Mode mixte CUCM avec Tokenless CTL

Contenu

Introduction

Ce document décrit la différence entre la Sécurité de Cisco Unified Communications Manager (CUCM) avec et sans l'utilisation des eTokens du matériel USB. Ce document décrit également les scénarios de base d'implémentation qui comportent la liste de confiance de certificat de Tokenless (CTL) et le processus qui est utilisé afin de s'assurer que les fonctions système correctement après les modifications.

Conditions préalables

Conditions requises

Cisco recommande que vous ayez la connaissance de la version 10.0(1) ou ultérieures CUCM. Supplémentaire, assurez cela :

- Vous avez l'accès administratif à l'interface de ligne de commande (CLI) du noeud CUCM Publisher.
- Vous avez accès aux eTokens du matériel USB et cela le module d'extension de client CTL est installé sur votre PC pour les scénarios qui exigent de vous de migrer de nouveau à l'utilisation des eTokens de matériel.
- Il y a de connectivité complète entre tous les Noeuds CUCM dans la batterie. C'est très important parce que le fichier CTL est copié sur tous les Noeuds dans la batterie par l'intermédiaire du protocole de transfert de fichiers de SSH (SFTP).

- La réplication de base de données (DB) dans la batterie fonctionne correctement et cela les serveurs répliquent les données en temps réel.

- Les périphériques dans votre déploiement prennent en charge la Sécurité par défaut (TV). Vous pouvez utiliser la *liste de caractéristique de téléphone d'Unified CM de* Cisco Unified signalant la page Web (IP ou FQDN>/cucreports/de https:// <CUCM) afin de déterminer les périphériques qui prennent en charge la Sécurité par défaut. Remarque: Le Cisco Jabber et beaucoup des Téléphones IP de TelePresence Cisco ou de gamme Cisco 7940/7960 ne prennent en charge pas actuellement la Sécurité par défaut. Si vous déployez Tokenless CTL avec les périphériques qui ne prennent en charge pas la Sécurité par défaut, n'importe quelle mise à jour à votre système qui change le certificat de CallManager sur l'éditeur empêchera ces périphériques de s'inscrire au système jusqu'à ce que leur CTL soit manuellement supprimé.

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Version 10.5.1.10000-7 (batterie CUCM de deux Noeuds)

- Les Téléphones IP de gamme Cisco 7975 se sont enregistrés par l'intermédiaire du Skinny Client Control Protocol (SCCP) avec la version SCCP75.9-3-1SR4-1S de micrologiciels

- Deux jetons de sécurité Cisco qui sont utilisés afin de placer la batterie au mode mixte avec l'utilisation du logiciel client CTL

Informations générales

Tokenless CTL est une nouvelle caractéristique dans des versions 10.0(1) et ultérieures CUCM qui permet le chiffrement de la signalisation et des medias d'appel pour des Téléphones IP sans nécessité d'utiliser des eTokens USB de matériel et le module d'extension de client CTL, qui était la condition requise dans les versions précédentes CUCM.

Quand la batterie est placée dans le mode mixte avec l'utilisation de la commande CLI, le fichier CTL est signé avec le certificat CCM+TFTP (serveur) du noeud de Publisher, et il y a aucun eToken des Certificats actuels dans le fichier CTL.

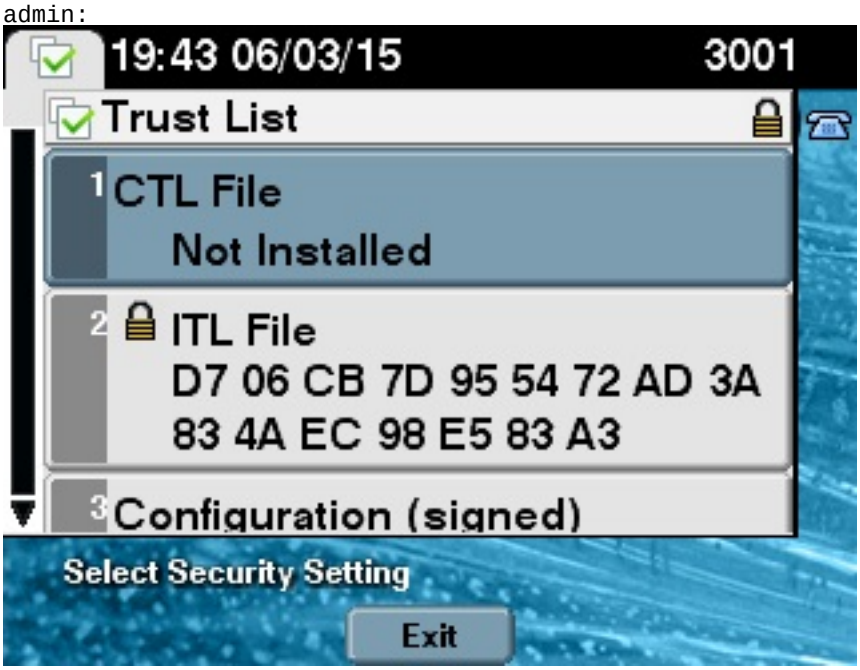
> Remarque: Quand vous régénérez le certificat du CallManager (CCM+TFTP) sur l'éditeur, il change le signataire du fichier. Les téléphones et les périphériques qui ne prennent en charge pas la Sécurité par défaut ne recevront pas le nouveau fichier CTL à moins que des fichiers CTL **soient manuellement supprimés de chaque périphérique**. Référez-vous à la dernière condition qui soit répertorié la section de conditions requises de ce pour en savoir plus de document.

Du mode Non-sécurisé au mode mixte (Tokenless CTL)

Cette section décrit le processus qui est utilisé afin d'entrer la Sécurité de batterie CUCM dans le mode mixte par l'intermédiaire du CLI.

Avant ce scénario, le CUCM était en mode Non-sécurisé, ainsi il signifie qu'il n'y avait aucun fichier CTL actuel sur les Noeuds l'uns des et que les Téléphones IP enregistrés ont eu seulement un fichier de la liste de confiance d'identité (ITL) installé, suivant les indications de ces sorties :

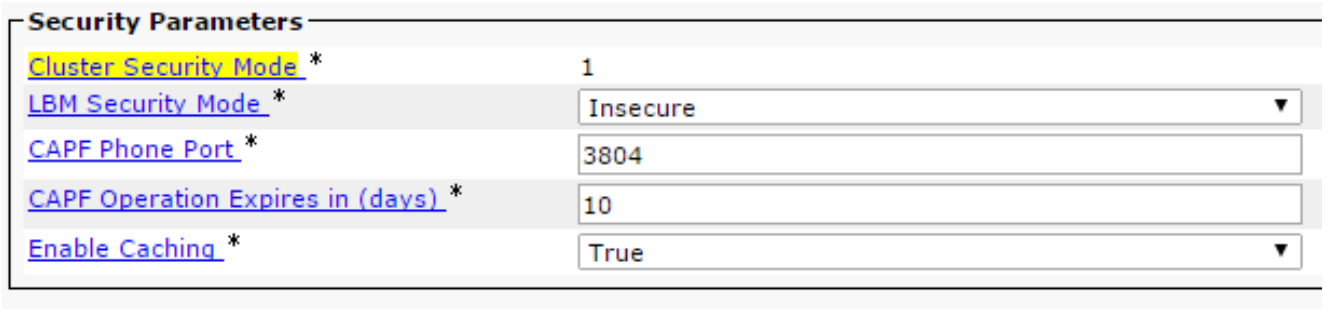
```
admin:show ctl
Length of CTL file: 0
CTL File not found. Please run CTLClient plugin or run the CLI - utils ctl.. to
generate the CTL file.
Error parsing the CTL File.
admin:
```



Afin d'entrer la Sécurité de batterie CUCM dans le mode mixte avec l'utilisation de la nouvelle caractéristique de Tokenless CTL, terminez-vous ces étapes :

1. Obtenez l'accès administratif au noeud CLI CUCM Publisher.

2. Sélectionnez la commande de **mode mixte de positionnement-batterie de ctl d'utils** dans le 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. Naviguez vers la **page > le System > Enterprise Parameters d'admin CUCM** et vérifiez si la batterie a été placée au mode mixte (une valeur de **1** indique le mode mixte) :



4. Redémarrez le TFTP et les services de Cisco CallManager sur tous les Noeuds dans la batterie qui dirigent ces services.

