CUCM Mixed Mode with Tokenless CTL

Contents

Introduction

This document describes differences between Cisco Unified Communications Manager (CUCM) security with / without the use of hardware USB eTokens.

Prerequisites

Requirements

Cisco recommends that you have knowledge of CUCM Version 10.0(1) or later. Additionally, ensure that:

- Your license server for CUCM version 11.5.1SU3 and higher must be Cisco Prime License Manager (PLM) 11.5.1SU2 or higher.

This is because CUCM version 11.5.1SU3 requires the Encryption License to enable mixed mode and PLM does not support the Encryption License until 11.5.1SU2.

For more information reference the Release Notes for Cisco Prime License Manager, Release 11.5(1)SU2.

- You have Administrative access to the Command Line Interface (CLI) of the CUCM Publisher node.
- You have access to the hardware USB eTokens and that the CTL Client Plugin is installed on your PC for scenarios that require you to migrate back to the use of hardware eTokens.

For additional clarity, this requirement is only if you, at any point, have a scenario where the USB eTokens are needed. The chances are very small that USB eTokens are needed for most people.

- There is full connectivity between all of the CUCM nodes in the cluster. This is very important because the CTL file is copied to all of the nodes in the cluster via SSH File Transfer Protocol (SFTP).
- The Database (DB) Replication in the cluster works properly and that the servers replicate the data in real-time.
- The devices in your deployment support Security by Default (TVS).

You can use the *Unified CM Phone Feature List* from the Cisco Unified Reporting webpage (https://<CUCM IP or FQDN>/cucreports/) in order to determine the devices that support Security by Default.

Note: Cisco Jabber and many Cisco TelePresence or Cisco 7940/7960 Series IP phones do not currently support Security by Default. If you deploy Tokenless CTL with devices that do not support Security by Default, any update to your system that changes the CallManager certificate on the publisher then prevents normal functionality of those devices until the CTL is manually deleted. Devices that support Security by Default, such as 7945 and 7965 phones or newer, are able to install CTL files when the CallManager certificate on the publisher is updated because they can use the Trust Verification Service (TVS).

Components Used

The information in this document is based on these software and hardware versions:

- CUCM Version 10.5.1.10000-7 (cluster of two nodes)
- Cisco 7975 Series IP Phones registered via Skinny Client Control Protocol (SCCP) with Firmware Version SCCP75.9-3-1SR4-1S
- Two Cisco Security Tokens that are used in order to set the cluster to Mixed mode with the use of CTL Client software

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

This document describes the difference between Cisco Unified Communications Manager (CUCM) security with and without the use of hardware USB eTokens.

This document also describes the basic implementation scenarios that involve Tokenless Certificate Trust List (CTL) and the process that is used in order to ensure that the system functions properly after the changes.

Tokenless CTL is a new feature in CUCM Versions 10.0(1) and later that allows the encryption of call signaling and media for IP Phones without the need to use hardware USB eTokens and the CTL Client plugin, which was the requirement in previous CUCM releases.

When the cluster is placed into Mixed mode with the use of the CLI command, the CTL file is signed with the CCM+TFTP (server) certificate of the Publisher node, and there are no eToken certificates present in the CTL file.

Note: When you regenerate the CallManager (CCM+TFTP) certificate on the publisher, it changes the signer of the file. The phones and devices that do not support Security by Default also do not accept the new CTL file unless CTL files are manually deleted from each device. Refer to the last requirement that is listed the Requirements section of this document for more information.

From Non-Secure Mode to Mixed Mode (Tokenless CTL)

This section describes the process that is used in order to move the CUCM cluster security into Mixed mode via the CLI.

Prior to this scenario, the CUCM was in Non-Secure mode, which means that there was no CTL file present on any of the nodes and that the registered IP Phones had only an Identity Trust List (ITL) file installed, as shown in these outputs:

<#root>

```
admin:
```

show ctl

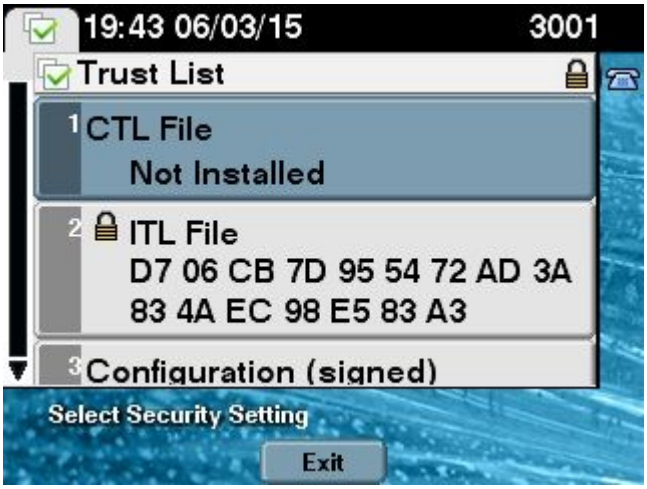
```
Length of CTL file: 0
```

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:
```

> **Note**: If there is a CTL file found on the server while the cluster is not in mixed mode, this means the cluster was once in mixed mode then moved back to non-mixed mode and the CTL file was not deleted from the cluster.
>
> The command **file delete activelog cm/tftpdata/CTLFile.tlv** deletes the CTL file from nodes in the CUCM cluster; however, the command needs to be entered on each node. To be clear, only use this command if your servers have a CTL file and the cluster is not in mixed mode.
>
> An easy way to confirm if a cluster is in mixed mode is to use the command **run sql select paramname,paramvalue from processconfig where paramname='ClusterSecurityMode'**. If the param value is 0, then the cluster is not in mixed mode.

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



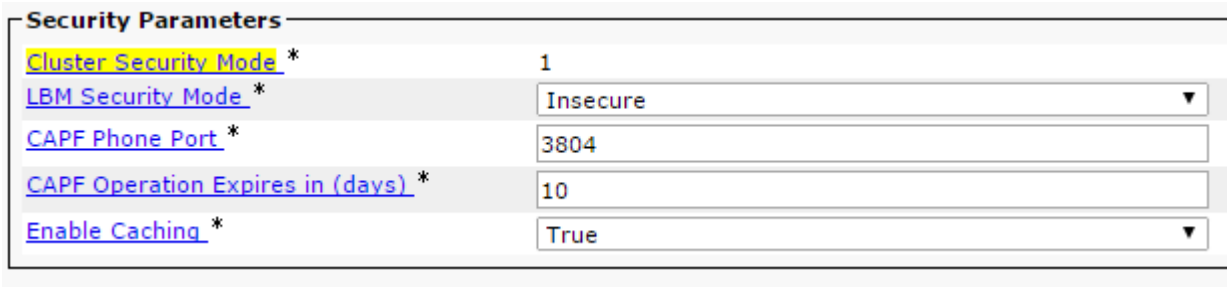
In order to move the CUCM cluster security into Mixed mode with the use of the new Tokenless CTL feature, complete these steps:

1. Obtain Administrative access to the CUCM Publisher node CLI.

2. Enter the **utils ctl set-cluster mixed-mode** command into the CLI:

```
<#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. Navigate to **CUCM Admin Page > System > Enterprise Parameters** and verify whether the cluster was set to Mixed mode (a value of **1** indicates Mixed mode):



4. Restart the TFTP and Cisco CallManager services on all of the nodes in the cluster that run these services.

5. Restart all of the IP Phones so that they can obtain the CTL file from the CUCM TFTP service.

6. In order to verify the content of the CTL file, enter the **show ctl** command into the CLI.
7. In the CTL file you can see that the CCM+TFTP (server) certificate for the CUCM Publisher node is used in order to sign the CTL file (this file is the same on all servers in the cluster). Here is a sample output:

```
<#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	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

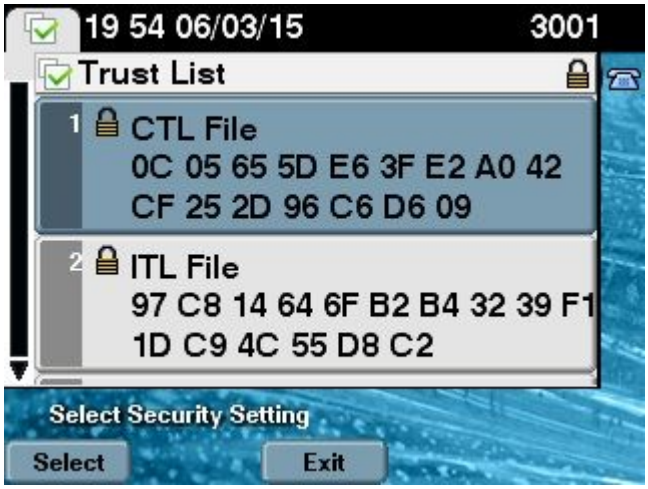
7	PUBLICKEY	140	
8	SIGNATURE	128	
9	CERTIFICATE	694	E9 D4 33 64 5B C8 8C ED 51 4D 8F E5 EA 5B 6D 21 A5 A3 8C 9C (SHA1 Hash HEX)
10	IPADDRESS	4	

[...]

