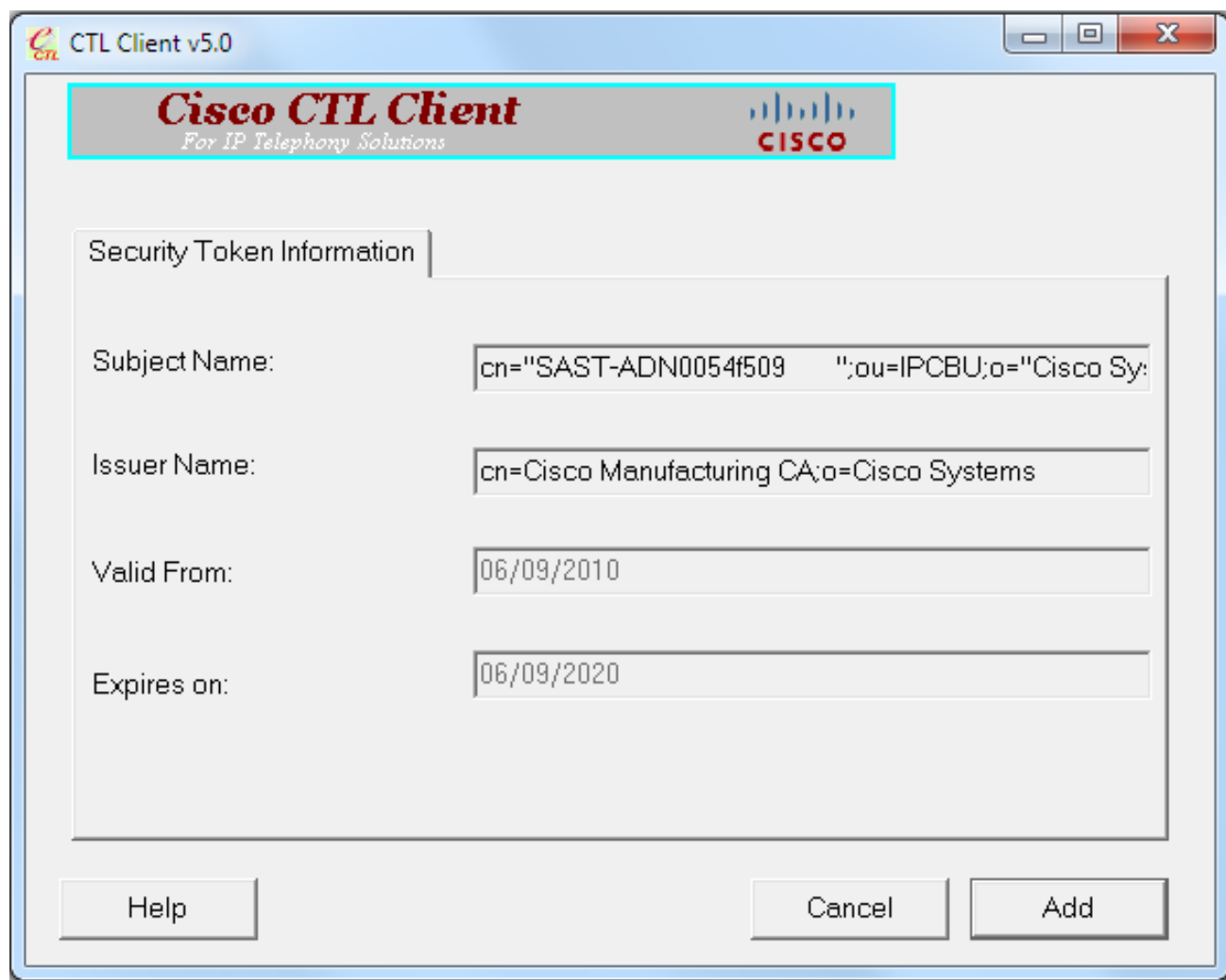




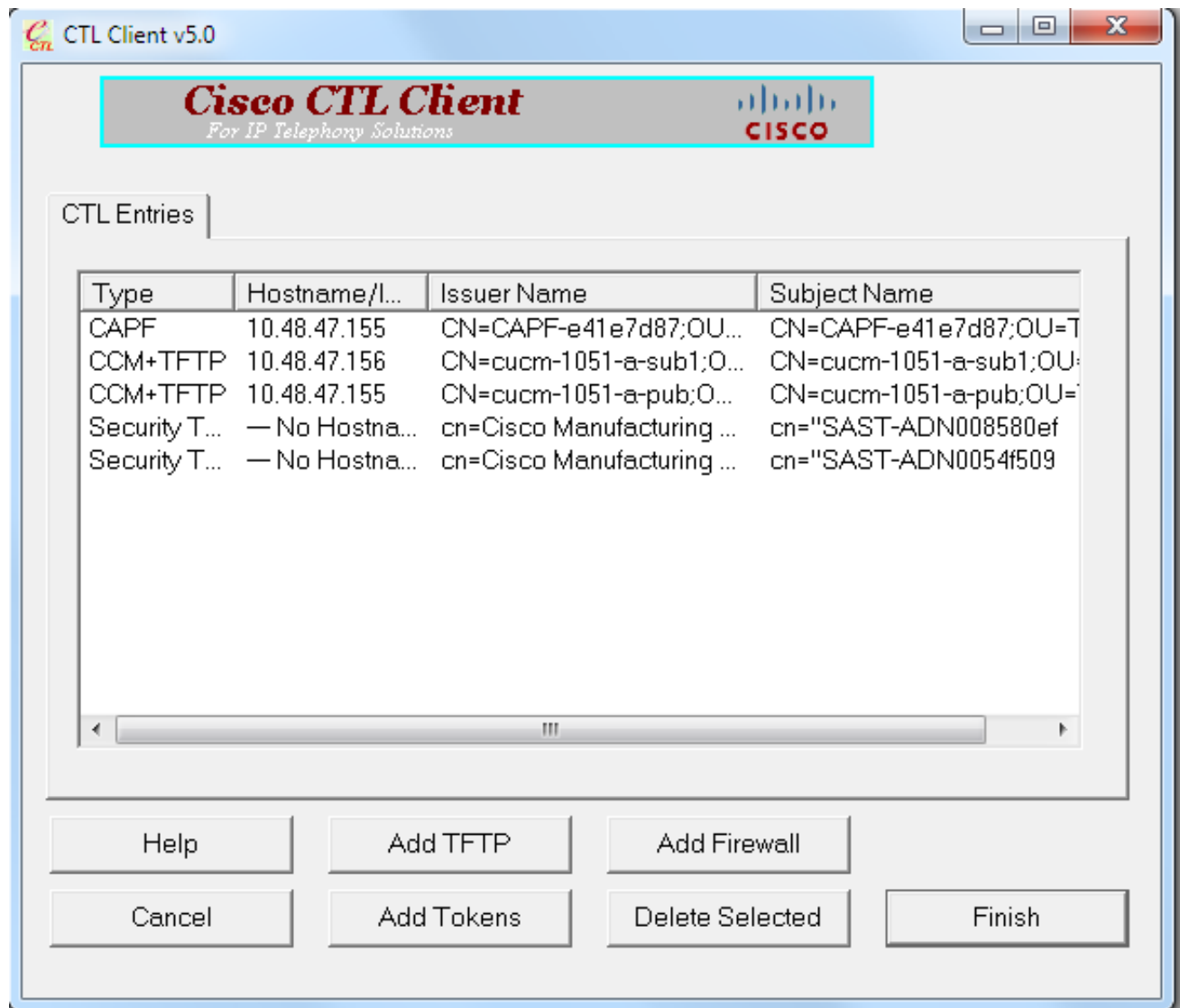
16. Zodra de inhoud van het CTL-bestand is weergegeven, klikt u op **Add Tokens** om de tweede USB Token toe te voegen:



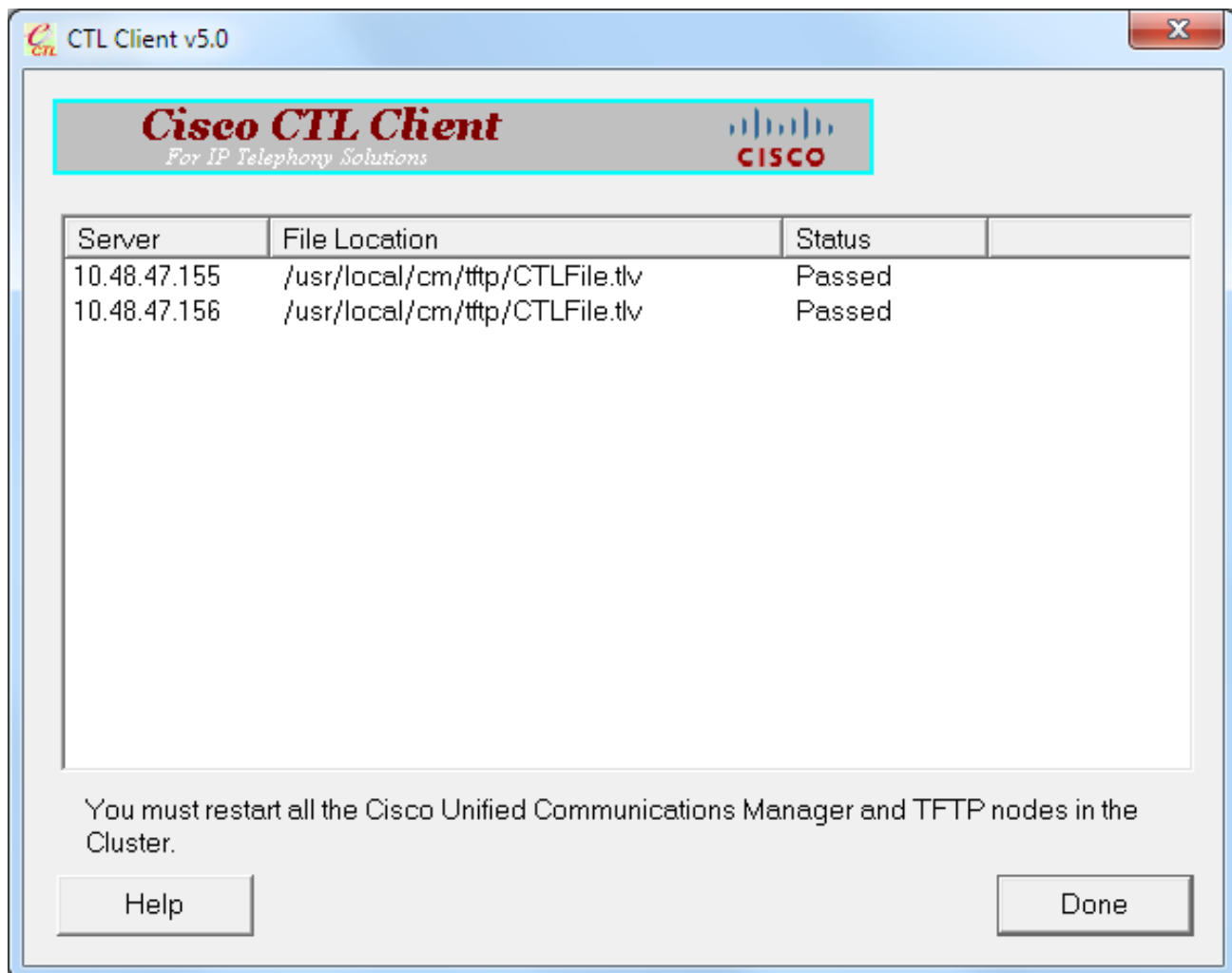
17. Klik na het verschijnen van de veiligheidstoken op **Toevoegen**:



18. Klik na de inhoud van het CTL-bestand op **Voltooien**. Voer na het oproepen van een wachtwoord **Cisco123** in:



19. Wanneer de lijst van CUCM-servers waarop het CTL-bestand bestaat, verschijnt, klikt u op **Gereedschap**:



20. Start de TFTP- en CallManager-services opnieuw op alle knooppunten in de cluster die deze services uitvoeren.
21. Start alle IP-telefoons opnieuw zodat ze de nieuwe versie van het CTL-bestand kunnen verkrijgen van de CUCM TFTP-service.
22. Om de inhoud van het CTL-bestand te verifiëren, voer de **show ctl**-opdracht in in de CLI. In het CTL-bestand kunt u de certificaten van beide USB-penningen zien (één ervan wordt gebruikt om het CTL-bestand te tekenen). Hier wordt een voorbeelduitvoer weergegeven:

```

admin:show ctl
The checksum value of the CTL file:
2e7a6113eadbdae67ffa918d81376902 (MD5)
d0f3511f10eef775cc91cce3fa6840c2640f11b8 (SHA1)

Length of CTL file: 5728
The CTL File was last modified on Fri Mar 06 22:53:33 CET 2015

[...]

CTL Record #:1
----
BYTEPOS TAG LENGTH VALUE
-----
1 RECORDLENGTH 2 1186
2 DNSNAME 1

```

```

3 SUBJECTNAME 56 cn="SAST-ADN0054f509 ";ou=IPCBU;o="Cisco Systems
4 FUNCTION 2 System Administrator Security Token
5 ISSUERNAM 42 cn=Cisco Manufacturing CA;o=Cisco Systems
6 SERIALNUMBER 10 3C:F9:27:00:00:00:AF:A2:DA:45
7 PUBLICKEY 140
9 CERTIFICATE 902 19 8F 07 C4 99 20 13 51 C5 AE BF 95 03 93 9F F2
CC 6D 93 90 (SHA1 Hash HEX)
10 IPADDRESS 4
This etoken was not used to sign the CTL file.

```

[...]

```

CTL Record #:5
----
BYTEPOS TAG LENGTH VALUE
-----
1 RECORDLENGTH 2 1186
2 DNSNAME 1
3 SUBJECTNAME 56 cn="SAST-ADN008580ef ";ou=IPCBU;o="Cisco Systems
4 FUNCTION 2 System Administrator Security Token
5 ISSUERNAM 42 cn=Cisco Manufacturing CA;o=Cisco Systems
6 SERIALNUMBER 10 83:E9:08:00:00:00:55:45:AF:31
7 PUBLICKEY 140
9 CERTIFICATE 902 85 CD 5D AD EA FC 34 B8 3E 2F F2 CB 9C 76 B0 93
3E 8B 3A 4F (SHA1 Hash HEX)
10 IPADDRESS 4
This etoken was used to sign the CTL file.

```

The CTL file was verified successfully.

23. Aan de kant van de IP-telefoon kunt u controleren dat nadat de IP-telefoons opnieuw zijn opgestart, ze de bijgewerkte versie van het CTL-bestand hebben gedownload (de MD5-checksum komt overeen met de uitvoer van CUCM):



Deze wijziging is mogelijk omdat u de Token-certificaten eerder naar de CUCM Certificate Trust Store hebt geëxporteerd en geüpload, en de IP-telefoons zijn in staat om dit onbekende certificaat te controleren dat is gebruikt om het CTL-bestand te tekenen tegen de TVS-versie (Trust Verification Service) die op het CUCM actief is. Dit logfragment illustreert hoe de IP-telefoon contact opneemt met de CUCM TVS met een verzoek om het onbekende eToken-certificaat te controleren, dat als **Phone-SAST-trust** wordt geüpload en op uw computer wordt vertrouwd:

**//In the Phone Console Logs we can see a request sent to TVS server to verify unknown certificate**

```
8074: NOT 23:00:22.335499 SECD: setupSocketToTvsProxy: Connected to TVS proxy server
8075: NOT 23:00:22.336918 SECD: tvsReqFlushTvsCertCache: Sent Request to TVS proxy,
len: 3708
```

