Безопасность CUCM по умолчанию и операцией ITL и устранением проблем

Содержание

Введение

Этот документ описывает функцию Безопасности по умолчанию (SBD) Версий Cisco Unified Communications Manager (CUCM) 8.0 и позже. Этот документ служит дополнением к официальной Безопасности Документами по умолчанию и предоставляет сведения о функционировании системы и советы по устранению проблем, чтобы помочь администраторам и упростить процесс устранения проблем.

Общие сведения

Версия 8.0 CUCM и позже представляет функцию SBD, которая состоит из файлов Идентификационного списка доверия (ITL) и службы проверки доверия (TVS). Каждый

кластер CUCM теперь использует основанную на ITL безопасность автоматически. Существует компромисс между безопасностью и простотой использования/простоты администрирования, о котором должны знать администраторы, прежде чем они будут делать определенные изменения в Версию 8.0 кластером CUCM.

Это - хорошая идея познакомиться с этими базовыми понятиями SBD: Асимметричная статья Key Cryptography Wikipedia и статья Public Key Infrastructure Wikipedia.

Обзор SBD

Этот раздел предоставляет краткий обзор точно, что предоставляет SBD. Для полных технических подробностей каждой функции посмотрите раздел Подробности и Сведений об устранении проблем SBD.

SBD предоставляет эти три функции для поддерживаемых IP-телефонов:

- Проверка подлинности по умолчанию загружаемых файлов TFTP (конфигурация, локаль, ringlist), которые используют ключ подписи
- Дополнительное шифрование файлов конфигурации TFTP, которые используют ключ подписи
- Проверка сертификата для инициируемых в телефон Подключений HTTPS, которые используют удаленную базу доверенных сертификатов сертификата на CUCM (TVS)

Этот документ предоставляет обзор каждой из этих функций.

Аутентификация загрузки TFTP

Когда Список надежных сертификатов (CTL) или файл ITL присутствуют, IP-телефон запрашивает файл конфигурации TFTP со знаком от сервера TFTP CUCM. Этот файл позволяет телефону проверять, что файл конфигурации прибыл из надежного источника. С подарком файлов CTL/ITL по телефонам файлы конфигурации должны быть подписаны доверяемым сервером TFTP. Файл является открытым текстом в сети, в то время как это передано, но идет со специальной подписью проверки.

Телефон запрашивает **SEP *<MAC-адрес>* .cnf.xml.sgn** для получения файла конфигурации со специальной подписью. Этот файл конфигурации подписан Частным ключом TFTP, который соответствует CallManager.pem на странице Administration Certificate Management Операционной системы (OS).

Файл со знаком имеет подпись наверху, чтобы аутентифицировать файл, но находится иначе в открытом тексте XML. Образ ниже показов, которые подписывающее лицо файла конфигурации **CN=CUCM8-Publisher.bbbburns.lab**, который в свою очередь подписан **CN=JASBURNS-AD**. Это означает, что телефон должен проверить подпись **CUCM8-Publisher.bbbburns.lab** против файла ITL, прежде чем будет принят этот файл конфигурации.

Вот схема, которая показывает, как секретный ключ используется наряду с Алгоритмом Дайджеста сообщения (MD) 5 или Защищенный алгоритм хэширования (SHA) 1 хэш-функция для создания файла со знаком.

Проверка подписи инвертирует этот процесс с помощью открытого ключа, который

совпадает для дешифрования хэша. Если хэши совпадают, это показывает:

- Этот файл не модифицировался в пути.
- Этот файл прибывает из стороны, перечисленной в подписи, так как что-либо дешифрованное успешно с открытым ключом, должно быть, было зашифровано с секретным ключом.

Шифрование файлов конфигурации TFTP

Если дополнительное шифрование конфигурации TFTP включено в связанном Телефонном Профиле безопасности, телефон запрашивает зашифрованный файл конфигурации. Этот файл подписан с Частным ключом TFTP и зашифрован с симметричным ключом, которым обмениваются между телефоном, и CUCM (обратитесь к Руководству по обеспечению безопасности Cisco Unified Communications Manager, Выпуску 8.5 (1) для полного изложения) так, чтобы его содержание не могло быть считано с сетевым анализатором, пока у наблюдателя нет необходимых ключей.

Телефон запрашивает **SEP *<MAC-адрес>* .cnf.xml.enc.sgn** для получения зашифрованного файла со знаком.

Зашифрованный файл конфигурации имеет подпись вначале также, но нет никаких данных открытого текста после, только зашифрованные данные (искаженные двоичные символы в этом текстовом редакторе). Образ показывает, что подписывающее лицо совпадает с в предыдущем примере, таким образом, это подписывающее лицо должно присутствовать в файле ITL, прежде чем телефон примет файл. Далее, ключи расшифровки должны быть корректными, прежде чем телефон сможет считать содержание файла.

Трастовый сервис проверки (Удаленная проверка сертификата и подписи)

IP-телефоны содержат ограниченное количество памяти, и может также быть большое число телефонов для управления в сети. CUCM действует как удаленная база доверенных сертификатов через TVS так, чтобы полная база доверенных сертификатов сертификата не была размещена в каждый IP-телефон. Любое время телефон не может проверить подпись или сертификат через CTL или файлы ITL, это спрашивает сервер для проверки TVS. Чем если бы база доверенных сертификатов присутствовала на всех IP-телефонах, этой центральной базой доверенных сертификатов легче управлять.

Подробность SBD и сведения об устранении проблем

Этот раздел детализирует процесс SBD.

Файлы ITL и подарок сертификатов на CUCM

Во-первых, существует много файлов, которые должны присутствовать на самом сервере CUCM. Самая важная часть является сертификатом TFTP и Частным ключом TFTP. Сертификат TFTP расположен под**> Certificate Management> Security администрирования ОС> CallManager.pem**.

Сервер CUCM использует секретные и открытые ключи сертификата CallManager.pem для Сервиса TFTP (а также для сервиса Cisco Call Manager (CCM)). Образ показывает, что сертификат CallManager.pem выполнен к **CUCM8-publisher.bbbburns.lab** и подписан **JASBURNS-AD**. Все файлы конфигурации TFTP подписаны секретным ключом ниже.

Все телефоны могут использовать открытый ключ TFTP в сертификате CallManager.pem для дешифрования любого файла, зашифрованного с Частным ключом TFTP, а также проверить любой файл, подписанный с Частным ключом TFTP.

В дополнение к секретному ключу сертификата CallManager.pem сервер CUCM также хранит файл ITL, который представлен телефонам. **Команда show itl** показывает полные содержимые этого файла ITL через доступ Secure Shell (SSH) к серверу CUCM CLI ОС.

Это разрывы раздела вниз часть частью файла ITL, потому что это имеет много важных компонентов, которые использует телефон.

Первая часть является данными о подписи. Даже файл ITL является файлом со знаком. Эти выходные данные показывают, что подписаны Частным ключом TFTP, который привязан к предыдущему сертификату CallManager.pem.

```
admin:show itl
Length of ITL file: 5438
The ITL File was last modified on Wed Jul 27 10:16:24 EDT 2011

       Parse ITL File
       ----------------

Version:       1.2
HeaderLength:  296 (BYTES)

BYTEPOS TAG            LENGTH VALUE
------- ---            ------ -----
3      SIGNERID       2      110
4      SIGNERNAME     76     CN=CUCM8-Publisher.bbbburns.lab;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
5      SERIALNUMBER   10     21:00:2D:17:00:00:00:00:00:05
6      CANAME         15     CN=JASBURNS-AD
*Signature omitted for brevity*
```

Следующие разделы каждый содержит их цель в параметре **специальной функции**. Первая функция является Маркером безопасности Системного администратора. Это - подпись открытого ключа TFTP.

```
       ITL Record #:1
                 ----
BYTEPOS TAG            LENGTH VALUE
------- ---            ------ -----
1      RECORDLENGTH   2      1972
2      DNSNAME        2
3      SUBJECTNAME    76     CN=CUCM8-Publisher.bbbburns.lab;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4      FUNCTION       2      System Administrator Security Token
5      ISSUERNAME     15     CN=JASBURNS-AD
6      SERIALNUMBER   10     21:00:2D:17:00:00:00:00:00:05
7      PUBLICKEY      140
8      SIGNATURE      256
9      CERTIFICATE    1442   0E 1E 28 0E 5B 5D CC 7A 20 29 61 F5
                                 8A DE 30 40 51 5B C4 89 (SHA1 Hash HEX)
This etoken was used to sign the ITL file.
```

Следующая функция является CCM+TFTP. Это - снова открытый ключ TFTP, который служит, чтобы аутентифицировать и дешифровать загруженные файлы конфигурации TFTP.

```
       ITL Record #:2
                 ----
BYTEPOS TAG             LENGTH VALUE
------- ---             ------ -----
1      RECORDLENGTH    2      1972
2      DNSNAME         2
3      SUBJECTNAME     76     CN=CUCM8-Publisher.bbbburns.lab;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4      FUNCTION        2      CCM+TFTP
5      ISSUERNAME      15     CN=JASBURNS-AD
6      SERIALNUMBER    10     21:00:2D:17:00:00:00:00:00:05
7      PUBLICKEY       140
8      SIGNATURE       256
9      CERTIFICATE     1442   0E 1E 28 0E 5B 5D CC 7A 20 29 61 F5
                                 8A DE 30 40 51 5B C4 89 (SHA1 Hash HEX)
```

Следующая функция является TVS. Существует запись для открытого ключа каждого сервера TVS, с которым соединяется телефон. Это позволяет телефону устанавливать сеанс Уровня защищенных сокетов (SSL) к серверу TVS.

```
       ITL Record #:3
                 ----
BYTEPOS TAG             LENGTH VALUE
------- ---             ------ -----
1      RECORDLENGTH    2      743
2      DNSNAME         2
3      SUBJECTNAME     76     CN=CUCM8-Publisher.bbbburns.lab;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4      FUNCTION        2      TVS
5      ISSUERNAME      76     CN=CUCM8-Publisher.bbbburns.lab;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
6      SERIALNUMBER    8      2E:3E:1A:7B:DA:A6:4D:84
7      PUBLICKEY       270
8      SIGNATURE       256
11     CERTHASH        20     C7 E1 D9 7A CC B0 2B C2 A8 B2 90 FB
                                 AA FE 66 5B EC 41 42 5D
12     HASH ALGORITHM 1       SHA-1
```

Заключительная функция, включенная в файл ITL, является функцией представительства сертифицирующей организации (CAPF). Этот сертификат позволяет телефонам устанавливать безопасное соединение с сервисом CAPF на сервере CUCM так, чтобы телефон мог установить или обновить логически значимый сертификат (LSC). Этот процесс будет охвачен в другом документе, который должен все же быть освобожден.

```
       ITL Record #:4
                 ----
BYTEPOS TAG             LENGTH VALUE
------- ---             ------ -----
1      RECORDLENGTH    2      455
2      DNSNAME         2
3      SUBJECTNAME     61     CN=CAPF-9c4cba7d;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4      FUNCTION        2      CAPF
5      ISSUERNAME      61     CN=CAPF-9c4cba7d;
                                 OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
6      SERIALNUMBER    8      0A:DC:6E:77:42:91:4A:53
7      PUBLICKEY       140
8      SIGNATURE       128
11     CERTHASH        20     C7 3D EA 77 94 5E 06 14 D2 90 B1
```

```
                              A1 43 7B 69 84 1D 2D 85 2E
12     HASH ALGORITHM 1      SHA-1

The ITL file was verified successfully.
```

Следующий раздел покрывает точно, что происходит, когда загружается телефон.

Телефон загружает ITL и файл конфигурации

После того, как телефон загружает и получает IP-адрес, а также адрес сервера TFTP, это просит CTL и файлы ITL сначала.

Этот захват пакета показывает телефонный запрос о файле ITL. Если вы фильтруете на **tftp.opcode == 1,** вы видите каждый Запрос Чтения TFTP с телефона:

Так как телефон получил CTL и файлы ITL от TFTP успешно, телефон просит файл конфигурации со знаком. Телефонные console log, которые показывают это поведение, доступны от веб-интерфейса телефона:

Сначала телефон запрашивает файл CTL, который успешно выполняется:

```
837: NOT 09:13:17.561856 SECD: tlRequestFile: Request CTLSEP0011215A1AE3.tlv
846: NOT 09:13:17.670439 TFTP: [27]:Requesting CTLSEP0011215A1AE3.tlv from
14.48.44.80
847: NOT 09:13:17.685264 TFTP: [27]:Finished --> rcvd 4762 bytes
```

Затем телефон также запрашивает файл ITL:

```
868: NOT 09:13:17.860613 TFTP: [28]:Requesting ITLSEP0011215A1AE3.tlv from
14.48.44.80
869: NOT 09:13:17.875059 TFTP: [28]:Finished --> rcvd 5438 bytes
```

Телефон проверяет ITL и файл конфигурации

После того, как файл ITL загружен, это должно быть проверенный. Существует много состояний, в которых может быть телефон на этом этапе, таким образом, этот документ покрывает их всех.

- Телефон не имеет никакого CTL или подарка файла ITL, или ITL является пробелом из-за **Подготовить Кластера для Отката к Пред 8.0** параметров. в этом состоянии телефон вслепую доверяет следующему CTL или загруженному файлу ITL и использует эту подпись впредь.
- Телефон уже имеет CTL, но никакой ITL. В этом состоянии телефон только доверяет ITL, если это может быть проверено функцией CCM+TFTP в файле CTL.
- Телефон уже имеет CTL и файл ITL. В этом состоянии телефон проверяет, что недавно загружаемые файлы совпадают с подписью или в CTL, ITL или в сервере TVS.

Вот блок-схема, которая описывает, как телефон проверяет подписанные файлы и сертификаты HTTPS:

В этом случае телефон в состоянии проверить подпись в файлах CTL и ITL. Телефон уже имеет и CTL и ITL, таким образом, это просто проверило против них и нашло корректную подпись.

```
877: NOT 09:13:17.925249 SECD: validate_file_envelope:
File sign verify SUCCESS; header length <296>
```

Так как телефон загрузил CTL и файлы ITL, с этого момента это, ONLY запрашивает подписанные файлы конфигурации. Это иллюстрирует, что логика телефона должна решить, что сервер TFTP безопасен, на основе присутствия CTL и ITL, и затем попросить файл со знаком:

```
917: NOT 09:13:18.433411 tftpClient: tftp request rcv'd from /usr/tmp/tftp,
srcFile = SEP0011215A1AE3.cnf.xml, dstFile = /usr/ram/SEP0011215A1AE3.cnf.xml
max size = 550001
918: NOT 09:13:18.457949 tftpClient: auth server - tftpList[0] = ::ffff:
14.48.44.80
919: NOT 09:13:18.458937 tftpClient: look up server - 0
920: NOT 09:13:18.462479 SECD: lookupCTL: TFTP SRVR secure
921: NOT 09:13:18.466658 tftpClient: secVal = 0x9 922: NOT 09:13:18.467762
tftpClient: ::ffff:14.48.44.80 is a secure server
923: NOT 09:13:18.468614 tftpClient: retval = SRVR_SECURE
924: NOT 09:13:18.469485 tftpClient: Secure file requested
925: NOT 09:13:18.471217 tftpClient: authenticated file approved - add .sgn
-- SEP0011215A1AE3.cnf.xml.sgn
926: NOT 09:13:18.540562 TFTP: [10]:Requesting SEP0011215A1AE3.cnf.xml.sgn
from 14.48.44.80 with size limit of 550001
927: NOT 09:13:18.559326 TFTP: [10]:Finished --> rcvd 7652 bytes
```

Как только файл конфигурации со знаком загружен, телефон должен аутентифицировать его против Функции для CCM+TFTP в ITL:

```
937: NOT 09:13:18.656906 SECD: verifyFile: verify SUCCESS
</usr/ram/SEP0011215A1AE3.cnf.xml>
```

Телефон связывается с TVS для неизвестного сертификата

Файл ITL предоставляет функцию TVS, которая содержит сертификат сервиса TVS, который работает на порте TCP сервера CUCM 2445. TVS работает на всех серверах, где активирован Сервис CallManager. Сервис TFTP CUCM использует настроенную Группу CallManager для построения списка серверов TVS, с которыми телефон должен связаться на файле конфигурации телефона.

Некоторые лабораторные работы используют только одиночный сервер CUCM. В мультиузле кластер CUCM может быть до трех записей TVS для телефона, один для каждого CUCM в CUCM Group телефона.

Данный пример показывает то, что происходит, когда нажата кнопка **Directories** на IP-телефоне. URL Каталогов настроен для HTTPS, таким образом, телефону предоставляют веб-сертификат Tomcat от сервера Каталогов. Этот веб-сертификат Tomcat (tomcat.pem в администрировании ОС) не загружен в телефоне, таким образом, телефон должен связаться с TVS для аутентификации сертификата.

См. предыдущий TVS Обзор схематически изображают для описания взаимодействия. Вот телефонная перспектива console log:

Сначала вы находите URL Каталога:

```
1184: NOT 15:20:55.219275 JVM: Startup Module Loader|cip.dir.TandunDirectories:
? - Directory url https://14.48.44.80:8443/ccmcip/xmldirectory.jsp
```

Это - SSL/Transport Layer Security (TLS) безопасный сеанс HTTP, который требует проверки.

```
1205: NOT 15:20:59.404971 SECD: clpSetupSsl: Trying to connect to IPV4, IP:
14.48.44.80, Port : 8443
```

```
1206: NOT 15:20:59.406896 SECD: clpSetupSsl: TCP connect() waiting,
<14.48.44.80> c:8 s:9 port: 8443
1207: NOT 15:20:59.408136 SECD: clpSetupSsl: TCP connected,
<14.48.44.80> c:8 s:9
1208: NOT 15:20:59.409393 SECD: clpSetupSsl: start SSL/TLS handshake,
<14.48.44.80> c:8 s:9
1209: NOT 15:20:59.423386 SECD: srvr_cert_vfy: Server Certificate
Validation needs to be done
```

Телефон сначала проверяет, что сертификат, представленный сервером SSL/TLS, присутствует в CTL. Затем телефон посмотрел на Функции в файле ITL, чтобы видеть, находит ли это соответствие. Это сообщение об ошибках говорит "свидетельство HTTPS не в CTL", что означает, "что сертификация не может быть найдена в CTL или ITL".

```
1213: NOT 15:20:59.429176 SECD: findByCertAndRoleInTL: Searching TL from CTL file
1214: NOT 15:20:59.430315 SECD: findByCertAndRoleInTL: Searching TL from ITL file
1215: ERR 15:20:59.431314 SECD: EROR:https_cert_vfy: HTTPS cert not in CTL,
<14.48.44.80>
```

После того, как прямое содержание CTL и файла ITL проверено для сертификата, следующая вещь, телефонные проверки являются кэшем TVS. Если телефон недавно попросил у сервера TVS того же сертификата, это сделано для сокращения сетевого трафика. Если сертификат HTTPS не найден в телефонном кэше, можно сделать TCP - подключение к самому серверу TVS.

```
1220: NOT 15:20:59.444517 SECD: processTvsClntReq: TVS Certificate
Authentication request
1221: NOT 15:20:59.445507 SECD: lookupAuthCertTvsCacheEntry: No matching
entry found at cache
1222: NOT 15:20:59.446518 SECD: processTvsClntReq: No server sock exists,
must be created
1223: NOT 15:20:59.451378 SECD: secReq_initClient: clnt sock fd 11 bound
to </tmp/secClnt_secd>
1224: NOT 15:20:59.457643 SECD: getTvsServerInfo: Phone in IPv4 only mode
1225: NOT 15:20:59.458706 SECD: getTvsServerInfo: Retreiving IPv4 address
1230: NOT 15:20:59.472628 SECD: connectToTvsServer: Successfully started
a TLS connection establishment to the TVS server: IP:14.48.44.80, port:2445
(default); Waiting for it to get connected.
```

Помните, что соединение с самим TVS является SSL/TLS (безопасный HTTP или HTTPS), таким образом, это - также сертификат, который должен аутентифицироваться против CTL ot ITL. Если все идет правильно, сертификат сервера TVS должен быть найден в функции TVS файла ITL. См. Запись ITL #3 в предыдущем примере файл ITL.

```
1244: NOT 15:20:59.529938 SECD: srvr_cert_vfy: Server Certificate Validation
needs to be done
1245: NOT 15:20:59.533412 SECD: findByIssuerAndSerialAndRoleInTL:
Searching TL from CTL file
1246: NOT 15:20:59.534936 SECD: findByIssuerAndSerialAndRoleInTL:
Searching TL from ITL file
1247: NOT 15:20:59.537359 SECD: verifyCertWithHashFromTL: cert hash and
hash in TL MATCH
1248: NOT 15:20:59.538726 SECD: tvs_cert_vfy: TVS cert verified with hash
from TL, <14.48.44.80>
```

Успешно! Телефон теперь имеет безопасное соединение с сервером TVS. Следующий шаг должен спросить сервер TVS "Hello, я доверяю этому серверному сертификату Каталогов?"

Данный пример показывает ответ на тот вопрос - ответ 0, что означает успех (никакая ошибка).

```
1264: NOT 15:20:59.789738 SECD: sendTvsClientReqToSrvr: Authenticate
Certificate : request sent to TVS server - waiting for response
```

```
1273: NOT 15:20:59.825648 SECD: processTvsSrvrResponse: Authentication Response
received, status : 0
```

С тех пор существует успешный ответ от TVS, результаты для того сертификата сохранены в кэш. Это означает, что при нажиме кнопки **Directories** снова в течение следующих 86,400 секунд вы не должны связываться с сервером TVS для проверки сертификата. Можно просто обратиться к локальному кэшу.

```
1279: NOT 15:20:59.837086 SECD: saveCertToTvsCache: Saving certificate
in TVS cache with default time-to-live value: 86400 seconds
1287: ERR 15:20:59.859993 SECD: Authenticated the HTTPS conn via TVS
```

Наконец, вы проверяете что ваше соединение с сервером Каталогов, за которым следуют.

```
1302: ERR 15:21:01.959700 JVM: Startup Module Loader|cip.http.ae:?
- listener.httpSucceed: https://14.48.44.80:8443/ccmcip/
xmldirectoryinput.jsp?name=SEP0011215A1AE3
```

Вот пример того, что происходит на сервере CUCM, куда выполняется TVS. Можно собрать журналы TVS с Cisco Унифицированное устройство контроля в реальном времени (RTMT).

Журналы TVS CUCM показывают, что вы, подтверждение связи SSL с телефоном, телефон спрашивает TVS о сертификате Tomcat, тогда TVS, отвечаете, чтобы указать, что с сертификатом совпадают в хранилище сертификата TVS.

```
15:21:01.954 |  debug 14.48.44.202: tvsSSLHandShake Session ciphers - AES256-SHA
15:21:01.954 |  debug TLS HS Done for ph_conn .
15:21:02.010 |  debug    MsgType              : TVS_MSG_CERT_VERIFICATION_REQ
15:21:02.011 |  debug tvsGetIssuerNameFromX509 - issuerName : CN=CUCM8-
Publisher.bbbburns.lab;OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US and Length: 75

15:21:02.011 |  debug CertificateDBCache::getCertificateInformation -
Certificate compare return =0
15:21:02.011 |  debug CertificateDBCache::getCertificateInformation -
Certificate found and equal
15:21:02.011 |  debug    MsgType              : TVS_MSG_CERT_VERIFICATION_RES
```

Хранилище сертификата TVS является списком всех сертификатов, содержавших на веб-странице **Administration> Certificate management OC**.

Вручную Проверьте что Телефонные Соответствия ITL CUCM ITL

Одно общее несовпадение, замеченное при устранении проблем проблем тенденция удалить файл ITL с надеждой, что это решит проблему проверки файла. Иногда удаление файла ITL требуется, но мог бы быть лучший путь.

Когда ALL этих условий встречен, файл ITL только должен быть удален.

- Подпись файла ITL по телефону не совпадает с подписью файла ITL на сервере TFTP CM.
- Подпись TVS в файле ITL не совпадает с сертификатом, представленным TVS.
- Телефон показывает "Проверку, Отказавшую" когда это attemps для загрузки файла ITL или файлов конфигурации.
- Никакая резервная копия не существует старого Частного ключа TFTP.

Вот то, как вы проверяете первые два из этих условий.

Во-первых, можно сравнить контрольную сумму подарка файла ITL на CUCM с контрольной суммой файл ITL телефона. В настоящее время нет никакого способа посмотреть на

MD5sum файла ITL на CUCM от самого CUCM, пока вы не выполняете версию с исправлением для этого идентификатора ошибки Cisco CSCto60209.

Тем временем выполните это со своим любимым GUI или программами CLI:

```
jasburns@jasburns-gentoo /data/trace/jasburns/certs/SBD $ tftp 14.48.44.80
tftp> get ITLSEP0011215A1AE3.tlv
Received 5438 bytes in 0.0 seconds
tftp> quit
jasburns@jasburns-gentoo /data/trace/jasburns/certs/SBD $ md5sum
ITLSEP0011215A1AE3.tlv
b61910bb01d8d3a1c1b36526cc9f2ddc ITLSEP0011215A1AE3.tlv
```

Это показывает, что MD5sum файла ITL в CUCM является **b61910bb01d8d3a1c1b36526cc9f2ddc**.

