

토큰리스 CTL이 포함된 CUCM 혼합 모드

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[비보안 모드에서 혼합 모드\(토큰리스 CTL\)로](#)

[하드웨어 토큰에서 토큰리스 솔루션으로](#)

[토큰리스 솔루션에서 하드웨어 토큰으로](#)

[토큰리스 CTL 솔루션의 인증서 재생성](#)

소개

이 문서에서는 CUCM(Cisco Unified Communications Manager) 보안과 하드웨어 USB eToken 사용/미사용 간의 차이점에 대해 설명합니다.

사전 요구 사항

요구 사항

CUCM 버전 10.0(1) 이상에 대해 알고 있는 것이 좋습니다. 또한 다음을 확인합니다.

- CUCM 버전 11.5.1SU3 이상의 라이선스 서버는 Cisco PLM(Prime License Manager) 11.5.1SU2 이상이어야 합니다.

이는 CUCM 버전 11.5.1SU3에서 혼합 모드를 활성화하려면 암호화 라이선스가 필요하며, PLM에서 11.5.1SU2까지 암호화 라이선스를 지원하지 않기 때문입니다.


자세한 내용은 [Cisco Prime License Manager 릴리스 11.5\(1\)SU2의 릴리스 정보를 참조하십시오](#).

- CUCM 게시자 노드의 CLI(Command Line Interface)에 대한 관리 액세스 권한이 있습니다.
- 하드웨어 USB eTokens에 액세스할 수 있으며 하드웨어 eTokens의 사용으로 다시 마이그레이션해야 하는 시나리오를 위해 CTL 클라이언트 플러그인이 PC에 설치되어 있습니다.

명확성을 위해 이 요건은 USB eToken이 필요한 시나리오를 가지고 있는 경우에만 해당됩니다. 대부분의 사람들에게 USB eToken이 필요할 가능성은 매우 적다.

- 클러스터의 모든 CUCM 노드 간에 완전한 연결이 있습니다. CTL 파일은 SSH SFTP(File Transfer Protocol)를 통해 클러스터의 모든 노드에 복사되기 때문에 이는 매우 중요합니다.
- 클러스터의 데이터베이스(DB) 복제가 제대로 작동하며 서버가 데이터를 실시간으로 복제합니다.
- 구축의 디바이스는 TVS(Security by Default)를 지원합니다.

Cisco Unified Reporting 웹 페이지(<https://<CUCM IP 또는 FQDN>/cucreports/>)에서 Unified CM Phone 기능 목록을 사용하여 기본적으로 보안을 지원하는 장치를 결정할 수 있습니다.

 참고: Cisco Jabber 및 많은 Cisco TelePresence 또는 Cisco 7940/7960 Series IP 전화는 현재 기본적으로 보안을 지원하지 않습니다. 기본적으로 보안을 지원하지 않는 장치와 함께 토큰리스 CTL을 배포하는 경우 게시자에서 CallManager 인증서를 변경하는 시스템 업데이트가 있으면 CTL이 수동으로 삭제될 때까지 해당 장치의 정상적인 기능을 사용할 수 없습니다. 7945 및 7965 전화 이상과 같이 기본적으로 보안을 지원하는 장치는 TVS(Trust Verification Service)를 사용할 수 있으므로 게시자의 CallManager 인증서가 업데이트될 때 CTL 파일을 설치할 수 있습니다.

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- CUCM 버전 10.5.1.10000-7(노드 2개로 구성된 클러스터)
- 펌웨어 버전 SCCP75.9-3-1SR4-1S를 사용하는 SCCP(Skinny Client Control Protocol)를 통해 등록된 Cisco 7975 Series IP Phone
- CTL Client 소프트웨어를 사용하여 클러스터를 혼합 모드로 설정하기 위해 사용되는 2개의 Cisco 보안 토큰

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.


배경 정보

이 문서에서는 하드웨어 USB eToken을 사용하는 경우와 사용하지 않는 경우의 Cisco Unified Communications Manager(CUCM) 보안의 차이점에 대해 설명합니다.

또한 이 문서에서는 CTL(Tokenless Certificate Trust List)과 관련된 기본 구현 시나리오와 변경 후 시스템이 제대로 작동하도록 하기 위해 사용되는 프로세스에 대해 설명합니다.

Tokenless CTL은 CUCM 버전 10.0(1) 이상의 새로운 기능으로, 하드웨어 USB eTokens 및 이전 CUCM 릴리스의 요구 사항이었던 CTL 클라이언트 플러그인을 사용하지 않고도 IP Phone용 통화 신호 및 미디어를 암호화할 수 있습니다.

CLI 명령을 사용하여 클러스터를 혼합 모드로 전환하면 CTL 파일은 게시자 노드의 CCM+TFTP(서버) 인증서로 서명되며 CTL 파일에 eToken 인증서가 없습니다.

 참고: 게시자에서 CallManager(CCM+TFTP) 인증서를 재생성하면 파일의 서명자가 변경됩니다. 기본적으로 보안을 지원하지 않는 전화기 및 장치도 각 장치에서 CTL 파일을 수동으로 삭제하지 않는 한 새 CTL 파일을 수락하지 않습니다. 자세한 내용은 이 문서의 요구 사항 섹션에 [나와](#) 있는 마지막 요구 사항을 참조하십시오.

비보안 모드에서 혼합 모드(토큰리스 CTL)로

이 섹션에서는 CLI를 통해 CUCM 클러스터 보안을 혼합 모드로 이동하는 데 사용되는 프로세스에 대해 설명합니다.

이 시나리오 이전에는 CUCM이 비보안 모드였습니다. 즉, 노드에 CTL 파일이 없고 등록된 IP Phone에 ITL(Identity Trust List) 파일만 설치되어 있었습니다. 이 내용은 다음 출력에서 확인할 수 있습니다.

```
<#root>
```

```
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:
```

 참고: 클러스터가 혼합 모드가 아닐 때 서버에 CTL 파일이 있는 경우, 이는 클러스터가 혼합 모드로 전환된 후 다시 혼합 모드가 아닌 모드로 전환되었고 CTL 파일이 클러스터에서 삭제되지 않았음을 의미합니다.

명령 파일 `delete activelog cm/tftpdata/CTLFile.tlv`는 CUCM 클러스터의 노드에서 CTL 파일을 삭제합니다. 그러나 각 노드에 명령을 입력해야 합니다. 명확히 하려면 서버에 CTL 파일이 있고 클러스터가 혼합 모드가 아닌 경우에만 이 명령을 사용합니다.

클러스터가 혼합 모드인지 확인하는 쉬운 방법은 `paramname='ClusterSecurityMode'`인 `processconfig`에서 `sql select paramname,paramvalue` 명령을 실행하는 것입니다. 매개변수 값이 0이면 클러스터는 혼합 모드가 아닙니다.

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



새 토큰리스 CTL 기능을 사용하여 CUCM 클러스터 보안을 혼합 모드로 전환하려면 다음 단계를 완료하십시오.

1. CUCM 게시자 노드 CLI에 대한 관리 액세스 권한을 얻습니다.
2. CLI에 `utils ctl set-cluster mixed-mode` 명령을 입력합니다.

```
<#root>
```

```
admin:
```

```
utils ctl set-cluster mixed-mode
```

```
This operation sets 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. CUCM Admin Page(CUCM 관리 페이지) > System(시스템) > Enterprise Parameters(엔터프

라이즈 매개변수로 이동하고 클러스터가 Mixed mode(혼합 모드)로 설정되었는지 확인합니다(1 값은 혼합 모드를 나타냄).

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

- 이러한 서비스를 실행하는 클러스터의 모든 노드에서 TFTP 및 Cisco CallManager 서비스를 다시 시작합니다.
- CUCM TFTP 서비스에서 CTL 파일을 가져올 수 있도록 모든 IP Phone을 다시 시작합니다.
- CTL 파일의 내용을 확인하려면 CLI에 show ctl 명령을 입력합니다.
- CTL 파일에서 CCM 게시자 노드에 대한 CCM+TFTP(서버) 인증서가 CTL 파일에 서명하는 데 사용됨을 확인할 수 있습니다(이 파일은 클러스터의 모든 서버에서 동일함). 다음은 샘플 출력입니다.

<#root>

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      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=Małopolska;C=PL
4      FUNCTION          2
CCM+TFTP

5      ISSUENAME         62      CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow;
                                      ST=Małopolska;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.

- IP Phone 측에서 서비스가 다시 시작된 후 TFTP 서버에 있는 CTL 파일을 다운로드하는지 확인할 수 있습니다(CUCM의 출력과 비교할 때 MD5 체크섬이 일치함).

 참고: 전화기의 체크섬을 확인할 때 전화기 유형에 따라 MD5 또는 SHA1이 표시됩니다.



하드웨어 토큰에서 토큰리스 솔루션으로

이 섹션에서는 CUCM 클러스터 보안을 하드웨어 eTokens에서 새로운 Tokenless 솔루션을 사용하도록 마이그레이션하는 방법에 대해 설명합니다.

경우에 따라 혼합 모드는 CTL 클라이언트를 사용하여 CUCM에 이미 구성되어 있으며, IP Phone은 하드웨어 USB eToken의 인증서가 포함된 CTL 파일을 사용합니다.

이 시나리오에서는 CTL 파일이 USB eToken 중 하나의 인증서로 서명되고 IP Phone에 설치됩니다. 예를 들면 다음과 같습니다.

```
<#root>
```

```
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	ISSUENAME	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.



CUCM 클러스터 보안을 토큰리스 CTL 사용으로 이동하려면 다음 단계를 완료하십시오.

1. CUCM 게시자 노드 CLI에 대한 관리 액세스 권한을 얻습니다.
2. `utils ctl update CTLFile` CLI 명령을 입력합니다.

```
<#root>
```

```
admin:
```

```
utils ctl update CTLFile
```

```
This operation updates 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. 이러한 서비스를 실행하는 클러스터의 모든 노드에서 TFTP 및 CallManager 서비스를 다시 시작합니다.

4. CUCM TFTP 서비스에서 CTL 파일을 가져올 수 있도록 모든 IP Phone을 다시 시작합니다.
5. CTL 파일의 내용을 확인하기 위해 CLI에 show ctl 명령을 입력합니다. CTL 파일에서 CUCM 게시자 노드의 CCM+TFTP(서버) 인증서가 하드웨어 USB eTokens의 인증서 대신 CTL 파일에 서명하는 데 사용됨을 확인할 수 있습니다.
6. 이 경우에 한 가지 더 중요한 차이점은 모든 하드웨어 USB eToken의 인증서가 CTL 파일에서 제거된다는 것입니다. 다음은 샘플 출력입니다.

```
<#root>
```

```
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=Małopolska;C=PL
4	FUNCTION	2	

```
System Administrator Security Token
```

5	ISSUENAME	62	CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow; ST=Małopolska;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      ISSUENAME      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.

 참고: 위 출력에서 CUCM 게시자의 CCM+TFTP(서버) 인증서가 서명자가 아닌 경우 하드웨어 토큰 기반 클러스터 보안 모드로 다시 이동한 다음 토큰리스 솔루션에 대해 변경 사항을 다시 반복합니다.

7. IP Phone 측에서 IP Phone이 다시 시작된 후 업데이트된 CTL 파일 버전을 다운로드했는지 확인할 수 있습니다(CUCM의 출력과 비교할 때 MD5 체크섬이 일치함).



토큰리스 솔루션에서 하드웨어 토큰으로

이 섹션에서는 CUCM 클러스터 보안을 새로운 Tokenless 솔루션에서 벗어나 하드웨어 eTokens를 사용하도록 다시 마이그레이션하는 방법에 대해 설명합니다.

CLI 명령을 사용하여 CUCM 클러스터 보안을 혼합 모드로 설정하고 CTL 파일이 CUCM 게시자 노드에 대한 CCM+TFTP(서버) 인증서로 서명된 경우 CTL 파일에 있는 하드웨어 USB eToken의 인증서가 없습니다.

따라서 CTL 파일을 업데이트하기 위해 CTL 클라이언트를 실행하면(하드웨어 eTokens를 사용하도록 다시 이동) 다음과 같은 오류 메시지가 나타납니다.

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.

이는 특히 시스템의 다운그레이드(버전이 다시 전환되는 경우)를 utils ctl 명령이 포함되지 않은 10.x 이전 버전으로 전환하는 시나리오에서 중요합니다.

이전 CTL 파일은 새로고침 또는 Linux에서 Linux(L2)로 업그레이드하는 과정에서 (내용을 변경하지

않고) 마이그레이션되며, 앞서 언급한 것처럼 eToken 인증서가 포함되지 않습니다. 다음은 샘플 출력입니다.

<#root>

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=Małopolska;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=Małopolska;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=Małopołska;C=PL
4	FUNCTION	2	System Administrator Security Token
5	ISSUERNAME	62	CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow; ST=Małopołska;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=Małopołska;C=PL
4	FUNCTION	2	
CCM+TFTP			
5	ISSUERNAME	62	CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow; ST=Małopołska;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=Małopołska;C=PL
4	FUNCTION	2	CAPF
5	ISSUERNAME	60	CN=CAPF-e41e7d87;OU=TAC;O=Cisco;L=Krakow; ST=Małopołska;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:

이 시나리오에서는 모든 IP Phone에서 CTL 파일을 수동으로 삭제하는 손실 eToken 절차를 사용할 필요 없이 CTL 파일을 안전하게 업데이트하려면 다음 단계를 완료하십시오.

1. CUCM 게시자 노드 CLI에 대한 관리 액세스 권한을 얻습니다.
2. CTL 파일을 삭제하려면 Publisher 노드 CLI에 `file delete tftp CTLFile.tlv` 명령을 입력합니다.

```
<#root>
```

```
admin:
```

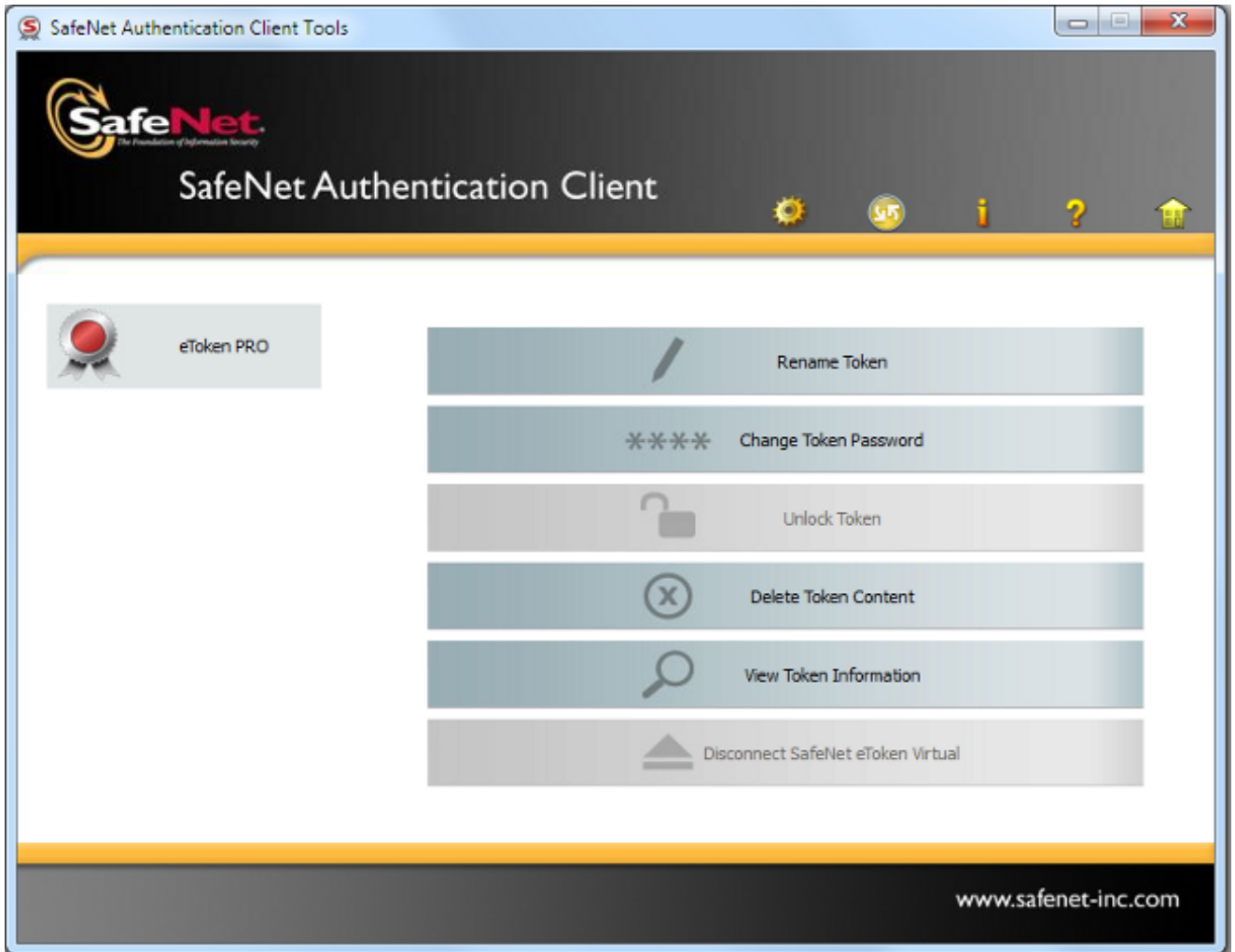
```
file delete tftp CTLFile.tlv
```

```
Delete the File CTLFile.tlv?
```

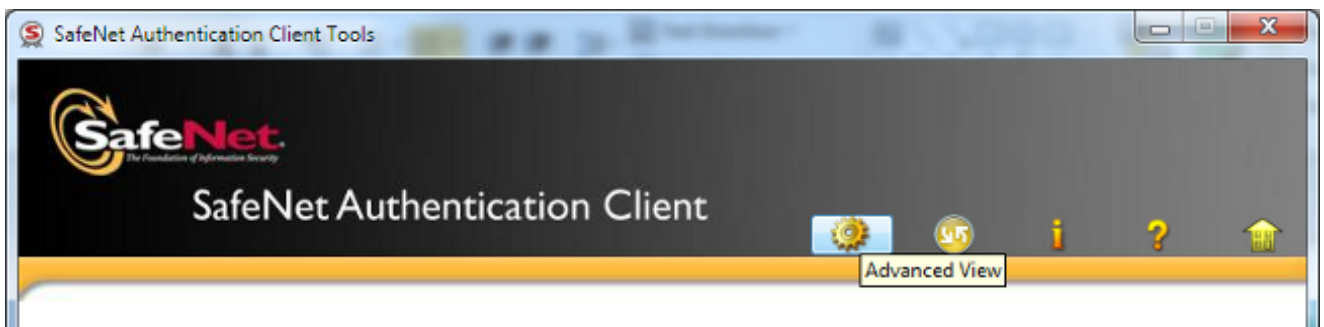
```
Enter "y" followed by return to continue: y
```

```
files: found = 1, deleted = 1
```

3. CTL 클라이언트가 설치된 Microsoft Windows 시스템에서 SafeNet 인증 클라이언트를 엽니다(CTL 클라이언트와 함께 자동으로 설치됨).

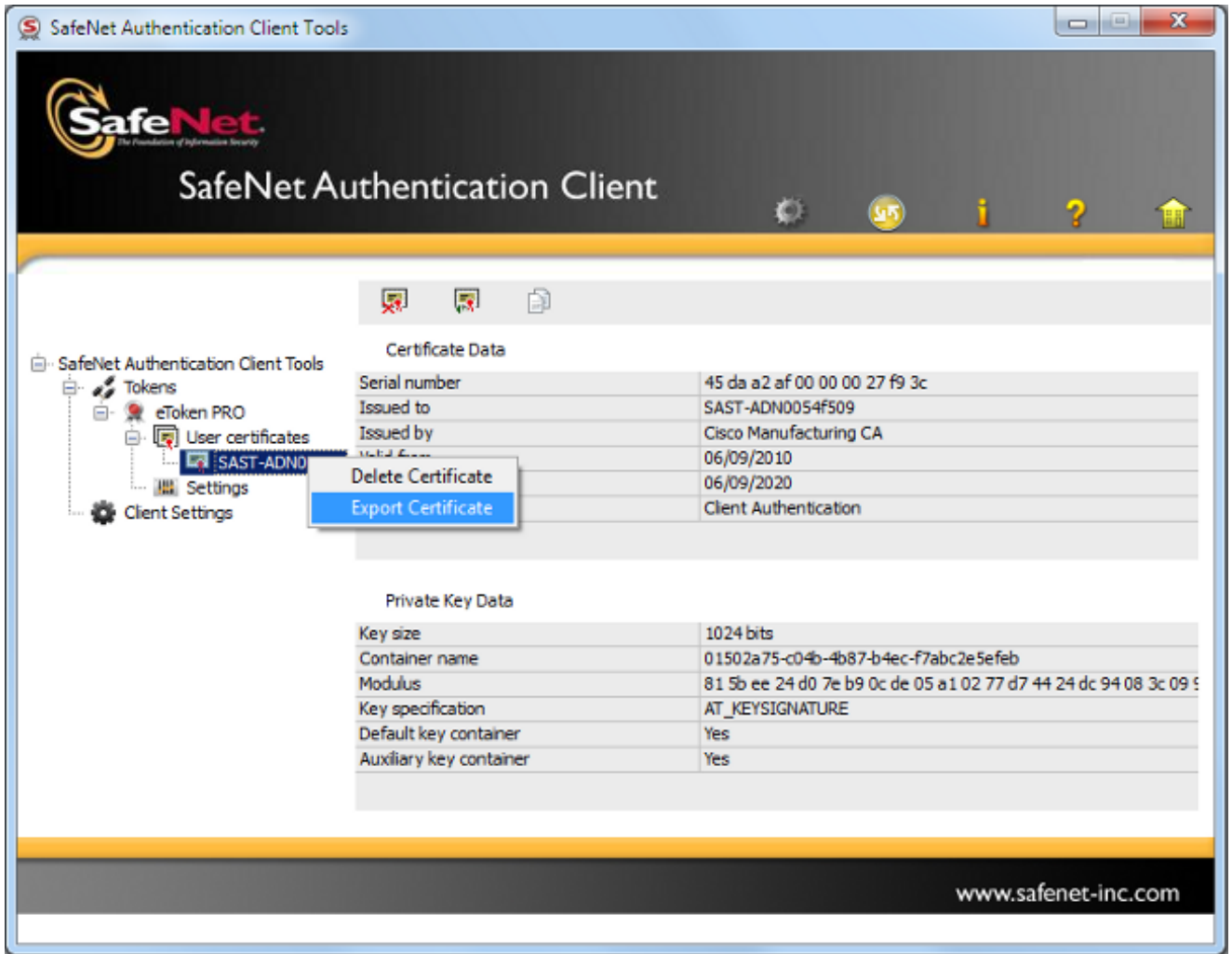


4. SafeNet 인증 클라이언트에서 고급 보기로 이동합니다.



5. 첫 번째 하드웨어 USB eToken을 삽입합니다.

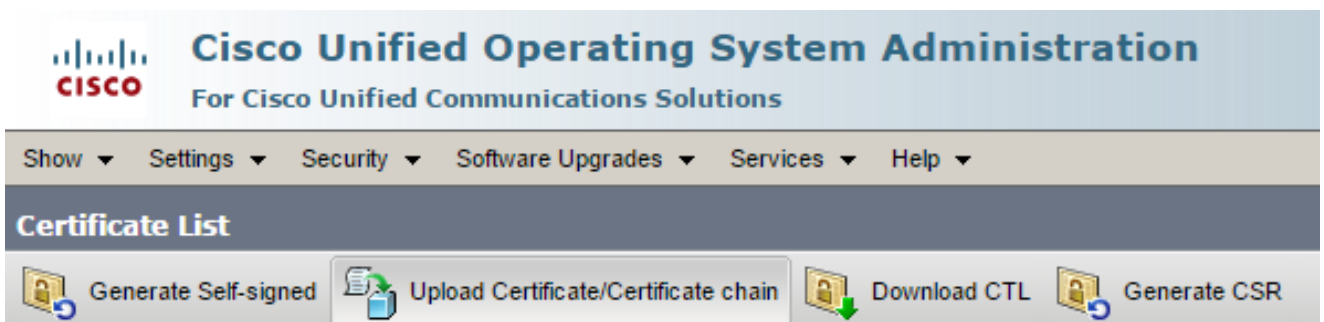
6. User certificates(사용자 인증서) 폴더에서 인증서를 선택하고 PC의 폴더로 내보냅니다. 비밀번호를 입력하라는 메시지가 표시되면 기본 비밀번호인 Cisco123을 사용합니다.



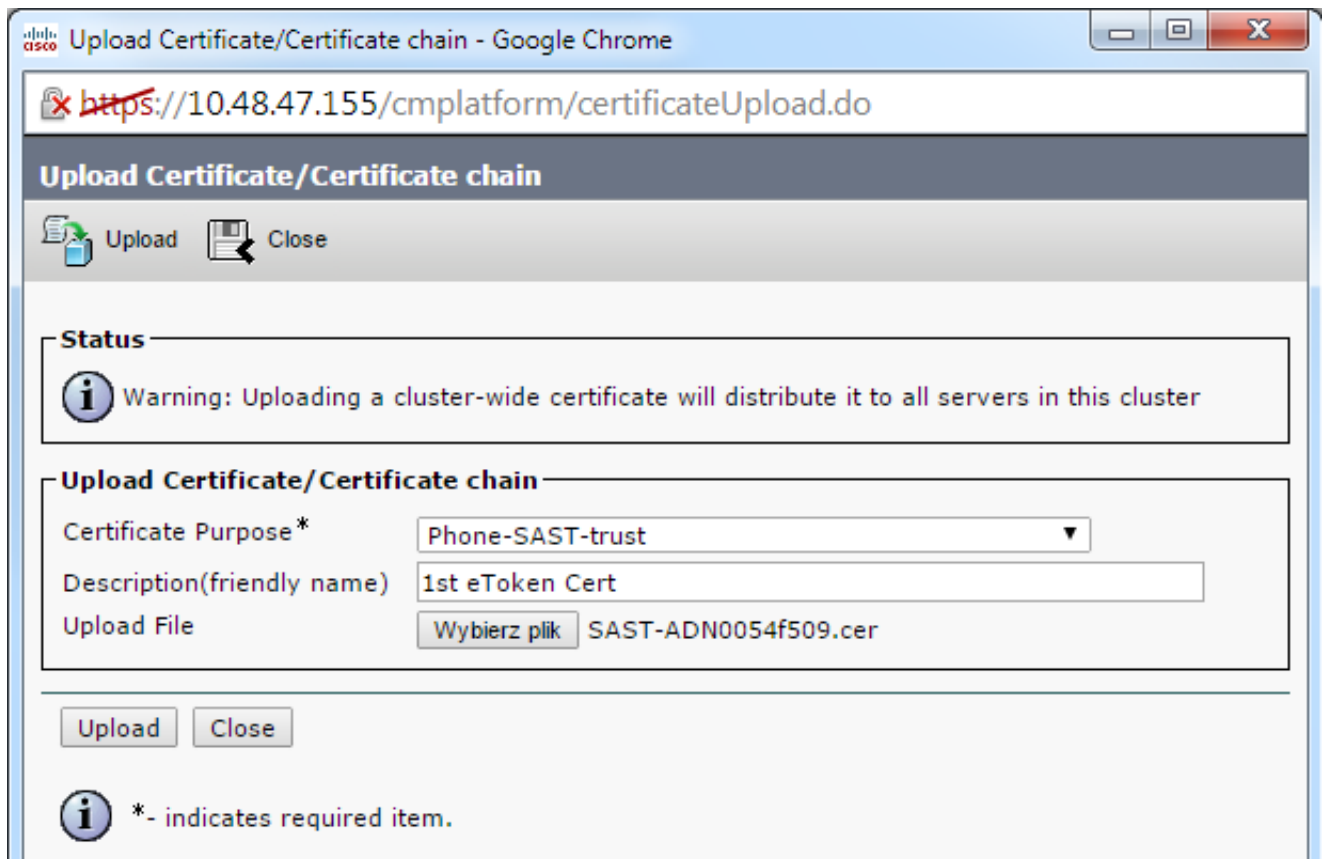
7. 두 인증서를 모두 PC로 내보내도록 두 번째 하드웨어 USB eToken에 대해 다음 단계를 반복합니다.

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. Cisco Unified OS(Operating System) Administration(Cisco Unified OS 관리)에 로그인하고 Security(보안) > Certificate Management(인증서 관리) > Upload Certificate(인증서 업로드)로 이동합니다.



9. 그러면 Upload Certificate 페이지가 나타납니다. Certificate Purpose(인증서 용도) 드롭다운 메뉴에서 Phone-SAST-trust를 선택하고 첫 번째 eToken에서 내보낸 인증서를 선택합니다.

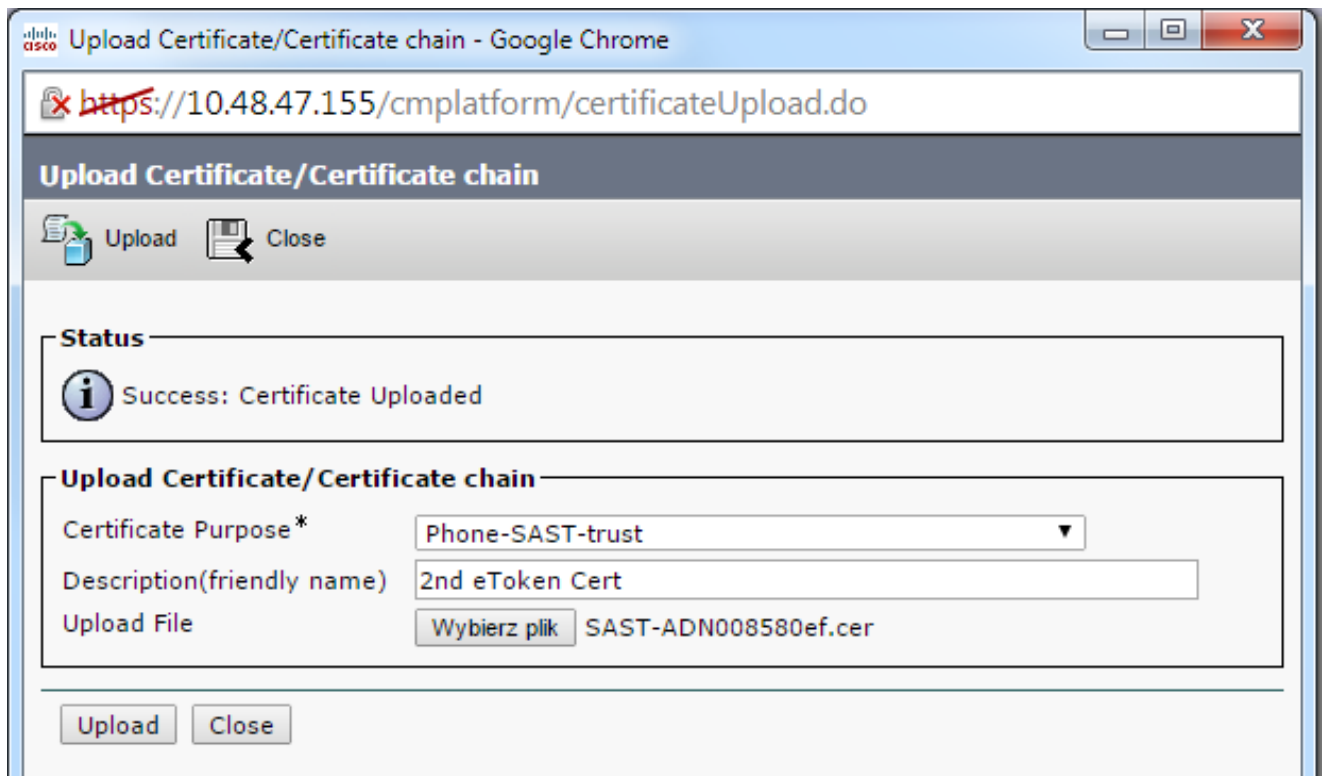


The screenshot shows a web browser window titled "Upload Certificate/Certificate chain - Google Chrome" with the URL "https://10.48.47.155/cmplatform/certificateUpload.do". The page header includes "Upload" and "Close" buttons. A status box displays a warning: "Warning: Uploading a cluster-wide certificate will distribute it to all servers in this cluster". The main form area is titled "Upload Certificate/Certificate chain" and contains the following fields:

- Certificate Purpose*: Phone-SAST-trust (dropdown menu)
- Description(friendly name): 1st eToken Cert (text input)
- Upload File: Wybierz plik SAST-ADN0054f509.cer (file selection button)

At the bottom of the form, there are "Upload" and "Close" buttons. A note below the form states: "* - indicates required item."

10. 두 번째 eToken에서 내보낸 인증서를 업로드하려면 이전 단계를 완료합니다.

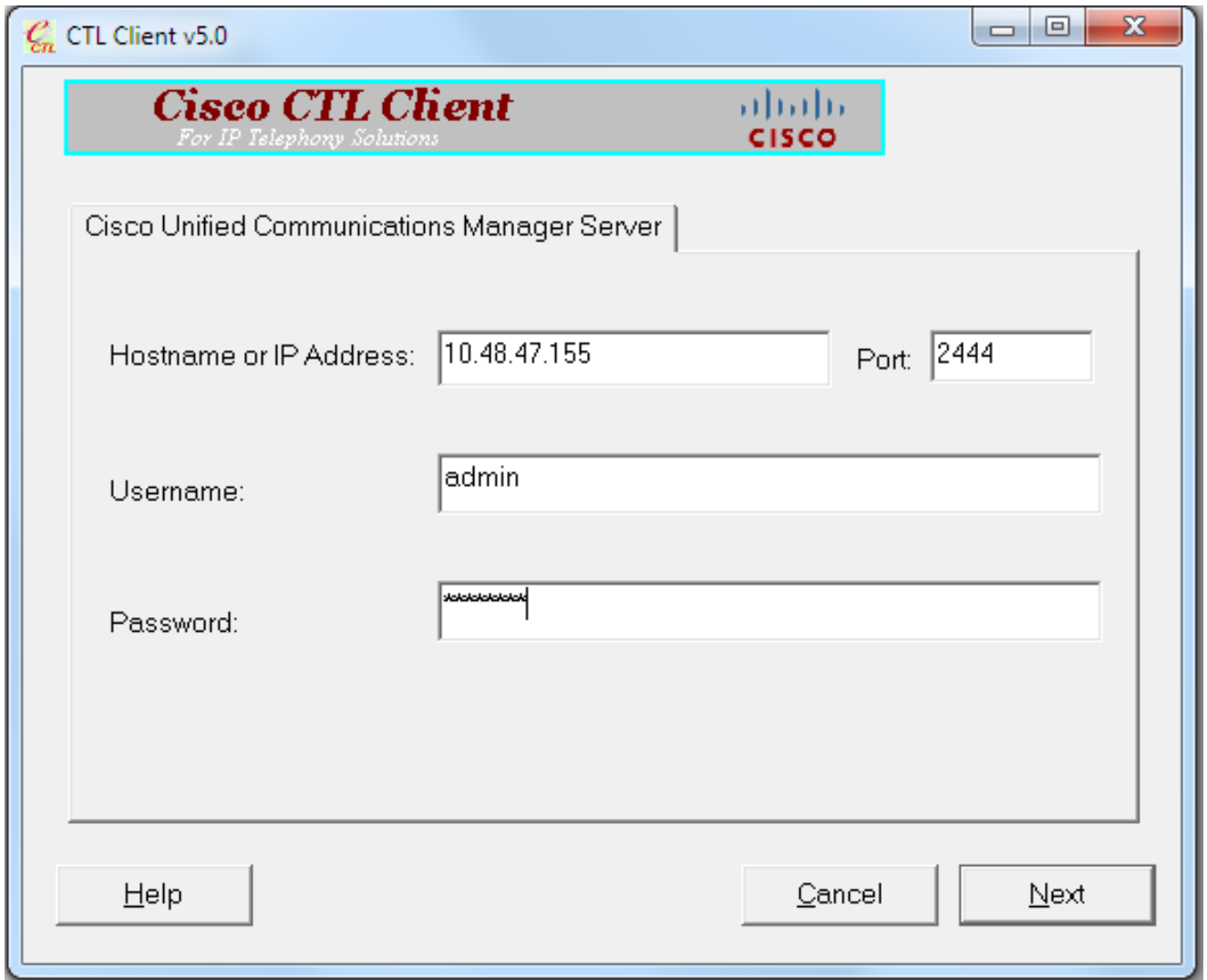


The screenshot shows the same web browser window, but the status box now displays a success message: "Success: Certificate Uploaded". The main form area is titled "Upload Certificate/Certificate chain" and contains the following fields:

- Certificate Purpose*: Phone-SAST-trust (dropdown menu)
- Description(friendly name): 2nd eToken Cert (text input)
- Upload File: Wybierz plik SAST-ADN008580ef.cer (file selection button)

At the bottom of the form, there are "Upload" and "Close" buttons.

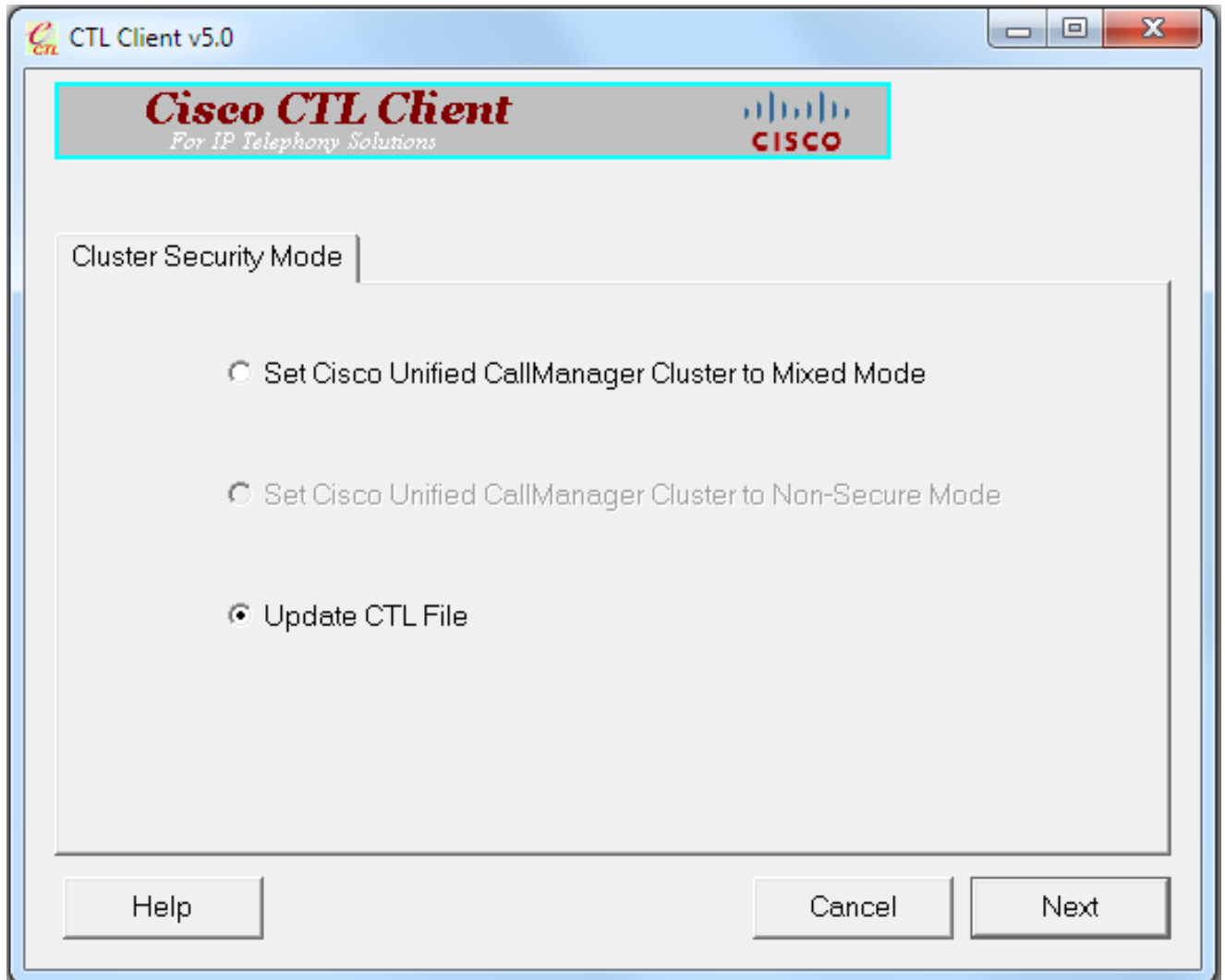
11. CTL 클라이언트를 실행하고 CUCM 게시자 노드의 IP 주소/호스트 이름을 제공한 다음 CCM 관리자 자격 증명을 입력합니다.



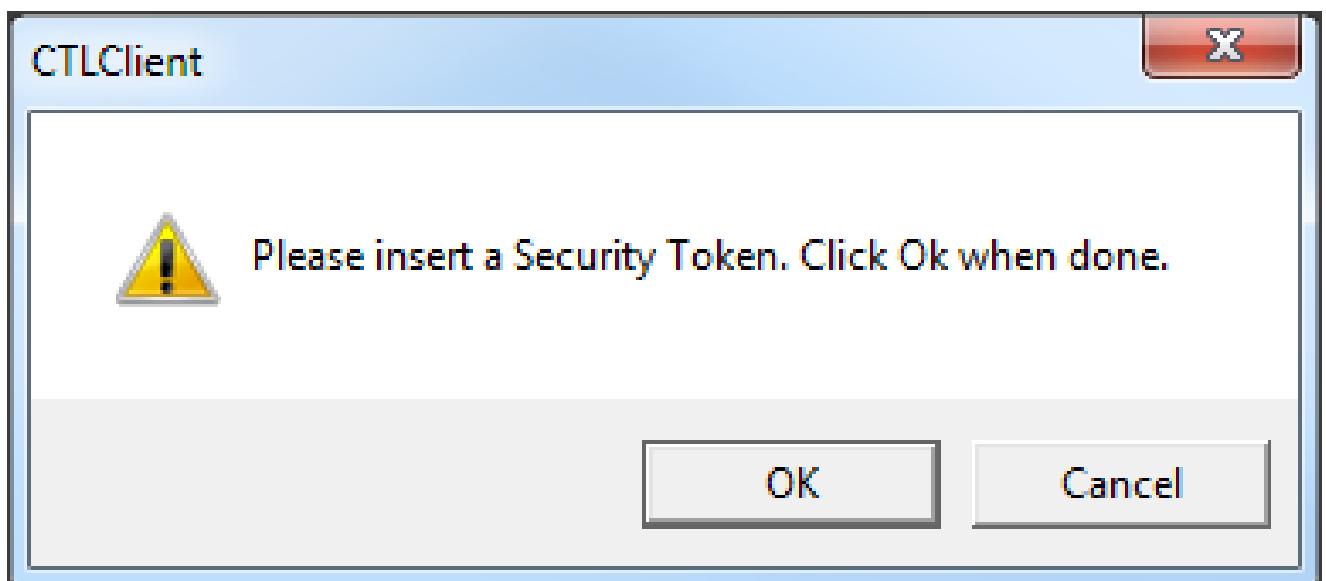
12. 클러스터가 이미 혼합 모드이지만 게시자 노드에 CTL 파일이 없으므로 이 경고 메시지가 나타납니다(무시하려면 OK를 클릭).

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.

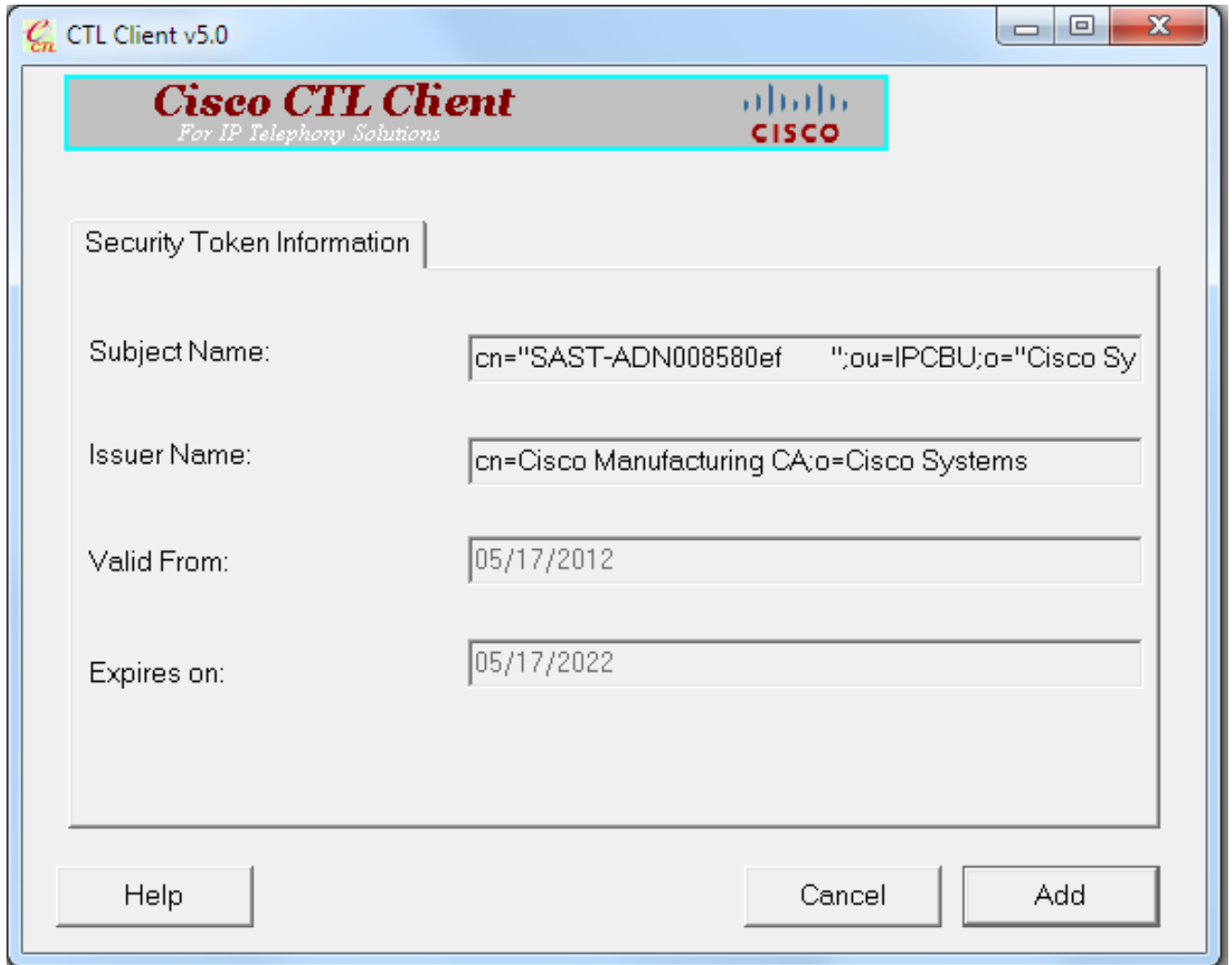
13. CTL 클라이언트에서 CTL 파일 업데이트 라디오 버튼을 클릭하고 다음 을 클릭합니다.



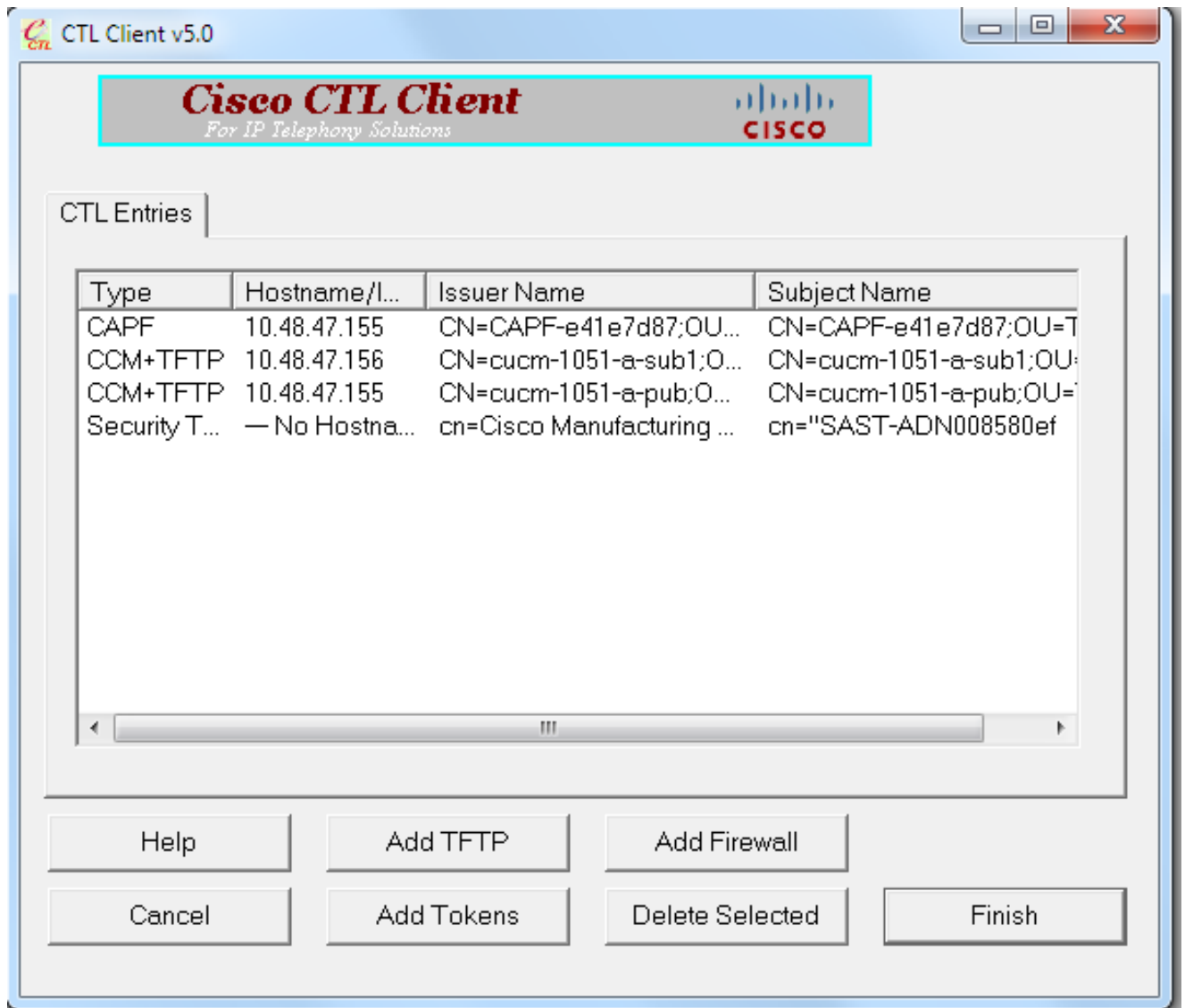
14. 첫 번째 보안 토큰을 삽입하고 OK(확인)를 클릭합니다.



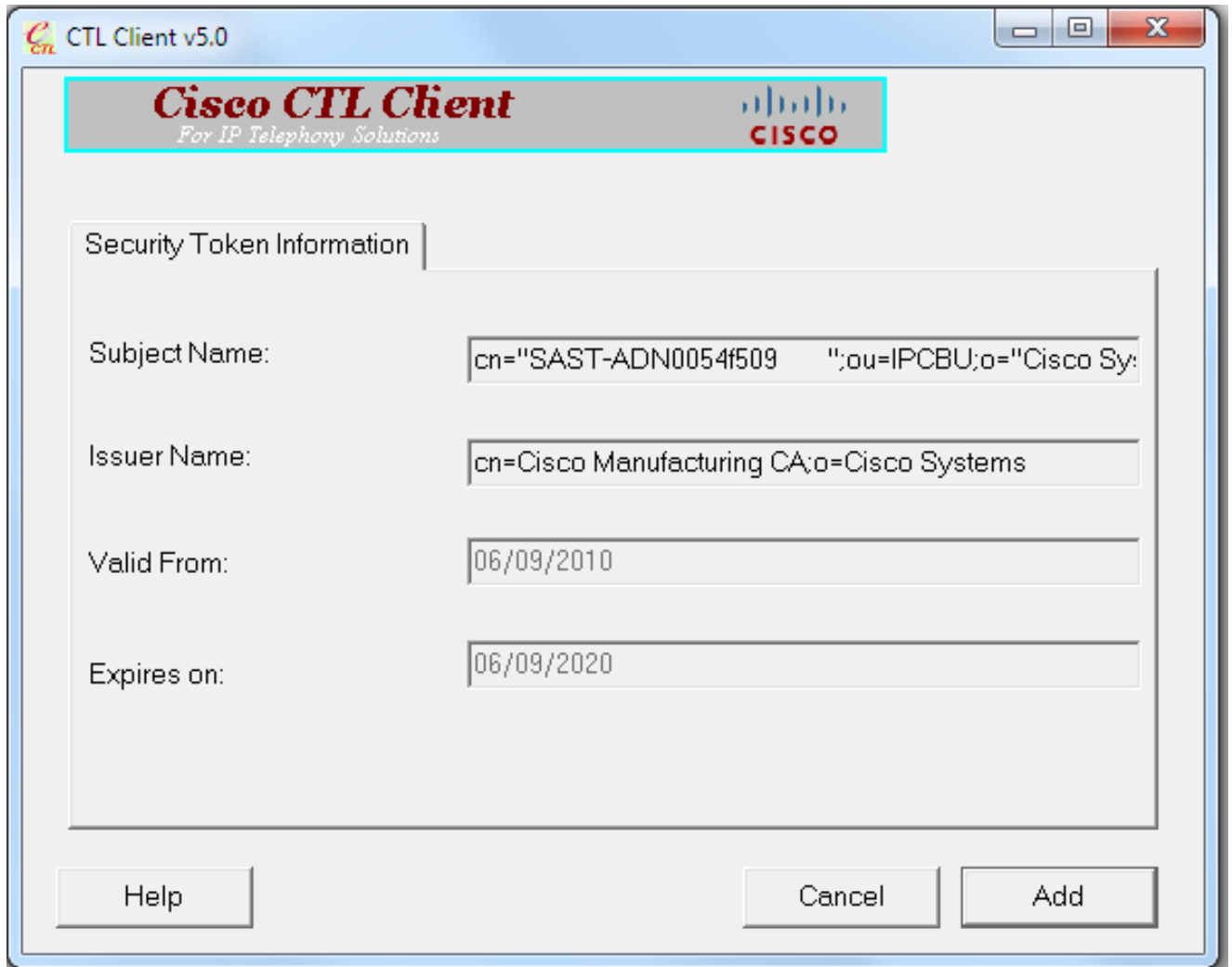
15. 보안 토큰 세부 정보가 표시되면 Add(추가)를 클릭합니다.