**//In the TVS logs on CUCM we can see the request coming from an IP Phone which is being successfully verified**

```
23:00:22.052 | debug tvsHandleQueryCertReq
23:00:22.052 | debug tvsHandleQueryCertReq : Subject Name is: cn="SAST-ADN008580ef
";ou=IPCBU;o="Cisco Systems
23:00:22.052 | debug tvsHandleQueryCertReq : Issuer Name is: cn=Cisco Manufacturing
CA;o=Cisco Systems
23:00:22.052 | debug tvsHandleQueryCertReq :subjectName and issuerName matches for
eToken certificate
23:00:22.052 | debug tvsHandleQueryCertReq : SAST Issuer Name is: cn=Cisco
Manufacturing CA;o=Cisco Systems
23:00:22.052 | debug tvsHandleQueryCertReq : This is SAST eToken cert
23:00:22.052 | debug tvsHandleQueryCertReq : Serial Number is: 83E9080000005545AF31
23:00:22.052 | debug CertificateDBCACHE::getCertificateInformation - Looking up the
certificate cache using Unique MAP ID : 83E9080000005545AF31cn=Cisco Manufacturing
CA;o=Cisco Systems
23:00:22.052 | debug ERROR:CertificateDBCACHE::getCertificateInformation - Cannot find
the certificate in the cache
23:00:22.052 | debug CertificateCTLCache::getCertificateInformation - Looking up the
certificate cache using Unique MAP ID : 83E9080000005545AF31cn=Cisco Manufacturing
CA;o=Cisco Systems, len : 61
23:00:22.052 | debug CertificateCTLCache::getCertificateInformation - Found entry
{rolecount : 1}
23:00:22.052 | debug CertificateCTLCache::getCertificateInformation - {role : 0}
23:00:22.052 | debug convertX509ToDER -x509cert : 0xa3ea6f8
23:00:22.053 | debug tvsHandleQueryCertReq: Timer started from tvsHandleNewPhConnection
```

**//In the Phone Console Logs we can see reply from TVS server to trust the new certificate (eToken Certificate which was used to sign the CTL file)**

```
8089: NOT 23:00:22.601218 SECD: clpTvsInit: Client message received on TVS proxy socket
8090: NOT 23:00:22.602785 SECD: processTvsClntReq: Success reading the client TVS
request, len : 3708
8091: NOT 23:00:22.603901 SECD: processTvsClntReq: TVS Certificate cache flush
request received
8092: NOT 23:00:22.605720 SECD: tvsFlushCertCache: Completed TVS Certificate cache
flush request
```

## **certificaatregeneratie voor kenloze CTL-oplossing**

In dit deel wordt beschreven hoe een CUCM-certificaat voor clusterbeveiliging moet worden regenereren wanneer de Tokenless CTL-oplossing wordt gebruikt.

Tijdens het proces van het onderhoud van CUCM, soms verandert het certificaat van de Uitgeverij CallManager van CUCM. De scenario's waarin dit kan gebeuren zijn de verandering van hostname, de verandering van domein, of simpelweg een regeneratie van certificaten (vanwege beëindigen van de verloopdatum van certificaat).

Nadat het CTL-bestand is bijgewerkt, wordt het ondertekend met een ander certificaat dan dat in het CTL-bestand bestaat dat op IP-telefoons is geïnstalleerd. Normaal gesproken wordt dit nieuwe CTL-bestand niet geaccepteerd; echter nadat de IP-telefoon het onbekende certificaat vindt dat wordt gebruikt om het CTL-bestand te ondertekenen, neemt het contact op met de TVS-service op CUCM.

Opmerking: De lijst met TVS-servers is in het configuratiebestand voor IP-telefoon en is in kaart gebracht in de CUCM-servers van het **IP-telefoonapparaat > CallManager-groep**.