```
The CTL file was verified successfully.
```

8. On the IP Phone side, you can verify that after the service is restarted, it downloads the CTL file, which is now present on the TFTP server (the MD5 checksum matches when compared to the output from the CUCM):

Note: When you verify the checksum on the phone, you see either **MD5** or **SHA1**, dependent upon the phone type.



From Hardware eTokens to Tokenless Solution

This section describes how to migrate the CUCM cluster security from hardware eTokens to the use of the new Tokenless solution.

In some situations, Mixed mode is already configured on the CUCM with the use of the CTL Client, and the IP Phones use CTL files that contain the certificates from the hardware USB eTokens.

With this scenario, the CTL file is signed by a certificate from one of the USB eTokens and is installed on the IP Phones. Here in an example:

<#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       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.
```

Complete these steps in order to move the CUCM cluster security to the use of Tokenless CTLs:

1. Obtain Administrative access to the CUCM Publisher node CLI.
2. Enter the **utils ctl update CTLFile** CLI command:

```
<#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. Restart the TFTP and CallManager services on all of the nodes in the cluster that run these services.

4. Restart all of the IP Phones so that they can obtain the CTL file from the CUCM TFTP service.

5. Enter the **show ctl** command into the CLI in order to verify the content of the CTL file. In the CTL file, you can see that the CCM+TFTP (server) certificate of the CUCM Publisher node is used in order to sign the CTL file instead of the certificate from the hardware USB eTokens.
6. One more important difference in this case is that the certificates from all of the hardware USB eTokens are removed from the CTL file. Here is a sample output:

<#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=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.
```

Note: In the above output,If CCM+TFTP (server) certificate of the CUCM Publisher is not signer, then move back to Hardware etoken based cluster security mode and repeat the changes again for tokenless solution.

---

7. On the IP Phone side, you can verify that after the IP Phones were restarted, they downloaded the updated CTL file version (the MD5 checksum matches when compared to the output from the CUCM):

From Tokenless Solution to Hardware eTokens

This section describes how to migrate the CUCM cluster security away from the new Tokenless solution and back to the use of hardware eTokens.

When the CUCM cluster security is set to Mixed mode with the use of the CLI commands, and the CTL file is signed with the CCM+TFTP (server) certificate for the CUCM Publisher node, there are no certificates from the hardware USB eTokens present in the CTL file.

For this reason, when you run the CTL Client in order to update the CTL file (move back to the use of hardware eTokens), this error message appears:

```
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.
```

This is particularly important in scenarios that include a downgrade (when the version is switched back) of the system to a pre-10.x version that does not include the **utils ctl** commands.

The previous CTL file is migrated (without changes in its content) in the process of a refresh or a Linux to Linux (L2) upgrade, and it does not contain the eToken certificates, as previously mentioned. Here is a sample output:

<#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=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:
```

For this scenario, complete these steps in order to securely update the CTL files without the need to use the procedure for lost eTokens, which ends up in manual deletion of the CTL file from all of the IP Phones:

1. Obtain Administrative access to the CUCM Publisher node CLI.

2. Enter the **file delete tftp CTLFile.tlv** command into the Publisher node CLI in order to delete the CTL file:

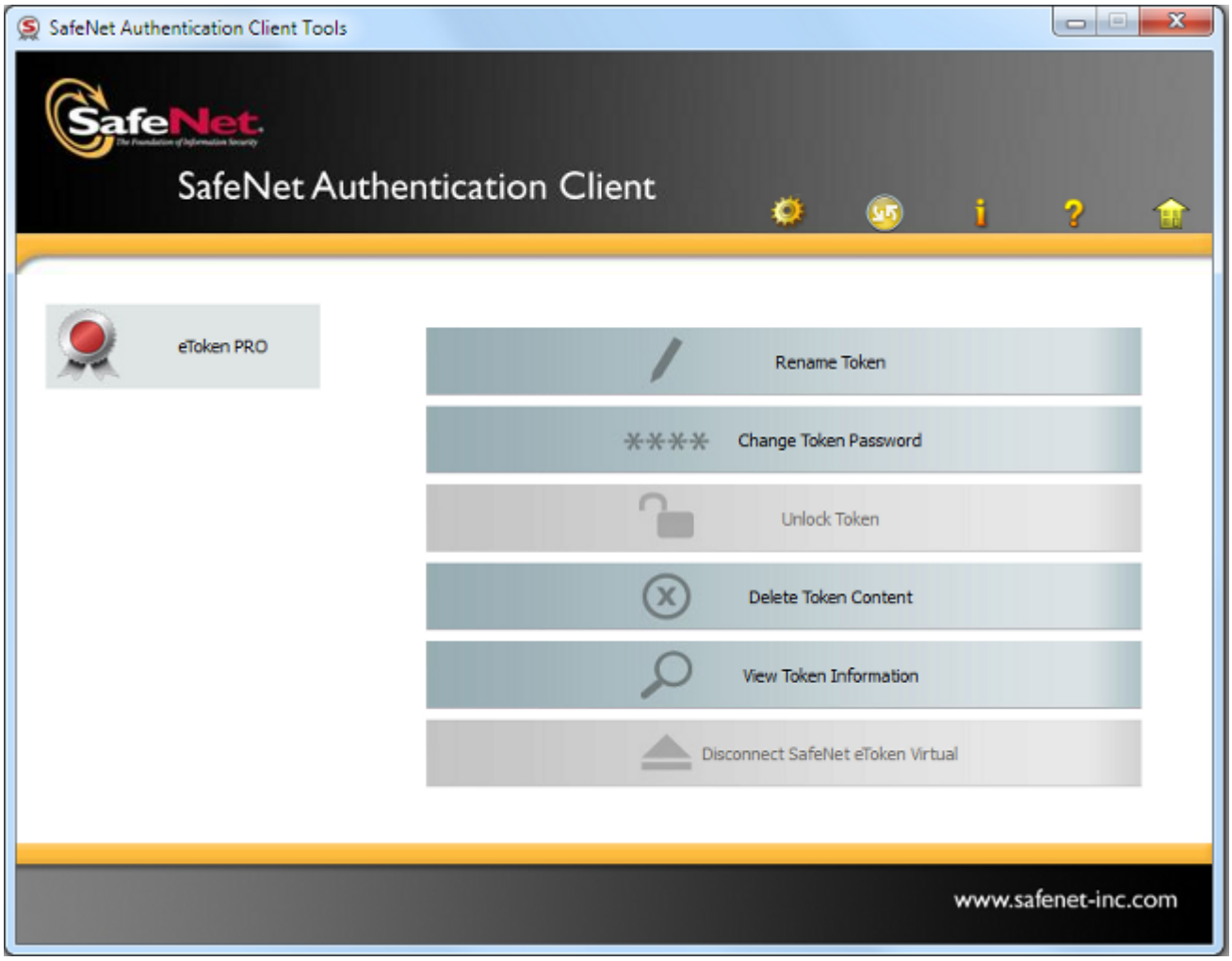
<#root>

```
admin:
```

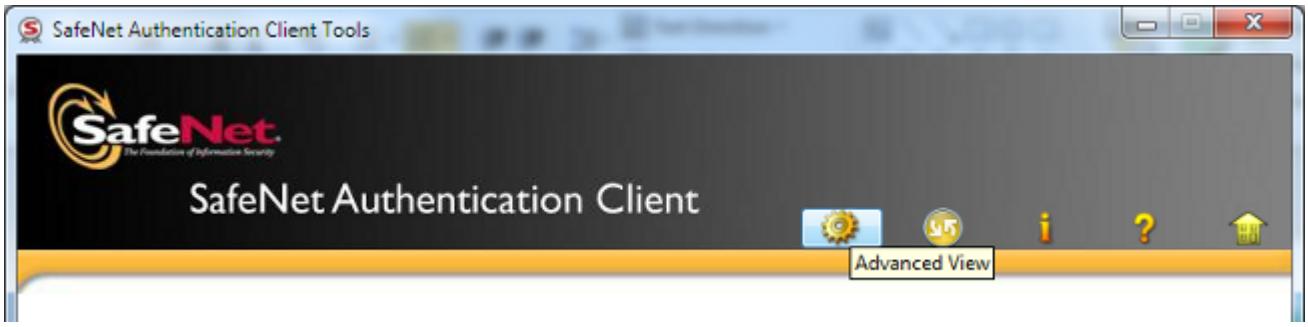
`file delete tftp CTLFile.tlv`

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

3. Open **SafeNet Authentication Client** on the Microsoft Windows machine that has the CTL Client installed (it is installed automatically with CTL Client):

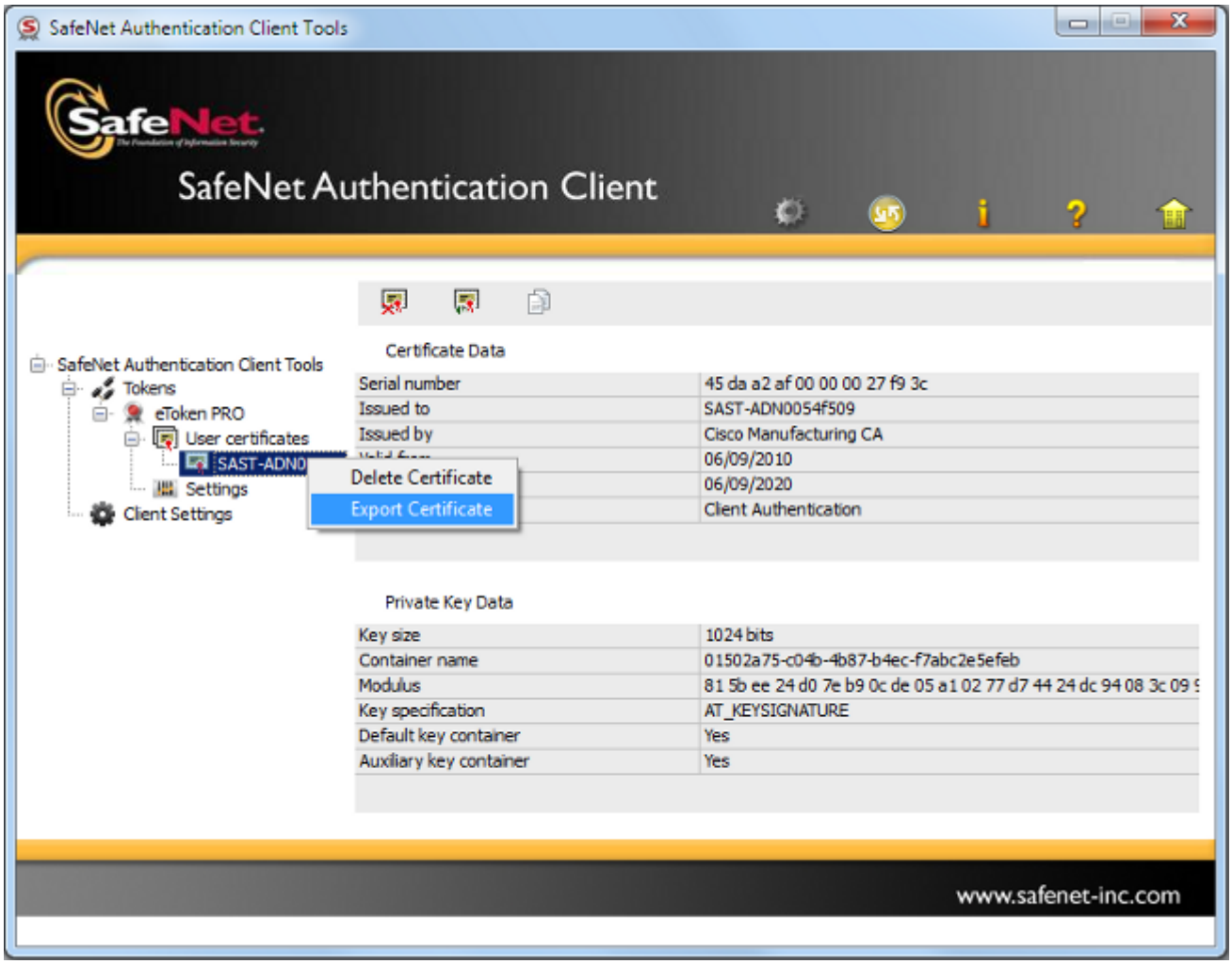


4. In SafeNet Authentication Client, navigate to the *Advanced View*:

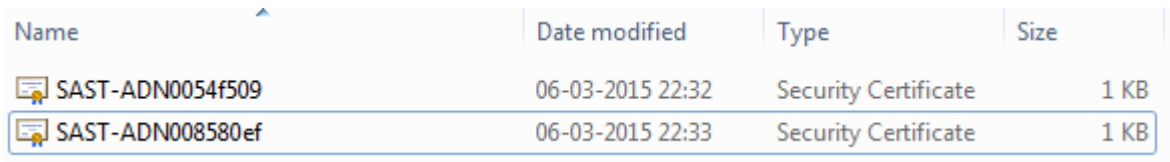


5. Insert the first hardware USB eToken.

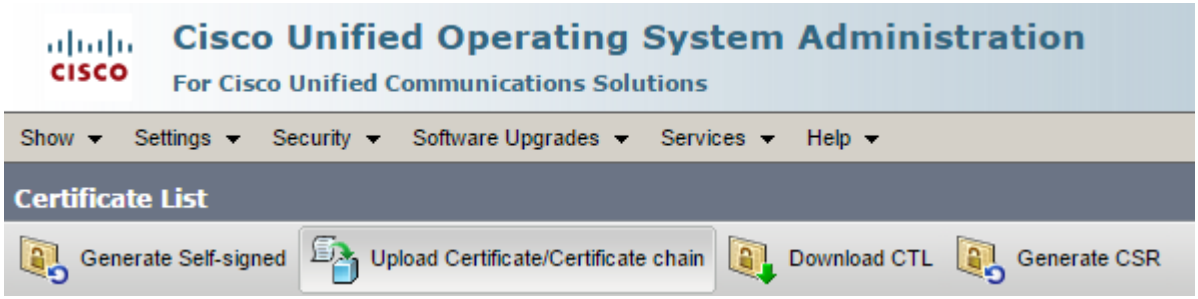