5. Redémarrez tous les Téléphones IP de sorte qu'ils puissent obtenir le fichier CTL du service TFTP CUCM.

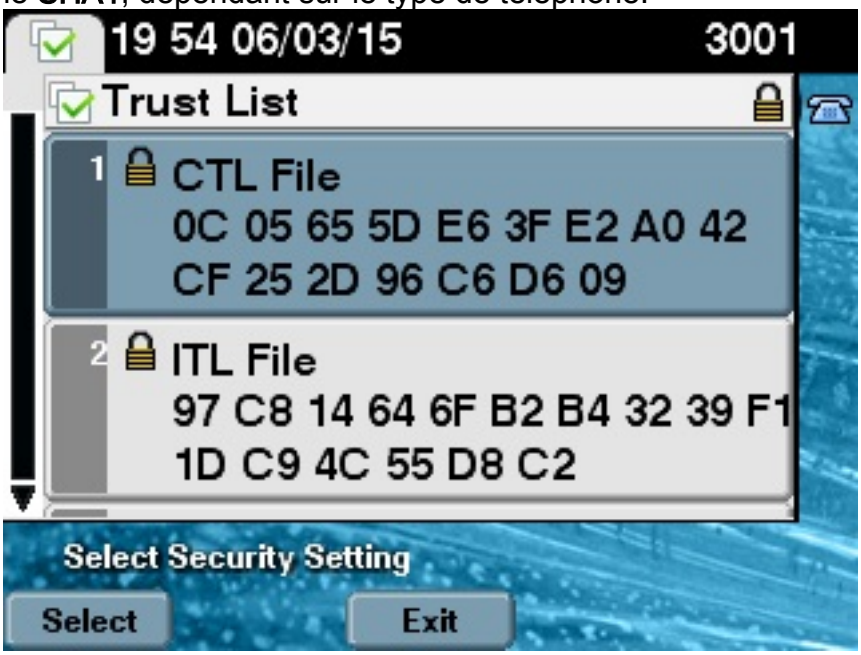
6. Afin de vérifier le contenu du fichier CTL, sélectionnez la commande de **ctl d'exposition** dans le CLI. Dans le fichier CTL vous pouvez voir que le certificat CCM+TFTP (serveur) pour le noeud CUCM Publisher est utilisé afin de signer le fichier CTL (ce fichier est identique sur tous les serveurs dans la batterie). Voici un exemple de sortie :

```
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. Du côté de téléphone IP, vous pouvez vérifier cela après que le service soit redémarré, il télécharge le fichier CTL, qui est maintenant présent sur le serveur TFTP (la somme de contrôle de MD5 s'assortit une fois comparé à la sortie du CUCM) :

   Remarque: Quand vous vérifiez la somme de contrôle au téléphone, vous voyez le **MD5** ou le **SHA1**, dépendant sur le type de téléphone.



Des eTokens de matériel à la solution de Tokenless

Cette section décrit comment migrer la Sécurité de batterie CUCM des eTokens de matériel vers l'utilisation de la nouvelle solution de Tokenless.

Dans certaines situations, le mode mixte est déjà configuré sur le CUCM avec l'utilisation du client CTL, et les fichiers CTL d'utilisation de Téléphones IP qui contiennent les Certificats des eTokens du matériel USB. Avec ce scénario, le fichier CTL est signé par un certificat d'un des eTokens USB et est installé sur les Téléphones IP. Ici dans un exemple :

```
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.
```

Terminez-vous ces étapes afin de déplacer la Sécurité de batterie CUCM à l'utilisation de Tokenless CTLs :

1. Obtenez l'accès administratif au noeud CLI CUCM Publisher.

2. Sélectionnez la commande de **CTLFile** CLI de **mise à jour de ctl d'utils** :
   ```
   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. Redémarrez le TFTP et les services de CallManager sur tous les Noeuds dans la batterie qui dirigent ces services.

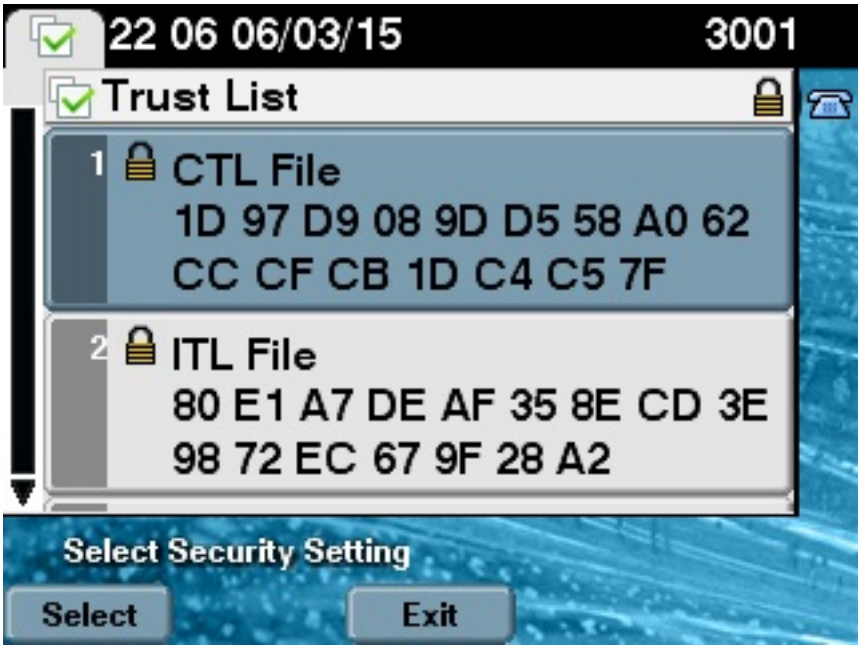
4. Redémarrez tous les Téléphones IP de sorte qu'ils puissent obtenir le fichier CTL du service TFTP CUCM.

5. Sélectionnez la commande de **ctl d'exposition** dans le CLI afin de vérifier le contenu du fichier CTL. Dans le fichier CTL, vous pouvez voir que le certificat CCM+TFTP (serveur) du noeud CUCM Publisher est utilisé afin de signer le fichier CTL au lieu du certificat des eTokens du matériel USB. Une différence plus importante est dans ce cas que les Certificats

de tous les eTokens du matériel USB sont retirés à partir du fichier CTL. Voici un exemple de sortie :

```
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.
```

6. Du côté de téléphone IP, vous pouvez vérifier cela après que les Téléphones IP aient été redémarrés, ils avez téléchargé la version mise à jour de fichier CTL (la somme de contrôle de MD5 s'assortit une fois comparé à la sortie du CUCM) :



De la solution de Tokenless aux eTokens de matériel

Cette section décrit comment migrer la Sécurité de batterie CUCM à partir de la nouvelle solution de Tokenless et de nouveau à l'utilisation des eTokens de matériel.

Quand la Sécurité de batterie CUCM est placée au mode mixte avec l'utilisation des commandes CLI, et le fichier CTL est signé avec le certificat CCM+TFTP (serveur) pour le noeud CUCM Publisher, il n'y a aucun Certificats des eTokens du matériel USB actuels dans le fichier CTL. Pour cette raison, quand vous exécutez le client CTL afin de mettre le fichier CTL à jour (mouvement de nouveau à l'utilisation des eTokens de matériel), ce message d'erreur apparaît :

```
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.
```

C'est particulièrement important dans les scénarios qui incluent un downgrade (quand la version est commutée de retour) du système à une version pre-10.x qui n'inclut pas les commandes de **ctl d'utils**. Le fichier CTL précédent est migré (sans changements de son contenu) en cours de régénération ou Linux vers la mise à jour du Linux (L2), et elle ne contient pas les Certificats d'eToken, comme précédemment mentionné. Voici un exemple de sortie :

```
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

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 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

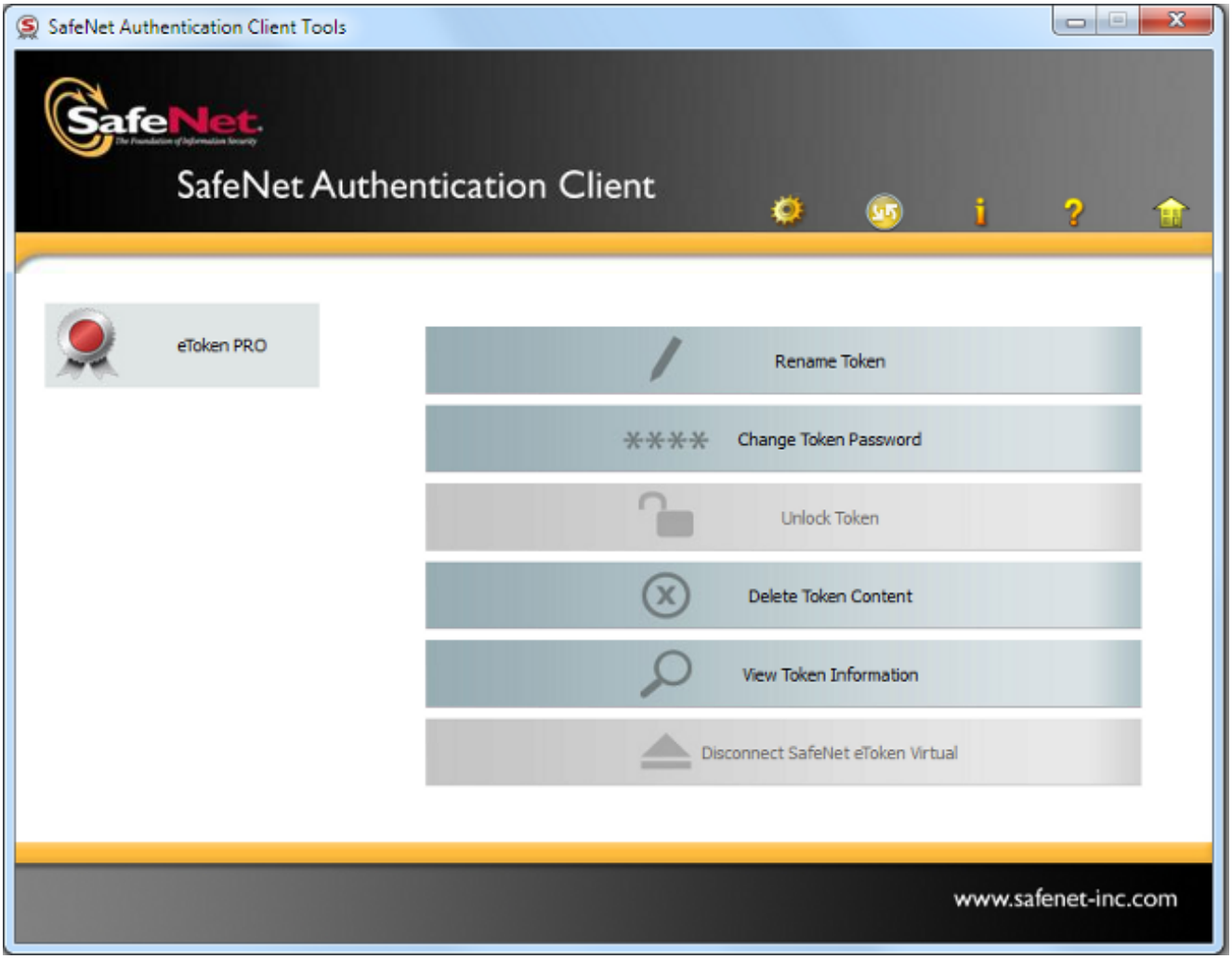
BYTEPOS TAG LENGTH VALUE
------- --- ------ -----
1 RECORDLENGTH 2 1156
2 DNSNAME 16 cucm-1051-a-pub
3 SUBJECTNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
4 FUNCTION 2 **CCM+TFTP**
5 ISSUERNAME 62 CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
ST=Malopolska;C=PL
6 SERIALNUMBER 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

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:
```