Na succesvolle verificatie met de TVS-server werkt de IP-telefoon zijn CTL-bestand met de nieuwe versie bij. Deze gebeurtenissen doen zich voor in een dergelijk scenario:

1. Het CTL-bestand bestaat op CUCM en op de IP-telefoon. Het CCM+TFT (server) certificaat voor het CUCM Publisher-knooppunt wordt gebruikt om het CTL-bestand te tekenen:

```
admin:show ctl
The checksum value of the CTL file:
7b7c10c4a7fa6de651d9b694b74db25f (MD5)
819841c6e767a59ecf2f87649064d8e073b0fe87 (SHA1)

Length of CTL file: 4947
The CTL File was last modified on Mon Mar 09 16:59:43 CET 2015
```

[...]

```
CTL Record #:1
----
BYTEPOS TAG LENGTH VALUE
-----
1 RECORDLENGTH 2 1156
2 DNSNAME 16 cucm-1051-a-pub
3 SUBJECTNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
4 FUNCTION 2 System Administrator Security Token
5 ISSUERNAM 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
6 SERIALNUMBER 16 70:CA:F6:4E:09:07:51:B9:DF:22:F4:9F:75:4F:C5:BB
7 PUBLICKEY 140
8 SIGNATURE 128
9 CERTIFICATE 694 E9 D4 33 64 5B C8 8C ED 51 4D 8F E5 EA 5B 6D
21 A5 A3 8C 9C (SHA1 Hash HEX)
10 IPADDRESS 4
This etoken was used to sign the CTL file.
```

```
CTL Record #:2
----
BYTEPOS TAG LENGTH VALUE
-----
1 RECORDLENGTH 2 1156
2 DNSNAME 16 cucm-1051-a-pub
3 SUBJECTNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
4 FUNCTION 2 CCM+TFTP
5 ISSUERNAM 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
6 SERIALNUMBER 16 70:CA:F6:4E:09:07:51:B9:DF:22:F4:9F:75:4F:C5:BB
7 PUBLICKEY 140
8 SIGNATURE 128
9 CERTIFICATE 694 E9 D4 33 64 5B C8 8C ED 51 4D 8F E5 EA 5B 6D
21 A5 A3 8C 9C (SHA1 Hash HEX)
10 IPADDRESS 4
```

[...]

The CTL file was verified successfully.



## Certificate Details for cucm-1051-a-pub, CallManager



Regenerate



Generate CSR



Download .PEM File



Download .DER File

### Status



Status: Ready

### Certificate Settings





File Name	CallManager.pem
Certificate Purpose	CallManager
Certificate Type	certs
Certificate Group	product-cm
Description(friendly name)	Self-signed certificate generated by system

### Certificate File Data

```
[
  Version: V3
  Serial Number: 70CAF64E090751B9DF22F49F754FC5BB
  SignatureAlgorithm: SHA1withRSA (1.2.840.113549.1.1.5)
  Issuer Name: L=Krakow, ST=Malopolska, CN=cucm-1051-a-pub, OU=TAC, O=Cisco, C=PL
  Validity From: Thu Jun 05 18:31:39 CEST 2014
  To: Tue Jun 04 18:31:38 CEST 2019
  Subject Name: L=Krakow, ST=Malopolska, CN=cucm-1051-a-pub, OU=TAC, O=Cisco, C=PL
  Key: RSA (1.2.840.113549.1.1.1)
  Key value:
  30818902818100950c9f8791e7677c5bf1a48f1a933549f73ef58d7c0c871b5b77d23a842aa14f5b293
  90e586e5945060b109bdf859b4c983cdf21699e3e4abdb0a47ba6f3c04cd7d4f59efeff4a60f6cf3c5db
  2ec32988605ae4352e77d647da25fae619dedf9ebb0e0bdd98f8ce70307ba106507a8919df8b8fd9f9
  03068a52640a6a84487a90203010001
  Extensions: 3 present
```


2. Het **CallManager.pem**-bestand (CCM+TFTP-certificaat) wordt opnieuw gegenereerd en u kunt zien dat het serienummer van het certificaat verandert:

### Certificate Details for cucm-1051-a-pub, CallManager

 Regenerate
  Generate CSR
  Download .PEM File
  Download .DER File

---

**Status**

 Status: Ready

---

**Certificate Settings**

File Name	CallManager.pem
Certificate Purpose	CallManager
Certificate Type	certs
Certificate Group	product-cm
Description(friendly name)	Self-signed certificate generated by system