6. Select the certificate under the *User certificates* folder and export it to the folder on the PC. When prompted for a password, use the default password of **Cisco123**:



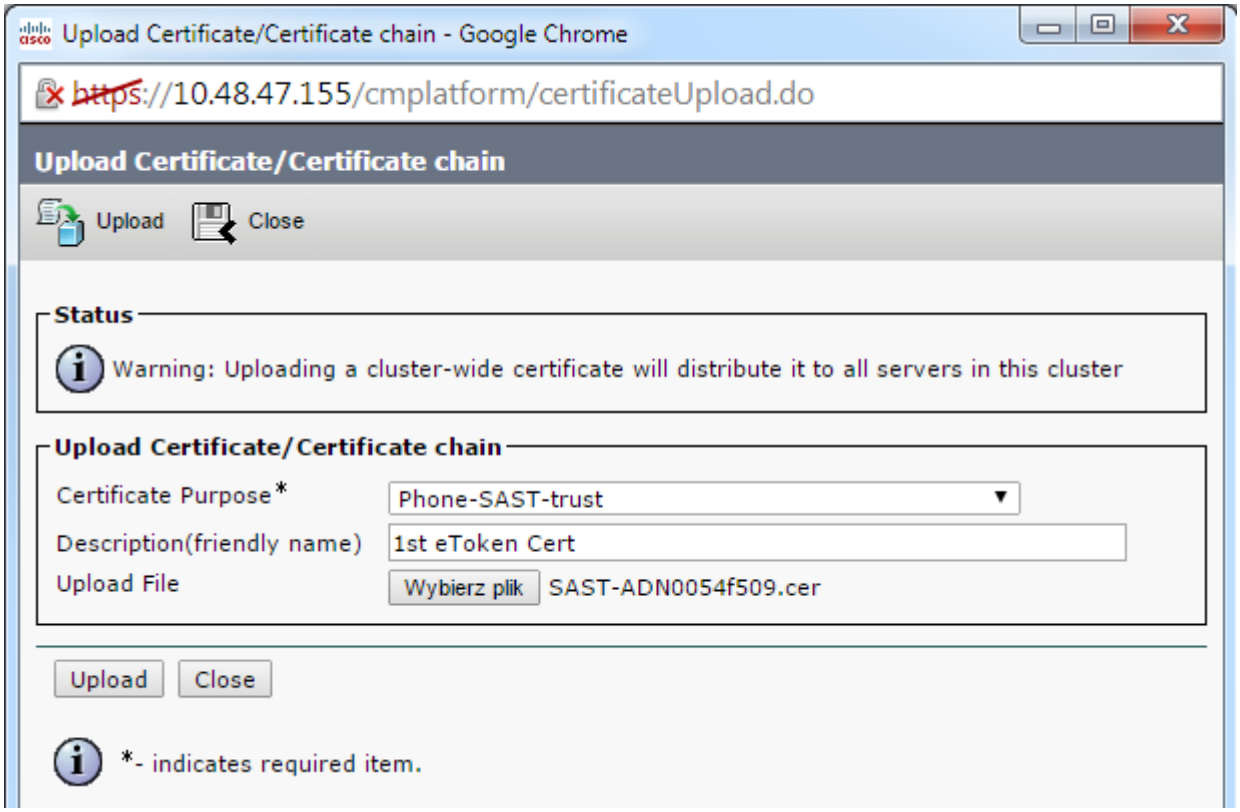
7. Repeat these steps for the second hardware USB eToken so that both certificates are exported to the PC:



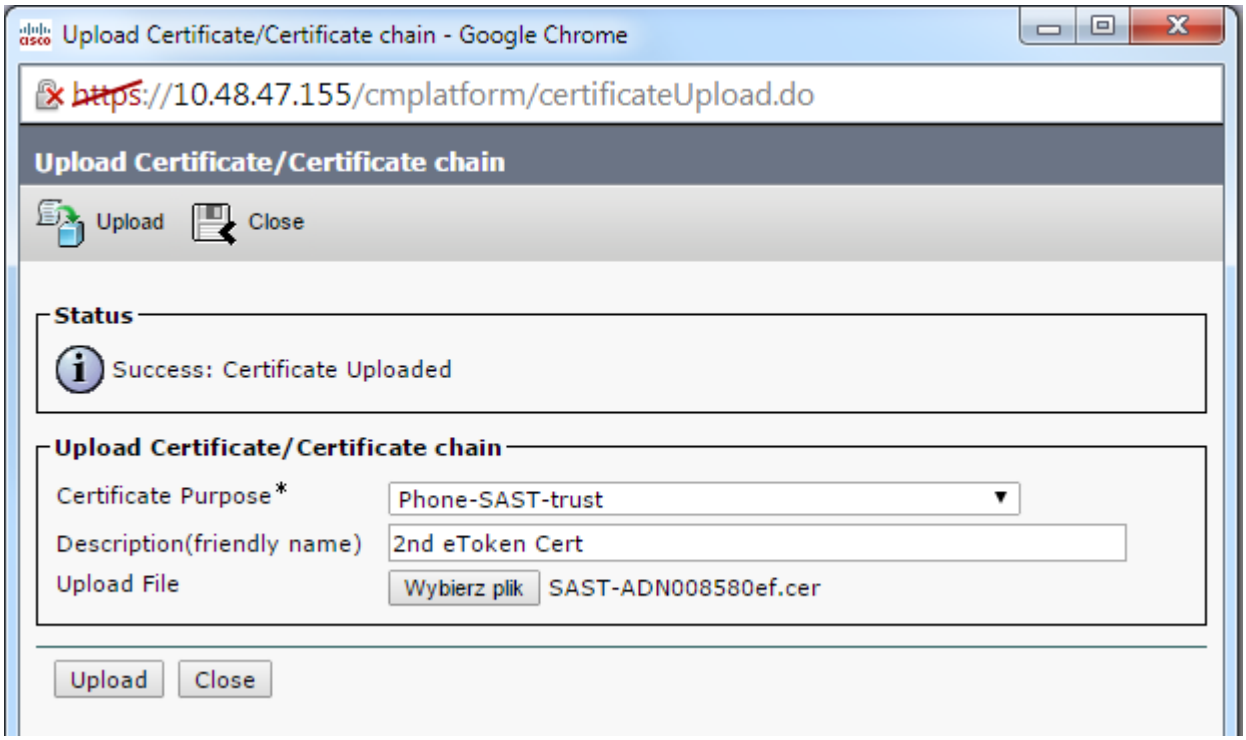
8. Log into the Cisco Unified Operating System (OS) Administration and navigate to **Security > Certificate Management > Upload Certificate**:



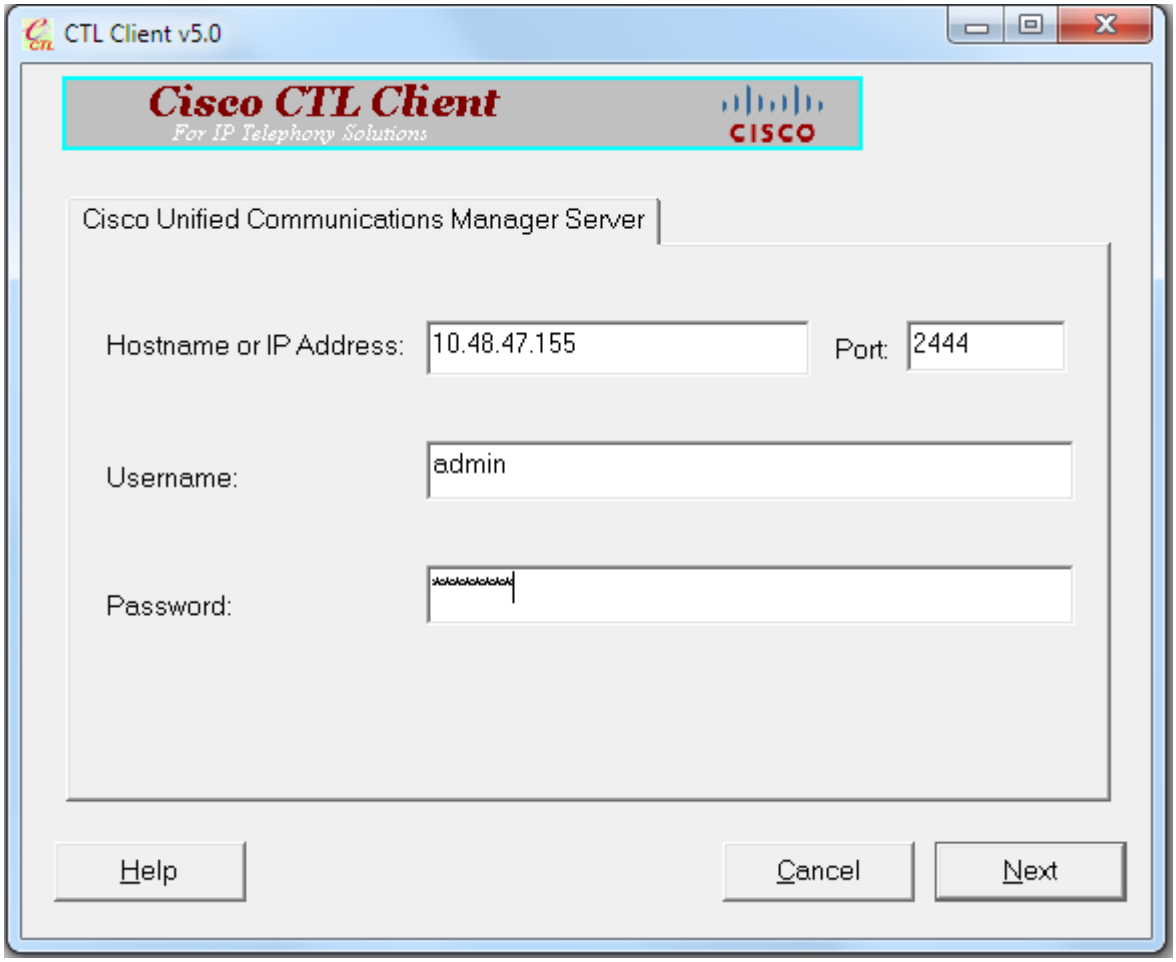
9. The Upload Certificate page then appears. Choose **Phone-SAST-trust** from the Certificate Purpose drop down menu and select the certificate that you exported from the first eToken:



10. Complete the previous steps in order to upload the certificate that you exported from the second eToken:



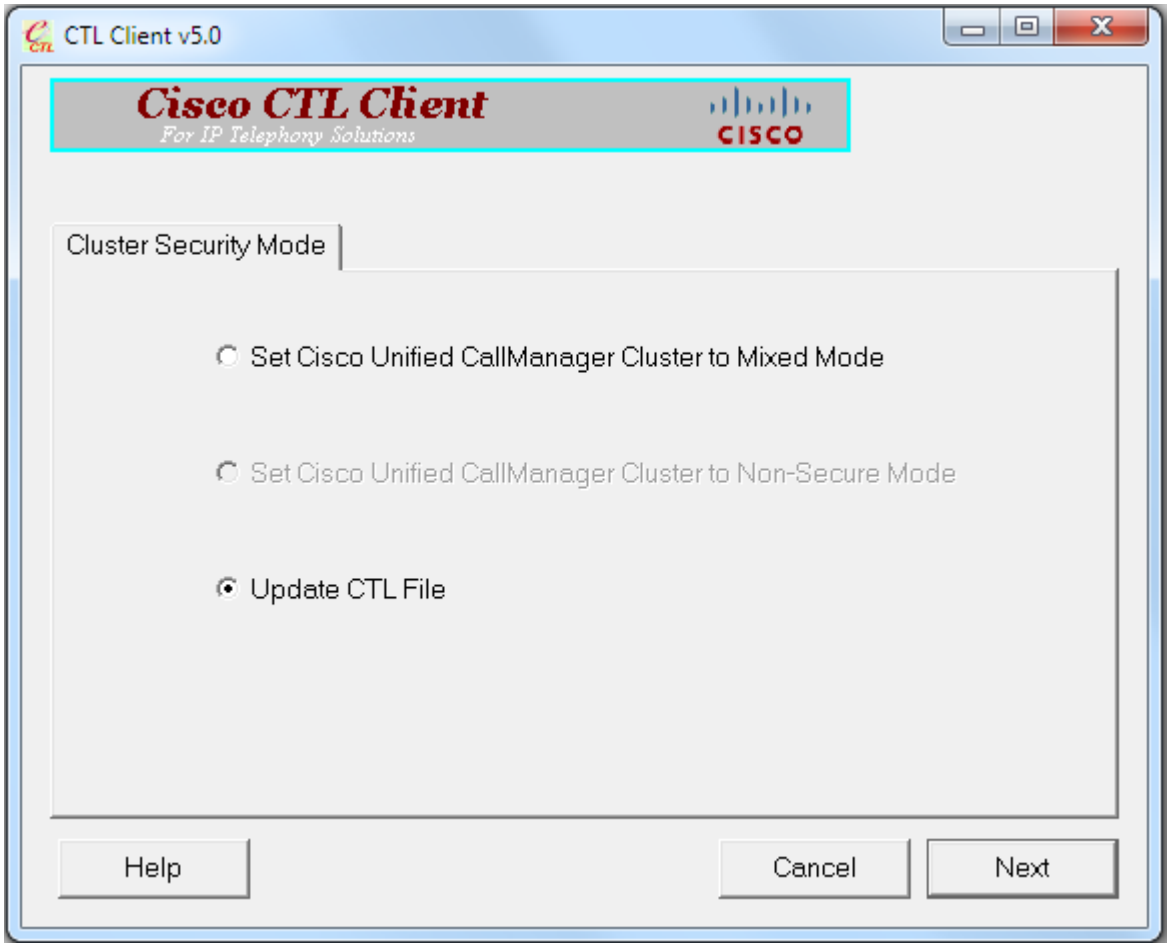
11. Run the CTL Client, provide the IP address/hostname of the CUCM Publisher node, and enter the CCM Administrator credentials:



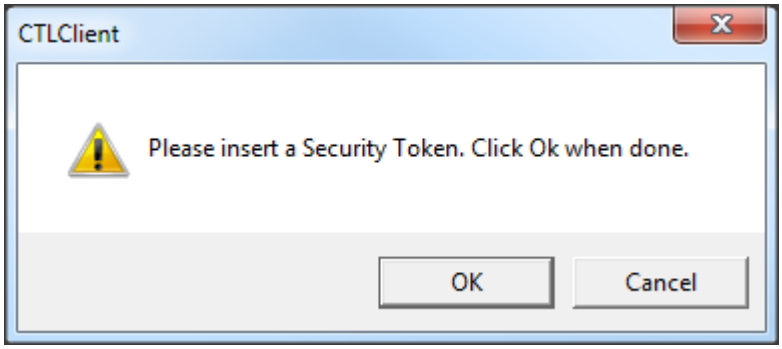
12. Since the cluster is in Mixed mode already, but no CTL file exists on the Publisher node, this warning message appears (click **OK** in order to ignore it):