Теперь можно посмотреть на сам телефон для определения хэша файла ITL, загруженного там:**> Security Параметров настройки Конфигурация> Трастовый Список**.

Это показывает, что совпадают MD5sums. Это означает, что файл ITL по телефону совпадает с файлом на CUCM, таким образом, это не должно быть удалено.

Если это соответствие DOES, необходимо перейти к следующей операции - определяют, совпадает ли сертификат TVS в ITL с сертификатом, представленным TVS. Эта операция немного более включена.

Во-первых, посмотрите на захват пакета телефона, который соединяется с сервером TVS на порте TCP 2445.

Щелкните правой кнопкой мыши на любом пакете в этом потоке в Wireshark, нажмите **Decode As** и выберите **SSL**. Найдите Серверный сертификат, который похож на это:

Посмотрите на сертификат TVS, содержавший в предыдущем файле ITL. Необходимо видеть запись с **серийным номером 2E3E1A7BDAA64D84**.

```
admin:show itl
       ITL Record #:3
                  ----
BYTEPOS TAG             LENGTH VALUE
------- ---             ------ -----
1       RECORDLENGTH    2      743
2       DNSNAME         2
3       SUBJECTNAME     76     CN=CUCM8-Publisher.bbbburns.lab;
                                  OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
4       FUNCTION        2      TVS
5       ISSUERNAME      76     CN=CUCM8-Publisher.bbbburns.lab;
                                  OU=TAC;O=Cisco;L=RTP;ST=North Carolina;C=US
6       SERIALNUMBER    8      2E:3E:1A:7B:DA:A6:4D:84
```

Успех, **TVS.pem** в файле ITL совпадает с сертификатом TVS, представленным в сети. Вы не должны удалять ITL, и TVS представляет корректный сертификат.

Если аутентификация файла все еще отказывает, проверьте остаток предыдущей блок-схемы.

Ограничения и взаимодействия

Восстановите Сертификаты / Восстанавливают Кластер / Окончание срока действия сертификата

Самый важный сертификат является теперь сертификатом CallManager.pem. Секретный ключ этого сертификата используется для подписания всех файлов конфигурации TFTP, который включает файл ITL.

Если файл CallManager.pem восстановлен, новый сертификат CCM+TFTP генерируется с новым секретным ключом. Дополнительно файл ITL теперь подписан этим новым ключом CCM+TFTP.

После того, как вы восстанавливаете CallManager.pem и перезапускаете TVS и Сервис TFTP, это происходит, когда загружается телефон.

1. Телефон пытается загрузить новый файл ITL, подписанный новым CCM+TFTP от сервера TFTP. Телефон имеет только старый файл ITL на этом этапе, и новые ключи не находятся в подарке файла ITL по телефону.
2. Так как телефон не мог найти новую подпись CCM+TFTP в старом ITL, это пытается связаться с сервисом TVS.
   Примечание: Эта часть чрезвычайно важна. Сертификат TVS от старого файла ITL должен все еще совпасть. Если и CallManager.pem и TVS.pem восстановлены в то же точное время, телефоны не в состоянии загрузить любые новые файлы, не удаляя ITL из телефона вручную.
3. Когда телефон связывается с TVS, сервер CUCM, который выполняет TVS, имеет новый сертификат CallManager.pem в Хранилище Сертификата ОС.
4. Сервер TVS возвращает успех и нагрузки телефона новый файл ITL в память.
5. Телефон теперь пытается загрузить файл конфигурации, который был подписан новым ключом CallManager.pem.
6. Так как новый ITL был загружен, недавно файл конфигурации со знаком успешно проверен ITL в памяти.

Ключевые точки:

- Никогда не восстанавливайте и CallManager.pem и сертификаты TVS.pem в то же время.
- Если или TVS.pem или CallManager.pem восстановлены, TVS и TFTP должны быть перезапущены и сброс телефонов для получения новых файлов ITL. Более новые версии CUCM обрабатывают этот телефонный сброс автоматически и предупреждают пользователя во время регенерации сертификата.
- Если несколько серверов TVS существуют (несколько серверов в Группе CallManager), дополнительные серверы могут аутентифицировать новый сертификат CallManager.pem.

Телефоны перемещения между кластерами

При перемещении телефонов от одного кластера до другого с ITLs на месте ITL и Частный ключ TFTP должны быть приняты во внимание. Любой новый файл конфигурации, представленный телефонному MUST, совпадает с подписью в CTL, ITL или подписи в текущем сервисе TVS телефона.

Этот документ объясняет, как удостовериться, файлу и файлам конфигурации нового кластера ITL может доверять текущий файл ITL по телефону. https://supportforums. cisco . com/docs/DOC-15799.

Резервное копирование и восстановление

Сертификат CallManager.pem и секретный ключ выполнены резервное копирование через систему аварийного восстановления (DRS). Если сервер TFTP восстановлен, это должно быть восстановленный от резервной копии так, чтобы мог быть восстановлен секретный ключ. Без секретного ключа CallManager.pem на сервере телефоны с текущими ITLs, которые используют старый ключ, не доверяют подписанным файлам конфигурации.

Если кластер восстановлен и не восстановлен от резервной копии, он точно походит на "Движущиеся Телефоны Между Кластерами" документ. Это вызвано тем, что кластер с новым ключом является другим кластером, насколько затронуты телефоны.

Существует один серьезный дефект, привязанный к резервной копии и восстановлению. Если кластер восприимчив к идентификатору ошибки Cisco CSCtn50405, резервные копии DRS не содержат сертификат CallManager.pem. Это заставляет любой сервер, восстановленный от этой резервной копии генерировать поврежденные файлы ITL, пока не генерируется новый CallManager.pem. Если нет никаких других функциональных серверов TFTP, которые не прошли резервную копию и восстановили операцию, это могло бы означать, что все файлы ITL должны быть удалены из телефонов.

Чтобы проверить, должен ли ваш файл CallManager.pem быть восстановлен, введите **команду show itl**, придерживавшуюся:

```
run sql select c.subjectname, c.serialnumber, c.ipv4address, t.name from
certificate as c, certificatetrustrolemap as r, typetrustrole as t where c.pkid =
r.fkcertificate and t.enum = r.tktrustrole
```

В выходных данных ITL ключевые ошибки искать:

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

и

```
Verification of the ITL file failed.
Error parsing the ITL file!!
```

Предыдущий запрос StructuredQuery Language (SQL) (язык структурированных запросов) ищет сертификаты, которые имеют роль "Проверки подлинности и авторизация". Сертификат CallManager.pem в предыдущем запросе базы данных, который имеет роль Проверки подлинности и авторизация, должен ALSO присутствовать в веб-странице Управления сертификатами администрирования ОС. Если с предыдущим дефектом встречаются, существует несоответствие между сертификатами CallManager.pem в запросе и в веб-странице ОС.

Имена хоста изменения или доменные имена

При изменении имени хоста или доменного имени сервера CUCM это восстанавливает все сертификаты сразу на том сервере. Раздел регенерации сертификата объяснил, что регенерация и TVS.pem и CallManager.pem является "плохой вещью".

Существует несколько сценариев, где изменение имени хоста отказывает и некоторые, где оно работает без проблем. Этот раздел покрывает всех их и связывает их назад с, что вы уже знаете о TVS и ITL от этого документа.

Кластер Одного узла с только ITL (проявляют осмотрительность, это ломается без подготовки),

- С Бизнес-Edition Server или развертываниями только для издателя, и CallManager.pem и TVS.pem восстановлены в то же время при изменении имен хоста.
- Если имя хоста изменено на кластере одного узла без первого использования Корпоративного параметра Отката, покрытого здесь, телефоны не в состоянии проверить новый файл ITL или файлы конфигурации против их текущего файла ITL. Кроме того, они не в состоянии соединиться с TVS, потому что также больше не доверяют сертификату TVS.
- Телефоны отображают ошибку о "Трастовой Отказавшей Проверке Списка", никакие новые изменения конфигурации не вступают в силу, и безопасный сервисный сбой URL.
- Единственное решение, если мера предосторожности в шаге 2 сначала не принята, состоит в том, чтобы вручную удалить ITL из каждого телефона.

Кластер Одного узла и с CTL и с ITL (это может быть временно сломано, но легко исправлено),

- После пробежки переименовывания серверов повторно выполните клиента CTL. Это размещает новый сертификат CallManager.pem в файл CTL, который загружает телефон.
- Новым файлам конфигурации, которые включают новые файлы ITL, можно доверять на основе функции CCM+TFTP в файле CTL.
- Это работает, потому что обновленному файлу CTL доверяют на основе секретного ключа eToken USB, который остается тем же.

Кластер мультиузла с только ITL (это обычно работает, но может быть постоянно сломано, если сделано торопливо),

- Поскольку кластер мультиузла имеет множественные серверы TVS, любому одиночному серверу можно было восстановить его сертификаты без проблемы. Когда телефону предоставляют эту новую, незнакомую подпись, он просит, чтобы другой из серверов TVS проверил новый серверный сертификат.
- Существует две основных проблемы, которые могут заставить это отказывать:
  Если все серверы переименованы и перезагружены в то же время, ни один из серверов TVS не достижим с известными сертификатами, когда серверы и телефоны возвращаются.Если телефон имеет только одиночный сервер в Группе CallManager, дополнительные серверы TVS не имеют никакого значения. См. "сценарий" Кластера Одного узла, чтобы решить это или добавить другой сервер к Группе CallManager телефона.

Кластер мультиузла и с CTL и с ITL (это не может быть постоянно сломано),

- После пробежки переименовывания сервис TVS аутентифицирует новые сертификаты.
- Даже если все серверы TVS недоступны по некоторым причинам, клиент CTL может все еще использоваться для обновления телефонов с новыми сертификатами CallManager.pem CCM+TFTP.

Централизованный TFTP

Когда телефон с ITL загружается, он запрашивает эти файлы: **CTLSEP *<MAC-адрес>* .tlv**, **ITLSEP *<MAC-адрес>* .tlv** и **SEP *<MAC-адрес>* .cnf.xml.sgn**.

Если телефон не может найти эти файлы, он запрашивает **ITLFile.tlv** и **CTLFile.tlv**, который централизованный сервер TFTP предоставляет любому телефону, который запрашивает его.

С централизованным TFTP существует одиночный кластер TFTP, который указывает ко многим другим кластерам sub. Часто это сделано, потому что телефоны на множественных кластерах CUCM совместно используют ту же область DHCP, и поэтому должны иметь тот же Параметр DHCP 150 серверов TFTP. Все IP-телефоны указывают к центральному кластеру TFTP, даже если они регистрируются к другим кластерам. Этот центральный сервер TFTP делает запрос удаленных серверов TFTP каждый раз, когда он получает запрос о файле, он не может найти.

Из-за этой операции централизованный TFTP только работает в гомогенной среде ITL. Все серверы должны выполнить Версию 8.x CUCM или позже, или все серверы должны выполнить версии до Версии 8. x.

Если ITLFile.tlv представлен от Централизованного сервера TFTP, телефоны не доверяют никаким файлам от удаленного сервера TFTP, потому что не совпадают подписи. Это происходит в неоднородном соединении. В гомогенном соединении телефон запрашивает ***<MAC>* ITLSEP .tlv**, который вытягивают от корректного удаленного кластера.

В разнородной среде с соединением версий ранее 8.x и кластеры Версии 8.x, "Готовят Кластер к Откату к Пред 8.0", должен быть включен на кластере Версии 8.x, как описано в идентификаторе ошибки Cisco CSCto87262 и "Защищенные Телефонные Параметры URL", настроенные с HTTP вместо HTTPS. Это эффективно отключает функции ITL по телефону.

Вопросы и ответы

Я могу выключить SBD?

Если SBD и ITL в настоящее время работают, можно только выключить SBD.

SBD может быть временно недоступным по телефонам с Подготовить Кластером для Отката к пред 8.0-дюймовый Корпоративный параметр и путем настройки "Защищенных Телефонных Параметров URL" с HTTP вместо HTTPS. При установке параметра Отката он создает файл ITL со знаком с пустыми функциональными записями. "Пустой" файл ITL все еще подписан, таким образом, кластер должен быть в полностью функциональном состоянии защиты, прежде чем сможет быть включен этот параметр.

После того, как этот параметр включен, и новый файл ITL с пустыми записями загружен и проверен, телефоны принимают любой файл конфигурации, независимо от того кто подписал его.

Не рекомендуется оставить кластер в этом состоянии, потому что ни одна из трех функций,

ранее упомянутых (аутентифицируемые файлы конфигурации, зашифрованные файлы конфигурации и URL HTTPS), не доступна.

Я могу легко удалить файл ITL из всех телефонов, как только потерян CallManager.pem?

В настоящее время нет никакого метода для удаления всего ITLs из телефона, удаленно предоставленного Cisco. Именно поэтому процедуры и взаимодействия, описанные в этом документе, так важны для принятия во внимание.

Существует в настоящее время нерешенное усовершенствование к идентификатору ошибки Cisco CSCto47052, который запрашивает эту функциональность, но это еще не было внедрено.

В промежуточный период новая характеристика была добавлена через идентификатор ошибки Cisco CSCts01319, который мог бы позволить Центру технической поддержки Cisco (TAC) возвращаться к ранее доверяемому ITL, если это все еще доступно на сервере. Это только работает в определенных экземплярах, где кластер находится на версии с этим исправлением дефекта, и где предыдущий ITL существует в резервной копии, сохраненной в специальном местоположении на сервере. Просмотрите дефект, чтобы видеть, имеет ли ваша версия исправление. Свяжитесь с Центром технической поддержки Cisco для пробежки потенциальной процедуры восстановления, объясненной в дефекте.

Если предыдущая процедура не доступна, на телефонные кнопки нужно нажать вручную по телефону для удаления файла ITL. Это - компромисс, который сделан между безопасностью и простотой администрирования. Для файла ITL, чтобы быть действительно безопасным, это не должно быть легко удалено удаленно.

Даже с заданными сценарием нажатиями кнопки с объектами XML Простого протокола доступа к объектам (SOAP), ITL не может быть удаленно удален. Это вызвано тем, что, на этом этапе, доступ TVS (и таким образом доступ через URL Безопасной аутентификации для проверки входящих объектов толчка кнопки SOAP XML) нефункционален. Если URL аутентификации не настроен как безопасный, могло бы быть возможно написать сценарий нажатий клавиш для удаления ITL, но этот сценарий не доступен от Cisco.

Другие методы для сценариев удаленных нажатий клавиш, не используя URL аутентификации могли бы быть доступными от третьей стороны, но эти приложения не предоставлены Cisco.

Наиболее часто используемый метод для удаления ITL является почтовым широковещанием всем телефонным пользователям, которое сообщает им об основной последовательности. Если доступ параметров настройки установлен в **Ограниченный** или **Отключенное**, телефон должен быть сбросом фабрики, поскольку у пользователей нет доступа к Меню Settings телефона.