---

**Certificate File Data**

```
[
Version: V3
Serial Number: 6B1D357B6841740B078FEE4A1813D5D6
SignatureAlgorithm: SHA256withRSA (1.2.840.113549.1.1.11)
Issuer Name: L=Krakow, ST=Malopolska, CN=cucm-1051-a-pub, OU=TAC, O=Cisco, C=PL
Validity From: Mon Mar 09 17:06:37 CET 2015
To: Sat Mar 07 17:06:36 CET 2020
Subject Name: L=Krakow, ST=Malopolska, CN=cucm-1051-a-pub, OU=TAC, O=Cisco, C=PL
Key: RSA (1.2.840.113549.1.1.1)
Key value:
3082010a0282010100c363617e37830eaf5312f4eb3fe68c74e7a037453d26a0514e52476e56d02f78
c19e83623952934279b8dee9b3944a2a43c21714502db749c4141edc4666358974f2248e001e58928
8a608e9a1bc8ef74267e413e03d5d53e61f0705fb564a1dd2744a53840f579a183cd29e9b3e0d5d689
e067b6426c8c8c49078c5c4cc1b6cb6fec83d31ee86661517bf560ef0c01f5ec056db0dcc9746402af2a
b3ed4d66521f6d0b795ac48f78deaafb324dc30962ffa9e96c8615cce6e1a68247f217c83bf324fb3d5c
]
```

3. Het opdracht om CTLFile bij te werken is in de CLI ingevoerd om het CTL-bestand bij te werken:

```
admin:utils ctl update CTLFile
This operation will update the CTLFile. Do you want to continue? (y/n):y

Updating CTL file
CTL file Updated
Please Restart the TFTP and Cisco CallManager services on all nodes in
the cluster that run these services
admin:
```

4. De TVS service werkt zijn certificaatcache bij met de nieuwe CTL-bestandsdetails:

```
17:10:35.825 | debug CertificateCache::localCTLCacheMonitor - CTLFile.tlv has been modified. Recaching CTL Certificate Cache
17:10:35.826 | debug updateLocalCTLCache : Refreshing the local CTL certificate cache
17:10:35.827 | debug tvs_sql_get_all_CTL_certificate - Unique Key used for Caching ::
6B1D357B6841740B078FEE4A1813D5D6CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL, length : 93
17:10:35.827 | debug tvs_sql_get_all_CTL_certificate - Unique Key used for Caching ::
6B1D357B6841740B078FEE4A1813D5D6CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL, length : 93
17:10:35.827 | debug tvs_sql_get_all_CTL_certificate - Unique Key used for Caching ::
```

```
744B5199770516E799E91E81D3C8109BCN=CAPF-e41e7d87;OU=TAC;O=Cisco;L=Krakow;  
ST=Malopolska;C=PL, length : 91  
17:10:35.827 | debug tvs_sql_get_all_CTL_certificate - Unique Key used for Caching ::  
6BEBFDCDCD8CA277CB2FD1D183A60E72CN=cucm-1051-a-sub1;OU=TAC;O=Cisco;L=Krakow;  
ST=Malopolska;C=PL, length : 94
```

## 5. Wanneer u de inhoud van het CTL-bestand bekijkt, kunt u zien dat het bestand getekend is met het nieuwe CallManager-servercertificaat voor het knooppunt Uitgever:

```
admin:show ctl  
The checksum value of the CTL file:  
ebc649598280a4477bb3e453345c8c9d(MD5)  
ef5c006b6182cad66197fac6e6530f15d009319d(SHA1)  
  
Length of CTL file: 6113  
The CTL File was last modified on Mon Mar 09 17:07:52 CET 2015
```

[..]

```
CTL Record #:1  
----  
BYTEPOS TAG LENGTH VALUE  
-----  
1 RECORDLENGTH 2 1675  
2 DNSNAME 16 cucm-1051-a-pub  
3 SUBJECTNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;  
ST=Malopolska;C=PL  
4 FUNCTION 2 System Administrator Security Token  
5 ISSUERNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;  
ST=Malopolska;C=PL  
6 SERIALNUMBER 16 6B:1D:35:7B:68:41:74:0B:07:8F:EE:4A:18:13:D5:D6  
7 PUBLICKEY 270  
8 SIGNATURE 256  
9 CERTIFICATE 955 5C AF 7D 23 FE 82 DB 87 2B 6F 4D B7 F0 9D D5  
86 EE E0 8B FC (SHA1 Hash HEX)  
10 IPADDRESS 4
```

