Защитите внешний пример конфигурации телефонных служб

Содержание

Введение

Этот документ описывает, как настроить Безопасную Внешнюю Телефонную службу. Эта конфигурация может работать с любым сервисом третьей стороны, но для демонстрации, Этот документ использует удаленный сервер Cisco Unified Communications Manager (CUCM).

Внесенный Хосе Виллэлобосом, специалистом службы технической поддержки Cisco.

Предварительные условия

Требования

Компания Cisco рекомендует предварительно ознакомиться со следующими предметами:

- CUCM
- Сертификаты CUCM
- Телефонные службы

Используемые компоненты

Сведения, содержащиеся в данном документе, касаются следующих версий программного обеспечения и оборудования:

- CUCM 10.5. X/CUCM 11. X
- Skinny Client Control Protocol (SCCP) и Протокол SIP звонят регистру с CUCM
- Лабораторная работа его сертификаты альтернативного имени субъекта (SAN) использования.
- Внешний каталог будет на SAN certs.
- Для всей системы на данном примере Центр сертификации (CA) будет тем же, все использование certs знак CA.
- Сервер доменных имен (DNS) и Протокол NTP должны быть настройкой свойства и работой.

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в данном документе, были запущены с конфигурацией по умолчанию. Если ваша сеть является оперативной, гарантируйте понимание потенциального воздействия любого изменения.

Родственные продукты

Данный документ также может использоваться со следующими версиями программного и аппаратного обеспечения:

- CUCM 9. X/10. X/11. X

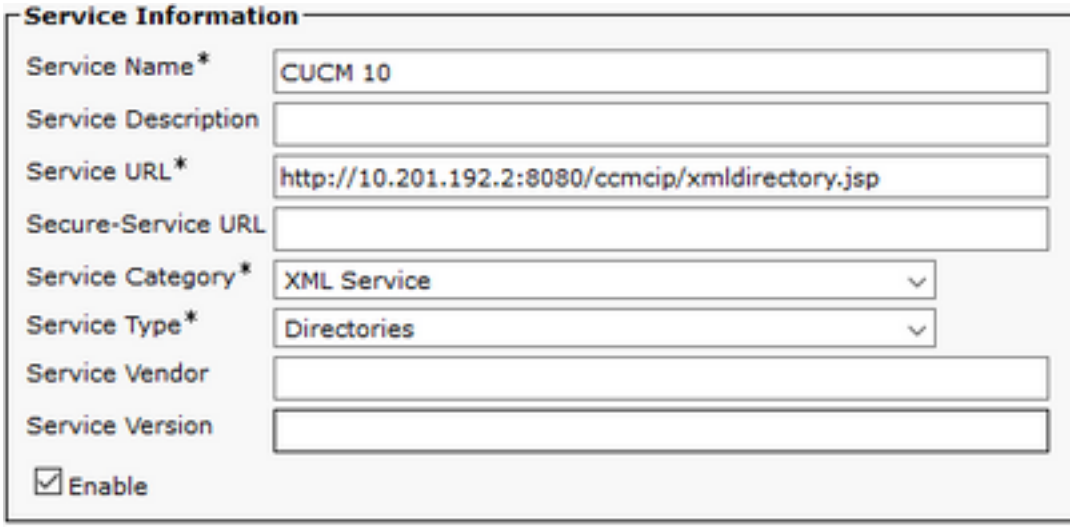
Порядок действий для настройки

Шаг 1. Установите сервисный URL в системе.

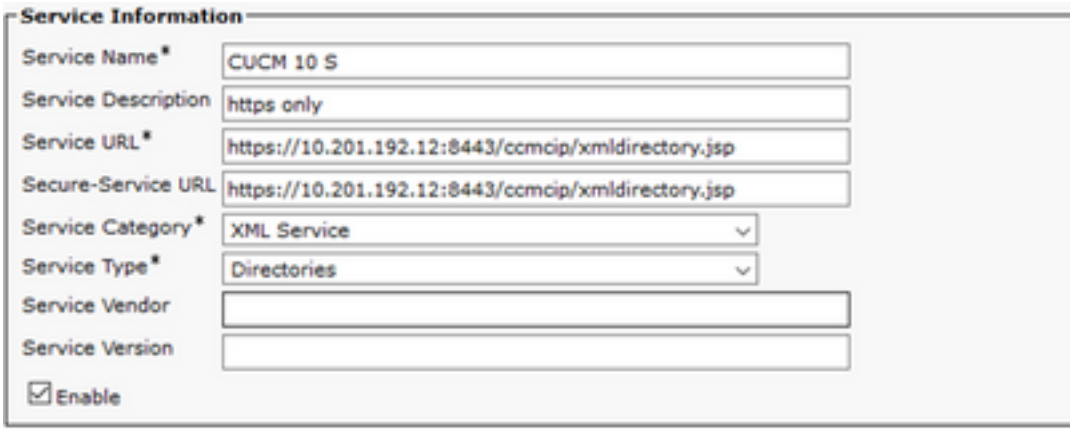
Протокол передачи гипертекста (HTTP) настройки и Протокол передачи гипертекстовых файлов, Безопасный (HTTPS) как доказательство понятий. Заключительная идея состоит в том, чтобы использовать только Безопасный трафик HTTP.

Перейдите к **Устройству> Настройки устройства>, Телефонная служба> Добавляет новый**

HTTP только



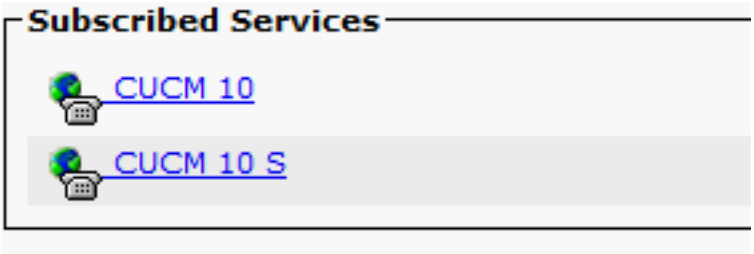
HTTPS только



% Warning: если вы добавляете проверку для **Подписки Предприятия,** шаг два может быть пропущен. Однако это изменение перезагружает все телефоны, поэтому гарантируйте понимание потенциального воздействия.

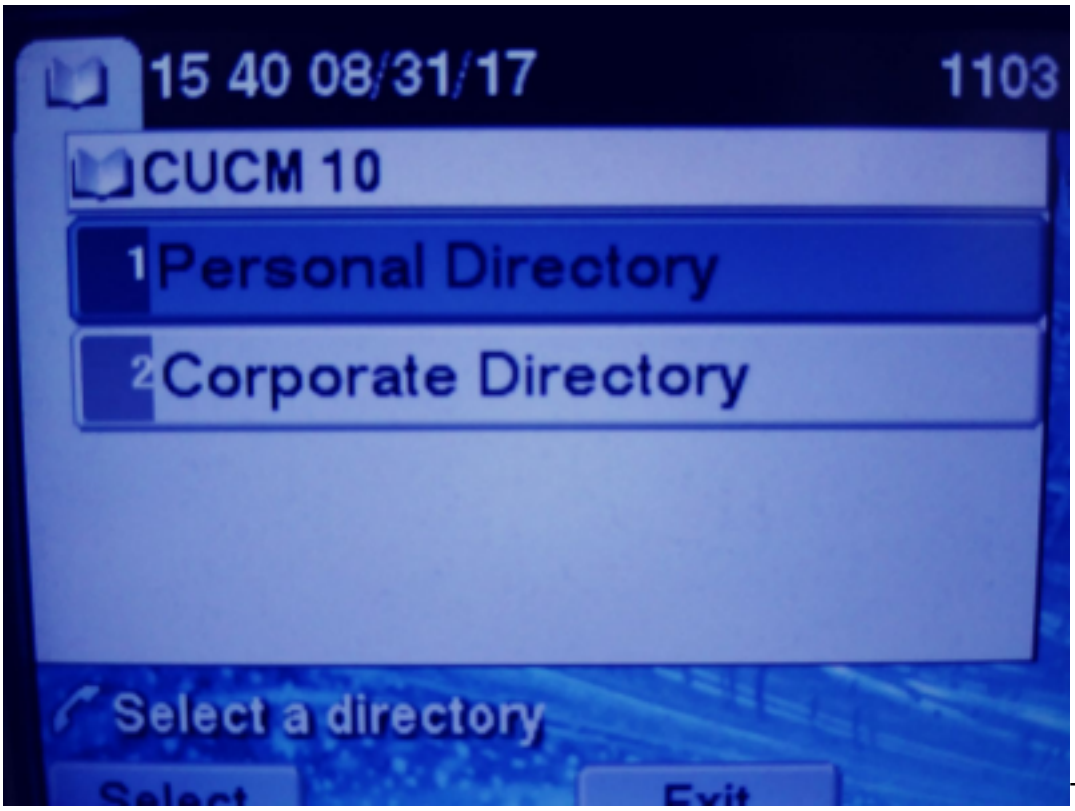
Шаг 2. Подпишите телефоны на сервисы.

Natigate к **Device> Phone>> сервис Абонента/Отменять подписку.**



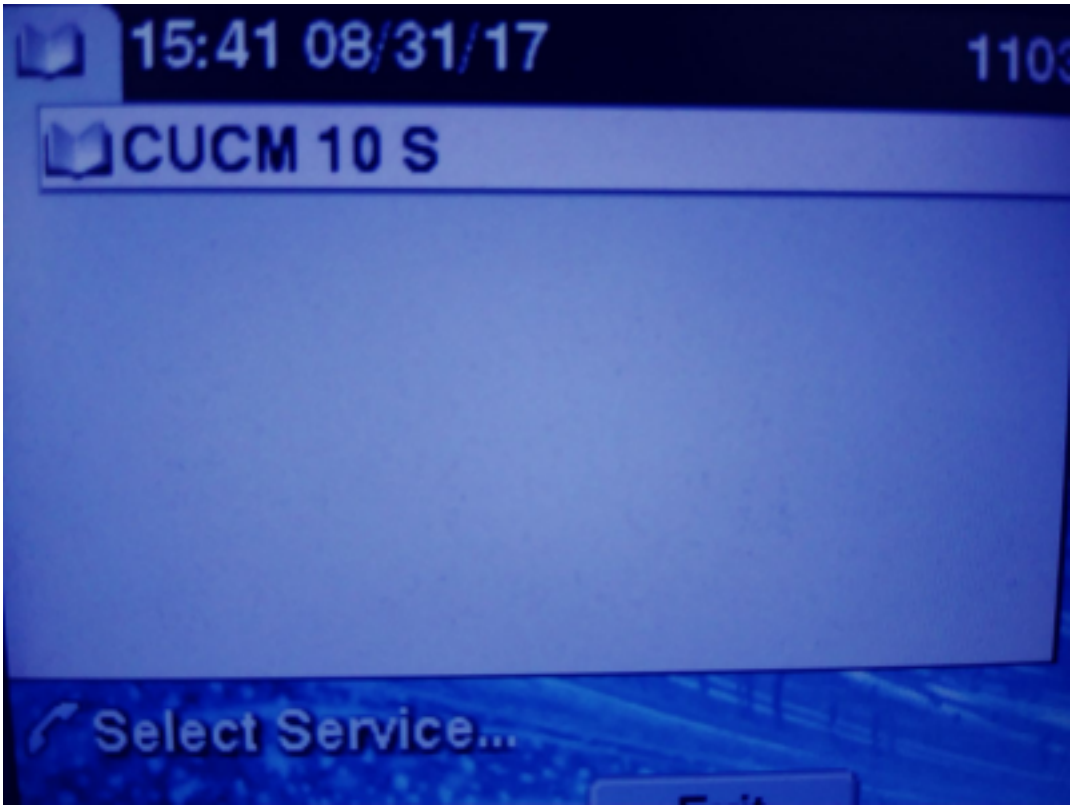
На этом этапе, если приложение предлагает HTTP, необходимо быть в состоянии достигнуть сервиса, но https подключен все еще.

HTTP



TTP

HTTPS



HTTPS покажет “Хост, не найденный” ошибка из-за факта, сервис TVS не может аутентифицировать это для телефона.

Шаг 3. Загрузите Внешние Трудовые книжки к CUCM.

Загрузите Внешний Сервис, поскольку **Tomcat доверяет только**. Гарантируйте, что сервисы перезагружены на всех узлах.

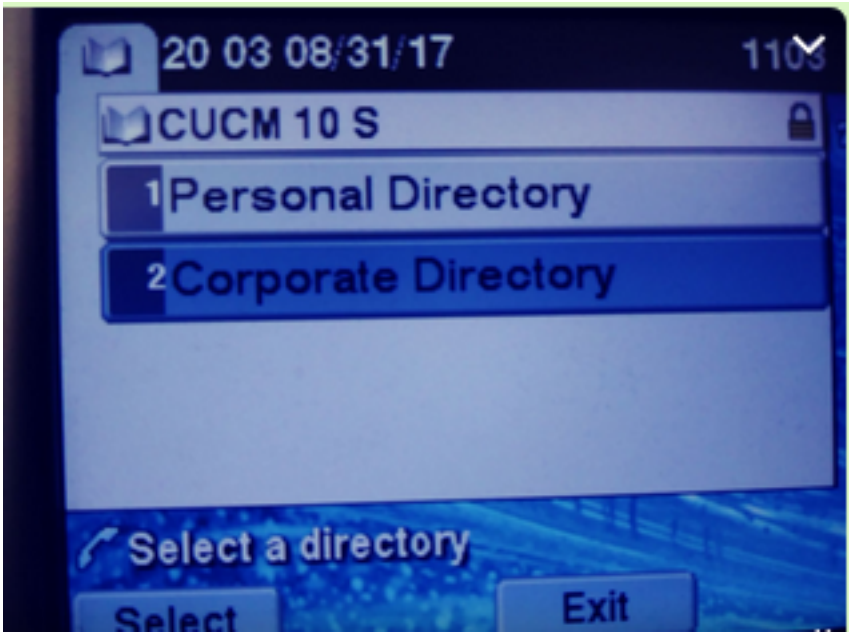
Этот тип certs не сохранен по телефону, скорее телефон должен свериться с сервисом TVS, чтобы видеть, устанавливает ли это Подключение HTTPS.

Перейдите **admin ОС> Сертификат> загрузка Сертификата.**

От SSH перезагружает сервис Tomcat CUCM на всех узлах.

```
admin:utils service restart Cisco Tomcat
Do not press Ctrl+C while the service is restarting. If the service has not rest
arted properly, execute the same command again.
Service Manager is running
```

После этих шагов телефоны должны быть в состоянии обратиться к сервису HTTPS без проблем



Частый задают вопросы (FAQ)

После того, как сертификатами обмениваются, HTTPS все еще отказывает с "хостом, не найденным".

- Проверьте узел, где телефон, который его регистр и гарантирует вам, видит сертификат третьей стороны на узле.

- Перезагрузите tomcat на определенном узле.

- Проверьте DNS, гарантируйте, что может быть решено Общее имя (CN) сертификата.

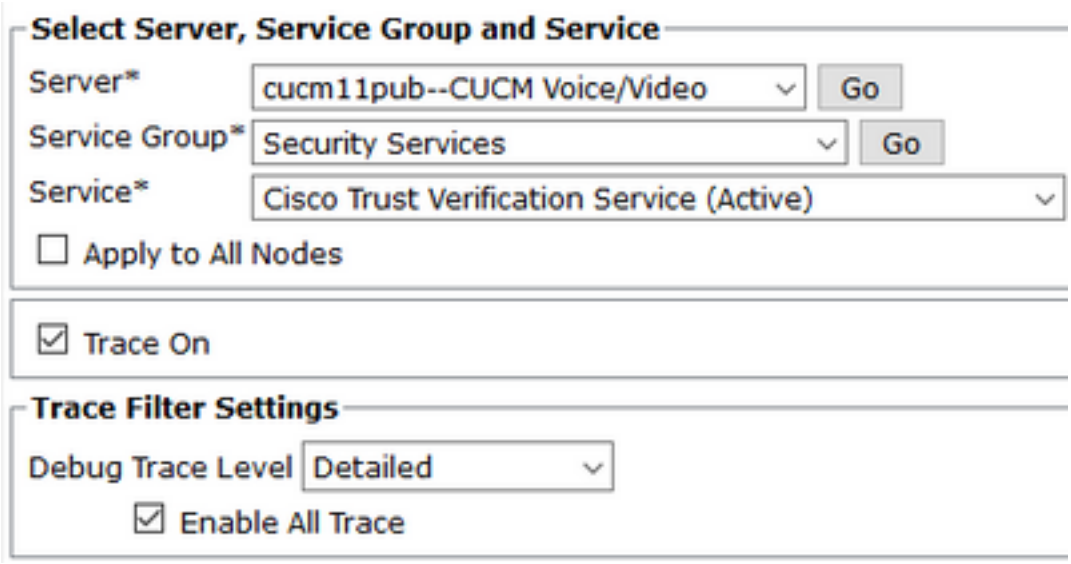
Устранение неисправностей

Соберите журналы TVS CUCM должны предоставить вас хорошая информация

Перейдите к **RTMT> Система> Трассировка и регистрируйте Центральный>, Собирают файлы журнала**

> **Примечание:** Соберите журналы от всех узлов и гарантируйте , что журналы TVS установлены в подробный.

TVS регистрирует набор к подробному



Пример трассировки

```
11:17:38.291 |   debug CTVSChangeNotifyServer::ProcessChangeNotification () -
CDBString=<msg><type>DBL</type><table>certificate</table><tableid>46</tableid><action>I</action>
<user>repl</user><time>1504203458</time><new><cdrserver>2</cdrserver><cdrtime>1504203457</cdrtim
e><pkid>e6148ee3-3eb5-e955-fa56-
2baa538a88fb</pkid><servername>cucm11pub</servername><subjectname>CN=10.201.192.12,OU=RCH,O=Cisc
o,L=RCH,ST=Tx,C=US</subjectname><issuername>CN=pablogon-
CA,DC=rcdncollab,DC=com</issuername><serialnumber>3d00000008230ded92f687ec03000000000008</serial
number><certificate></certificate><ipv4address>10.201.192.13</ipv4address><ipv6address></ipv6add
ress><timetolive>NULL</timetolive><tkcertificatedistribution>1</tkcertificatedistribution><ifx_r
eplcheck>6460504654345273346</ifx_replcheck></new></msg>
11:17:38.291 |   debug CTVSChangeNotifyServer::ProcessChangeNotification () - Database table
"certificate" has been changed
11:17:38.291 |   debug CTVSChangeNotifyServer::ProcessChangeNotification () - Looking up the
roles for
11:17:38.291 |   debug Pkid : fead9987-66b5-498f-4e41-c695c54fac98
11:17:38.291 |   debug CTVSChangeNotifyServer::ProcessThreadProc () - Waiting for DBChange
Notification
11:17:38.300 |   debug CTVSChangeNotifyServer::ProcessThreadProc () - DBChange Notification
received
11:17:38.300 |   debug CTVSChangeNotifyServer::ProcessChangeNotification () -
CDBString=<msg><type>DBL</type><table>certificatetrustrolemap</table><tableid>50</tableid><actio
n>I</action><user>repl</user><time>1504203458</time><new><cdrserver>2</cdrserver><cdrtime>150420
3457</cdrtime><pkid>5ae6e1d2-63a2-4590-bf40-1954bfa79a2d</pkid><fkcertificate>e6148ee3-3eb5-
e955-fa56-
2baa538a88fb</fkcertificate><tktrustrole>7</tktrustrole><ifx_replcheck>6460504654345273346</ifx_
replcheck></new></msg>
11:17:38.300 |   debug CTVSChangeNotifyServer::ProcessChangeNotification () - Database table
"certificatetrustrolemap" has been changed
11:17:38.300 |   debug CTVSChangeNotifyServer::ProcessThreadProc () - Waiting for DBChange
Notification
11:17:46.811 |   debug updateLocalDBCache : Refreshing the local DB certificate cache
11:34:00.131 |   debug Return value after polling is 1
11:34:00.131 |   debug FD_ISSET i=0, SockServ=14

11:34:00.131 |   debug Accepted TCP connection from socket 0x00000014
```