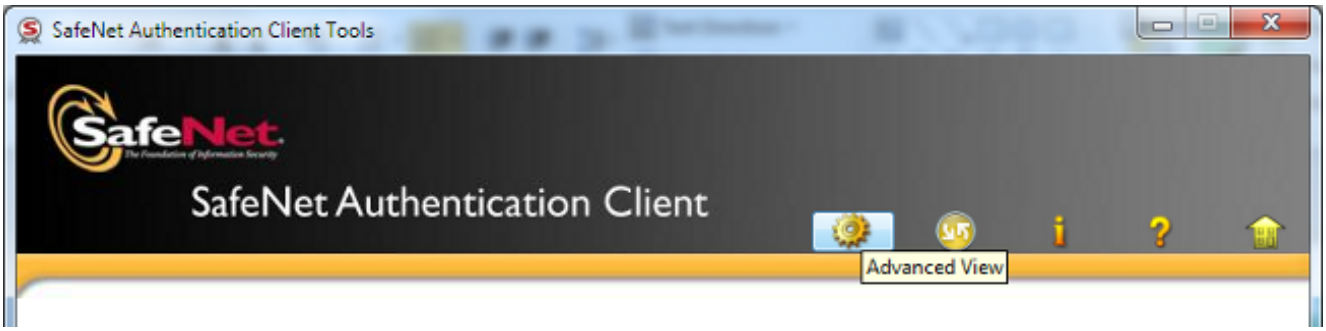
Pour ce scénario, terminez-vous ces étapes afin de mettre sécurisé les fichiers CTL à jour sans nécessité d'utiliser la procédure pour les eTokens perdus, qui finit par dans la suppression manuelle du fichier CTL de tous les Téléphones IP :

1. Obtenez l'accès administratif au noeud CLI CUCM Publisher.

2. Sélectionnez la **commande du tftp CTLFile.tlv d'effacement de fichier** dans le noeud CLI de Publisher afin de supprimer le fichier CTL :
   ```
   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. **L'authentication client** ouverte de **SafeNet** sur l'ordinateur de Microsoft Windows qui a le client CTL installé (il est installée automatiquement avec le client CTL) :

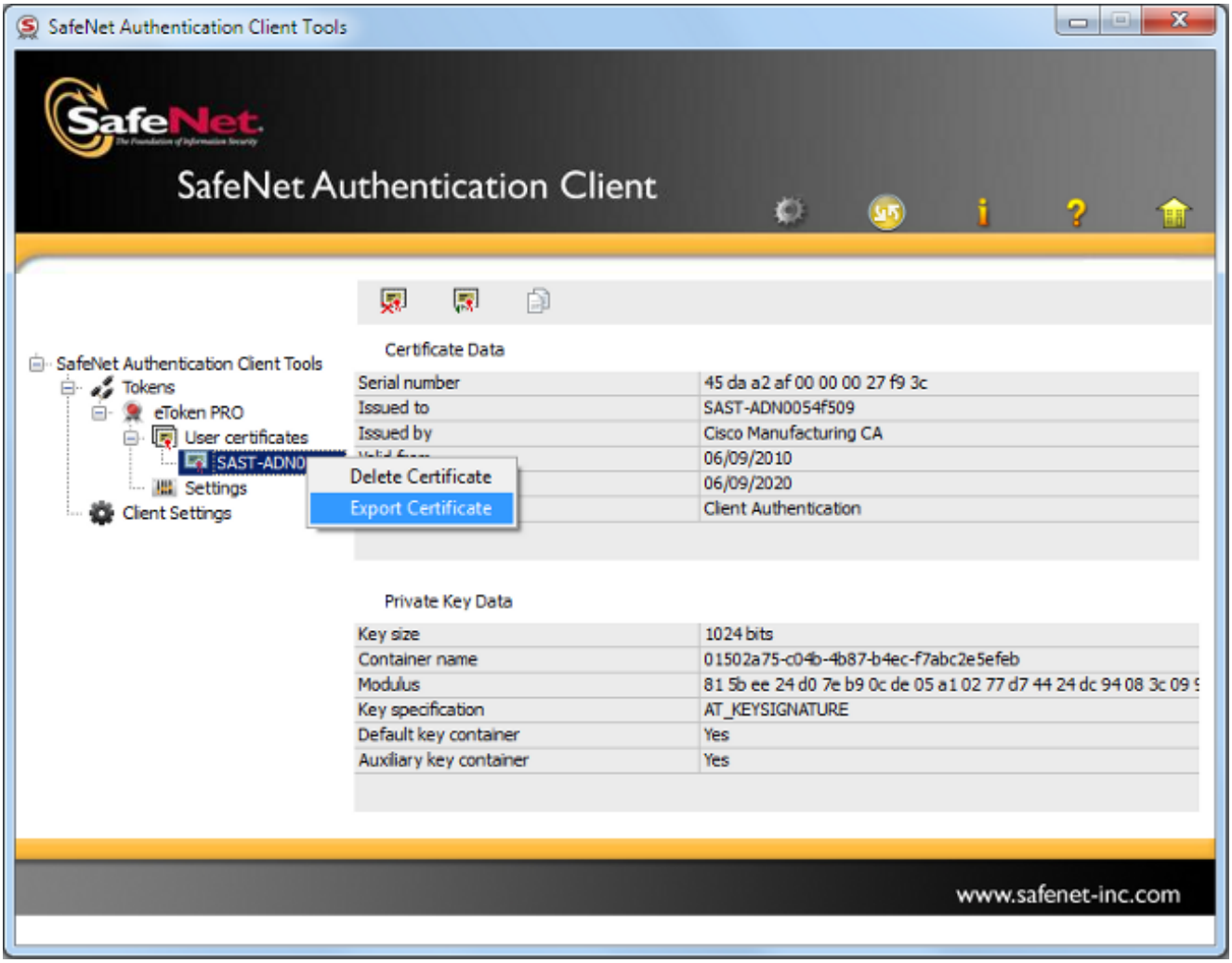


4. Dans l'authentication client de SafeNet, naviguez vers la *vue avancée* :

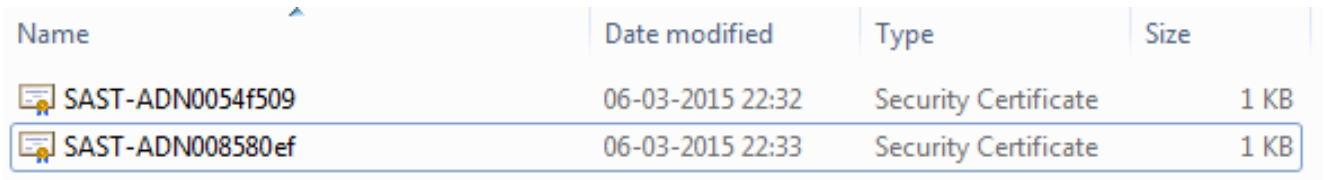


5. Insérez le premier matériel USB eToken.

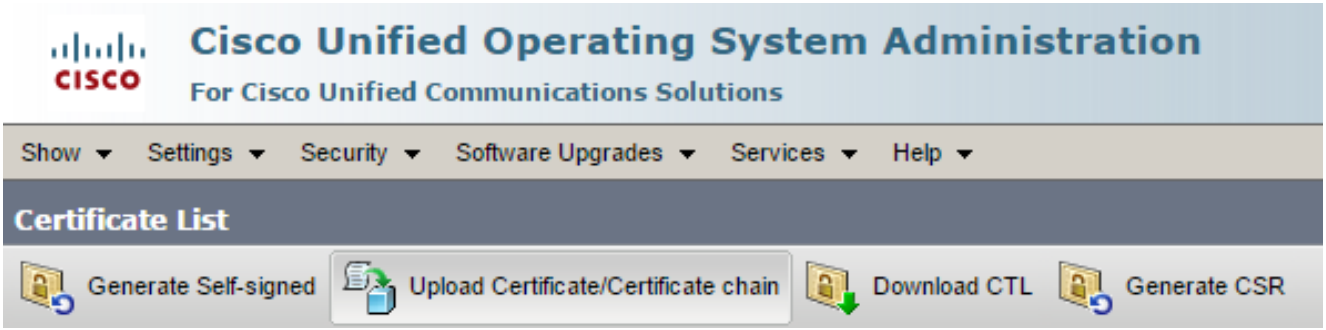
6. Sélectionnez le certificat sous le répertoire de *certificats utilisateurs* et exportez-le au répertoire sur le PC. Une fois incité pour un mot de passe, utilisez le mot de passe par défaut de **Cisco123** :



7. Répétez ces étapes pour le deuxième matériel USB eToken de sorte que les deux Certificats soient exportés au PC :

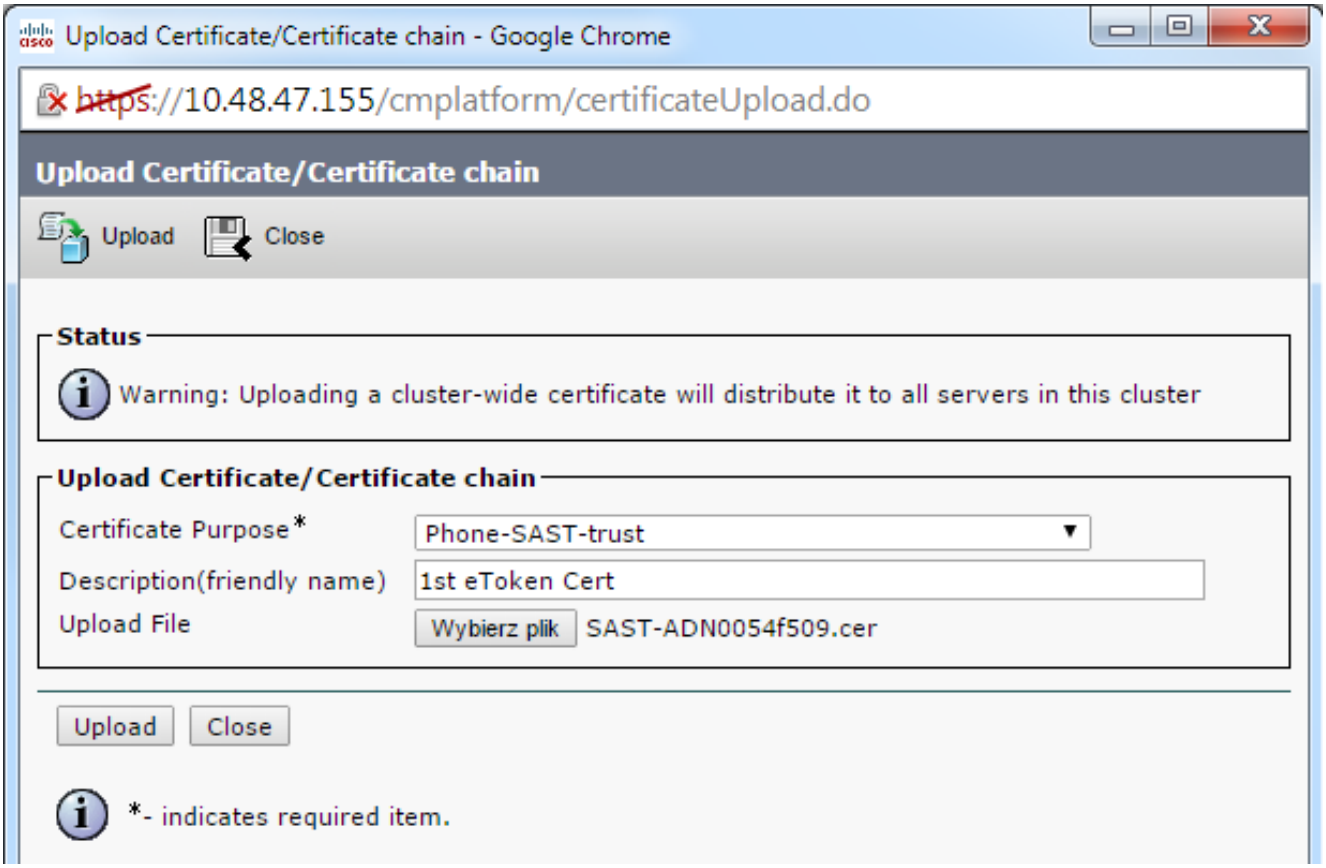