**This etoken was used to sign the CTL file.**

```
CTL Record #:2  
----  
BYTEPOS TAG LENGTH VALUE  
-----  
1 RECORDLENGTH 2 1675  
2 DNSNAME 16 cucm-1051-a-pub  
3 SUBJECTNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;  
ST=Malopolska;C=PL  
4 FUNCTION 2 CCM+TFTP  
5 ISSUERNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;  
ST=Malopolska;C=PL  
6 SERIALNUMBER 16 6B:1D:35:7B:68:41:74:0B:07:8F:EE:4A:18:13:D5:D6  
7 PUBLICKEY 270  
8 SIGNATURE 256  
9 CERTIFICATE 955 5C AF 7D 23 FE 82 DB 87 2B 6F 4D B7 F0 9D D5  
86 EE E0 8B FC (SHA1 Hash HEX)  
10 IPADDRESS 4
```

[...]

The CTL file was verified successfully.

## 6. Van de pagina Unified Service wordt de TFTP- en Cisco CallManager-services opnieuw gestart op alle knooppunten in de cluster die deze services uitvoeren.

7. De IP-telefoons worden opnieuw opgestart en ze nemen contact op met de TVS-server om het onbekende certificaat te controleren dat nu wordt gebruikt om de nieuwe versie van het CTL-bestand te tekenen:

```
// In the Phone Console Logs we can see a request sent to TVS server to verify
unknown certificate
2782: NOT 17:21:51.794615 SECD: setupSocketToTvsProxy: Connected to TVS proxy server
2783: NOT 17:21:51.796021 SECD: tvsReqFlushTvsCertCache: Sent Request to TVS
proxy, len: 3708
```

```
// In the TVS logs on CUCM we can see the request coming from an IP Phone which is
being successfully verified
17:21:51.831 | debug tvsHandleQueryCertReq
17:21:51.832 | debug tvsHandleQueryCertReq : Subject Name is: CN=cucm-1051-a-pub;
OU=TAC;O=Cisco;L=Krakow;ST=Malopolska
17:21:51.832 | debug tvsHandleQueryCertReq : Issuer Name is: CN=cucm-1051-a-pub;
OU=TAC;O=Cisco;L=Krakow;ST=Malopolska;
17:21:51.832 | debug tvsHandleQueryCertReq : Serial Number is:
6B1D357B6841740B078FEE4A1813D5D6
17:21:51.832 | debug CertificateDBCACHE::getCertificateInformation - Looking up the
certificate cache using Unique MAPco;L=Krakow;ST=Malopolska;C=PL
17:21:51.832 | debug CertificateDBCACHE::getCertificateInformation - Found entry
{rolecount : 2}
17:21:51.832 | debug CertificateDBCACHE::getCertificateInformation - {role : 0}
17:21:51.832 | debug CertificateDBCACHE::getCertificateInformation - {role : 2}
17:21:51.832 | debug convertX509ToDER -x509cert : 0xf6099df8
17:21:51.832 | debug tvsHandleQueryCertReq: Timer started from
tvsHandleNewPhConnection
```

```
// In the Phone Console Logs we can see reply from TVS server to trust the new
certificate (new CCM Server Certificate which was used to sign the CTL file)
2797: NOT 17:21:52.057442 SECD: clpTvsInit: Client message received on TVS
proxy socket
2798: NOT 17:21:52.058874 SECD: processTvsClntReq: Success reading the client TVS
request, len : 3708
2799: NOT 17:21:52.059987 SECD: processTvsClntReq: TVS Certificate cache flush
request received
2800: NOT 17:21:52.062873 SECD: tvsFlushCertCache: Completed TVS Certificate
cache flush request
```

8. Tot slot kunt u op de IP-telefoons controleren dat het CTL-bestand met de nieuwe versie is bijgewerkt en dat de MD5-checksum van het nieuwe CTL-bestand overeenkomt met dat van CUCM:

