

# Unified Communications Manager ITL Enhancements in Version 10.0(1)



Document ID: 117598

Contributed by Joe Martini, Cisco TAC Engineer.  
Apr 08, 2014

## Contents

### Introduction

### Background

### Problem Symptoms

### Solution – Bulk ITL Reset

- ITLRecovery with the Local Recovery Key
- ITLRecovery with the Remote Recovery Key
- Verify Current Signer with the "show itl" Command
- Verify that the ITLRecovery Key is Used

### Enhancements to Decrease the Possibility of Phones Losing Trust

### Back Up the ITL Recovery

### Verify

### Caveats

## Introduction

This document describes a new feature in Cisco Unified Communications Manager (CUCM) Version 10.0(1) that enables the bulk reset of Identity Trust List (ITL) files on Cisco Unified IP Phones. The bulk ITL reset feature is used when phones no longer trust the ITL file signer and also cannot authenticate the ITL file provided by the TFTP service locally or with the use of the Trust Verification Service (TVS).

## Background

The ability to bulk reset ITL files prevents the need to perform one or many of these steps to reestablish trust between IP phones and the CUCM servers.

- Restore from a backup in order to upload an old ITL file that the phones trust
- Change the phones in order to use a different TFTP server
- Delete the ITL file from the phone manually through the settings menu
- Factory reset the phone in the event settings so that access is disabled in order to erase the ITL

This feature is not intended to move phones between clusters; for that task, use one of the methods described in *Migrating IP Phones Between Clusters with CUCM 8 and ITL Files*. The ITL reset operation is used only to reestablish trust between IP phones and the CUCM cluster when they have lost their trust points.

Another security-related feature available in CUCM Version 10.0(1) that is not covered in this document is the Tokenless Certificate Trust List (CTL). The Tokenless CTL replaces the hardware USB security tokens with a software token used in order to enable encryption on the CUCM servers and endpoints. For additional information, refer to the *IP Phone Security and CTL (Certificate Trust List)* document.

Additional information on the ITL files and security by default can be found in the *Communications Manager Security By Default and ITL Operation and Troubleshooting* document.

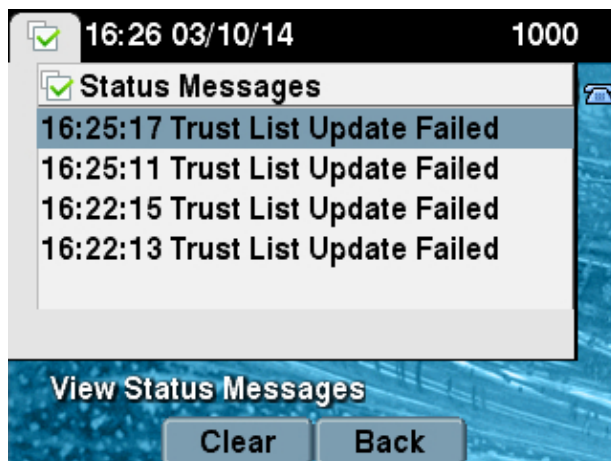
## Problem Symptoms

When phones are in a *locked* or *untrusted* state, they do not accept the ITL file or TFTP configuration provided by the TFTP service. Any configuration change that is contained in the TFTP configuration file is not applied to the phone. Some examples of settings that are contained in the TFTP configuration file are:

- Settings Access
- Web Access
- Secure Shell (SSH) Access
- Switched Port Analyzer (SPAN) to PC Port

If any of these settings are changed for a phone on the CCM Admin page and, after the phone is reset, the changes do not take effect, the phone might not trust the TFTP server. Another common symptom is when you access the corporate directory or other phone services, the message *Host Not Found* displays. In order to verify that the phone is in a locked or untrusted state, check the phone status messages from the phone itself or the phone web page in order to see if a *Trust List Update Failed* message displays. The *ITL Update Failed* message is an indicator that the phone is in a locked or untrusted state because it has failed to authenticate the trust list with its current ITL and failed to authenticate it with TVS.

The *Trust List Update Failed* message can be seen from the phone itself if you navigate to *Settings > Status > Status Messages*:



The *Trust List Update Failed* message can also be seen from the phone web page from the *Status Messages* as shown here:



## Solution – Bulk ITL Reset

CUCM Version 10.0(1) uses an additional key that can be used in order to reestablish trust between phones and the CUCM servers. This new key is the ITL Recovery key. The ITL Recovery key is created during the install or upgrade. This recovery key does not change when hostname changes, DNS changes, or other

changes are performed that might lead to problems where the phones get into a state where they no longer trust the signer of their configuration files.

The new *utils itl reset* CLI command can be used in order to reestablish trust between a phone or phones and the TFTP service on CUCM when phones are in a state where the *Trust List Update Failed* message is seen. The *utils itl reset* command:

1. Takes the current ITL file from the publisher node, strips the signature of the ITL file, and signs the contents of the ITL file again with the ITL Recovery private key.
2. Automatically copies the new ITL file to the TFTP directories on all of the active TFTP nodes in the cluster.
3. Automatically restarts the TFTP services on every node where TFTP runs.

The administrator must manually reset all of the phones. The reset causes the phones to request the ITL file upon boot up from the TFTP server and the ITL file the phone receives is signed by the ITLRecovery key instead of the *callmanager.pem* private key. There are two options to run an ITL reset: *utils itl reset localkey* and *utils itl reset remotekey*. The ITL reset command can only be run from the publisher. If you issue an ITL reset from a subscriber, it results in the *This is not a Publisher Node* message. Examples of each command are detailed in the next sections.

## ITLRecovery with the Local Recovery Key

The localkey option uses the ITL Recovery private key contained in the ITLRecovery.p12 file present on the Publisher hard drive as the new ITL file signer.

```
admin:utils itl reset localkey
Enter CCM Administrator password :

Locating active Tftp servers in the cluster.....

Following is the list of Active tftp servers in the cluster

['test10pub', 'test10sub']
The reset ITL file was generated successfully

Transferring new reset ITL file to the TFTP server nodes in the cluster.....

Restarting Cisco Tftp service on host test10pub
Cisco Tftp service restarted on host test10pub
Successfully transferred reset ITL to node test10sub

Restarting Cisco Tftp service on host test10sub
Cisco Tftp service restarted on host test10sub
```

## ITLRecovery with the Remote Recovery Key

The remotekey option allows the external SFTP server from which the ITLRecovery.p12 file has been saved to be specified.

```
admin:utils itl reset remotekey joemar2-server.cisco.com joemar2
/home/joemar2/ITLRecovery.p12
Enter Sftp password :Processing token in else 0 tac
countn is 1
Processing token in else 0 tac
countn is 1
```

Enter CCM Administrator password :

Locating active Tftp servers in the cluster.....

Following is the list of Active tftp servers in the cluster

```
['test10pub', 'test10sub']
```

The reset ITL file was generated successfully

Transferring new reset ITL file to the TFTP server nodes in the cluster.....

```
Restarting Cisco Tftp service on host test10pub
Cisco Tftp service restarted on host test10pub
Successfully transferred reset ITL to node test10sub
```

```
Restarting Cisco Tftp service on host test10sub
Cisco Tftp service restarted on host test10sub
```

**Note:** If an ITL reset is done with the remotekey option, the localkey (on disk file) on the publisher is replaced with the remotekey.

## Verify Current Signer with the "show itl" Command

If you view the ITL file with the *show itl* command before you issue an ITL reset command, it shows that the ITL contains an *ITLRECOVERY\_<publisher\_hostname>* entry. Every ITL file that is served by any TFTP server in the cluster contains this ITL recovery entry from the publisher. The output of the *show itl* command is taken from the publisher in this example. The token used in order to sign the ITL is in bold:

```
admin:show itl
```

The checksum value of the ITL file:

```
b331e5bfb450926e816be37f2d8c24a2(MD5)
```

```
9d7da73d16c1501b4d27dc1ed79211f390659982(SHA1)
```

Length of ITL file: 5302

The ITL File was last modified on Wed Feb 26 10:24:27 PST 2014

Parse ITL File

-----

Version: 1.2

HeaderLength: 324 (BYTES)

BYTEPOS TAG LENGTH VALUE

-----

3 SIGNERID 2 139

**4 SIGNERNAME 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US**

**5 SERIALNUMBER 16 61:40:52:E9:1E:E9:7C:77:9B:7B:5E:81:0A:B1:46:A5**

6 CANAME 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US

7 SIGNATUREINFO 2 15

8 DIGESTALGORTITHM 1

9 SIGNATUREALGOINFO 2 8

10 SIGNATUREALGORTITHM 1

11 SIGNATUREMODULUS 1

12 SIGNATURE 128

8f d4 0 cb a8 23 bc b0

f 75 69 9e 25 d1 9b 24

49 6 ae d0 68 18 f6 4

52 f8 1d 27 7 95 bc 94

d7 5c 36 55 8d 89 ad f4

88 0 d7 d0 db da b5 98

12 a2 6f 2e 6a be 9a dd

da 38 df 4f 4c 37 3e f6

ec 5f 53 bf 4b a9 43 76  
35 c5 ac 56 e2 5b 1b 96  
df 83 62 45 f5 6d 0 2f  
c d1 b8 49 88 8d 65 b4  
34 e4 7c 67 5 3f 7 59  
b6 98 16 35 69 79 8f 5f  
20 f0 42 5b 9b 56 32 2b  
c0 b7 1a 1e 83 c9 58 b  
14 FILENAME 12  
15 TIMESTAMP 4

ITL Record #:1

-----

BYTEPOS TAG LENGTH VALUE

-----

1 RECORDLENGTH 2 1115  
2 DNSNAME 2  
3 **SUBJECTNAME 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US**  
4 FUNCTION 2 System Administrator Security Token  
5 ISSUERNAM 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US  
6 **SERIALNUMBER 16 61:40:52:E9:1E:E9:7C:77:9B:7B:5E:81:0A:B1:46:A5**  
7 PUBLICKEY 140  
8 SIGNATURE 128  
9 CERTIFICATE 684 01 9C 4C 3D 27 D7 4D 82 CB B1 84 84 D4 2A 63 9F 71 78 BE A9  
(SHA1 Hash HEX)

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

ITL Record #:2

-----

BYTEPOS TAG LENGTH VALUE

-----

1 RECORDLENGTH 2 1115  
2 DNSNAME 2  
3 SUBJECTNAME 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US  
4 FUNCTION 2 TFTP  
5 ISSUERNAM 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US  
6 SERIALNUMBER 16 61:40:52:E9:1E:E9:7C:77:9B:7B:5E:81:0A:B1:46:A5  
7 PUBLICKEY 140  
8 SIGNATURE 128  
9 CERTIFICATE 684 01 9C 4C 3D 27 D7 4D 82 CB B1 84 84 D4 2A 63 9F 71 78 BE A9  
(SHA1 Hash HEX)

ITL Record #:3

-----

BYTEPOS TAG LENGTH VALUE

-----

1 RECORDLENGTH 2 439  
2 DNSNAME 2  
3 SUBJECTNAME 49 CN=CAPF-75638ad9;OU=tac;O=cisco;L=rtp;ST=nc;C=US  
4 FUNCTION 2 CAPF  
5 ISSUERNAM 49 CN=CAPF-75638ad9;OU=tac;O=cisco;L=rtp;ST=nc;C=US  
6 SERIALNUMBER 16 42:BF:37:77:9B:16:AF:1C:D2:30:88:C2:17:24:9D:AA  
7 PUBLICKEY 140  
8 SIGNATURE 128  
11 CERTHASH 20 BB C3 4A 1D DE 17 39 C5 36 1A 15 6B F0 65 FE BE D1 E6 19 03  
12 HASH ALGORITHM 1 SHA-1

ITL Record #:4

-----

BYTEPOS TAG LENGTH VALUE

-----

1 RECORDLENGTH 2 455  
2 DNSNAME 2  
3 SUBJECTNAME 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US  
4 FUNCTION 2 TVS  
5 ISSUERNAM 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US  
6 SERIALNUMBER 16 76:59:84:D8:B7:30:60:63:36:E5:C7:B6:A7:DD:B9:D6

```
7 PUBLICKEY 140
8 SIGNATURE 128
11 CERTHASH 20 7A BF CE B6 BE E2 06 02 74 D9 EB AE 58 48 52 93 7A 1E A5 55
12 HASH ALGORITHM 1 SHA-1
```

ITL Record #:5

----

BYTEPOS TAG LENGTH VALUE

```
-----
1 RECORDLENGTH 2 1141
2 DNSNAME 2
3 SUBJECTNAME 66 CN=ITLRECOVERY_test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;
ST=nc;C=US
4 FUNCTION 2 System Administrator Security Token
5 ISSUENAME 66 CN=ITLRECOVERY_test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;
ST=nc;C=US
6 SERIALNUMBER 16 6E:3F:65:C2:97:B3:E0:1F:E7:42:81:AB:52:CC:55:DC
7 PUBLICKEY 140
8 SIGNATURE 128
9 CERTIFICATE 692 CD C7 4A 39 87 87 52 FA B0 89 0C 28 AB CB 36 32 EB 87 16 DC
(SHA1 Hash HEX)
```

***This etoken was not used to sign the ITL file.***

ITL Record #:6

----

BYTEPOS TAG LENGTH VALUE

```
-----
1 RECORDLENGTH 2 713
2 DNSNAME 2
3 SUBJECTNAME 57 CN=test10sub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US
4 FUNCTION 2 TVS
5 ISSUENAME 57 CN=test10sub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US
6 SERIALNUMBER 16 67:D1:A1:9E:F8:AB:91:E6:C0:A6:10:35:26:30:EE:02
7 PUBLICKEY 270
8 SIGNATURE 256
11 CERTHASH 20 AF 28 89 16 F9 94 E6 62 A3 65 C2 88 3B 94 08 1C 66 7C 49 C9
12 HASH ALGORITHM 1 SHA-1
```

The ITL file was verified successfully.

## Verify that the ITLRecovery Key is Used

If you view the ITL file with the *show itl* command after you perform an ITL reset, it shows that the ITLRecovery entry has signed the ITL as shown here. The ITLRecovery remains the signer of the ITL until the TFTP is restarted, at which time the *callmanager.pem* or TFTP certificate is used in order to sign the ITL again.

admin:*show itl*

```
The checksum value of the ITL file:
c847df047cf5822c1ed6cf376796653d(MD5)
3440f94f9252e243c99506b4bd33ea28ec654dab(SHA1)
```

Length of ITL file: 5322

The ITL File was last modified on Wed Feb 26 10:34:46 PST 2014<

Parse ITL File

-----

Version: 1.2

HeaderLength: 344 (BYTES)

BYTEPOS TAG LENGTH VALUE

-----

3 SIGNERID 2 157  
4 **SIGNERNAME 66 CN=ITLRECOVERY\_test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US**  
5 **SERIALNUMBER 16 6E:3F:65:C2:97:B3:E0:1F:E7:42:81:AB:52:CC:55:DC**  
6 CANAME 66 CN=ITLRECOVERY\_test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US  
7 SIGNATUREINFO 2 15  
8 DIGESTALGORTITHM 1  
9 SIGNATUREALGOINFO 2 8  
10 SIGNATUREALGORTITHM 1  
11 SIGNATUREMODULUS 1  
12 SIGNATURE 128  
58 ff ed a ea 1b 9a c4  
e 75 f0 2b 24 ce 58 bd  
6e 49 ec 80 23 85 4d 18  
8b d0 f3 85 29 4b 22 8f  
b1 c2 7e 68 ee e6 5b 4d  
f8 2e e4 a1 e2 15 8c 3e  
97 c3 f0 1d c0 e 6 1b  
fc d2 f3 2e 89 a0 77 19  
5c 11 84 18 8a cb ce 2f  
5d 91 21 57 88 2c ed 92  
a5 8f f7 c 0 c1 c4 63  
28 3d a3 78 dd 42 f0 af  
9d f1 42 5e 35 3c bc ae  
c 3 df 89 9 f9 ac 77  
60 11 1f 84 f5 83 d0 cc  
14 FILENAME 12  
15 TIMESTAMP 4

ITL Record #:1

-----

BYTEPOS TAG LENGTH VALUE

-----

1 RECORDLENGTH 2 1115  
2 DNSNAME 2  
3 SUBJECTNAME 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US  
4 FUNCTION 2 System Administrator Security Token  
5 ISSUERNAME 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US  
6 SERIALNUMBER 16 61:40:52:E9:1E:E9:7C:77:9B:7B:5E:81:0A:B1:46:A5  
7 PUBLICKEY 140  
8 SIGNATURE 128  
9 CERTIFICATE 684 01 9C 4C 3D 27 D7 4D 82 CB B1 84 84 D4 2A 63 9F 71 78 BE A9  
(SHA1 Hash HEX)

***This etoken was not used to sign the ITL file.***

ITL Record #:2

-----

BYTEPOS TAG LENGTH VALUE

-----

1 RECORDLENGTH 2 1115  
2 DNSNAME 2  
3 SUBJECTNAME 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US  
4 FUNCTION 2 TFTP  
5 ISSUERNAME 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US  
6 SERIALNUMBER 16 61:40:52:E9:1E:E9:7C:77:9B:7B:5E:81:0A:B1:46:A5  
7 PUBLICKEY 140  
8 SIGNATURE 128  
9 CERTIFICATE 684 01 9C 4C 3D 27 D7 4D 82 CB B1 84 84 D4 2A 63 9F 71 78 BE A9  
(SHA1 Hash HEX)

ITL Record #:3

-----

BYTEPOS TAG LENGTH VALUE

-----

1 RECORDLENGTH 2 439

```
2 DNSNAME 2
3 SUBJECTNAME 49 CN=CAPF-75638ad9;OU=tac;O=cisco;L=rtp;ST=nc;C=US
4 FUNCTION 2 CAPF
5 ISSUENAME 49 CN=CAPF-75638ad9;OU=tac;O=cisco;L=rtp;ST=nc;C=US
6 SERIALNUMBER 16 42:BF:37:77:9B:16:AF:1C:D2:30:88:C2:17:24:9D:AA
7 PUBLICKEY 140
8 SIGNATURE 128
11 CERTHASH 20 BB C3 4A 1D DE 17 39 C5 36 1A 15 6B F0 65 FE BE D1 E6 19 03
12 HASH ALGORITHM 1 SHA-1
```

ITL Record #:4

----

BYTEPOS TAG LENGTH VALUE

-----

```
1 RECORDLENGTH 2 455
2 DNSNAME 2
3 SUBJECTNAME 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US
4 FUNCTION 2 TVS
5 ISSUENAME 57 CN=test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US
6 SERIALNUMBER 16 76:59:84:D8:B7:30:60:63:36:E5:C7:B6:A7:DD:B9:D6
7 PUBLICKEY 140
8 SIGNATURE 128
11 CERTHASH 20 7A BF CE B6 BE E2 06 02 74 D9 EB AE 58 48 52 93 7A 1E A5 55
12 HASH ALGORITHM 1 SHA-1
```

ITL Record #:5

----

BYTEPOS TAG LENGTH VALUE

-----

```
1 RECORDLENGTH 2 1141
2 DNSNAME 2
3 SUBJECTNAME 66 CN=ITLRECOVERY_test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US
4 FUNCTION 2 System Administrator Security Token
5 ISSUENAME 66 CN=ITLRECOVERY_test10pub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US
6 SERIALNUMBER 16 6E:3F:65:C2:97:B3:E0:1F:E7:42:81:AB:52:CC:55:DC
7 PUBLICKEY 140
8 SIGNATURE 128
9 CERTIFICATE 692 CD C7 4A 39 87 87 52 FA B0 89 0C 28 AB CB 36 32 EB 87 16 DC
(SHA1 Hash HEX)
```

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

ITL Record #:6

----

BYTEPOS TAG LENGTH VALUE

-----

```
1 RECORDLENGTH 2 713
2 DNSNAME 2
3 SUBJECTNAME 57 CN=test10sub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US
4 FUNCTION 2 TVS
5 ISSUENAME 57 CN=test10sub.joemar2.lab;OU=tac;O=cisco;L=rtp;ST=nc;C=US
6 SERIALNUMBER 16 67:D1:A1:9E:F8:AB:91:E6:C0:A6:10:35:26:30:EE:02
7 PUBLICKEY 270
8 SIGNATURE 256
11 CERTHASH 20 AF 28 89 16 F9 94 E6 62 A3 65 C2 88 3B 94 08 1C 66 7C 49 C9
12 HASH ALGORITHM 1 SHA-1
```

The ITL file was verified successfully.

## Enhancements to Decrease the Possibility of Phones Losing Trust

In addition to the ITL reset capability, CUCM Version 10.0(1) includes administrator features that help prevent phones from entering an untrusted state. The two trust points the phone has are the TVS certificate



(*TVS.pem*) and the TFTP certificate (*callmanager.pem*). In the simplest environment with only one CUCM server, if an administrator regenerates the *callmanager.pem* certificate and the *TVS.pem* certificate one right after another, the phone resets and upon bootup displays the **Trust List Update Failed** message. Even with an automatic device reset sent from CUCM to the phone due to a certificate contained in the ITL that is regenerated, the phone can enter a state where it does not trust CUCM.

In order to help prevent the scenario where multiple certificates are regenerated at the same time (typically hostname change or DNS domain name modifications), CUCM now has a hold timer. When a certificate is regenerated, CUCM prevents the administrator from regenerating another certificate on the same node within five minutes of the previous certificate regeneration. This process causes the phones to be reset upon regenerating the first certificate, and they should be back up and registered before the next certificate is regenerated.

Regardless of which certificate is generated first, the phone has its secondary method to authenticate files. Additional details about this process can be found in Communications Manager Security By Default and ITL Operation and Troubleshooting.

This output shows a situation where CUCM prevents the administrator from regenerating another certificate within five minutes of a previous certificate regeneration as viewed from the CLI:

```
admin:set cert regen CallManager
```

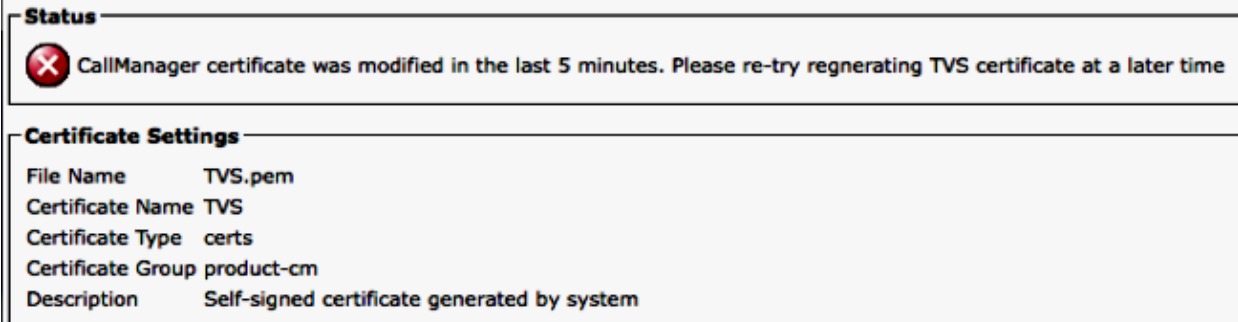
```
WARNING: This operation will overwrite any CA signed certificate
previously imported for CallManager
Proceed with regeneration (yes|no)? yes
```

```
Successfully Regenerated Certificate for CallManager.
Please do a backup of the server as soon as possible. Failure to do
so can stale the cluster in case of a crash.
You must restart services related to CallManager for the regenerated
certificates to become active.
```

```
admin:set cert regen TVS
```

```
CallManager certificate was modified in the last 5 minutes. Please re-try
regenerating TVS certificate at a later time
```

The same message can be seen from the operating system (OS) Administration page as shown here:



The screenshot shows a web interface with two sections. The top section, titled "Status", contains a red error icon and the text: "CallManager certificate was modified in the last 5 minutes. Please re-try regenerating TVS certificate at a later time". The bottom section, titled "Certificate Settings", lists the following details:

File Name	TVS.pem
Certificate Name	TVS
Certificate Type	certs
Certificate Group	product-cm
Description	Self-signed certificate generated by system

The publisher ITL recovery key is the only one in use by the entire cluster, even though each node has its own ITLRecovery certificate issued to the Common Name (CN) of *ITLRecovery\_<node name>*. The publisher ITLRecovery key is the only one used in the ITL files for the entire cluster as seen from the *show itl* command. This is why the only *ITLRecovery\_<hostname>* entry seen in an ITL file contains the hostname of the publisher.

If the hostname of the publisher is changed, the ITLRecovery entry in the ITL continues to show the old hostname of the publisher. This is done intentionally because the ITLRecovery file should never change to ensure the phones always trust the ITL recovery.

This applies for when domain names are changed too; the original domain name is seen in the ITLRecovery entry in order to ensure that the recovery key does not change. The only time the ITLRecovery certificate should change is when it expires due to the five-year validity and must be regenerated.

The ITL recovery keypairs can be regenerated with either the CLI or the OS Administration page. IP phones are not reset when the ITLRecovery certificate is regenerated on the publisher or any of the subscribers. Once the ITLRecovery certificate has been regenerated, the ITL file does not update until the TFTP service is restarted. After ITLRecovery certificate regeneration on the publisher, restart the TFTP service on every node that runs the TFTP service in the cluster in order to update the ITLRecovery entry in the ITL file with the new certificate. The final step is to reset all devices from *System > Enterprise Parameters* and to use the reset button in order to make all devices download the new ITL file that contains the new ITLRecovery certificate.

## Back Up the ITL Recovery

The ITL Recovery key is required in order to recover phones when they enter an untrusted state. Due to this, new Real-Time Monitoring Tool (RTMT) alerts are generated daily until the ITL Recovery key is backed up. A Disaster Recovery System (DRS) backup does not suffice to stop the alerts. Although a backup is recommended in order to save the ITL Recovery key, a manual backup of the key file is needed as well.

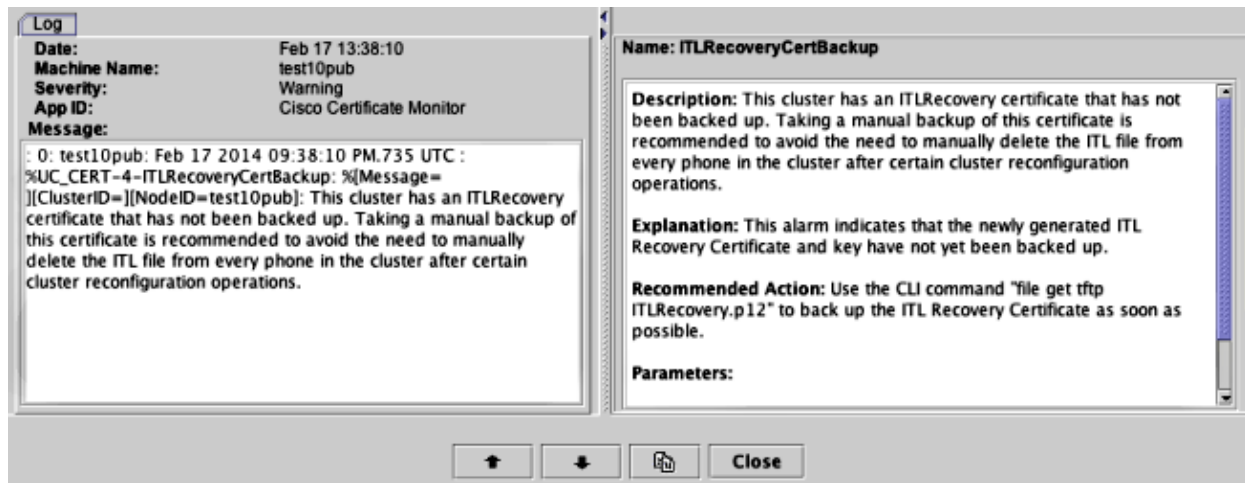
In order to back up the recovery key, log in to the CLI of the publisher and enter the *file get tftp ITLRecovery.p12* command. An SFTP server is needed in order to save the file to as shown here. Subscriber nodes do not have an ITL recovery file, so if you issue the *file get tftp ITLRecovery.p12* command on a subscriber, it results in *file not found*.

```
admin:file get tftp ITLRecovery.p12
Please wait while the system is gathering files info ...done.
Sub-directories were not traversed.
Number of files affected: 1
Total size in Bytes: 1709
Total size in Kbytes: 1.6689453
Would you like to proceed [y/n]? y
SFTP server IP: joemar2-server.cisco.com
SFTP server port [22]:
User ID: joemar2
Password: *****

Download directory: /home/joemar2/

The authenticity of host 'joemar2-server.cisco.com (172.18.172.254)' can't be
established.
RSA key fingerprint is 2c:8f:9b:b2:ff:f7:a6:31:61:1b:bc:95:cc:bc:ba:bd.
Are you sure you want to continue connecting (yes/no)? yes
.
Transfer completed.
Downloading file: /usr/local/cm/tftp/ITLRecovery.p12
```

Until the manual backup is performed from the CLI in order to back up the ITLRecovery.p12 file, a warning is printed in the CiscoSyslog (Event Viewer – Application Log) every day as shown here. A daily email might also be received until the manual backup is performed if email notification is enabled from the OS Administration page, *Security > Certificate Monitor*.



While a DRS backup contains the ITLRecovery, it is recommended to still store the ITLRecovery.p12 file in a safe location in case the backup files are lost or corrupted or in order to have the option to reset the ITL file without the need to restore from a backup. If you have the ITLRecovery.p12 file from the publisher saved, it also allows the publisher to be rebuilt without a backup with the use the DRS restore option to restore the database from a subscriber and reestablish trust between the phones and CUCM servers by resetting the ITL with the *utils itl reset remotekey* option.

Remember that if the publisher is rebuilt, the cluster security password should be the same as the publisher where the ITLRecovery.p12 file was taken from because the ITLRecovery.p12 file is password-protected with a password based off of the cluster security password. For this reason, if the cluster security password is changed, the RTMT alert that indicates the ITLRecovery.p12 file has not been backed up is reset and triggers daily until the new ITLRecovery.p12 file is saved with the *file get tftp ITLRecovery.p12* command.

## Verify

The bulk ITL reset feature only works if phones have an ITL installed that contains the ITLRecovery entry. In order to verify that the ITL file installed on the phones contains the ITLRecovery entry, enter the *show itl* command from the CLI on each of the TFTP servers to find the checksum of the ITL file. The output from the *show itl* command displays the checksum:

```
admin:show itl
The checksum value of the ITL file:
b331e5bfb450926e816be37f2d8c24a2(MD5)
9d7da73d16c1501b4d27dc1ed79211f390659982(SHA1)
```

The checksum is different on each TFTP server because each server has its own *callmanager.pem* certificate in its ITL file. The ITL checksum of the ITL installed on the phone can be found if you view the ITL on the phone itself under *Settings > Security Configuration > Trust List*, from the phone web page, or from the DeviceTLInfo alarm reported by phones that run newer firmware.

Most phones that run firmware Version 9.4(1) or later report the SHA1 hash of their ITL to CUCM with the DeviceTLInfo alarm. The information sent by the phone can be viewed in the Event Viewer – Application Log from RTMT and compared to the SHA1 hash of the ITL hash of the TFTP servers the phones use in order to find any phones that do not have the current ITL installed, which contains the ITLRecovery entry.

## Caveats

- CSCun18578 – ITL reset localkey/remotekey fails in certain scenarios
- CSCun19112 – ITL reset remotekey error in SFTP bad authentication type