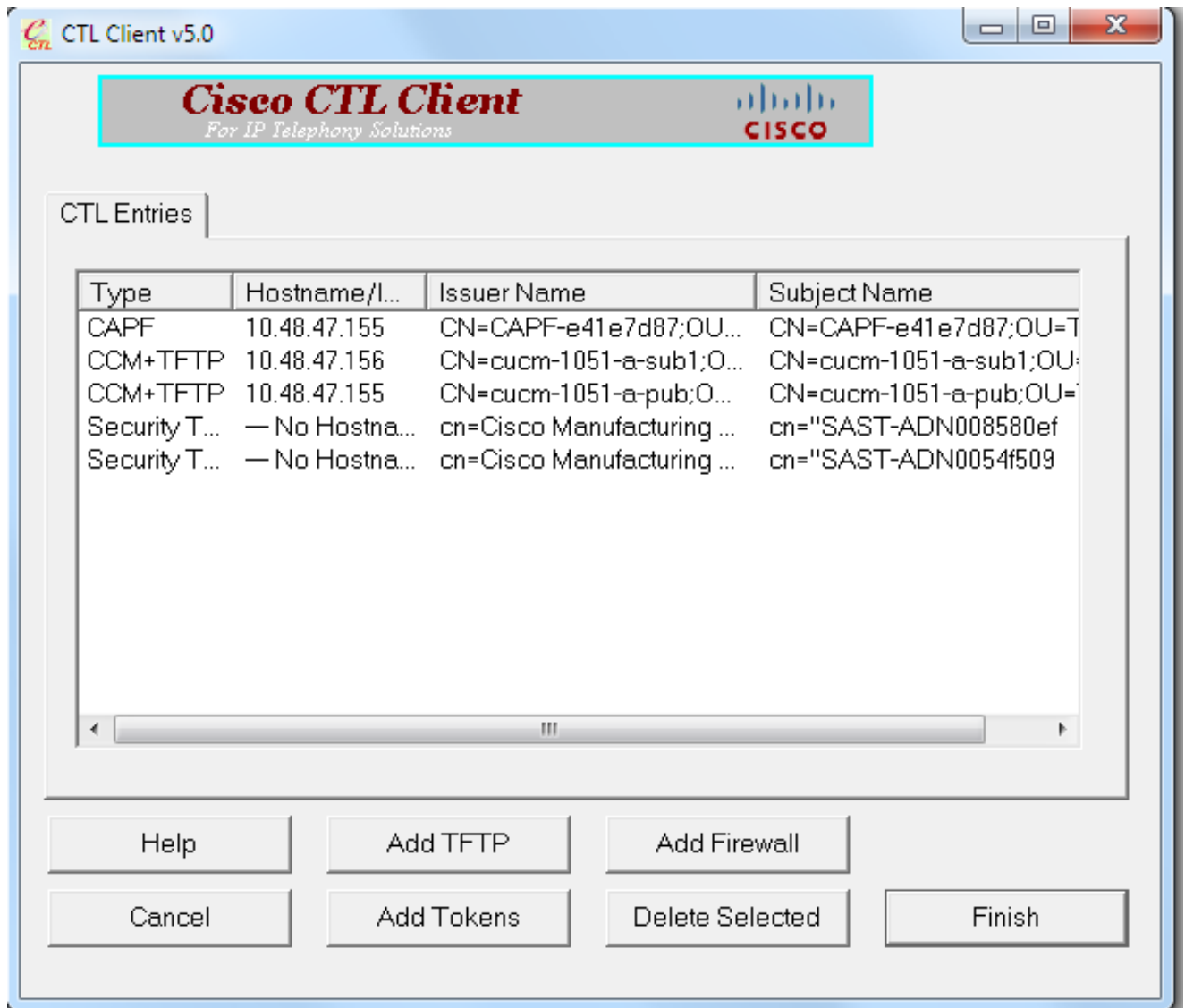
16. CTL 파일의 내용이 나타나면 두 번째 USB eToken을 추가하려면 Add Tokens를 클릭합니다.



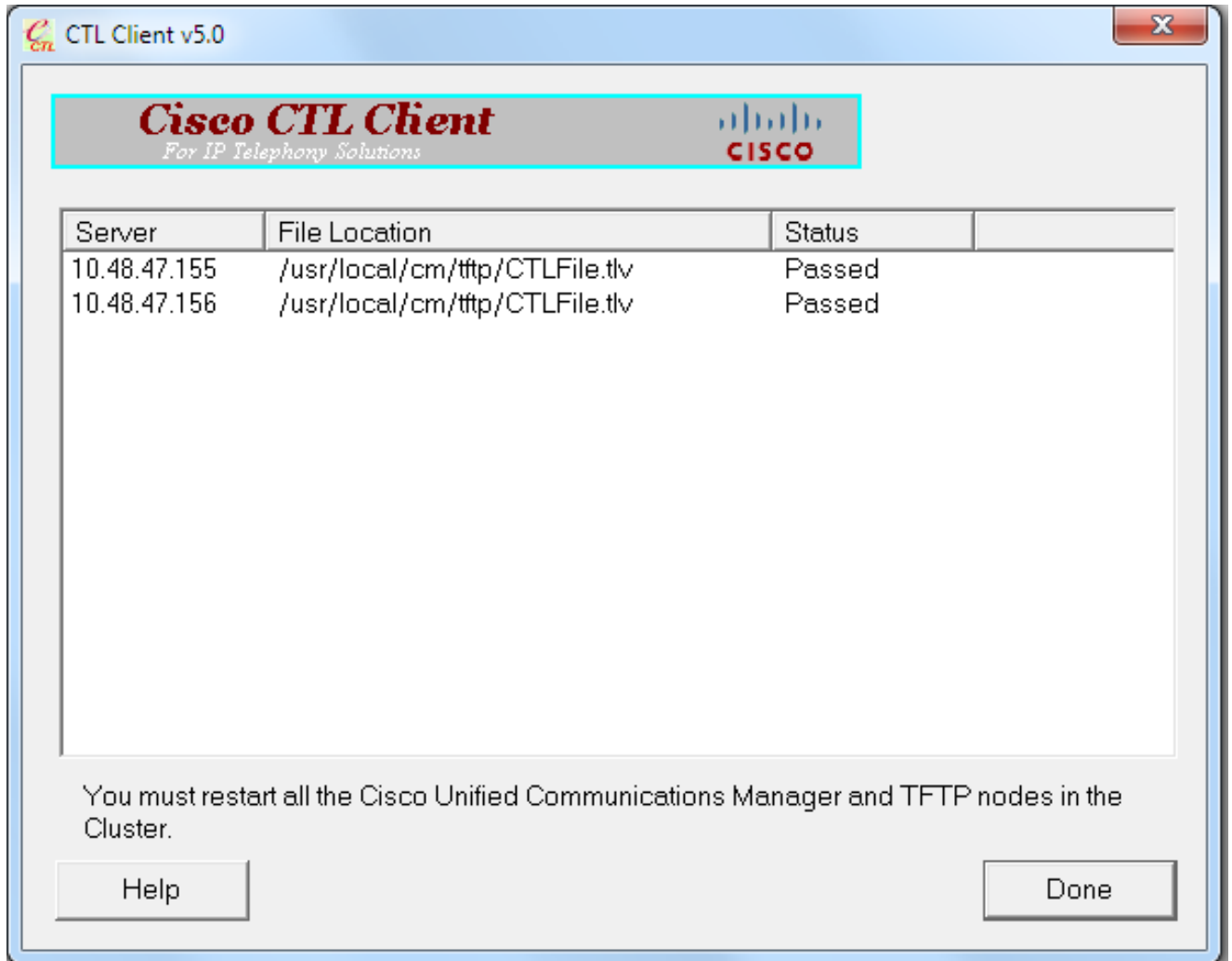
17. 보안 토큰 세부사항이 나타나면 Add(추가)를 클릭합니다.



18. CTL 파일의 내용이 나타나면 Finish(마침)를 클릭합니다. 비밀번호를 입력하라는 프롬프트가 표시되면 Cisco123을 입력합니다.



19. CTL 파일이 있는 CUCM 서버 목록이 나타나면 Done(완료)을 클릭합니다.



20. 이러한 서비스를 실행하는 클러스터의 모든 노드에서 TFTP 및 CallManager 서비스를 다시 시작합니다.
21. CUCM TFTP 서비스에서 새 버전의 CTL 파일을 가져올 수 있도록 모든 IP Phone을 다시 시작합니다.
22. CTL 파일의 내용을 확인하려면 CLI에 show ctl 명령을 입력합니다. CTL 파일에서 두 USB eToken의 인증서를 볼 수 있습니다(둘 중 하나는 CTL 파일에 서명하는 데 사용됨). 다음은 샘플 출력입니다.

```
<#root>
```

```
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	ISSUENAME	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	ISSUENAME	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. IP Phone 측에서 IP Phone이 다시 시작된 후 업데이트된 CTL 파일 버전을 다운로드했는지 확인할 수 있습니다(CUCM의 출력과 비교할 때 MD5 체크섬이 일치함).



이전에 eToken 인증서를 CUCM Certificate Trust Store로 내보내고 업로드했으며 IP Phone에서 CUCM에서 실행되는 TVS(Trust Verification Service)에 대해 CTL 파일을 서명하는 데 사용된 이 알 수 없는 인증서를 확인할 수 있으므로 이러한 변경이 가능합니다.

이 로그 스니펫은 IP Phone이 알 수 없는 eToken 인증서를 확인하는 요청과 함께 CUCM TVS에 연결하는 방법을 보여줍니다. 이 인증서는 Phone-SAST-trust로 업로드되며 신뢰할 수 있습니다.

```
<#root>
```

```
//
```

```
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: 83E908000005545AF31
23:00:22.052 | debug CertificateDBCACHE::getCertificateInformation - Looking up the
certificate cache using Unique MAP ID : 83E908000005545AF31cn=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 : 83E908000005545AF31cn=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
```

토크리스 CTL 솔루션의 인증서 재생성


이 섹션에서는 토크리스 CTL 솔루션이 사용될 때 CUCM 클러스터 보안 인증서를 재생성하는 방법에 대해 설명합니다.

CUCM 유지 관리 프로세스에서 CUCM 게시자 노드 CallManager 인증서가 변경되는 경우가 있습니다.

이러한 상황이 발생할 수 있는 시나리오에는 호스트 이름 변경, 도메인 변경 또는 인증서 만료 날짜 종료로 인한 인증서 재생성이 포함됩니다.

CTL 파일이 업데이트되면 IP Phone에 설치된 CTL 파일에 있는 인증서와 다른 인증서로 서명됩니다.

일반적으로 이 새 CTL 파일은 허용되지 않지만, IP Phone에서 CTL 파일을 서명하는 데 사용되는 알 수 없는 인증서를 찾은 후 CUCM의 TVS 서비스에 연결합니다.

 참고: TVS 서버 목록은 IP Phone 컨피그레이션 파일에 있으며 IP Phone Device Pool(IP Phone 디바이스 풀) > CallManager Group(CallManager 그룹)의 CUCM 서버에 매핑되어 있습니다.

TVS 서버에 대한 확인에 성공하면 IP Phone은 CTL 파일을 새 버전으로 업데이트합니다. 이러한 이벤트는 다음과 같은 시나리오에서 발생합니다.

1. CTL 파일은 CUCM 및 IP Phone에 있습니다. CCM 게시자 노드에 대한 CCM+TFT(서버) 인증서는 CTL 파일에 서명하는 데 사용됩니다.

```
<#root>
```

```
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      ISSUENAME     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=Małopolska;C=PL
4	FUNCTION	2	

CCM+TFTP

5	ISSUENAME	62	CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow; ST=Małopolska;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. CallManager.pem 파일(CCM+TFTP 인증서)이 다시 생성되고 인증서의 일련 번호가 변경되는 것을 확인할 수 있습니다.

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. CTL 파일을 업데이트하기 위해 `utils ctl update CTLFile` 명령을 CLI에 입력합니다.

```
<#root>
```

```
admin:
```

```
utils ctl update CTLFile
```

```
This operation updates 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. TVS 서비스는 새 CTL 파일 세부사항으로 인증서 캐시를 업데이트합니다.

```
<#root>
```


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 ::

6B1D357B6841740B078FEE4A1813D5D6

CN=

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 ::

6B1D357B6841740B078FEE4A1813D5D6

CN=

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. CTL 파일 내용을 볼 때 게시자 노드에 대한 새 CallManager 서버 인증서로 파일이 서명된 것을 확인할 수 있습니다.

<#root>

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

[...]

[...]

The CTL file was verified successfully.

6. Unified Serviceability(Unified 서비스 가용성) 페이지에서 TFTP 및 Cisco CallManager 서비스는 이러한 서비스를 실행하는 클러스터의 모든 노드에서 다시 시작됩니다.
7. IP Phone이 다시 시작되고 TVS 서버에 연결하여 현재 새 버전의 CTL 파일에 서명하는 데 사용되는 알 수 없는 인증서를 확인합니다.

```
<#root>
```

```
//
```

```
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
```

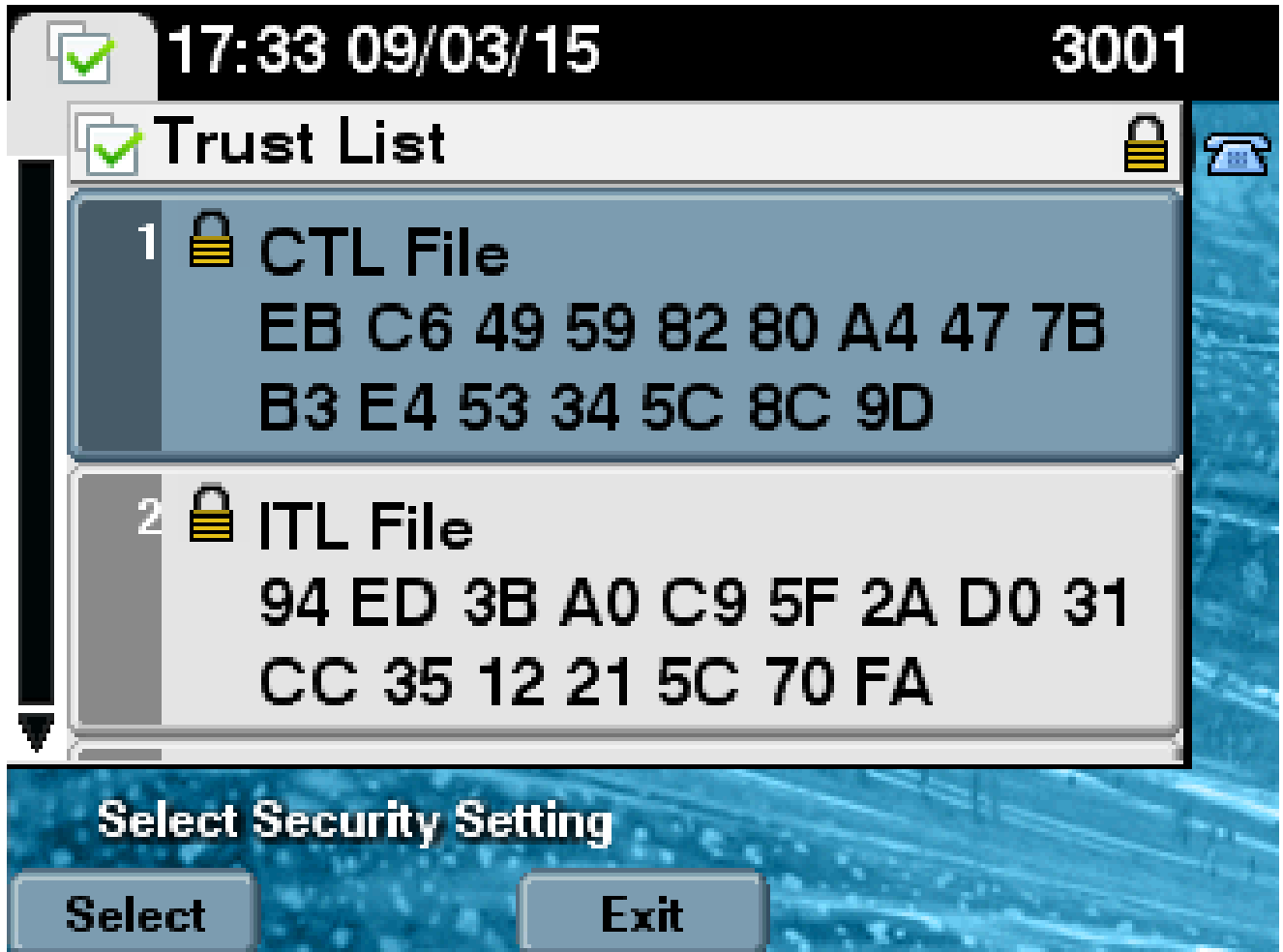
```
//
```

```
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. 마지막으로, IP Phone에서 CTL 파일이 새 버전으로 업데이트되고 새 CTL 파일의 MD5 체크섬이 CUCM의 체크섬과 일치하는지 확인할 수 있습니다.



이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.