```
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.
```

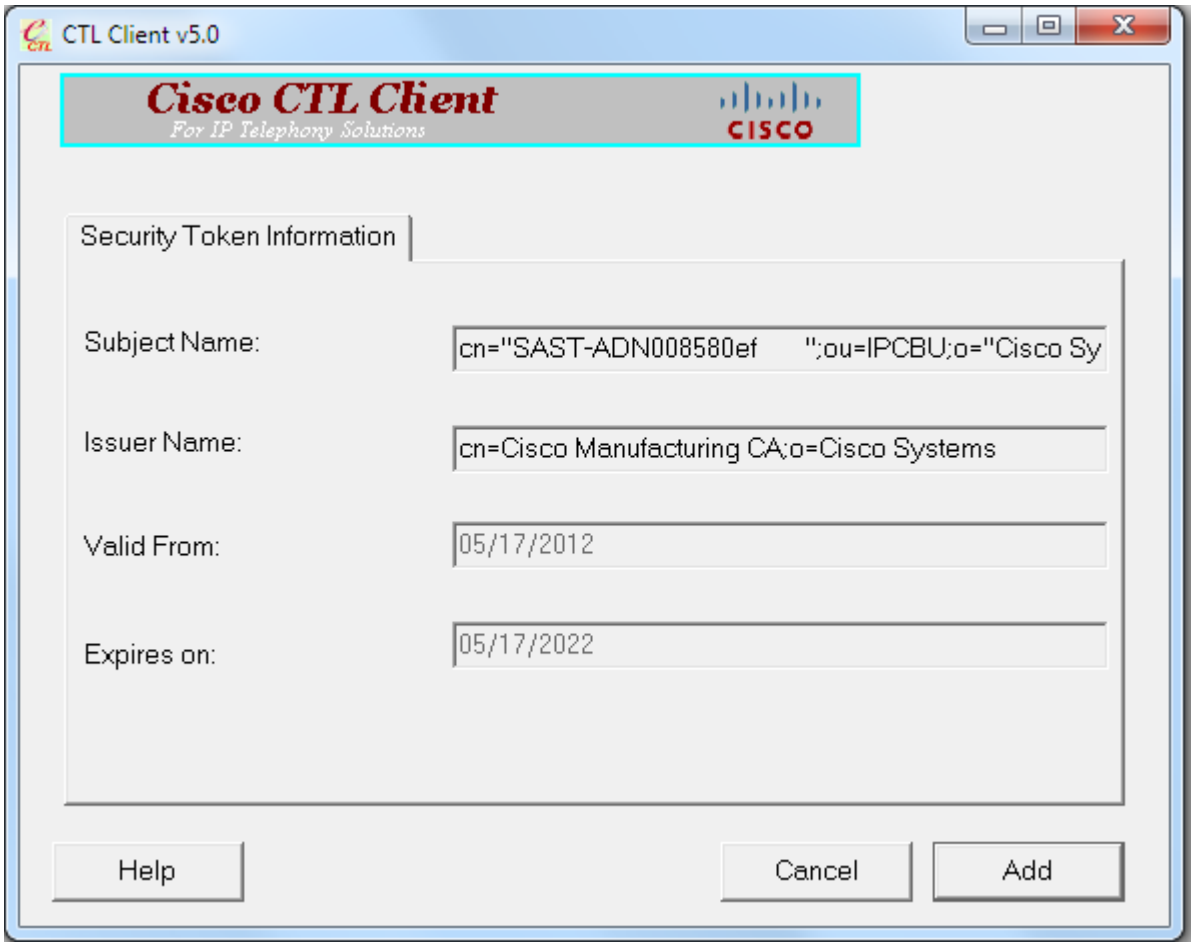
13. From the CTL Client, click the **Update CTL File** radio button, and then click **Next**:



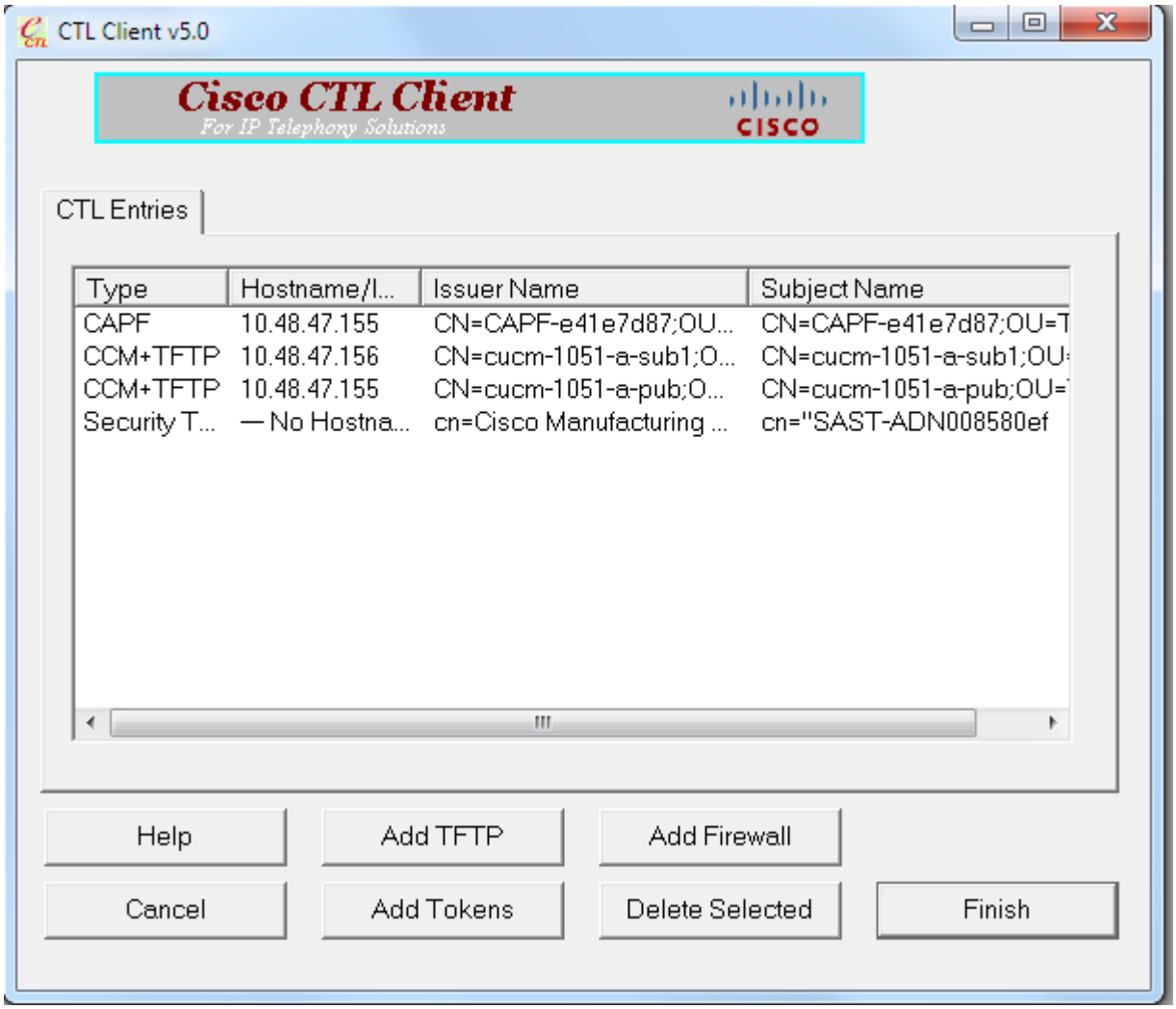
14. Insert the first security token and click **OK**:



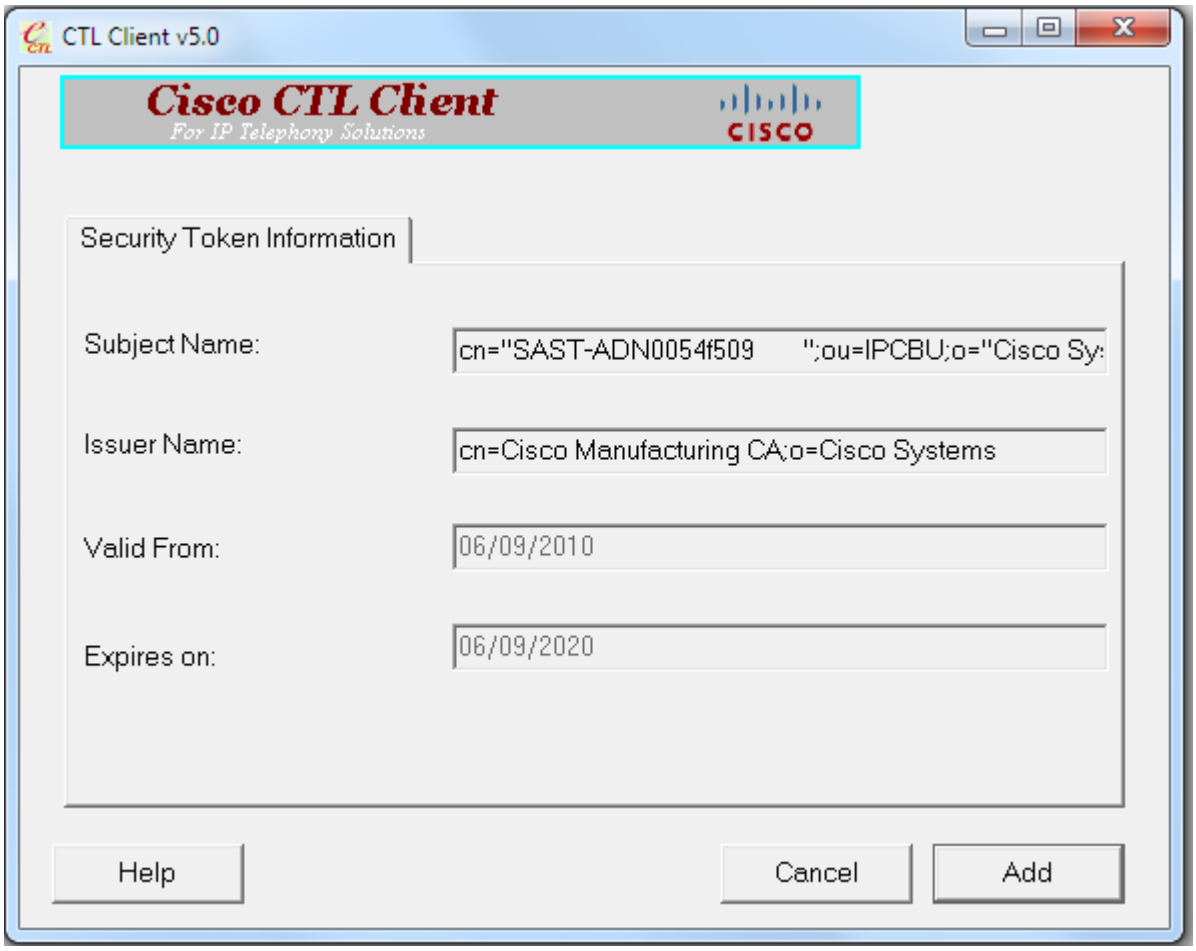
15. After the security Token details are displayed, click **Add**:



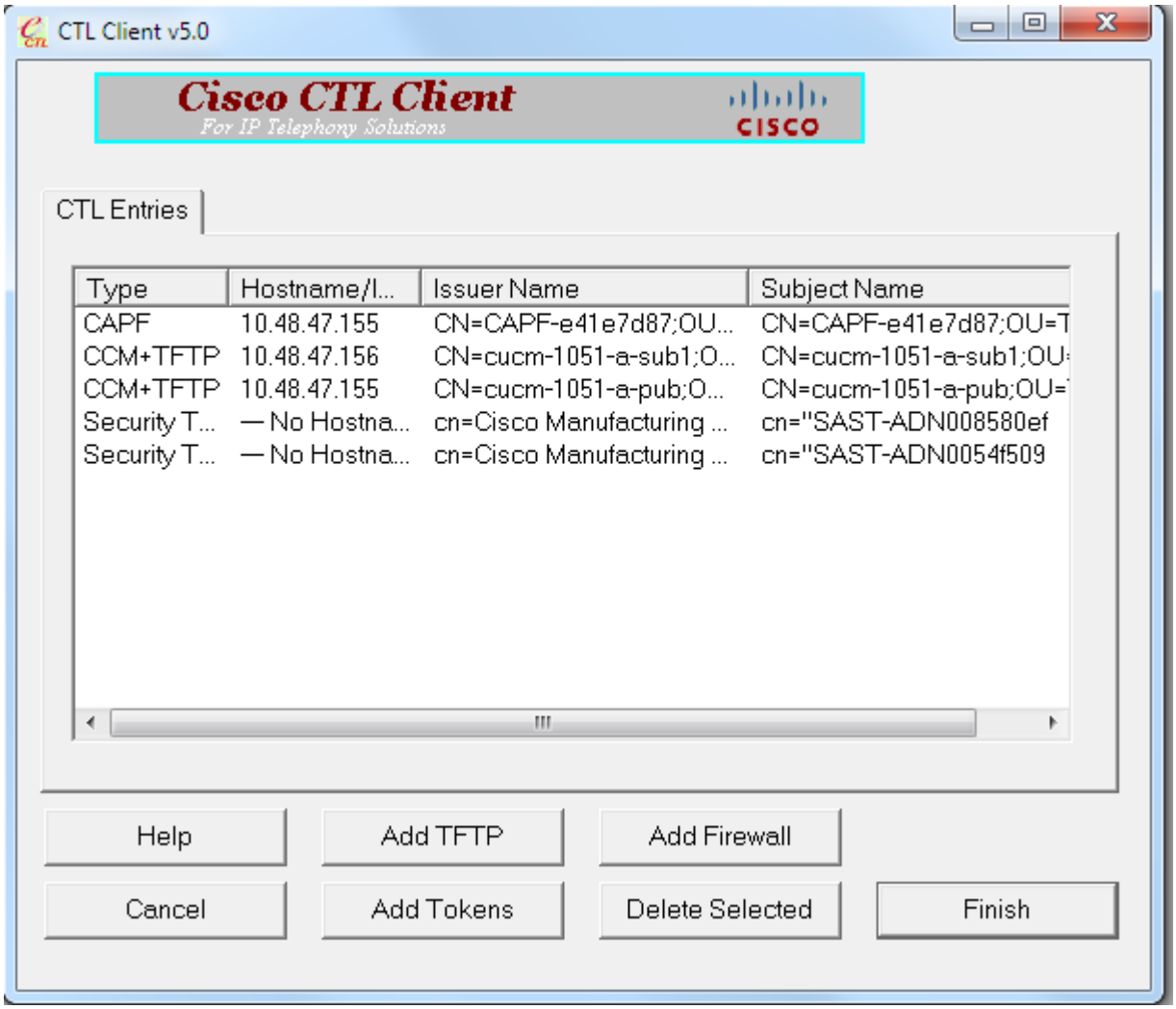
16. Once the content of the CTL file appears, click **Add Tokens** in order to add the second USB eToken:



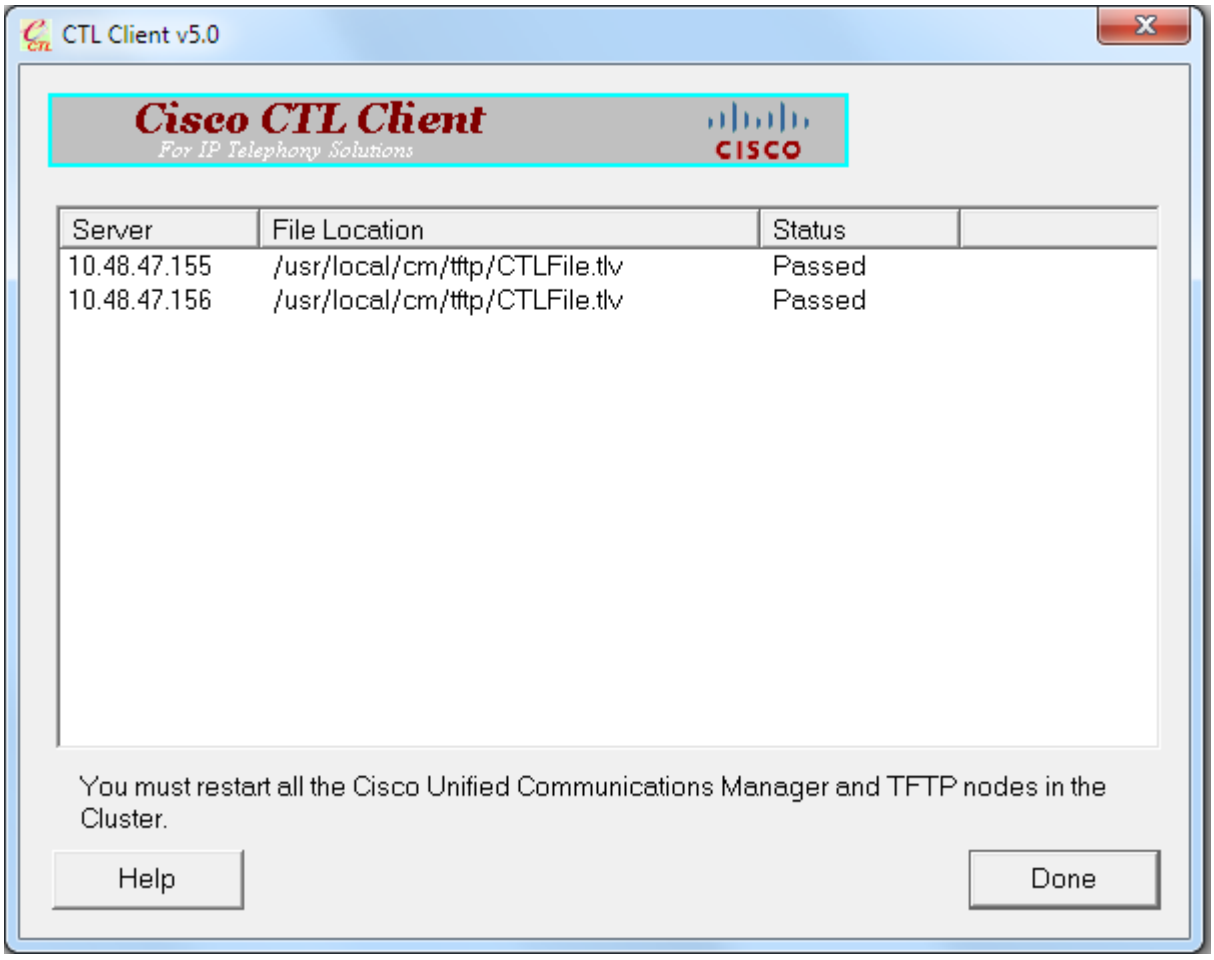
17. After the security Token details appear, click **Add**:



18. After the content of the CTL file appears, click **Finish**. When prompted for a password, enter **Cisco123**:



19. When the list of CUCM Servers on which the CTL file exists appears, click **Done**:



20. Restart the TFTP and CallManager services on all of the nodes in the cluster that run these services.

21. Restart all of the IP Phones so that they can obtain the new version of the CTL file from the CUCM TFTP service.

22. In order to verify the content of the CTL file, enter the **show ctl** command into the CLI. In the CTL file you can see the certificates from both of the USB eTokens (one of them is used in order to sign the CTL file). Here is a sample output:

```
<#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       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. On the IP Phone side, you can verify that after the IP Phones were restarted, they downloaded the updated CTL file version (the MD5 checksum matches when compared to the output from the CUCM):

This change is possible because you previously exported and uploaded the eToken certificates to the CUCM Certificate Trust Store, and the IP Phones are able to verify this unknown certificate that was used in order to sign the CTL file against the Trust Verification Service (TVS) that runs on the CUCM.

This log snippit illustrates how the IP Phone contacts the CUCM TVS with a request to verify the unknown eToken certificate, which is uploaded as **Phone-SAST-trust** and is trusted:

<#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: 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
```

Certificate Regeneration for Tokenless CTL Solution

This section describes how to regenerate a CUCM cluster security certificate when the Tokenless CTL solution is used.

In the process of CUCM maintenance, sometimes the CUCM Publisher node CallManager certificate changes.

The scenarios in which this can happen include the change of hostname, the change of domain, or simply a certificate regeneration (due to close certificate expiration date).

After the CTL file is updated, it is signed with a different certificate than those that exist in the CTL file that is installed on the IP Phones.

Normally, this new CTL file is not accepted; however, after the IP Phone finds the unknown certificate that is used in order to sign the CTL file, it contacts the TVS service on the CUCM.

> **Note**: The TVS server list is in the IP Phone configuration file and is mapped in the CUCM servers from the IP Phone **Device Pool > CallManager Group**.

Upon successful verification against the TVS server, the IP Phone updates its CTL file with the new version. These events occur in such a scenario:

1. The CTL file exists on the CUCM and on the IP Phone. The CCM+TFT (server) certificate for the CUCM Publisher node is used in order to sign the CTL file:

   <#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	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	

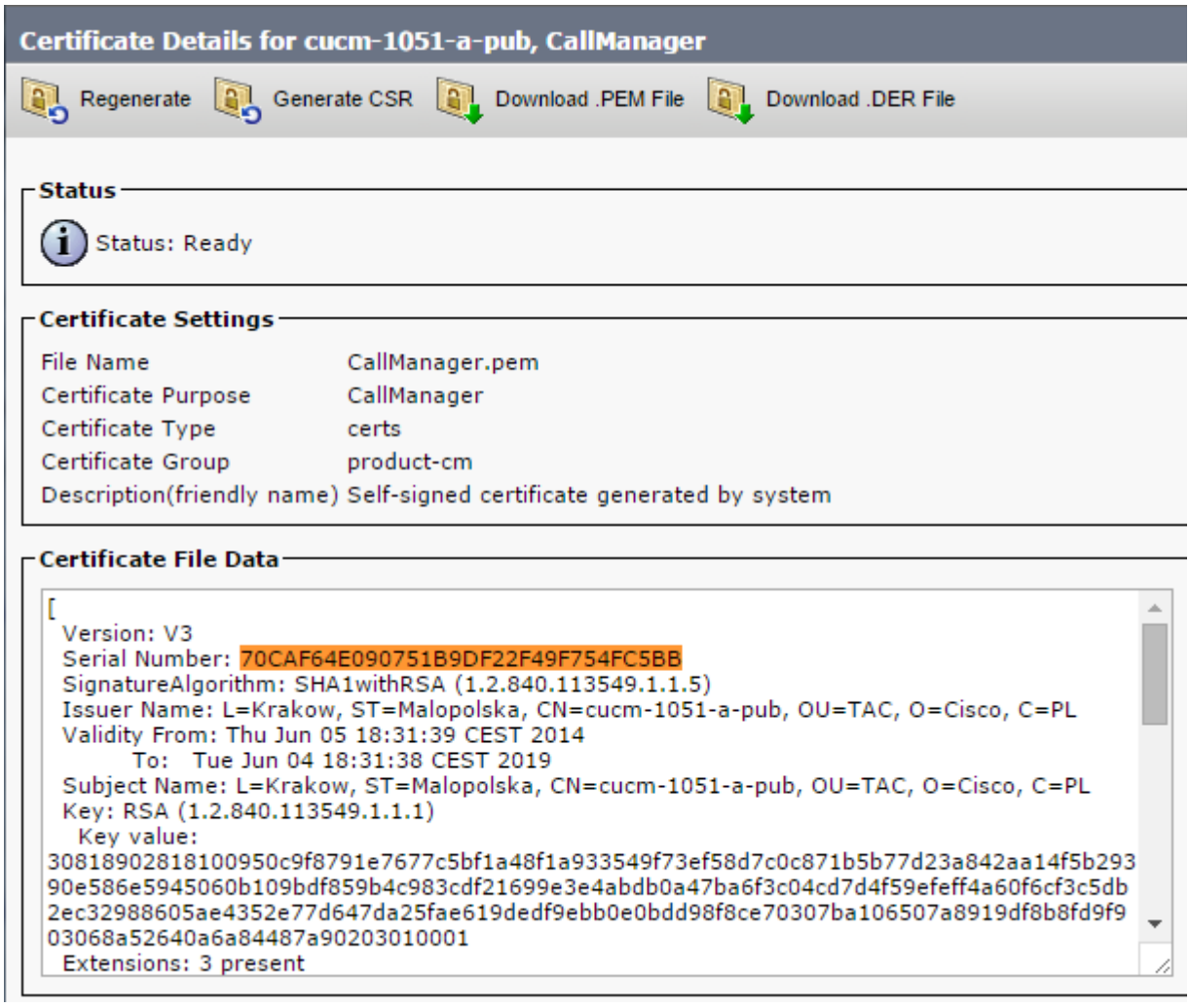
cucm-1051-a-pub

3	SUBJECTNAME	62	CN=cucm-1051-a-pub;OU=TAC;O=Cisco;L=Krakow; ST=Malopolska;C=PL
4	FUNCTION	2	

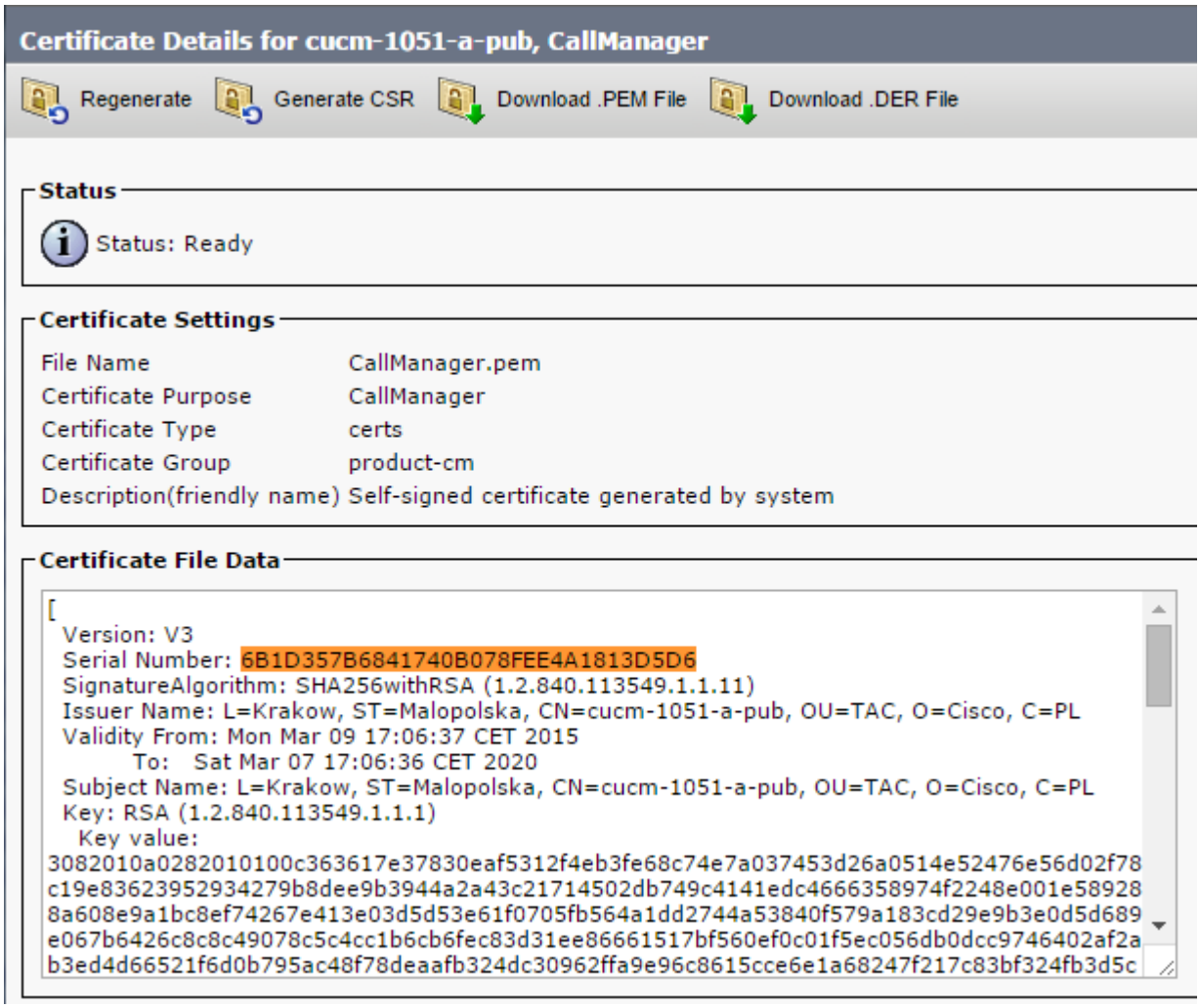
CCM+TFTP

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. The **CallManager.pem** file (CCM+TFTP certificate) is regenerated, and you can see that the serial number of the certificate changes:



3. The **utils ctl update CTLFile** command is entered into the CLI in order to update the CTL file:

```
<#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. The TVS service updates its certificate cache with the new CTL file details:

```
<#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. When you view the CTL file content, you can see that the file is signed with the new CallManager server certificate for the Publisher node:

```
<#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


[..]



         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. From the Unified Serviceability page, the TFTP and Cisco CallManager services are restarted on all of the nodes in the cluster that run these services.

7. The IP Phones are restarted, and they contact the TVS server in order to verify the unknown certificate that is now used in order to sign the new version of the CTL file:

<#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. Finally, on the IP Phones, you can verify that the CTL file is updated with the new version and that the MD5 checksum of the new CTL file matches with that of the CUCM: