TRUSTED TLS для APACHE v2.2.x

РУКОВОДСТВО ПО УСТАНОВКЕ И НАСТРОЙКЕ

СОДЕРЖАНИЕ

ОПИСАНИЕ ПРОДУКТА

Программный продукт Trusted TLS - это доработанное программное решение mod_ssl (встраивается взамен имеющегося модуля mod_ssl веб-сервера Apache), позволяющее использовать российские стандарты криптографической защиты информации в веб-сервере Apache.

Без использования криптопровайдера "КриптоПро CSP" модуль обеспечивает все функции аутентификации и криптографии с использованием RSA и Diffie-Helman шифров. Установка криптопровайдера позволяет применять и сертифицированные алгоритмы шифрования. Trusted TLS работает через протоколы TLS/SSL с помощью набора инструментов OpenSSL. Он поставляется как динамически загружаемая библиотека (mod_digt_tls.so обеспечивает работу по протоколу TLS на ГОСТ алгоритмах, используя низкоуровневые вызовы функции криптопровайдера КриптоПро CSP).

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

1. Совместимость с продуктом КриптоПро CSP 3.6 и КриптоПро TLS от компании «Крипто-Про», а также другими реализациями TLS, выполненными в соответствии с проектом рекомендаций IETF «GOST 28147-89 Cipher Suites for Transport Layer Security (TLS)».

2. Поддержка протоколов TLS/SSL и алгоритмов

- TLS v.1 для сертифицированных алгоритмов
- SSL v.2/v.3 для не сертифицированных алгоритмов
- Поддержка шифросюиты (CipherSuite) 0x0081 (TLS_GOSTR341001_WITH_28147_CNT_IMIT)
- 256-битное шифрование для веб-приложений в соответствии с ГОСТ 28147-89
- Поддержка сертификатов стандарта X.509 версии 3 с ключами по ГОСТ Р 34.10-2001

3. Поддержка платформ

- Windows XP/Vista/7, Windows Server 2003/2008/2008 R2 32 bit (64 bit в тестовых целях)
- Red Hat Enterprise Linux 4/5/6 32/64 bit, Ubuntu 10.04.3 LTS (lucid) 64 bit
- FreeBSD 7/8 32 bit
- Solaris 10 SPARC/Intel 32/64 bit

4. Возможность односторонней (Сервера) и двусторонней (Клиента и Сервера) аутентификации.

5. Поддержка аутентификации с использованием ГОСТ-сертификата при проксировании.

6. Поддержка разных ГОСТ-сертификатов для нескольких виртуальных хостов.

ИНСТРУКЦИЯ ПО УСТАНОВКЕ

Данное руководство описывает установку и настройку продукта Trusted TLS, который собран на основе Apache HTTP Server версии 2.2.21.

Данный продукт поставляется в виде следующих пакетов:

TrustedTLS_<version>_<platform>_<release>.zip (для Windows)
TrustedTLS_<version>_<platform>_<release>.tgz (для Unix-систем с КриптоПро CSP 3.6 и 3.6 R2)
TrustedTLS_<version>-**R3**_<platform>_<release>.tgz (для Unix-систем с КриптоПро CSP 3.6 **R3** и 3.6 **R4**)
mod_digt_tls-<version>-<release>.<platform>.rpm (для Unix-систем)

Перед установкой Trusted TLS необходимо установить криптопровайдер КриптоПро CSP с клиентской (если такая предусмотрена для используемой Вами ОС) или серверной лицензией.

Дистрибутив криптопровайдера необходимой версии можно взять по следующему адресу: http://www.cryptopro.ru/cryptopro/download/default.asp?n=1

Для приема TLS-соединений на сервере необходимо сформировать (или подключить существующую) ключевую пару и установить серверный сертификат, выданный в Удостоверяющем центре.

Установка СКЗИ КриптоПро CSP, Apache HTTP Server и Trusted TLS

Unix-системы

Для удобства в дальнейшем будет использоваться следующее обозначение *<**arch**>* для указания идентификатора платформы, на которой установлено СКЗИ КриптоПро CSP:

- для 32bit Solaris Intel, Linux, FreeBSD: ***ia32***,
- для 64bit Solaris Intel, Linux: ***amd64***,
- для 32bit Solaris SPARC: ***sparc***,
- для 64bit Solaris SPARC: ***sparcv9***,
- для 32bit AIX: ***ppc***,
- для 64bit AIX: ***ppc64***.

1. Для автоматического скачивания СОС установите пакет curl – библиотеку для скачивания файлов по FTP, HTTP и некоторым другим протоколам.
2. Установите криптопровайдер КриптоПро CSP из дистрибутива.
   2.1. Для систем с поддержкой RPM-пакетов установите RPM-пакеты командой *“rpm -ivh <файл_пакета>”* в следующем порядке:
      2.1.1. пакеты, устанавливаемые предварительно при необходимости:
      libstdc++ версии 3.4 - GNU Standard C++ Library 3.4
      cprocsp-compat-altlinux – дополнение к LSB для AltLinux
      cprocsp-compat-splat – дополнение к LSB для Linux SPLAT
      2.1.2. основные пакеты криптопровайдера (для варианта исполнения КС1):
      lsb-cprocsp-base
      lsb-cprocsp-rdr – базовые считыватели
      lsb-cprocsp – библиотеки криптопровайдера
      lsb-cprocsp-capilite – библиотеки CryptoAPI 2.0 Lite и утилиты
      2.1.3. дополнительно может потребоваться установка вспомогательных пакетов, например:
      lsb-cprocsp-rdr-pcsc – считыватели токенов и смарт-карт
   2.2. На платформе **AIX** для установки **32-битной** версии КриптоПро CSP распакуйте файл aix-ppc.zip по команде

      unzip aix-ppc.zip

Перед запуском скрипта **install.sh** на платформе **AIX** (**ppc64**, 64-бит) (команда `uname -m` дает не пустой результат), измените его для установки 32-битных пакетов. Для этого нужно закомментировать строку

```
# ARG=64
```

и в следующей строке исправьте **ppc64** на ppc

```
arch=ppc
```

Запустите скрипт **install.sh**.

2.3. Для систем на платформе Solaris требуется:

2.3.1. Распаковать файл дистрибутива, например, **solaris-sparc.zip**. Для этого выполните команды:

```
unzip solaris-sparc.zip
cd solaris-sparc/release/
tar xvf sparc-32.tar
```

В процессе распаковки будут извлечены следующие пакеты, доступные для установки.

CPROcspb	Base directories and scripts (базовый пакет)
CPROcspd	Crypto service provider documentation and headers (основная документация и h-файлы)
CPROrdr	Readers and support library for CSP (32 bit) (базовые считыватели)
CPROrdg	GUI components of CSP readers (32 bit) (графические утилиты для работы со считывателями)
CPROrdp	PC/SC components of CSP readers (32 bit)
CPROcp2	Crypto service provider library (32 bit). Level KC2. (библиотеки криптопровайдера уровня KC2)
CPROcsp	Crypto service provider library (32 bit) (библиотеки криптопровайдера)
CPROdrv	Crypto service provider kernel loadable module (32 bit) (загружаемые на уровня ядра модули)
CPROcpl	Crypto API lite (32 bit) (библиотеки CryptoAPI 2.0 Lite и утилиты)
CPROstnl	Stunnel
CPROp11	CryptoPro PKCS#11 (32 bit) (считыватели токенов и смарт-карт)
CPROp11d	CryptoPro PKCS#11 devel (пакет для разработчика)
CPROdrvd	Crypto service provider sources and executables for tests of kernel loadable module (32 bit) (пакет для разработчика)

2.3.2. Для изменения состава устанавливаемых пакетов криптопровайдера нужно найти и отредактировать в файле solaris-sparc/release/setup.sh одну строчку, в которой перечислены все пакеты:

```
…
for i in CPROcspb CPROcspd CPROrdr CPROrdg CPROrdp CPROcp2 CPROcsp CPROdrv CPROcpl
CPROstnl CPROp11 CPROp11d CPROdrvd; do
…
```

Минимальный требуемый набор пакетов указан в строке ниже:

```
…
for i in CPROcspb CPROrdr CPROrdp CPROcsp CPROdrv CPROcpl; do
```

…

2.3.3. После определения списка устанавливаемых пакетов для инсталляции криптопровайдера выполните команды:

```
cd solaris-sparc/release
./setup.sh
```

3. Возможно, в файле /etc/opt/cprocsp/config.ini (config64.ini для 64-разрядной сборки Trusted TLS) потребуется подправить строку

   "libcurl.so" = "/usr/local/lib/libcurl.so",

   указав правильное расположение в системе файла libcurl.so (см п. 1).

4. Введите лицензию на КриптоПро CSP с помощью команды

   */opt/cprocsp/sbin/<**arch**>/cpconfig -license -set XXXXX-XXXXX-XXXXX-XXXXX-XXXXX*

5. Без установленной лицензии КриптоПро CSP будет работать 3 месяца в полнофункциональном режиме.
6. Перед установкой Trusted TLS на платформе Solaris 10 sparc32 необходимо убедиться в наличии установленных библиотек GCC 3.4 (для функционирования Trusted TLS требуются **libgcc_s.so.1** и **libstdc++.so.6**). При их отсутствии можно установить пакеты **CSWgcc3corert** и **CSWgcc3g++rt** (согласно инструкции с http://www.blastwave.org).
7. Перед установкой Trusted TLS на платформе AIX 5.3 необходимо убедиться в наличии установленных библиотек GCC 4.2 (для функционирования Trusted TLS требуется **libgcc_s.a**). При их отсутствии можно установить пакет libgcc-4.2.0.
8. Установите веб-сервер Apache с поддержкой TLS-шифрования по ГОСТ путем распаковки архива в корень раздела. Архив представляет из себя tgz-файл, включающий в себя Apache HTTP Server и подгружаемый модуль mod_digt_tls. **Внимание!** Дистрибутив Trusted TLS надо брать соответствующий используемой версии КриптоПро CSP (см. начало раздела «Инструкция по установке»).
9. Альтернативно этому варианту установить Apache HTTP Server на ОС **RHEL** можно из rpm-пакета, поставляемого с дистрибутивом ОС Linux. Установку Trusted TLS в этом случае можно провести из rpm-пакета, который нужно непосредственно скачать из репозитория http://repos.trusted.ru/yum/centos/ и использовать команду для установки

   rpm -ivh mod_digt_tls-2.2.21-XXX.rhelX.i386.rpm

   или, предварительно скачав файл trusted.repo, настроить используемые на хосте репозитории для утилиты yum, и установить Trusted TLS командой

   yum install mod_digt_tls.

   Файл trusted.repo обычно размещается в каталоге **/etc/yum.repos.d**.

10. На платформе AIX 5.3 можно установить Trusted TLS из rpm-пакета по команде

    rpm -ivh mod_digt_tls-2.2.17-362.aix5.3.ppc.rpm

    Перед этим требуется провести установку Apache-сервера из rpm-файла, который можно получить с сайта AIX 5L Open Source Packages. Возможно, для его этого придется установить в указанном порядке следующие пакеты с указанного ресурса:

    ```
    rpm -i apr-1.4.2-2.aix5.2.ppc.rpm
    rpm -i info-4.13a-2.aix5.1.ppc.rpm
    rpm -i readline-6.1-3.aix5.1.ppc.rpm
    rpm -i unixODBC-2.3.0-1.aix5.1.ppc.rpm
    ```

```
rpm -i libgpg-error-1.10-1.aix5.1.ppc.rpm
rpm -i libgcrypt-1.4.6-1.aix5.1.ppc.rpm
rpm -i libtasn1-2.9-1.aix5.1.ppc.rpm
rpm -i lzo-2.04-1.aix5.1.ppc.rpm
rpm -i gnutls-2.10.4-1.aix5.1.ppc.rpm
rpm -i freetds-0.82-2.aix5.1.ppc.rpm
rpm -i sqlite-3.7.3-1.aix5.1.ppc.rpm
rpm -i apr-util-1.3.10-2.aix5.1.ppc.rpm
rpm -U openssl-1.0.0c-1.aix5.1.ppc.rpm
rpm -ihv httpd-2.2.17-1.aix5.1.ppc.rpm
```

При дальнейшей настройке конфигурационных файлов Apache-сервера, установленных таким образом, следует запомнить их расположение: **/opt/freeware/etc/httpd/conf/httpd.conf** и **/opt /freeware/etc/httpd/conf/extra/httpd-ssl.conf**. В каталог **/opt/freeware/etc/httpd/conf/ssl.key** выкладывается ГОСТ-сертификат сервера. В каталоге **/opt/freeware/etc/httpd/conf/ssl.crt** размещаются сертификаты УЦ, клиенты которых будут обслуживаться сервером.
Запускать Apache-сервер по команде

```
/etc/rc.d/init.d/httpd start
```

Windows

1. Установите КриптоПро CSP с помощью программы-инсталлятора. Для целей тестирования следует устанавливать криптопровайдер в исполнении КС1, т.к. он не требует использования аппаратного ДСЧ для формирования ключевой пары.
2. Введите лицензию на КриптоПро CSP в соответствии с документацией. Без лицензии СКЗИ будет работать в полнофункциональном режиме в течение одного или трех месяцев (в зависимости от версии).
3. Настройте требуемые считыватели КриптоПро CSP через панель управления в соответствии с Инструкцией по использованию КриптоПро CSP.
4. Установите Trusted TLS путем распаковки архива в корень выбранного для установки диска.
5. Если в системе отсутствует **Microsoft Visual C++ 2008 SP1 Redistributable Package**, то установите его перед запуском Trusted TLS. Скачать Microsoft Visual C++ 2008 SP1 Redistributable Package можно по ссылкам:
   http://www.microsoft.com/download/en/details.aspx?id=5582 – файл vcredist_x86.exe и
   http://www.microsoft.com/download/en/details.aspx?id=2092 - файл vcredist_x64.exe.

Создание ключевой пары и запроса на сертификат, выпуск серверного сертификата в УЦ и установка его в хранилище

Информация данного раздела адресована в основном администраторам, которые развертывают веб-сервер в тестовом режиме, а также тем, кто обслуживается в Удостоверяющих центрах, которые предоставляют возможность самостоятельно сформировать ключевую пару и запрос на сертификат. Многие Удостоверяющие центры, выдающие сертификаты для использования в информационных системах общего пользования, работают по схеме, при которой ключевой контейнер формируется оператором УЦ на отчуждаемый носитель (например, дискету, флеш-диск или токен), который затем передается клиенту. Поэтому если ключевой контейнер и сертификат сервера имеется в наличии, то можно перейти к следующему разделу данного Руководства.

Сертификат TLS-сервера отличается от сертификатов пользователей и других сервисов следующими характеристиками:

- В имени владельца (т.е. в атрибуте «Субъект» / «Subject») сертификата элемент CN должен содержать доменное (DNS) имя сервера, на котором развертывается TLS-сервер, например, web-portal.yourcompany.ru. Для серверов, не имеющих доменного имени, возможно использование статического ip-адреса. Если данное требование не будет соблюдено, то некоторые версии браузеров без дополнительной настройки параметров безопасности могут отказаться устанавливать защищенное по SSL/TLS-протоколу соединение с сервером.

- Ключевая пара должна обеспечивать возможность шифрования данных, что определяется наличием значений вариантов использования «Шифрование ключей, Шифрование данных» / «Key Encipherment, Data Encipherment» в расширении «Использование ключа» / «Key Usage» (KU) серверного сертификата.

- В расширении «Улучшенный ключ» / «Enhanced Key Usage» (EKU) должен содержаться объектный идентификатор 1.3.6.1.5.5.7.3.1, обозначающий вариант использования сертификата «Проверка подлинности сервера» / «Server Authentication».

Unix-системы

Генерация запроса и установка серверного сертификата должна производиться под учетной записью пользователя, от имени которого будет функционировать веб-сервер.

Перед генерацией ключевого контейнера требуется определиться с его именем и местоположением. В данном примере описывается создание ключевого контейнера с именем «tlsserver» и расположенном на жестком диске (считыватель «HDIMAGE»). Имя ключевого контейнера должно быть уникальным, поэтому предварительно рекомендуется убедиться в отсутствии контейнера с выбранным именем. Список существующих ключевых контейнеров можно посмотреть командой:

`/opt/cprocsp/bin/<arch>/csptestf -keyset -provtype 75 -enum_containers -verifycontext -fqcn`

Для создания нового ключевого контейнера с формированием запроса на сертификат TLS-сервера выполните команду:

`/opt/cprocsp/bin/<arch>/cryptcp -creatrqst -provtype 75 -ex -cont "\\\\.\\HDIMAGE\\tlsserver" -dn "CN=web-portal.yourcompany.ru, O=My Company, C=RU, E=test@test.ru" -certusage "1.3.6.1.5.5.7.3.1" /tmp/server-gost.csr`

Параметры –cont и –dn измените в соответствии с вашими данными. Параметр –cont должен быть уникальным для каждого нового запроса. В случае, если Вы укажите имя существующего контейнера, то операция завершится с ошибкой 0x8009000f («Объект уже существует» / «Object already exists»). В данном примере ключевой контейнер будет создан в хранилище CSP на файловой системе, если Вам нужно сгенерировать ключ на токене или дискете, измените его согласно руководству для КриптоПро CSP.

В процессе формирования ключа для инициализации датчика случайных чисел Вам может потребоваться нажать поочередно несколько клавиш на клавиатуре. После генерации закрытого ключа будет предложено защитить его паролем, который в дальнейшем можно менять командой:

`/opt/cprocsp/bin/<arch>/csptestf -passwd -change <новый_пин_код> -provtype 75 -container <имя_контейнера>`

Сгенерированный файл запроса следует обработать в Удостоверяющем центре. При тестировании можно использовать тестовый УЦ компании КриптоПро: http://www.cryptopro.ru/certsrv/certrqxt.asp. Сертификат, выданный по запросу в УЦ, сохраните (в DER или Base64 формате, последний в данном случае удобнее) в файл /tmp/server-gost.cer на сервере.

Для установки полученного сертификата выполните следующую команду:

`/opt/cprocsp/bin/<arch>/cryptcp -instcert -provtype 75 /tmp/server-gost.cer`

и в появившемся пронумерованном списке укажите номер соответствующего контейнера и пароль (если был задан на данный контейнер).

В случае если предыдущая команда завершится с ошибкой, можно попробовать установить серверный сертификат другим способом:

`/opt/cprocsp/bin/<arch>/certmgr -inst -store uMy -file /tmp/server-gost.cer -cont "\\\\.\\HDIMAGE\\tlsserver"`

Если одна из предыдущих команд установки сертификата выполнилась с кодом ошибки 0x00000000, то серверный сертификат установлен в личное хранилище текущего пользователя со ссылкой на закрытый ключ.

Windows

Генерация запроса и установка серверного сертификата должна производиться под учетной записью

пользователя, от имени которого будет осуществляться запуск веб-сервера.

В тестовых целях создать новый ключевой контейнер и сформировать запрос на сертификат TLS-сервера Вы можете через веб-интерфейс тестового Удостоверяющего центра компании КриптоПро: http://www.cryptopro.ru/certsrv/certrqma.asp. Данная функциональность доступна только из браузера Internet Explorer с разрешенными элементами ActiveX и сценариями.

Откройте форму запроса на сертификат, дождитесь, когда загрузятся необходимые модули ActiveX и заполнится список CSP в разделе «Параметры ключа» и заполните форму следующим образом:

- Заполните соответствующие сведения в полях раздела «Идентифицирующие сведения». Например, в поле «Имя» введите «web-portal.yourcompany.ru», и оставьте значение «*RU*» в поле «Страна».
- В списке «Нужный тип сертификата» выберите «Сертификат проверки подлинности сервера».
- В разделе «Параметры ключа» рекомендуется включить режим «Заданное пользователем имя ключевого контейнера» и в появившемся поле «Container Name» ввести, например, «*tlsserver*» (только предварительно рекомендуется убедиться, что контейнер с таким именем отсутствует в системе, посмотреть список существующих контейнеров можно через Панель управления КриптоПро CSP на закладке «Сервис», например, с помощью операции «Скопировать контейнер»).
- Остальные поля можно оставить по умолчанию при условии, что в выпадающем списке «CSP» выбран один из криптопровайдеров компании КриптоПро, и ни один из них не содержит в имени «KC2».

После заполнения формы нажмите кнопку «Выдать», при необходимости предоставьте информацию для инициализации ДСЧ. После генерации закрытого ключа будет предложено защитить его паролем, который в дальнейшем можно менять через Панель управления КриптоПро CSP на закладке «Сервис».

После отправки запроса и его автоматической обработки в УЦ откроется страница установки выданного сертификата. Дождитесь окончания загрузки данной страницы и нажмите на ссылку «Установить этот сертификат». При первой установке будет сначала предложено установить сертификат Центра сертификации, с установкой которого следует согласиться. Если для ключевого контейнера был задан пароль, то появится окно, в котором надо будет его ввести для записи выданного сертификата в контейнер.

Теперь серверный сертификат установлен в личное хранилище текущего пользователя со ссылкой на закрытый ключ.

Подготовка к настройке веб-сервера, установка сертификатов УЦ

Для того чтобы стартовать Trusted TLS на ГОСТ сертификатах, необходимо чтобы на стороне сервера были соблюдены следующие условия:

- Веб-сервер должен запускаться из-под учетной записи пользователя, отличной от root.
- Под учетной записью пользователя, из-под которой будет запускаться веб-сервер, должен быть доступен контейнер с ключевой парой, управляемый СКЗИ КриптоПро CSP. Для поддержки ключевых контейнеров, находящихся на отчуждаемых носителях (флеш-дисках, токенах и др.), потребуется подключить к криптопровайдеру соответствующие считыватели. Описание их подключения приводится в руководстве на CSP.
- В личное ("my") хранилище сертификатов пользователя, из-под учетной записи которого будет запускаться веб-сервер, должен быть установлен сертификат TLS-сервера, соответствующий ключевому контейнеру из предыдущего пункта. Данный сертификат должен быть установлен со ссылкой на ключевой контейнер.
- Серверный сертификат должен быть дополнительно сохранен на сервере в файл в формате Base64.
- Сертификат корневого УЦ, выдавшего серверный сертификат, должен быть в хранилище корневых сертификатов ("root"). В случае если УЦ, выдавший сертификат сервера, является подчиненным, его сертификат должен быть установлен в хранилище сертификатов промежуточных УЦ ("ca"), а корневой сертификат вышестоящего УЦ – в хранилище корневых сертификатов ("root").

- Должны быть скорректированы настройки в файлах конфигурации веб-сервера.

Получение и установка серверного сертификата в личное ("my") хранилище пользователя с привязкой к ключевому контейнеру описана в предыдущем разделе, настройка файлов конфигурации веб-сервера – в следующем. Описание остальных настроек приводится далее.

Unix-системы

1. Проверка наличия установленного серверного сертификата

   На данном этапе рекомендуется удостовериться, что в личном ("my") хранилище сертификатов пользователя установлен серверный сертификат с привязкой к ключевому контейнеру. Для этого выполните следующую команду:

   */opt/cprocsp/bin/<**arch**>/certmgr -list -store uMy -cert*

   В выведенном на экран списке в блоке соответствующего сертификата должна быть строчка:

   PrivateKey Link: Yes. Container: <имя_контейнера>

2. Получение файла серверного сертификата

   При настройке файлов конфигурации веб-сервера потребуется указать имя файла серверного сертификата в формате DER или Base64. Для варианта Base64 первая и последняя строки файла сертификата должны быть соответственно «-----BEGIN CERTIFICATE-----» и «-----END CERTIFICATE-----» (без кавычек). Если у Вас отсутствует файл с данным сертификатом, но он установлен в хранилище, то его можно экспортировать следующей командой:

   */opt/cprocsp/bin/<**arch**>/certmgr -export -store uMy -dest* /tmp*/server-gost.cer -cert -dn "CN=web-portal.yourcompany.ru"*

   Значение параметра –dn измените на соответствующее. Если сертификат в хранилище единственный, то данный параметр можно не указывать. Если в хранилище установлено несколько сертификатов, у которых совпадает значение поля «CN», то значение параметра –dn следует конкретизировать. Список установленных сертификатов можно посмотреть командой:

   */opt/cprocsp/bin/<**arch**>/certmgr -list -store uMy -cert*

3. Установка сертификата Удостоверяющего центра

   Для выполнения данной операции загрузите на сервер сертификат УЦ (в формате DER или Base64).

   Сертификат корневого Удостоверяющего центра должен быть установлен в хранилище корневых ("root") сертификатов.

   Для установки корневого сертификата УЦ в машинное хранилище корневых сертификатов получите привилегии супер-пользователя и выполните следующую команду:

   */opt/cprocsp/bin/<**arch**>/certmgr -inst -store mRoot -file /path/to/root-gost.cer -cert*

   В случае если серверный сертификат издан промежуточным УЦ, то сертификат последнего должен быть установлен в хранилище сертификатов промежуточных УЦ ("ca"), а корневой, которым подписан промежуточный, - в хранилище корневых сертификатов ("root"). Установка сертификата корневого УЦ описана выше.

   Установка промежуточных сертификатов в машинное хранилище должна выполняться с привилегиями супер-пользователя командой:

   */opt/cprocsp/bin/<**arch**>/certmgr -inst -store mCa -file /path/to/intermediate-gost.cer -cert*

Windows

1. Проверка наличия установленного серверного сертификата

   На данном этапе рекомендуется удостовериться, что в личном ("my") хранилище сертификатов пользователя установлен серверный сертификат с привязкой к ключевому контейнеру. Для этого запустите «Internet Explorer» – откройте меню «Сервис» – «Свойства обозревателя» – переедите на закладку «Содержание» – нажмите кнопку «Сертификаты» – на закладке «Личное» выберите серверный сертификат и откройте диалог просмотра его свойств. Удостоверьтесь, что в нижней

части закладки «Общие», под сроком действия сертификата есть надпись «Есть закрытый ключ, соответствующий этому сертификату».

2. Получение файла серверного сертификата

   При настройке файлов конфигурации веб-сервера потребуется указать имя файла серверного сертификата (сохраненного в формате DER или Base64). Если у Вас отсутствует файл с данным сертификатом, но он установлен в хранилище, то его можно экспортировать из диалога просмотра свойств сертификата в хранилище или mmc-оснастке. Как открыть диалог свойств сертификата через IE, описано в предыдущем пункте данного раздела. Затем перейдите на закладку «Состав», нажмите кнопку «Копировать в файл» и пройдете мастер экспорта сертификата.

3. Установка сертификата Удостоверяющего центра

   Для установки сертификата корневого УЦ откройте диалог просмотра его свойств, и на закладке «Общие» нажмите кнопку «Установить сертификат». Если достаточно установки сертификата УЦ в хранилище корневых сертификатов текущего пользователя, то в Мастере импорта сертификатов можно оставить режим «Автоматически выбирать хранилище на основе типа сертификата». Если же требуется, чтобы сертификат УЦ использовался сразу всеми пользователями данного компьютера, то следует выбрать режим «Поместить все сертификаты в следующее хранилище», нажать кнопку «Обзор», включить режим «Показать физические хранилища» и выбрать в дереве элемент «Доверенные корневые центры сертификации» - «Локальный компьютер». Учтите, что установка сертификатов в машинное хранилище должна производиться пользователем с правами администратора.

   В случае если серверный сертификат издан промежуточным УЦ, то сертификат последнего должен быть установлен в хранилище «Промежуточные центра сертификации», а корневой, которым подписан промежуточный, - в хранилище «Доверенные корневые центры сертификации», аналогично тому, как описано выше.

Настройка файлов конфигурации веб-сервера для односторонней аутентификации Trusted TLS по ГОСТ сертификату

Под ОС Windows путь /opt/DIGT/Trusted_Web_Server22 в примерах конфигурационных файлов, описываемых ниже, должен быть заменен на путь к каталогу, в который был установлен продукт, например, C:/Apache22 (или /Apache22 в случае, если запуск веб-сервера производится из того же диска ОС Windows, на котором установлен Trusted TLS).

Редактирование файла конфигурации conf/httpd.conf:

1. LoadModule

   Удостоверьтесь, что включена загрузка модуля mod_digt_tls.so продукта Trusted TLS:

   LoadModule ssl_module modules/mod_digt_tls.so

2. User и Group

   Только для Unix-систем: в директивах User и Group задайте имя пользователя, под которым установлен ключевой контейнер и соответствующий ему серверный сертификат, и его группу.
   В ОС Windows значения данных директив игнорируются.

3. LogLevel

   При первоначальной настройке рекомендуется задать максимальный уровень журналирования «LogLevel debug», который после успешного запуска веб-сервера можно будет понизить до «LogLevel warn».

4. Подключение conf/extra/httpd-ssl.conf

   Удостоверьтесь, что в базовом файле конфигурации подключается файл конфигурации httpd-ssl.conf (в противном случае удалите символ комментария в следующей строке):

   Include conf/extra/httpd-ssl.conf

Редактирование файла конфигурации conf/extra/httpd-ssl.conf:

Перед корректировкой данного файла необходимо определиться, на каком порту будет осуществляться прием защищенных соединений. Стандартный SSL/TLS-порт – 443, но если на нем запущен другой веб-сервер, то необходимо, либо тот остановить, либо выбрать другой порт, например, 4433. Если защита веб-контента с использованием SSL/TLS-шифрования требуется одновременно для нескольких сайтов, расположенных одном веб-сервере, то для каждого сайта должен быть использован уникальный номер порта.

1. Listen

   Задайте директиве Listen номер SSL/TLS-порта, на котором будет осуществляться прием защищенных соединений. Для работы одновременно на нескольких портах задайте каждый дополнительный порт директивой Listen в отдельной строке, например:

   ```
   Listen 443
   Listen 4433
   ```

2. SSLSessionCache

   Современные браузеры в процессе загрузки веб-страницы параллельно запрашивают вложенные в нее картинки и другие объекты, однако текущая версия Trusted TLS не поддерживает кэширование, используемое для оптимизации обработки таких запросов, поэтому оно должно быть выключено:

   ```
   SSLSessionCache none
   ```

3. VirtualHost

   При описании секции виртуального хоста укажите DNS-имя (или ip-адрес) сервера и SSL/TLS-порт в формате <VirtualHost *адрес*:*порт*>. Для обработки данным виртуальным хостом защищенных соединений на всех сетевых интерфейсах, подключенных к серверу, вместо DNS-имени достаточно указать идентификатор _default_. Для защиты по SSL/TLS-протоколу нескольких порталов создайте для каждого из них собственную секцию виртуального хоста.

   Для каждого из виртуальных хостов:

   3.1. ServerName

   Описание директивы ServerName имеет следующий формат:

   ```
   ServerName [схема://]полное-доменное-имя[:порт]
   ```

   Рекомендуется как можно детальнее задавать значение данной директивы, т.к. она используется для формирования URL-адресов при генерации страниц веб-портала.

   В качестве «полного-доменного-имени» должно быть задано внешнее DNS-имя, которое пользователь будет вводить в строке браузера при обращении к Вашему веб-серверу. Например, если Trusted TLS развертывается на сервере с именем host123.yourhoster.org, а будет обрабатывать запросы к порталу web-portal.yourcompany.ru, то последний и должен быть указан в качестве «полного-доменного-имени».

   Внутри виртуального хоста, обрабатывающего только защищенные соединения, желательно указать все компоненты данной директивы, например:

   ```
   ServerName https://web-portal.yourcompany.ru:443
   ```

   3.2. SSLCertificateFile и SSLCertificateKeyFile

   Скопируйте файл с серверным ГОСТ сертификатом (в формате DER или Base64) в каталог, доступный для чтения пользователю, из-под которого будет работать веб-сервер, и укажите к нему путь в директивах SSLCertificateFile и SSLCertificateKeyFile, например:

   ```
   SSLCertificateFile "/opt/DIGT/Trusted_Web_Server22/conf/server-gost.cer"
   SSLCertificateKeyFile "/opt/DIGT/Trusted_Web_Server22/conf/server-gost.cer"
   ```

Примечание: SSLCertificateKeyFile указывает на сертификат, ввиду того, что криптопровайдер КриптоПро CSP не реализует хранение закрытых ключей в файлах. Поэтому ссылка на закрытый ключ ищется по сертификату в личном хранилище средствами CryptoAPI.

3.3. SSLVerifyClient

Настройка двухсторонней аутентификации будет описана в одном из следующих разделов, а для работы в режиме односторонней аутентификации директива SSLVerifyClient должна быть закомментирована или иметь значение none:

```
SSLVerifyClient none
```

Остальные параметры каждого виртуального хоста, а также глобальные директивы, установите в соответствии с документацией на веб-сервер Apache и его модули. Для первого запуска Trusted TLS в тестовом режиме подойдут значения, уже установленные в файле конфигурации из дистрибутива.

Установка лицензии

При запуске (см. раздел «Старт сервера») с неустановленной или некорректной лицензией на консоль будет выдано сообщение:

Can not load license. Please set the correct license (press <Ctrl+C> to quit):

Введите корректную строку лицензии в формате XXXXX-XXXXX-XXXXX-XXXXX-XXXXX. После этого будет выдано сообщение *License accepted,* и сервер запустится.

Для тестирования продукта Вы можете получить временную лицензию, отправив заявку на email-адрес info@digt.ru или support@digt.ru. При запросе временной лицензии, пожалуйста, указывайте версию Trusted TLS и операционной системы, а для ОС Solaris еще и архитектуру процессора (Intel или SPARC).

Примечание: сохранение введенной лицензии возможно только в случае, если запуск сервера был выполнен с правами администратора.

Переустановка временной лицензии на бессрочную

Если установленная ранее лицензия еще действует, то сначала ее необходимо удалить из системы. Новую лицензию следует устанавливать в соответствии с инструкциями в предыдущем разделе.

Порядок удаления лицензии зависит от типа ОС:

Unix-системы

Удалите (или переименуйте) файл /opt/DIGT/etc/Trusted/Web/license.lic.

Windows

Удалите (или переименуйте) ветки реестра:

- HKEY_LOCAL_MACHINE\SOFTWARE\DIGT\Trusted Web 1.0\License (для 32- и 64-разрядных ОС);

- HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\DIGT\Trusted Web 1.0\License (только для 64 bit).

Старт сервера

Синхронизируйте время на сервере и на клиентских рабочих станциях, с которых будет производиться обращение к серверу.

Если веб-сервер настроен на прием какого-либо соединения по порту с номером до 1024 (по умолчанию как раз используются 80 и 443 порты), то его запуск должен выполняться пользователем, обладающим правами администратора.

При первом запуске Trusted TLS, а также в случае наличия истекшей лицензии, будет запрошен ввод лицензии. Для корректного сохранения введенной лицензии запуск сервера необходимо выполнять с правами администратора.

Unix-системы

Для старта сервера выполните следующую команду:

/opt/DIGT/Trusted_Web_Server22/bin/apachectl start

Если серверный ключ защищен паролем и директиве «SSLPassPhraseDialog» установлено значение «builtin», то при запуске сервера в консоли отобразится запрос ввода пароля.

Windows

Для старта сервера выполните команду:

C:\Apache22\bin\httpd.exe

Если серверный ключ защищен паролем и директиве «SSLPassPhraseDialog» установлено значение «builtin», то при запуске сервера отобразится окно для ввода пароля.

Проверка работы односторонней аутентификации

Перед обращением к серверу по TLS-протоколу установите в хранилище "Доверенные корневые Центры Сертификации" сертификат УЦ, которым подписан серверный сертификат. В случае если на клиентской стороне используется ОС семейства Windows 2000 совместно с КриптоПро CSP 2.0, то на ней дополнительно необходимо установить дистрибутив «КриптоПро TLS 2.0».

Запустите браузер Internet Explorer, включите в его настройках поддержку протокола TLS, и введите в адресной строке адрес https://your-server-name. Если сервер принимает защищенные соединения на порту, отличающемся от стандартного – 443, то в адресной строке дополнительно укажите порт, например, https://your-server-name:4433.

Организация двусторонней аутентификации

Под двухсторонней TLS-аутентификацией понимается установка защищенного соединения между клиентом и веб-сервером с использованием сертификата и закрытого ключа не только веб-сервера, но и клиента. Также данный режим называют «клиентской TLS-аутентификаций».

Дополнительно к аутентификации поддерживается авторизация доступа пользователя к веб-ресурсам (разграничение по URL-адресам), а при наличии поддержки со стороны веб-приложения возможно упростить авторизацию на нем за счет перехода от авторизации по логину-паролю к авторизации по сертификатам или одновременного использования обеих схем для разных категорий пользователей.

Для настройки двухсторонней аутентификации и авторизации доступа (с разграничением по URL-адресам сайта) в Trusted TLS необходимо:

1. Установить сертификаты Удостоверяющих центров, которыми изданы клиентские сертификаты TLS-аутентификации. Их установка производится аналогично инструкциям в п.3 раздела «Подготовка к настройке веб-сервера, установка сертификатов УЦ».

2. Отредактировать параметры виртуального хоста в файле конфигурации conf/extra/httpd-ssl.conf.

Директивы, используемые для настройки двухсторонней аутентификации:

1. SSLVerifyClient

   Допустимые значения:

   1.1. none - двухсторонняя аутентификация отключена (значение по умолчанию).

1.2. optional – в этом режиме браузер предложит пользователю предоставить сертификат клиентской аутентификации, если пользователь откажется от его предоставления, то защищенное TLS-соединение будет установлено в режиме односторонней аутентификации.

1.3. require – защищенные соединения будут приниматься сервером только при условии успешного выполнения двухсторонней аутентификации.

2. SSLVerifyDepth

Trusted TLS аналогично mod_ssl поддерживает проверку клиентских сертификатов по COC (спискам отозванных сертификатов, CRL, certificate trusted list), которые периодически публикуется Удостоверяющим центром, и в которые помещается информация об отозванных и приостановленных сертификатах данного УЦ.

Допустимые значения:

2.1. Значение 1 - при проверке клиентского сертификата для него строится только путь сертификации без проверки статусов сертификатов по COC (CRL).

2.2. Значение 2 - при проверке клиентского сертификата для него строится цепочка сертификатов, статусы которых затем проверяются по COC (CRL) из локального хранилища сертификатов (а также кэша). При этом попытка обновления COC по CDP (из точки распространения COC) не выполняется.

При использовании данного режима наличие актуальных COC в хранилище обязательно, поскольку в случае их отсутствия или истечения их периода действия невозможно судить о статусах сертификатов.

Для установки COC вручную под unix-системами сохраните COC в файл (в формате DER или Base64) и выполните с привилегиями супер-пользователя команду:
*/opt/cprocsp/bin/<**arch**>/certmgr -inst -store mCa -file <имя_файла_COC> -crl*

Для установки COC вручную под ОС Windows сохраните его в файл (в формате DER или Base64) с расширением «.crl», откройте на полученном файле контекстное меню, выберите в нем пункт «Установить список отзыва (CRL)» и пройдите до конца открывшийся «Мастер импорта сертификатов» (который также позволяет импортировать и COC).

Обновление COC возможно автоматизировать. Под unix-системами – добавьте в cron скрипт, который бы скачивал COC (например, с помощью утилиты wget) и устанавливал его в хранилище (с помощью утилиты certmgr как описано выше). Под ОС Windows можно из встроенного планировщика («Назначенные задания», «Scheduled Tasks») запускать VBS или JS-скрипт, реализующий обновление COC с использованием COM-объекта CPCRLUpdate от компании КриптоПро или COM-объекта, предоставляемого ПО КриптоАРМ.

2.3. Значение 3 - при проверке клиентского сертификата для него строится путь сертификации, статусы сертификатов которого затем проверяются по COC (CRL) из локального хранилища (и кэша), а также при необходимости производится получение COC по CDP. CDP – это расширение сертификата «Точка распространения COC», в котором указываются http, ftp, file и ldap-ссылки, по которым выложены актуальные файлы COC.

2.4. Все другие значения интерпретируются аналогично SSLVerifyDepth 3.

3. SSLCACertificateFile и SSLCACertificatePath

Данные директивы используются при двухсторонней аутентификации для ограничения списка используемых клиентских сертификатов, выданных различными УЦ. Если ни одна из этих директив не задана, то клиенту предлагается для выбора список из всех сертификатов, установленных в личное хранилище со ссылкой на закрытый ключ и содержащих объектный идентификатор 1.3.6.1.5.5.7.3.2 в расширении «Улучшенный ключ» (EKU).

3.1. Директивой SSLCACertificateFile задается файл, содержащий корневые и промежуточные сертификаты УЦ, клиентские сертификаты от которых будут приниматься веб-сервером. Сертификаты Удостоверяющих центров перечисляются друг за другом в формате Base64:

```
-----BEGIN CERTIFICATE-----
MIIC8TCCAp6gAwIBAgIKJTwXwwACAAAeLDAKBgYqhQMCAgMFADBlMSAwHgYJKoZI
hvcNAQkBFhFpbmZvQGNyeXB0b3Byby5ydTELMAkGA1UEBhMCUlUxEzARBgNVBAoT
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIC8TCCAgcqhQMCAh4BA0MABEAbq2WGBnHH/fqsGWlYBDGvd07kyxB9IVSTPQln
BgNVHR8ETjBMMEqgSKBGhkRodHRwOi8vd3d3LmNyeXB0b3Byby5ydS9jZXJ0ZW5y
...
-----END CERTIFICATE-----
```

3.2. Действие директивы SSLCACertificatePath аналогично SSLCACertificateFile с тем отличием, что в ней задается каталог с файлами, в каждом из которых должен находиться только один сертификат УЦ в формате Base64.

4. SSLCARevocationPath и SSLCARevocationFile

Данные директивы используются только для проверки статусов сертификатов с алгоритмами, отличными от ГОСТ. Проверка по COC ГОСТ-сертификатов регулируется директивой SSLVerifyDepth.

5. SSLRequire

Для разграничения доступа к различным частям сайта (по URL-адресам) можно использовать директиву SSLRequire. Она должна размещаться внутри каталога сайта, к которому она относится. Полное описание данной директивы и список переменных сервера, которые могут совместно с ней использоваться, приводится в документации к mod_ssl.

Дополнительно к ним в Trusted TLS поддерживается переменная сервера SSL_CLIENT_EKU (не путать с функцией OID() базового модуля mod_ssl, которая ищет строку, указанную в левой части выражения, в значении атрибута с заданным объектным идентификатором). В SSL_CLIENT_EKU заносится строка с объектными идентификаторами (OID) из расширения «Улучшенный ключ» / «Enhanced Key Usage» (EKU) клиентского сертификата разделяемыми пробелами. В случае если клиентский сертификат не имеет расширения EKU, то SSL_CLIENT_EKU принимает значение SSL_EKU_ANY.

Пример настройки, при которой разрешается доступ к подсистеме статистики сайта (расположенной по адресу https://portal/stat) только сотрудникам московских филиалов, по сертификатам с объектным идентификатором TLS-клиентской аутентификации (1.3.6.1.5.5.7.3.2) и любым OID из подмножества 1.2.643.2.2.34.*:

```
<Location /stat/>
SSLRequire %{SSL_CLIENT_S_DN_L} =~ m/\A\xD0\x9C\xD0\xBE\xD1\x81\xD0\xBA\xD0\xB2\xD0\xB0\Z/ \
        and %{SSL_CLIENT_EKU} =~ m/(\A| )1\.3\.6\.1\.5\.5\.7\.3\.2( |\Z)/ \
        and %{SSL_CLIENT_EKU} =~ m/(\A| )1\.2\.643\.2\.2\.34\./
</Location>
```

Вместо «%{SSL_CLIENT_S_DN_L} =~ m/\A\xD0\x9C\xD0\xBE\xD1\x81\xD0\xBA\xD0\xB2\xD0\xB0\Z/» можно использовать «%{SSL_CLIENT_S_DN_L} eq "Москва"», при этом русские слова должны быть записаны в кодировке UTF-8 (под Windows можно редактировать с помощью Блокнота). И тогда если открыть файл конфигурации на просмотр в кодировке Win1251, то наименование будет выглядеть следующим образом: «%{SSL_CLIENT_S_DN_L} eq "РњРѕСЃРєРІР°"».

Внимание! Метасимволы регулярных выражений \A и \Z используются для определения соответственно начала и конца требуемого объектного идентификатора. Если проверка окончания OID не всегда критична, то его начало проверять следует обязательно, – чтобы не пропускать сертификаты, в которых требуемый объектный идентификатор является составной частью OID из другого подмножества, например: 1.2.xx.**1.2.643.2.2.34**.xx.

Остальные параметры каждого виртуального хоста, а также глобальные директивы, при необходимости задайте в соответствии с документацией на модуль mod_ssl веб-сервера Apache.

Проверка работы двусторонней аутентификации

Проверка работы двухсторонней аутентификации выполняется аналогично проверке, описанной в разделе «Проверка работы односторонней аутентификации» со следующими отличиями:

1. Перед обращением к серверу дополнительно получите в УЦ и установите на клиентской машине сертификат клиентской аутентификации (он должен иметь объектный идентификатор 1.3.6.1.5.5.7.3.2 в расширении «Улучшенный ключ» (EKU)).

2. В процессе обращения к серверу укажите сертификат клиентской аутентификации из списка «подходящих» сертификатов, предлагаемого браузером.

   Примечание: по умолчанию, если в списке подходящих сертификатов имеется только один сертификат или их не содержится вовсе, то браузер не отображает диалог выбора сертификата аутентификации. Для диагностики ошибок, возникающих при тестовой настройке, может оказаться полезным постоянно отображать этот диалог. Для этого требуется выполнить следующую настройку браузера: в меню «Сервис» выберите пункт «Свойства обозревателя», перейдите на закладку «Безопасность», выберите зону соответствующую веб-серверу, нажмите кнопку «Другой» и в разделе «Разное» отключите настройку «Не запрашивать сертификат клиента, если он отсутствует или имеется только один».

Работа с двумя серверными сертификатами

Trusted TLS поддерживает режим работы одновременно с двумя серверными сертификатами стандартов RSA и ГОСТ, сконфигурированными для одного виртуального сервера. Чтобы использовать данный режим, в конфигурационном файле для данного виртуального сервера должно быть указано две пары директив *SSLCertificateFile* и *SSLCertificateKeyFile* соответственно для каждого типа сертификата (см. документацию на mod_ssl для подробностей конфигурации RSA сертификатов).

Данный режим имеет следующую особенность: клиенты с установленным КриптоПро CSP 3.0 и выше смогут взаимодействовать с веб-сервером по TLS-соединению только с использованием ГОСТ алгоритмов, а все остальные (у которых КриптоПро CSP не установлен или установлена версия 2.0) будут поднимать защищенное соединение только с использованием алгоритмов RSA.

Использование Trusted TLS в режиме HTTP/HTTPS прокси-сервера

Продукт Trusted TLS может быть использован в качестве HTTP/HTTPS прокси-сервера для защищенного доступа через Интернет к корпоративным информационным системам, не поддерживающим сертифицированные в РФ средства криптозащиты (Рис. 1), а также для организации защищенного канала между клиентами, не имеющим возможность шифрования с использованием ГОСТ-алгоритмов, и сервером, использующим их (Рис. 2).

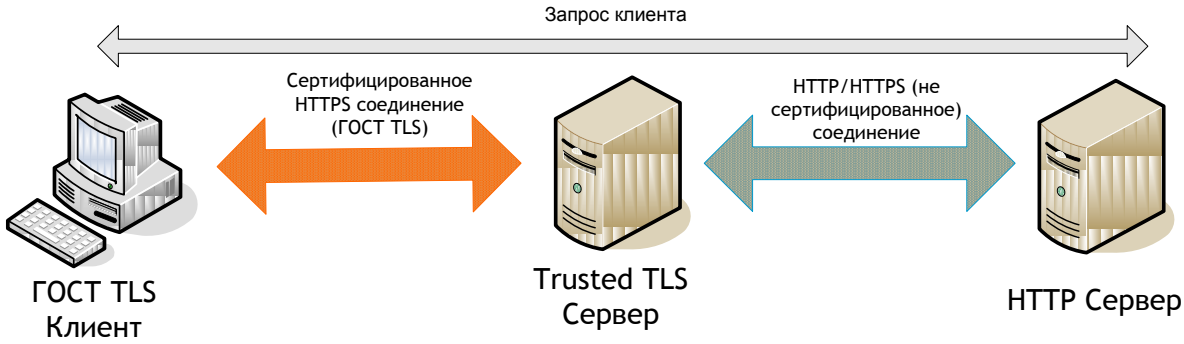

Рис. 1

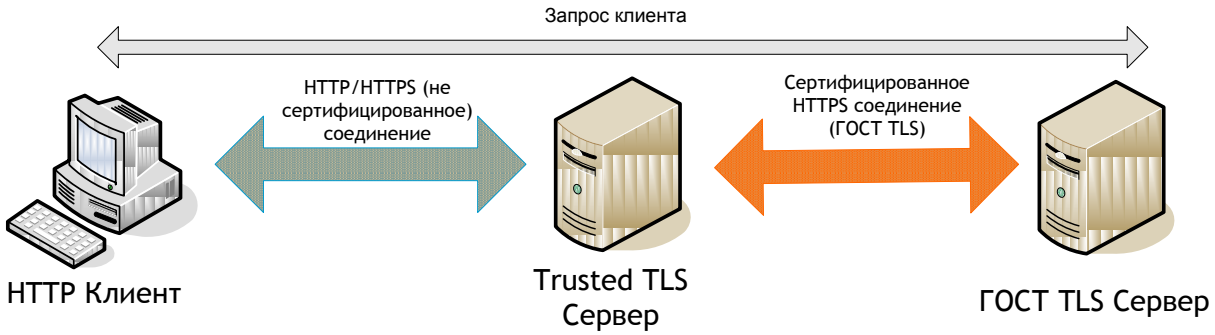

Рис. 2

Дополнительные преимущества подобной схемы заключаются в том, что один сервер, на котором используется Trusted TLS, может обслуживать запросы клиентов одновременно к нескольким серверам. Это, с одной стороны, облегчает администрирование, а с другой стороны - позволяет использовать его как средство балансировки нагрузки, так как вычислительные затраты при работе по защищенному сертифицированному каналу обычно довольно велики.

Описываемая функциональность достигается за счет совместного использования криптографических возможностей работы с сертифицированными в РФ алгоритмами модуля mod_digt_tls, а также стандартных модулей Apache mod_proxy и mod_proxy_http, поставляемых в составе Trusted TLS.

Для включения функциональности вышеописанных модулей, их использование должно быть разрешено в файле конфигурации conf/httpd.conf следующим образом:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

В глобальной конфигурации либо в конфигурации соответствующего виртуального сайта необходимо сконфигурировать использование прокси с помощью директив ProxyPass и ProxyPassReverse, например:

```
ProxyPass        /mirror/portal/forbidden-area !
ProxyPass        /mirror/portal/ http://backend.some-portal.local/
ProxyPassReverse /mirror/portal/ http://backend.some-portal.local/
```

В случае если необходимо проксировать запрос на сервер, так же работающий по https протоколу (в том числе по ГОСТ TLS), в директивах ProxyPass и ProxyPassReverse вместо http-префикса следует указывать https, и дополнительно требуется разрешить данную функциональность с помощью директивы:

```
SSLProxyEngine on
```

В случае если защищенный ГОСТ TLS сервер требует авторизации по клиентским сертификатам, они могут быть указаны с помощью директивы:

```
SSLProxyMachineCertificateFile conf/proxy.pem
```

где proxy.pem - это файл с одним или несколькими клиентскими сертификатами (а для RSA-сертификатов – еще и ключами) в формате Base64:

```
-----BEGIN CERTIFICATE-----
MIIDkDCCAnigAwIBAgIKN7ozGAABAAAAkzANBgkqhkiG9w0BAQUFADBGMRUwEwYK
/RSA/CERT/RSA/CERT/RSA/CERT/RSA/CERT/RSA/CERT/RSA/CERT/RSA/CERT/
...
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCjyU9XwlZ8lk/DJKo0CPDYM+aPovBU9HbCyK2RPkflRtoNYBnW
/RSA/KEY/RSA/KEY/RSA/KEY/RSA/KEY/RSA/KEY/RSA/KEY/RSA/KEY/RSA/KEY
...
-----END RSA PRIVATE KEY-----
```

```
-----BEGIN CERTIFICATE-----
MIIDAjCCAq+gAwIBAgIKF3uunAACAAAnbTAKBgYqhQMCAgMFADBlMSAwHgYJKoZI
/GOST/CERT/GOST/CERT/GOST/CERT/GOST/CERT/GOST/CERT/GOST/CERT/GOS
...
-----END CERTIFICATE-----
```

ГОСТ-сертификаты из proxy.pem должны быть установлены на сервере аналогично ГОСТ TLS-серверному сертификату. Защита паролем их закрытых ключей в этом режиме не поддерживается.

Более детальная информация по директивам, перечисленным в данном разделе, приводится в документации к модулю mod_ssl.

Использование переменных сервера для аутентификации в веб-приложении

Модуль mod_digt_tls аналогично mod_ssl формирует переменные веб-сервера. Дополнительно к набору стандартного модуля в нем поддерживается переменная SSL_CLIENT_EKU, описание которой приводится в пункте «5. SSLRequire» раздела «Организация двусторонней аутентификации».

Переменные сервера удобно использовать для обеспечения аутентификации пользователя в веб-приложении по сертификату дополнительно к аутентификации по логину и паролю. Для этого необходимо на веб-сервере включить возможность (*SSLVerifyClient optional*) или требование (*SSLVerifyClient require*) установления двухсторонней аутентификации, а также добавить в веб-приложение определение соответствия между пользователем и его сертификатами. Наиболее простой способ – это помещать логин пользователя в одно из полей имени владельца сертификата аутентификации, а наиболее гибкий – хранить сертификаты или только уникальную информацию (имя издателя + серийный номер) из них в таблице веб-системы с привязкой к учетным записям пользователей.

Если Trusted TLS используется на отдельном от веб-приложения сервере, то на нем необходимо настроить передачу переменных сервера, например, с использованием директивы RequestHeader модуля mod_headers. Пример частичной конфигурации виртуального хоста:

```
ProxyPass        / http://backend.some-portal.local/
ProxyPassReverse / http://backend.some-portal.local/

RequestHeader add Forwarded-Ssl-Client-I-Dn     "%{SSL_CLIENT_I_DN}s"
RequestHeader add Forwarded-Ssl-Client-M-Serial "%{SSL_CLIENT_M_SERIAL}s"
RequestHeader add Forwarded-Ssl-Client-S-Dn     "%{SSL_CLIENT_S_DN}s"
```

Название заголовков запроса должно удовлетворять RFC 822, п.3.1 (для HTTP/1.1 – RFC 2616, 4.2).

В данном примере пересылаемым переменным с целью отделения от локальных добавляется префикс «FORWARDED_». А доступ к ним должен производиться с префиксом «HTTP_FORWARDED_», например, «HTTP_FORWARDED_SSL_CLIENT_I_DN».

Пример обращения к переменным сервера из PHP:

```
<?php
function WriteServerVariable($sVarName)
{
    echo $sVarName . " = '" . $_SERVER[$sVarName] . "'<br>";
}

WriteServerVariable("SSL_CLIENT_I_DN");
WriteServerVariable("SSL_CLIENT_M_SERIAL");
WriteServerVariable("SSL_CLIENT_S_DN");

WriteServerVariable("HTTP_FORWARDED_SSL_CLIENT_I_DN");
WriteServerVariable("HTTP_FORWARDED_SSL_CLIENT_M_SERIAL");
WriteServerVariable("HTTP_FORWARDED_SSL_CLIENT_S_DN");
?>
```

Пример обращения к переменным сервера из ASP:

```
<%
sub WriteServerVariable(ByVal sVarName)
    Response.Write sVarName & " = '" & Request.ServerVariables(sVarName) & "'"
    Response.Write "<br>"
end sub

WriteServerVariable("CERT_ISSUER")
WriteServerVariable("CERT_SERIALNUMBER")
WriteServerVariable("CERT_SUBJECT")

WriteServerVariable("HTTP_FORWARDED_SSL_CLIENT_I_DN")
WriteServerVariable("HTTP_FORWARDED_SSL_CLIENT_M_SERIAL")
WriteServerVariable("HTTP_FORWARDED_SSL_CLIENT_S_DN")
%>
```

Детальное описание директивы RequestHeader приведено в документации к mod_headers по адресу http://httpd.apache.org/docs/2.2/mod/mod_headers.html.

Запуск Trusted TLS в режиме сервиса

Unix-системы

Если ключевой контейнер сервера защищен паролем, то для запуска Apache в режиме сервиса пароль необходимо сбросить (если это допускается регламентом), либо автоматизировать его передачу серверу.

Сбросить пароль можно следующей командой, выполняемой из-под пользователя, который имеет доступ к ключевому контейнеру и от имени которого функционирует веб-сервер:

*/opt/cprocsp/bin/<**arch**>/csptestf -passwd -change "" -container <имя_контейнера>*

Один из способов автоматизации передачи пароля серверу – это создать скрипт, который бы печатал пароль на экран, установить ему права на запуск и указать его имя в директиве SSLPassPhraseDialog, например:

```
SSLPassPhraseDialog exec:/opt/DIGT/Trusted_Web_Server22/conf/server-password.sh
```

Пример файла скрипта:

```
#!/bin/sh
echo 12345678
# Uncomment and update follow lines for several virtual hosts:
#if [ "$1:$2" = "www.example.com:443:GOST" ]; then echo "1234" # specific host
#elif [ "$2" = "GOST" ]; then echo "4321" # all other GOST hosts
#elif [ "$2" = "RSA" ]; then echo "11111111" # all RSA hosts
#fi
```

Затем убедитесь, что сервер успешно стартует без ввода каких-либо либо данных с клавиатуры. После этого добавьте его в автоматический запуск стандартным для используемой ОС способом.

Windows

Если ключевой контейнер сервера защищен паролем, то для запуска Apache в режиме сервиса пароль необходимо сбросить (если это допускается регламентом), либо автоматизировать его передачу серверу.

Сбросить пароль можно через Панель управления КриптоПро CSP на закладке «Сервис».

Один из способов автоматизации передачи пароля серверу – это создать скрипт, который бы печатал пароль на экран, установить ему права на запуск и указать его имя в директиве SSLPassPhraseDialog, например:

```
SSLPassPhraseDialog exec:/Apache22/conf/server-password.bat
```

Пример файла скрипта:

```
@echo off
echo 12345678
rem Uncomment and update follow lines for several virtual hosts:
rem if "%1:%2" == "www.example.com:443:GOST" (
rem   rem specific host
rem   echo 1234
rem ) else if "%2" == "GOST" (
rem   rem all other GOST hosts
rem   echo 4321
rem ) else if "%2" == "RSA" (
rem   rem all RSA hosts
rem   echo 11111111
rem )
```

Для добавления сервера в качестве сервиса ОС Windows выполните команду:

httpd.exe -k install

После этого найдите добавленный сервис с именем «Apache2.2» в списке сервисов, откройте его свойства, на закладке «Вход в систему» / «Log On» активируйте режим входа в систему «С учетной записью» / «This account», выберите учетную запись пользователя, под которым установлен серверный сертификат и имеется доступ к ключевому контейнеру, и в два соответствующих поля введите его пароль. Затем примените сделанные изменения и запустите сервис.

Сервер может быть помечен на удаление из списка сервисов с помощью команды:

httpd.exe -k uninstall

При этом окончательное его удаление произойдет при перезагрузке системы.

Обновление сертификата сервера

Обычно серверный сертификат выпускается в Удостоверяющем центре сроком на 1 год. Для его обновления необходимо создать новую ключевую пару, сформировать запрос, выпустить по нему сертификат, установить последний в хранилище личных сертификатов пользователя, из-под учтенной записи которого работает веб-сервер, и указать его в файле конфигурации сервера. Возможно, также потребуется установить новые сертификаты УЦ.

Описание всех этих операций приведено в предыдущих разделах данного Руководства. Единственное затруднение, которое может возникнуть в процессе обновления серверного сертификата – это присвоение новому контейнеру для закрытого ключа имени, совпадающего с одним из уже присутствующих в системе.

СЕРВЕРА ПРИЛОЖЕНИЙ: ИНТЕГРАЦИЯ TRUSTED JAVA И TRUSTED TLS

Введение

В предлагаемом ниже материале рассматриваются методики построения защищенного канала до серверов приложений (Tomcat, IBM Websphere, Oracle Weblogic) на базе продукта Trusted TLS (веб-сервера Apache версии 2.2) для обеспечения конфиденциальности информации и аутентификации пользователей в приложениях, разворачиваемых на этих серверах. (Методики были проверены на операционной системе RedHat Enterprise Linux/CentOS 5.3 x86.)
В частности, рассматриваются вопросы обеспечения взаимодействия в цепочке <Клиент> - <Apache-сервер> и <Apache-сервер> - <Сервер приложений>, а также доставки сертификатов пользователей (и Apache-сервера) до сервера приложений.
Продукт Trusted Java также может быть интегрирован в составе рассматриваемых решений как мост для доступа приложений на Java к криптографическим операциям при использовании следующих связок:

- Apache HTTP Server + Trusted TLS + КриптоПро CSP + Tomcat 5.5/6.0/7.0 + Trusted Java
- Apache HTTP Server + Trusted TLS + КриптоПро CSP + IBM Websphere 6.1/7.0 + Trusted Java
- Apache HTTP Server + Trusted TLS + КриптоПро CSP + Oracle WebLogic Server 10.3.2 + Trusted Java

Передача сертификатов от Apache-сервера серверам приложений

Для использования сертификатов клиента (и сервера) на стороне развернутого приложения предлагается использовать на стороне Apache-сервера возможность модуля **mod_headers** вставлять в заголовок запроса переменную с содержимым из SSL-переменных. Следует подчеркнуть, что таким образом в Apache версии 2.0 корректно передать многострочное содержимое сертификатов не удастся.
Предполагается, что на Apache-сервере уже настроен модуль mod_digt_tls.so (продукт Trusted TLS) согласно прилагаемым к нему инструкциям.
В конфигурационном файле **httpd.conf** добавим строку загрузки модуля **mod_headers**

```
LoadModule headers_module modules/mod_headers.so
```

и в конфигурационном файле **ssl.conf** в секции *<VirtualHost _default_:4433>* прописываем строки

```
<IfModule mod_headers.c>
      RequestHeader set Forwarded-SSL-CLIENT-CERT "%{SSL_CLIENT_CERT}s"
      RequestHeader set Forwarded-SSL-SERVER-CERT "%{SSL_SERVER_CERT}s"
</IfModule>
```

Пересылка запросов серверу приложений от Apache-сервера

Пересылку запросов на сервер приложений можно организовать с использованием

- Apache-модуля mod_proxy,
- Специализированного WebServer Plug-ins

Apache-модуль Mod_proxy

Процесс интеграции

В конфигурационном файле **httpd.conf** добавим строку загрузки модуля **mod_proxy.so**

```
LoadModule proxy_module modules/mod_proxy.so
```

и в конфигурационном файле **ssl.conf** в секции *<VirtualHost _default_:4433>* прописываем строки

```
<Location />
        <IfModule mod_proxy.c>
            ProxyPass http://AS-host:AS-port/
            ProxyPassReverse http://AS-host:AS-port/
        </IfModule>
 </Location>
```

As-host и AS-port – DNS-имя и порт хоста, на котором принимает запросы сервер приложений.

Mod_proxy и ApacheTomcat 5/6/7

Для интеграции с ApacheTomcat 5/6/7 в качестве порта **AS-port** нужно использовать значение **8080**. Например,

```
<Location />
      <IfModule mod_proxy.c>
            ProxyPass http://localhost:8080/
            ProxyPassReverse http://localhost:8080/
      </IfModule>
 </Location>
```

Mod_proxy и IBM Websphere 6.1/7.0

Для интеграции с IBM Websphere 6.1/7.0 в качестве порта **AS-port** нужно использовать значение **9080**. Например,

```
<Location />
      <IfModule mod_proxy.c>
            ProxyPass http://localhost:9080/
            ProxyPassReverse http://localhost:9080/
      </IfModule>
</Location>
```

Mod_proxy и Oracle Weblogic 10.3.2

Для интеграции с Oracle Weblogic 10.3.2 в качестве порта **AS-port** нужно использовать значение **7001**. Например,

```
<Location />
      <IfModule mod_proxy.c>
            ProxyPass http://localhost:7001/
            ProxyPassReverse http://localhost:7001/
      </IfModule>
</Location>
```

Apache-Tomcat mod_jk connector

Для построения связки между Apache и Tomcat серверами будем использовать Apache Tomcat Connector. Из хранилища выкачиваем для Apache 2.2 (например, под Linux) mod_jk-1.2.28-httpd-2.2.X.so. Копируем его под именем **mod_jk.so** в APACHE_HOME/modules.

Конфигурирование сервера Apache

Проверяем, что в файле **APACHE_HOME/conf/httpd.conf** задана директива

Include conf.d/.conf*

Создаем конфигурационный файл **APACHE_HOME/conf.d/mod_jk.conf** со следующим содержимым:

```
LoadModule jk_module modules/mod_jk.so

JkWorkersFile "conf.d/workers.properties"

# Where to put jk shared memory
JkShmFile "logs/mod_jk.shm"

JkLogFile "logs/mod_jk.log"
JkLogLevel info
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]”
```

В соответствие директиве **JkWorkersFile** создаем далее конфигурационный файл **APACHE_HOME/conf.d/workers.properties** со следующим содержимым:

```
[channel.socket:localhost:8009]
port=8009
host=localhost
worker=ajp13:localhost:8009
```

Он описывает параметры AJP-соединения: идентификатор соединения с именем хоста, используемое значение порта и тип соединения.
В файле **APACHE_HOME/conf.d/ssl.conf** настраиваем в секции виртуального хоста:

```
<VirtualHost _default:443>
…
    JkMount /* ajp13
    JkLogLevel  info
…
</VirtualHost>
```

Перезапускаем сервер Apache.

Конфигурирование сервера Tomcat

Согласно содержимому конфигурационного файла **APACHE_HOME/conf.d/workers.properties** в конфигурационном файле **TOMCAT_HOME/conf/server.xml** сервера Tomcat проверяем наличие строк

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" enableLookups="false" redirectPort="8443" proto-
col="AJP/1.3" />
```

Перезапускаем сервер Tomcat и заходим на ресурсы сервера TOMCAT по защищенному каналу на порту 443.

Oracle WebLogic WebServer Plug-Ins

Oracle WebLogic WebServer Plug-Ins входит в поставку Oracle WebLogic 10.3.2.
После установки Oracle WebLogic

1. Взять из каталога **WL_HOME/server/plugin** модуль **mod_wl_22.so** (или mod_wl128_22.so) из подкаталога, соответствующего платформе, на которой установлен web-сервер Apache (например **linux/i686**).

2.Скопировать его в **APACHE_HOME/modules**.
3.В конфигурационный файл APACHE **httpd.conf** прописать загрузку этого модуля:

```
LoadModule weblogic_module modules/mod_wl_22.so
```

4.Установить проксирование на базе путей (можно указать в секции <Virtualhost _default_:**4433**>):

```
<Location />
   SetHandler weblogic-handler
   WLLogFile   /tmp/wl_root_log.log
</Location>
```

5.Установить глобальные параметры:

```
<IfModule mod_weblogic.c>
   WebLogicHost localhost
   WebLogicPort 7001
   Debug            OFF
   WLLogFile        /tmp/wl_global_proxy.log
   #WLTempDir       "/tmp/wl"
   DebugConfigInfo    On
   KeepAliveEnabled   ON
   KeepAliveSecs      15
</IfModule>
```

После проделанных действий можно использовать доступ к серверу Weblogic по порту **4433**.

IBM WebSphere 7.0 WebServer Plug-Ins

Далее описывается процесс настройки Apache 2.2 WebServer Plug-Ins для WebSphere 7.0. Установка плагина производится штатным образом согласно документации с дополнительного установочного диска.
После установки плагина нужно проверить в файле **httpd.conf** наличие следующих строк для **Linux** или **Solaris x32**

```
LoadModule was_ap22_module /opt/IBM/WebSphere/Plugins/bin/mod_was_ap22_http.so
WebSpherePluginConfig /opt/IBM/WebSphere/Plugins/config/webserver_ap22/plugin-cfg.xml
```

Файл **plugin-cfg.xml** генерируется в процессе установки или через консоль управления сервером (https://servername:9043/ibm/console/logon.jsp) после внесения различных изменений в конфигурацию сервера приложений. По умолчанию в результате этой настройки по портам **80** (для http) и **443** (для https) запросы будут транслироваться от Apache-сервера на сервер приложений. Для конфигурирования https-протокола на нестандартный порт, например, **4433** требуется проделать следующие действия, которые описаны для варианта размещения сервера приложений и apache-сервера на одном хосте.

- Зарегистрироваться в консоли управления сервером.
- В пункте «**Среда**» выбрать «**Виртуальные хосты**».
- В списке виртуальных хостов выбрать, например, «**default_host**».

- В «**Дополнительных свойствах**» выбрать «**Псевдонимы хоста**».
- Выбрав пункт «**Создать**», в поле «**Имя хоста**» внести DNS-имя хоста, в поле «**Порт**» указать значение **4433** и нажать «**Ок**».
- Далее требуется сохранить конфигурацию.
- После определения виртуального хоста требуется в пункте «**Среда**» выбрать «**Обновить глобальную конфигурацию модуля Web-сервера**» и затем нажать кнопку «**Ок**».
- Далее в пункте «**Среда**» выбрать «**Серверы**» - «**Типы серверов**» - «**Web-серверы**».
- В поле «**Выбрать**» напротив требуемого сервера выставить **галочку**.
- Последовательно выбрать пункты «**Сгенерировать модуль**» и «**Распространить модуль**».
- Перезапустить сервер приложений.
- В конфигурационном файле **ssl.conf** apache-сервера нужно проверить наличие строк

```
...
Listen 4433
...
<Virtualhost [DNS-имя хоста|_default_]:4433>
...
</Virtualhost>
```

- Перезапустить apache-сервер.

После проделанных действий будут доступны развернутые приложения на виртуальном сервере **default_host** по защищенному каналу на порту **4433**.

УСТРАНЕНИЕ ОШИБОК

В данном разделе перечислены ошибки, которые могут возникнуть на этапе развертывания или в процессе использования Trusted TLS, с описанием их решения. Для улучшения диагностики ошибок следует задать максимальный уровень журналирования (“LogLevel debug” в httpd.conf), перезапустить сервер и воспроизвести ошибку повторно.

Описание ошибки	Описание решения
"Init: Crypto-Pro Cryptographic Service Provider is NOT found"	Не установлен дистрибутив криптопровайдера КриптоПро CSP
"Cannot load /opt/…/mod_digt_tls.so into server: libcapi10.so.3: cannot open shared object file: No such file or directory"	Не установлены все необходимые пакеты дистрибутива криптопровайдера КриптоПро CSP
"Init: <имя_сервера> Unable to configure GOST server certificate"	Для выбранного ГОСТ сертификата не был найден соответствующий личный ключ. Вероятно, вы пытаетесь запустить Trusted TLS в учетной записи, отличной от той, в которой был получен ГОСТ сертификат сервера.
Ошибка "0x8009000f («Объект уже существует» / «Object already exists»)", возникающая при генерации ключевой пары и запроса на сертификат утилитой cryptcp	Наиболее вероятно, что в команде генерации нового ключевого контейнера Вы указали имя существующего контейнера. Попробуйте изменить значение параметра –cont. Список существующих контейнеров можно посмотреть командой */opt/cprocsp/bin/<**arch**>/csptestf -keyset -provtype 75 -enum_containers -verifycontext -fqcn* Удалить ненужный контейнер можно командой: */opt/cprocsp/bin/<**arch**>/csptestf -keyset -provtype 75 -deletekeyset -container <имя_контейнера>*
При использовании метода получения сертификата через CryptoPro Test CA команда */opt/cprocsp/bin/<**arch**>/cryptcp -creatcert -provtype 75 -ex -cont "\\\\.\\HDIMAGE\\tlsserver" -dn "CN*=web-portal.yourcompany.ru*, O=My Company, C=RU, E=test@test.ru" -certusage "1.3.6.1.5.5.7.3.1"* завершается ошибкой *Creating request...* *Sending request to CA...* *Installing certificate...* *Error: Отказано в доступе. (0x5)* *[ErrorCode: 0x00000005]* которая в дальнейшем приводит к неработоспособности Trusted TLS	Причина в том, что метод "cryptcp –createcert .." кроме установки сертификата пользователя пытается установить корневой сертификат УЦ в системное корневое хранилище, запись в которое по умолчанию запрещена непривилегированным пользователям. Варианты решения: а) Производить получение сертификата через "*cryptcp -creatrqst ...*" и устанавливать CRL(COC) и корневой сертификат вручную от пользователя, которому это разрешено (обычно только root) б) Дать права на запись в файлы (крайне не рекомендуется использовать в реальных условиях из соображений безопасности): */var/opt/cprocsp/users/stores/root.sto* */var/opt/cprocsp/users/stores/ca.sto*
При открытии https://servername в IE отображается страница, сообщающая об ошибке. Если просмотреть серверный сертификат на рабочем месте клиента, то в диалоге свойств отображается ошибка «Этот сертификат не удалось проверить, проследив до его доверенного центра сертификации»	Установите на клиентском рабочем месте сертификат УЦ, выпустившего серверный сертификат.
При открытии https://servername в IE отображается страница, сообщающая об ошибке. Если просмотреть	Установите на клиентском рабочем месте тот сертификат УЦ, которым был подписан

серверный сертификат на рабочем месте клиента, то в диалоге свойств отображается ошибка «Целостность этого сертификата не гарантирована. Возможно, он поврежден или изменен.	серверный сертификат. Или установите все сертификаты Уполномоченных лиц данного УЦ.
Не стартует сервер на 64-битной ОС семейства Unix.	Попробовать перед запуском сервера указать: *LD_LIBRARY_PATH=* */opt/cprocsp/lib/<arch>/:$LD_LIBRARY_PATH* *export LD_LIBRARY_PATH* *LD_PRELOAD=/opt/cprocsp/lib/<arch>/libcapi20.so* *export LD_PRELOAD*
Ошибки проверки лицензий. Сервер не стартует, в логе присутствуют следующие ошибки:	
[error] Init: Initialization OpenSSL library failed (license not found)	Отсутствует лицензия
[error] Init: Initialization OpenSSL library failed (invalid license)	Лицензия подверглась несанкционированному изменению. Внимание! Если вы заметили данное сообщение, сообщите об этом компании-разработчику или поставщику.
[error] Init: Initialization OpenSSL library failed (license expired)	У вас истекла временная лицензия, обратитесь в компанию-разработчика для приобретения постоянной лицензии или прекратите использование продукта.
[error] Init: Initialization OpenSSL library failed (internal error, please contact with support@digt.ru)	Произошла внутренняя ошибка системы лицензирования, обратитесь в службу поддержки компании-разработчика.
Ошибки, записываемые в лог в процессе работы веб-сервера:	
[error] [client xx.xx.xx.xx] Invalid method in request \x80U\x01\x03\x01	Задайте правильное dns-имя (или ip-адрес) и порт виртуальному хосту. Возможно, порт также потребуется указать и в директиве ServerName.
[error] SSL_use_certificate_CP() failed: 0x80092004 или [error] SSL_CTX_AcquirePrivateKey_CP() failed: 0x80092004	В личном хранилище пользователя, под учетной записью которого работает веб-сервер, не установлен серверный сертификат или он не соответствует файлу, на который ссылается директива SSLCertificateFile. Либо он установлен без привязки к закрытому ключу. Также убедитесь, что веб-сервер запускается от нужного пользователя (для Unix-систем проверьте значение директивы User, а для Windows-сервисов – имя учетной записи, установленное в свойствах сервиса).
[error] Certificate Verification: Error (-2146762486): error number -2146762486 а также [error] Certificate Verification: Error (-2146762487): error number -2146762487 и [error] Certificate Verification: Error (-2146869244): error number -2146869244	Установите на сервере все сертификаты УЦ, которые присутствуют в пути сертификации клиентского сертификата (путь сертификации удобно смотреть на последней закладке диалога просмотра клиентского сертификата в ОС Windows). Их установка производится в соответствии с инструкциями в п.3 раздела «Подготовка к настройке веб-сервера, установка сертификатов УЦ». В некоторых случаях оказывается, что у пользователя, из-под которого работает Apache, отсутствуют права на вход в подкаталог /var/opt/cprocsp/users/stores или на чтение файлов, находящихся в нем. В этом случае надо предоставить возможность чтения (но не записи!) всех файлов в указанном подкаталоге.
[error] Certificate Verification: Error (-2146762482): error number -2146762482 [error] Certificate Verification: Error (-2146885613):	При проверке клиентского сертификата не был доступен актуальный СОС: - при SSLVerifyDepth 2 – настройте собственноручное обновление СОС в хранилище;

error number -2146885613 [error] Certificate Verification: Error (-2146885614): error number -2146885614	- при SSLVerifyDepth 3 – убедитесь, что СОС доступен с сервера УЦ без дополнительной авторизации, с помощью следующей команды, выполняемой на веб-сервере: *curl http://server/.../xxx.crl*
[info] [client xx.xx.xx.xx] (70014)End of file found: SSL handshake interrupted by system [Hint: Stop button pressed in browser?!]	Данное предупреждение обычно не является ошибкой, т.к. запись появляется в тот момент, когда браузер разрывает соединение и предлагает выбрать пользователю клиентский сертификат для двухсторонней аутентификации.
[info] SSL Library Error: 336113689 error:1408B019:SSL routines:SSL3_GET_CLIENT_KEY_EXCHANGE:reason(25) или [info] SSL Library Error: 336113771 error:1408B06B:SSL routines:SSL3_GET_CLIENT_KEY_EXCHANGE:bad decompression	При старте веб-сервера указан неверный пароль на ключевой контейнер. Если он считывается через директиву SSLPassPhraseDialog, то проверьте корректность ее значения. Либо если Apache запущен в режиме сервиса Windows, то на ключевой контейнер необходимо установить пустой пароль (см. «Запуск Trusted TLS в режиме сервиса») и перезапустить сервис. Также проверьте срок действия лицензии КриптоПро CSP командой: */opt/cprocsp/sbin/<**arch**>/cpconfig -license -view*
[info] SSL Library Error: 336113695 error:1408B019:SSL routines:SSL3_GET_CLIENT_KEY_EXCHANGE:reason(31)	Проблема с доступом к ключевому контейнеру. Возможно, поможет переустановка серверного сертификата в личном хранилище пользователя, из-под которого запущен веб-сервер. Закрытый ключ при этом удалять не требуется!
[info] SSL Library Error: 336113769 error:1408B069:SSL routines:SSL3_GET_CLIENT_KEY_EXCHANGE:bad hello request	Проблема с доступом к ключевому контейнеру. Если он расположен на отчуждаемом носителе, то, возможно, данный носитель не вставлен в соответствующее устройство чтения.
[info] SSL Library Error: 336113774 error:1408B06E:SSL routines:SSL3_GET_CLIENT_KEY_EXCHANGE:bad dh p length	Проблема с доступом к ключевому контейнеру. Возможно, в другой сессии отображалось окно для ввода пароля, которое закрылось по истечению интервала ожидания.
[info] SSL Library Error: 336408641 error:140D3041:SSL routines:TLS1_SETUP_KEY_BLOCK:malloc failure	Установите директиве «SSLSessionCache» значение «none»: SSLSessionCache none
[info] SSL Library Error: 218595468 error:0D07808C:asn1 encoding routines:ASN1_ITEM_EX_D2I:mstring wrong tag [info] SSL Library Error: 218640442 error:0D08303A:asn1 encoding routines:ASN1_TEMPLATE_D2I:nested asn1 error [info] SSL Library Error: 336105485 error:1408900D:SSL routines:SSL3_GET_CLIENT_CERTIFICATE:ASN1 lib	Обновите Trusted TLS, т.к. в сборках ниже 584 не поддерживается работа с квалифицированными сертификатами.

При решении проблем можно использовать описание кодов ошибок вида 0xXXXXXXXX, расположенное по адресу http://msdn.microsoft.com/en-us/library/ms819775.aspx (англ.).

Если в данном описании нет информации, необходимой для решения Вашей проблемы, то поищите его в документации к модулю mod_ssl по адресу http://httpd.apache.org/docs/2.2/mod/mod_ssl.html (англ.), на форуме компании-разработчика или обратитесь в службу поддержки.

СПРАВКА О КОМПАНИИ

Компания ООО «Цифровые технологии» занимается разработкой программного обеспечения в области конфиденциального юридически значимого электронного документооборота. Разработанные нами программные продукты находят широкое применение в различных отраслях российской экономики: они используются государственными и коммерческими организациями:

- Государственной Думой РФ
- Министерством финансов республики Бурятия
- Управлением федерального казначейства Тверской области
- Управлением федерального казначейства Тульской области
- БКИ «Южное»
- ОАО «Центральный телеграф»
- ЗАО «Дельтабанк»
- ОАО «Белгородэнерго»
- МТС
- ЗАО «Центел»
- и другими

ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

ООО «Цифровые технологии»

Почтовый адрес: 424033, Россия, Республика Марий Эл, г. Йошкар-Ола, ул. Петрова, д.1, а/я 67

Юридический адрес: 424033, Россия, Республика Марий Эл, г. Йошкар-Ола, ул. Петрова, д.1

Телефон: 8 (8362) 33-70-50

Интернет: http://www.trusted.ru/support/

E-mail: support@trusted.ru