8. Connectez-vous dans Cisco Unified la gestion du système d'exploitation (de SYSTÈME D'EXPLOITATION) et naviguez vers la **Gestion de Sécurité > de certificat > le certificat de téléchargement** :

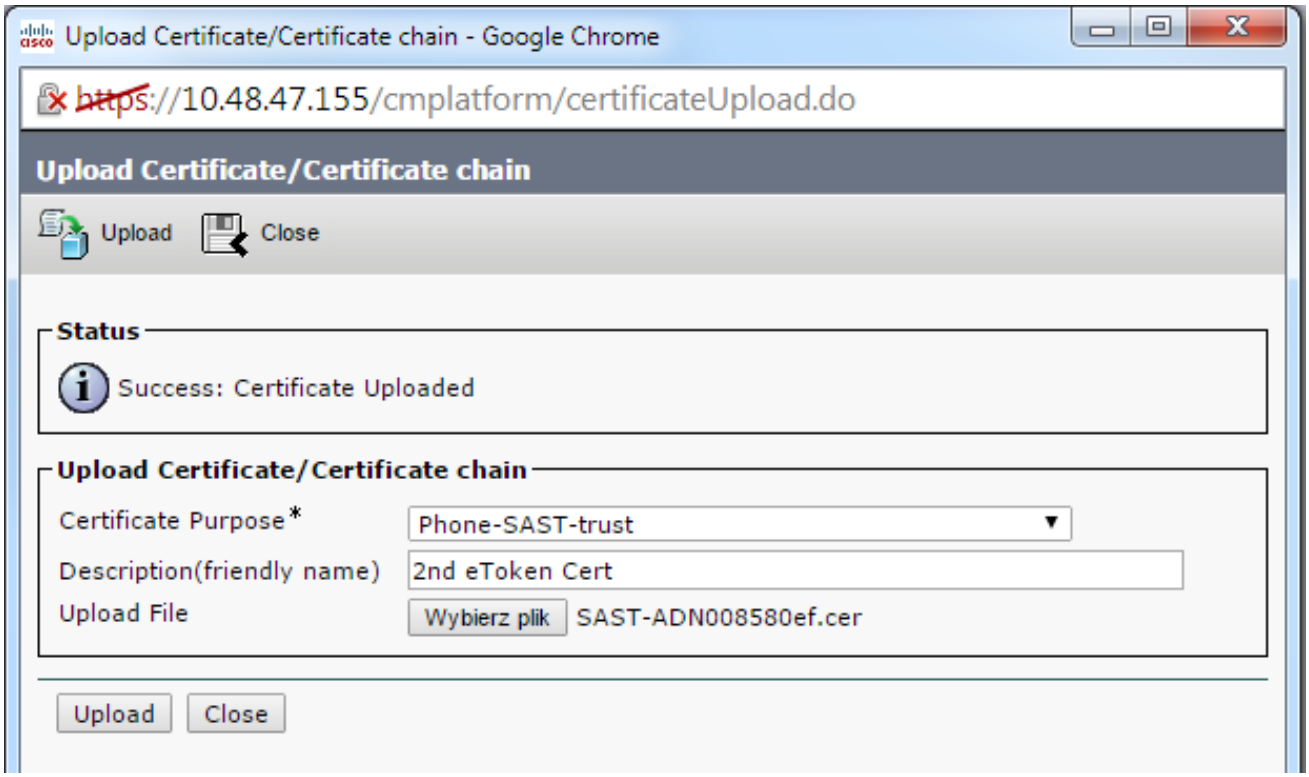


9. La page de certificat de téléchargement paraît alors. Choisissez la Téléphone-SAST-

confiance du but de certificat relâchent vers le bas le menu et sélectionnent le certificat que vous avez exporté dès le début eToken :

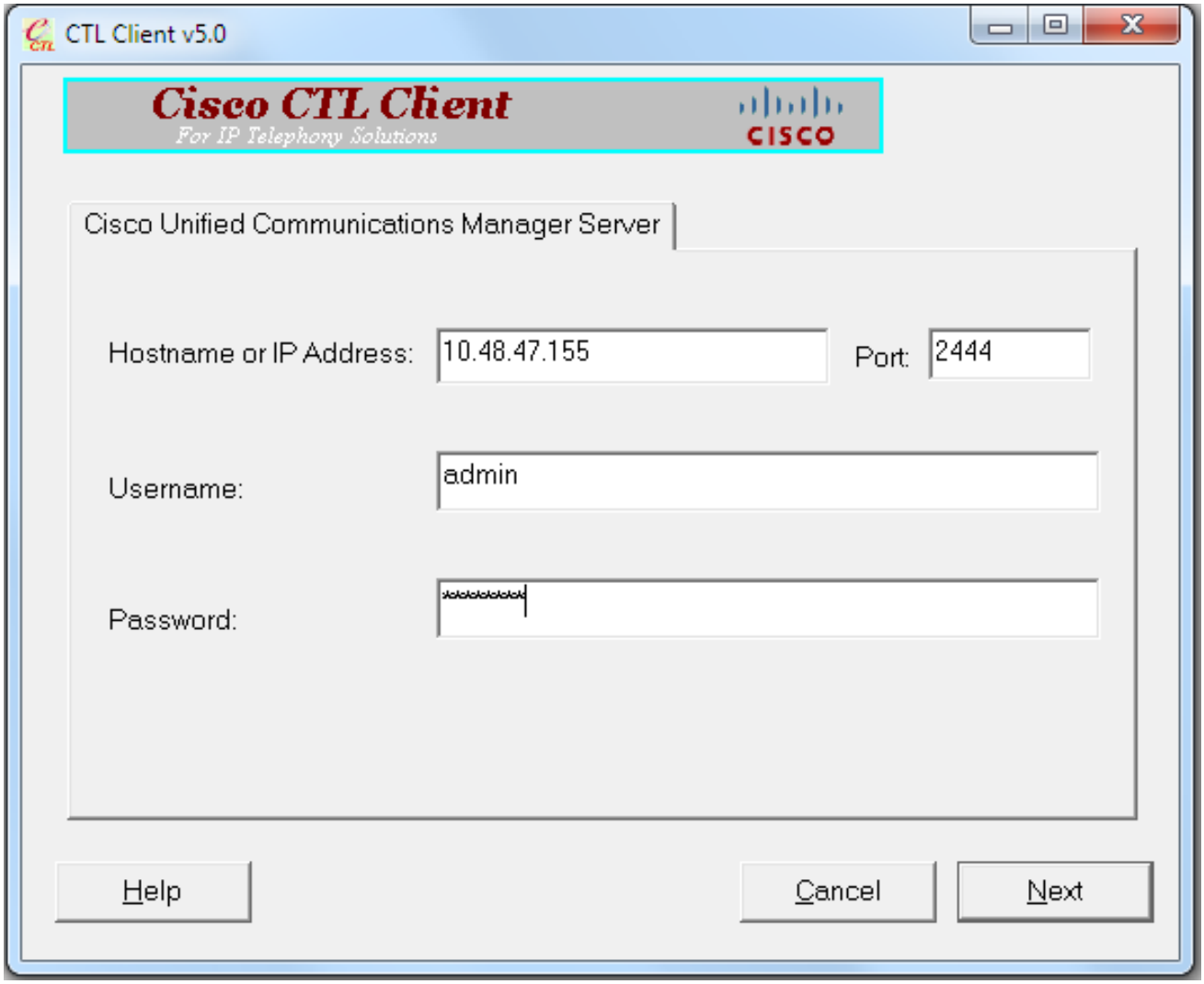


10. Terminez-vous les étapes précédentes afin de télécharger le certificat que vous avez exporté de la deuxième eToken :



11. Exécutez le client CTL, fournissez l'adresse IP/adresse Internet du noeud CUCM Publisher,

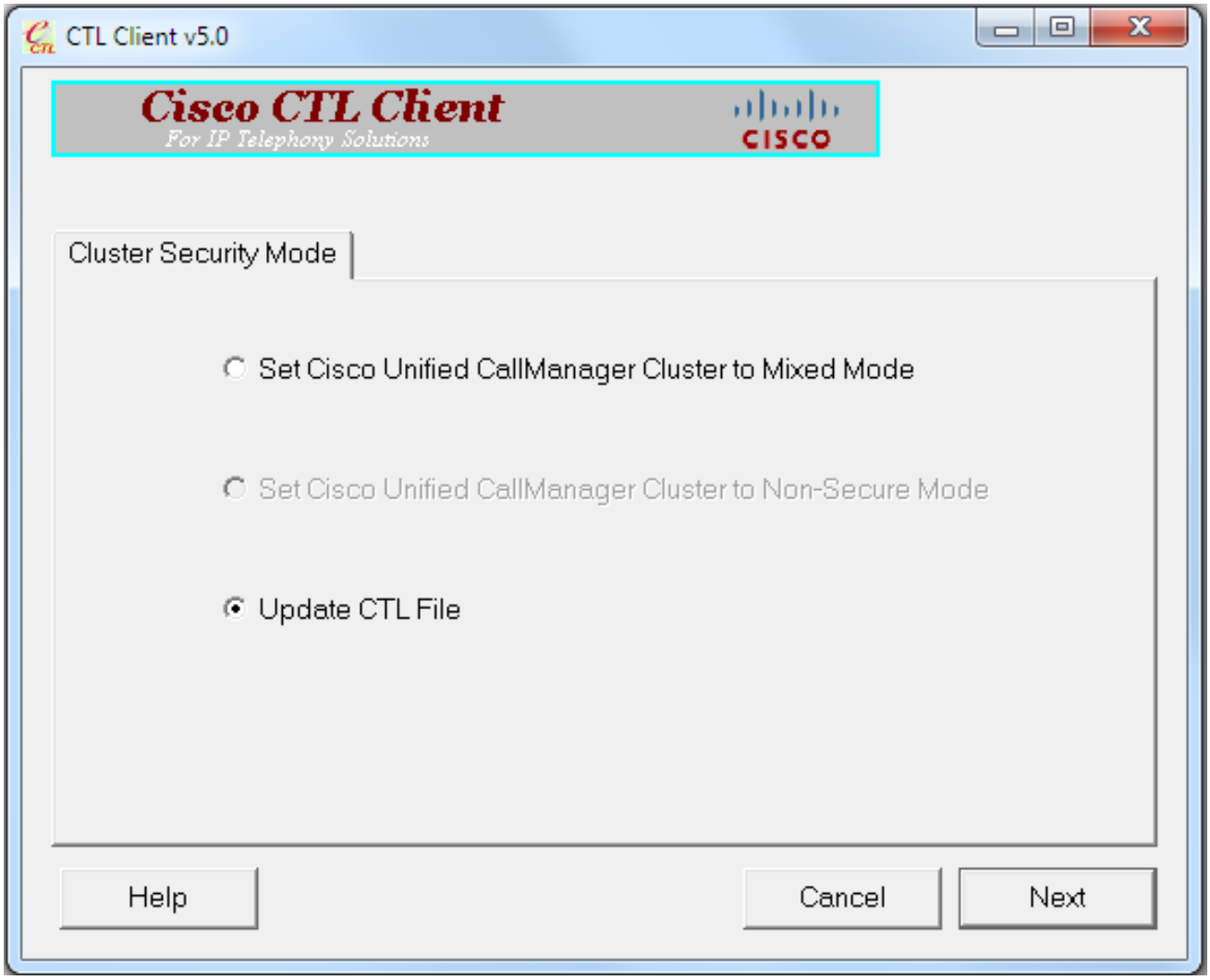
et entrez dans les qualifications de l'administrateur CCM :



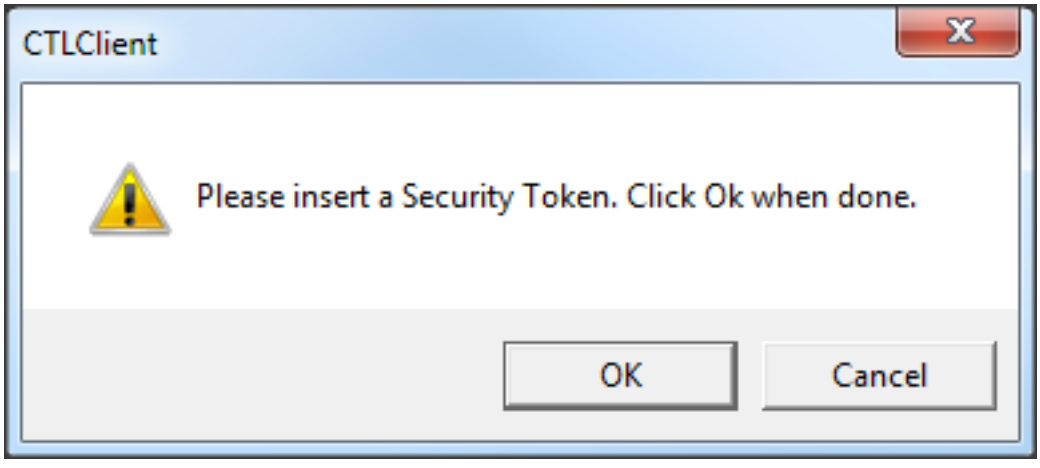
12. Puisque la batterie est dans le mode mixte déjà, mais aucun fichier CTL n'existe sur le noeud de Publisher, ce message d'avertissement apparaît (cliquez sur OK afin de l'ignorer) :

```
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.
```

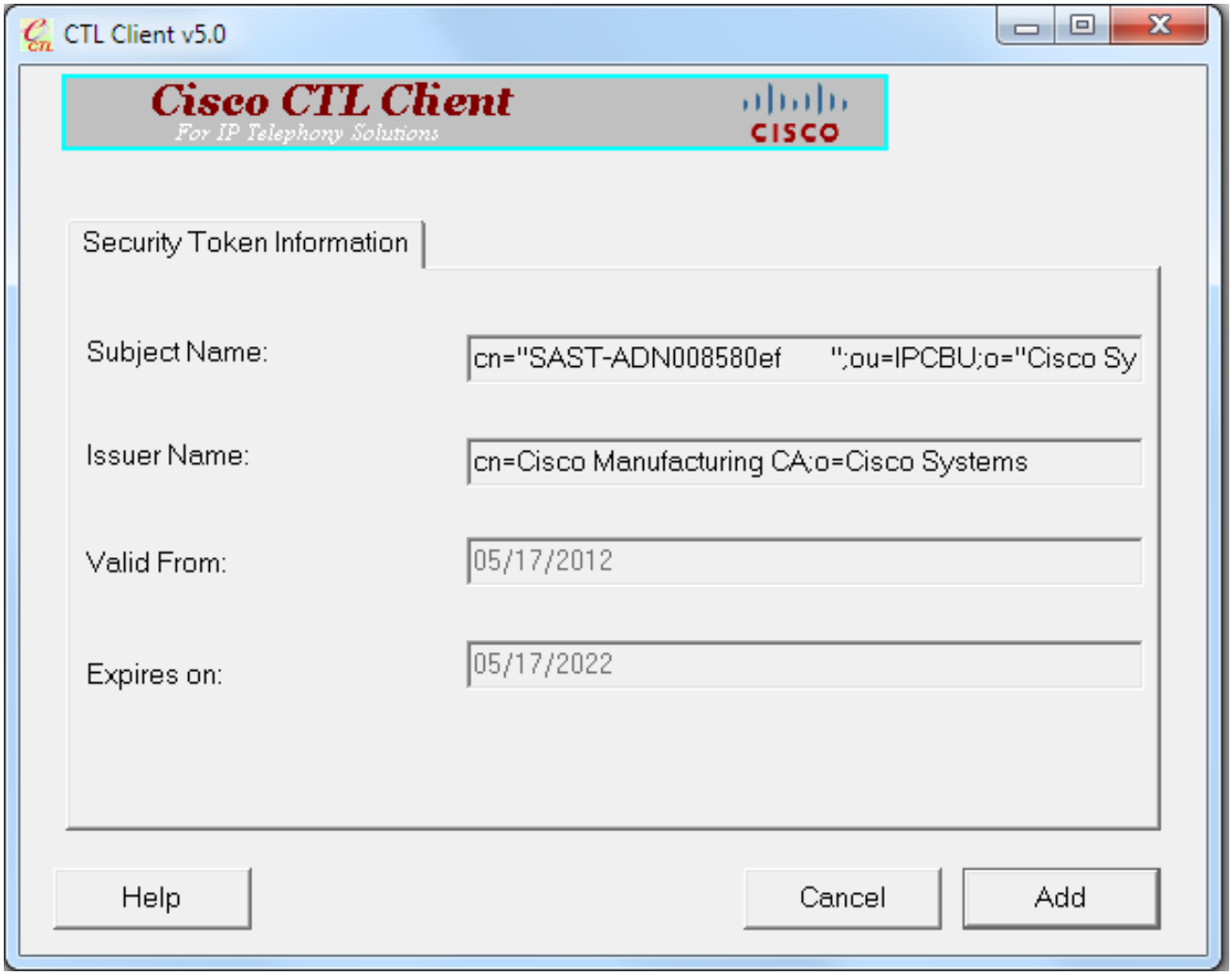
13. Du client CTL, cliquez sur la case d'option de **fichier CTL de mise à jour**, et puis cliquez sur Next :



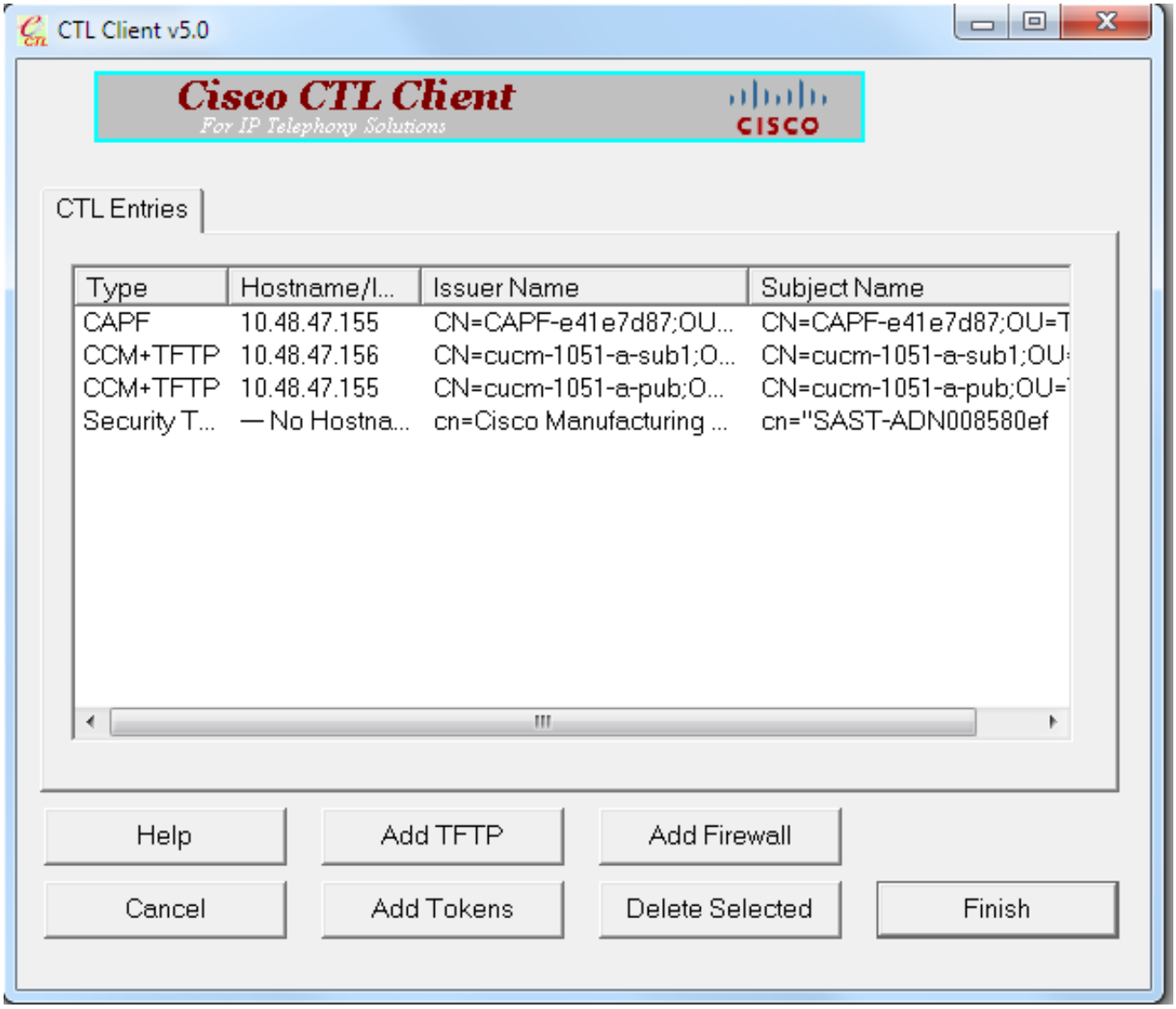
14. Insérez le premier jeton de Sécurité et cliquez sur OK :



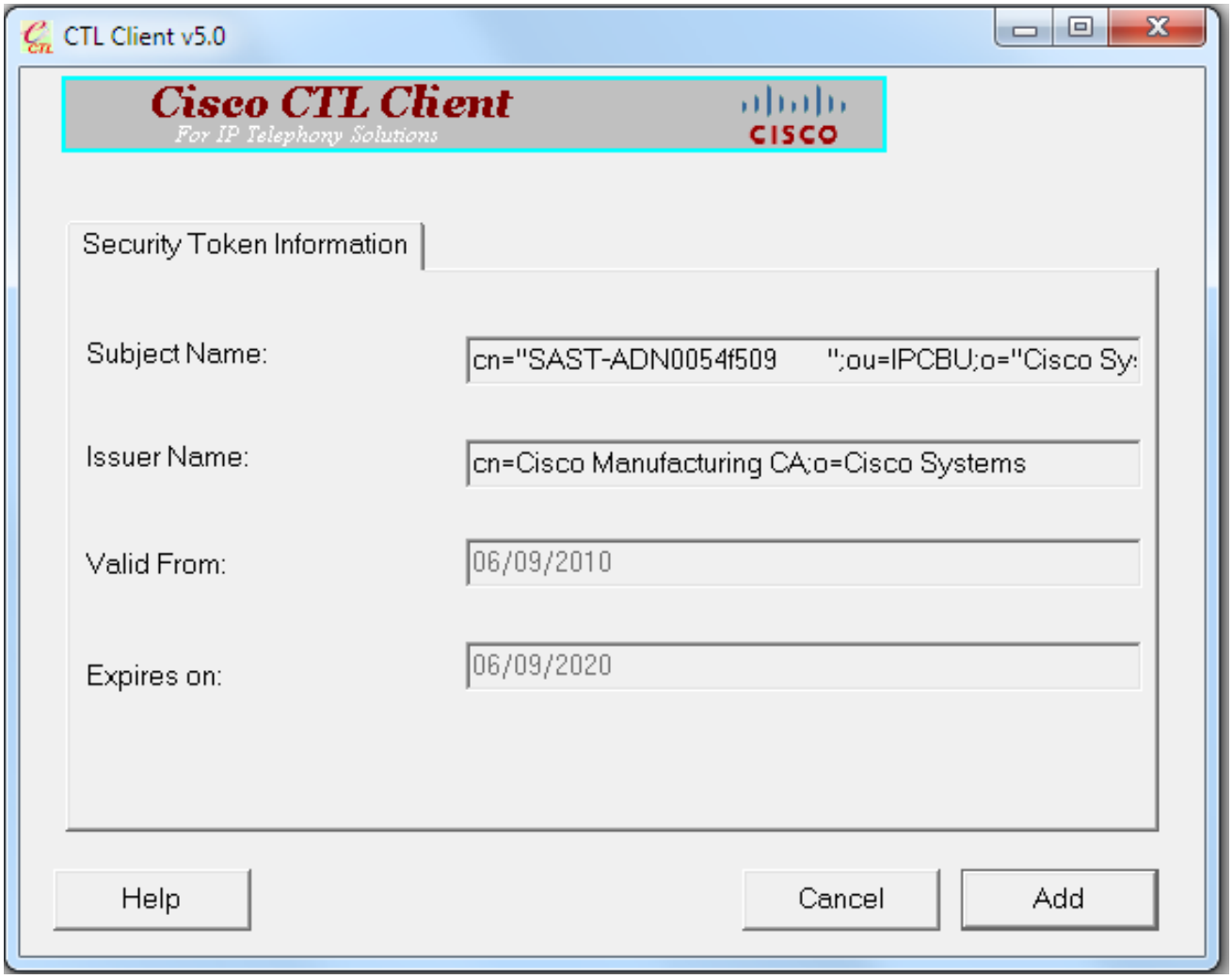
15. Après que les détails symboliques de Sécurité soient affichés, cliquez sur Add :



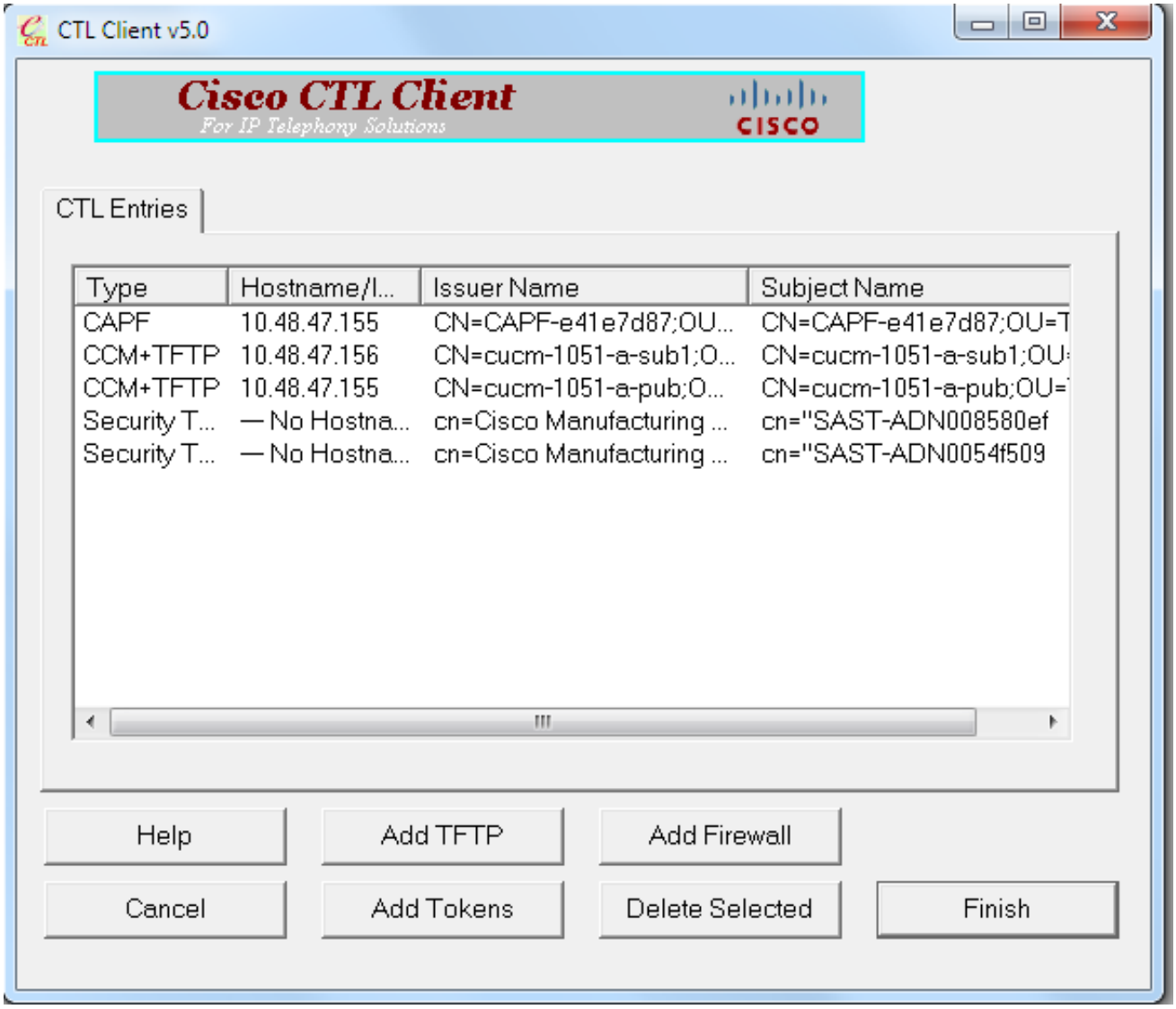
16. Une fois le contenu du fichier CTL apparaît, clique sur Add des **jetons** afin d'ajouter le deuxième USB eToken :



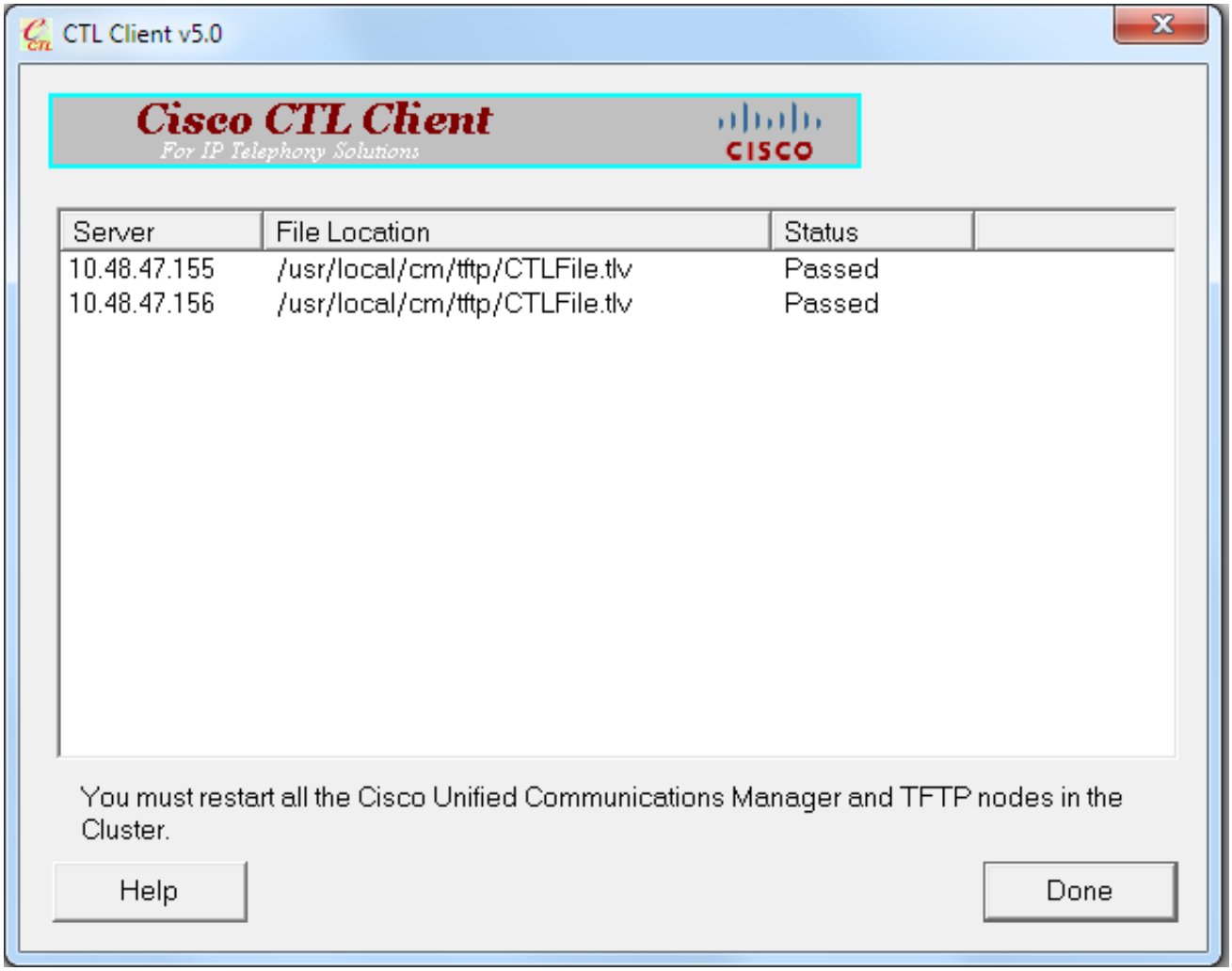
17. Après que les détails symboliques de Sécurité apparaissent, cliquez sur Add :



18. Après que le contenu du fichier CTL apparaisse, cliquez sur Finish. Une fois incité pour un mot de passe, écrivez **Cisco123** :



19. Quand la liste de serveurs CUCM sur lesquels le fichier CTL existe apparaît, clic **fait** :



20. Redémarrez le TFTP et les services de CallManager sur tous les Noeuds dans la batterie qui dirigent ces services.

21. Redémarrez tous les Téléphones IP de sorte qu'ils puissent obtenir la nouvelle version du fichier CTL du service TFTP CUCM.

22. Afin de vérifier le contenu du fichier CTL, sélectionnez la commande de **ctl d'exposition** dans le CLI. Dans le fichier CTL vous pouvez voir les Certificats de chacun des deux eTokens USB (l'un d'entre eux est utilisé afin de signer le fichier CTL). Voici un exemple de sortie :

```
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 ISSUERNAME 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 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.
```

23. Du côté de téléphone IP, vous pouvez vérifier cela après que les Téléphones IP aient été redémarrés, ils avez téléchargé la version mise à jour de fichier CTL (la somme de contrôle de MD5 s'assortit une fois comparé à la sortie du CUCM) :

Cette modification est possible parce que vous avez précédemment exporté et avez téléchargé les Certificats d'eToken à la mémoire de confiance de certificat CUCM, et les Téléphones IP peuvent vérifier ce certificat inconnu qui a été utilisé afin de signer le fichier CTL contre le service de vérification de confiance (TV) ces passages sur le CUCM. Ce snippit de log illustre comment le téléphone IP entre en contact avec le CUCM TV avec une demande de vérifier l'inconnu eToken le certificat, qui est téléchargé comme Téléphone-SAST-confiance et est de confiance :

```
//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
```

Régénération de certificat pour la solution de Tokenless CTL

Cette section décrit comment régénérer un Security Certificate de batterie CUCM quand la solution de Tokenless CTL est utilisée.

En cours de maintenance CUCM, parfois le certificat de CallManager de noeud CUCM Publisher change. Les scénarios dans lesquels ceci peut se produire incluent la modification de l'adresse Internet, la modification du domaine, ou simplement une régénération de certificat (devant fermer la date d'expiration de certificat).

Après que le fichier CTL soit mis à jour, il est signé avec un certificat différent que ceux qui existent dans le fichier CTL qui est installé sur les Téléphones IP. Normalement, ce nouveau fichier CTL n'est pas reçu ; cependant, après que le téléphone IP trouve le certificat inconnu qui est utilisé afin de signer le fichier CTL, il entre en contact avec le service TV sur le CUCM.

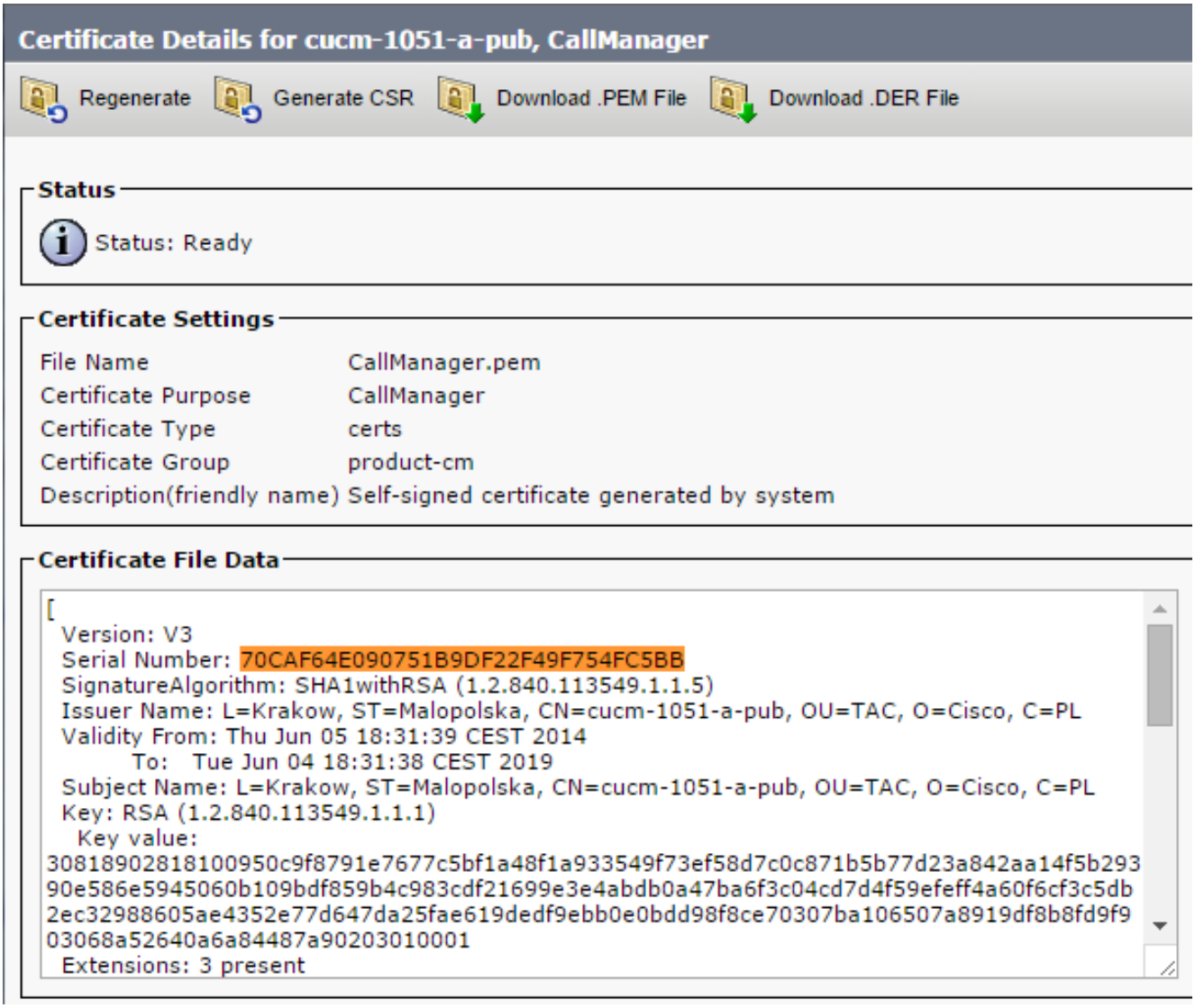
> Remarque: La liste de serveur TV est dans le fichier de configuration de téléphone IP et est tracée dans les serveurs CUCM du **groupe de Pool d'appareils > de CallManager de**

téléphone IP.

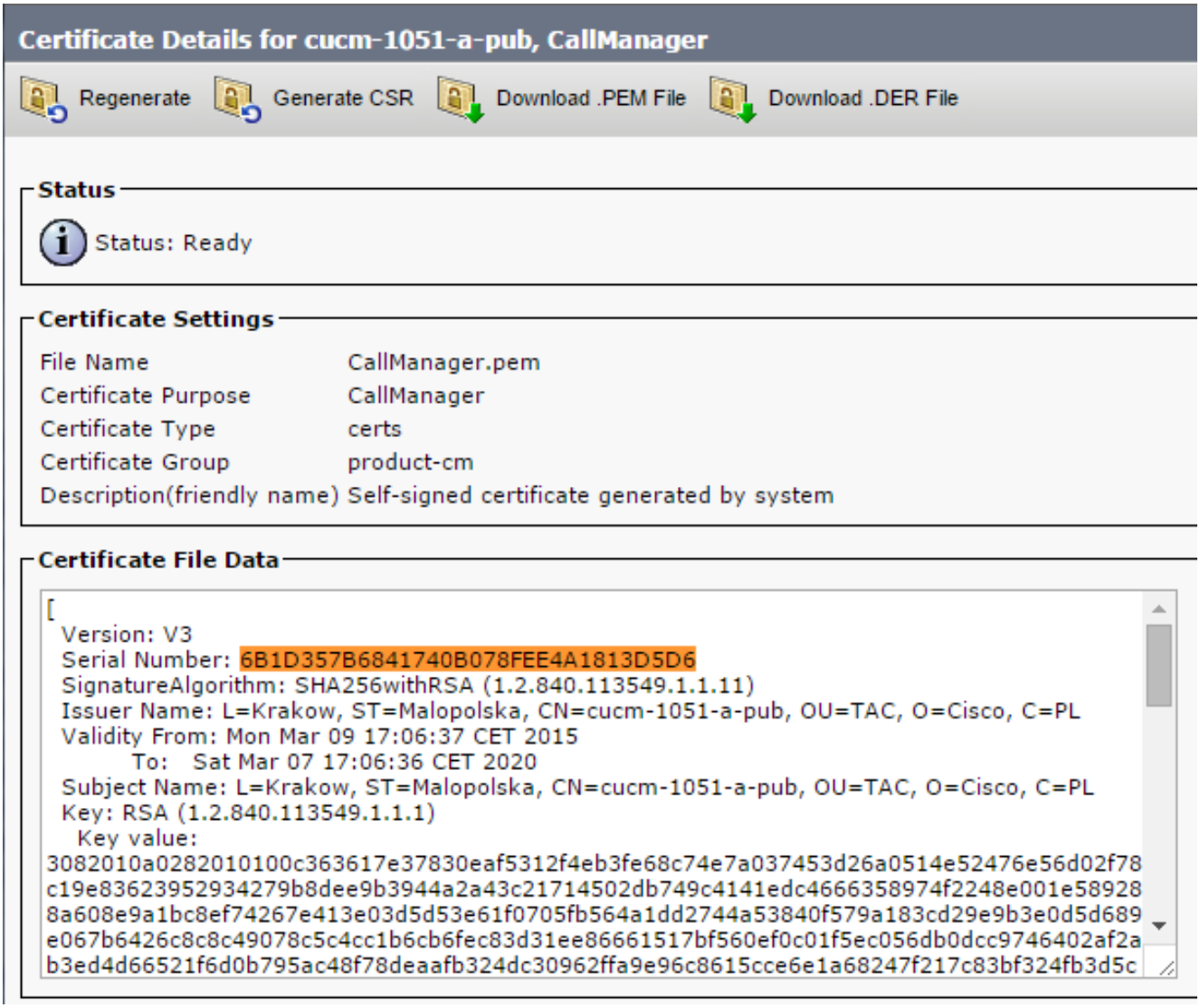
Sur la vérification réussie contre le serveur TV, le téléphone IP met son fichier CTL à jour avec la nouvelle version. Ces événements se produisent dans un tel scénario :

1. Le fichier CTL existe sur le CUCM et sur le téléphone IP. Le certificat CCM+TFT (serveur) pour le noeud CUCM Publisher est utilisé afin de signer le fichier CTL :

```
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 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.
```



2. **Le fichier CallManager.pem** (certificat CCM+TFTP) est régénéré, et vous peut voir que le numéro de série du certificat change :



3. La commande de **CTLFile de mise à jour de ctl d'utils** est sélectionnée dans le CLI afin de mettre le fichier CTL à jour :

```
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. Le service TV met à jour son cache de certificat avec les nouveaux détails de fichier CTL :

```
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. Quand vous visualisez le contenu de fichier CTL, vous pouvez voir que le fichier est signé avec le nouveau certificat de serveur CallManager pour le noeud de Publisher :

```
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. De la page unifiée d'utilité, le TFTP et des services de Cisco CallManager sont redémarrés sur tous les Noeuds dans la batterie qui dirigent ces services.

7. Les Téléphones IP sont redémarrés, et ils contactent le serveur TV afin de vérifier le certificat inconnu qui est maintenant utilisé afin de signer la nouvelle version du fichier CTL :

```
// 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. En conclusion, sur les Téléphones IP, vous pouvez vérifier que le fichier CTL est mis à jour avec la nouvelle version et que la somme de contrôle de MD5 du nouveau fichier CTL s'assortit avec celle du CUCM :