


TRUSTED TLS 3.0

Руководство по установке и настройке
с веб-серверами Apache 2.4 и nginx 1.18

Содержание

| | |
|---|-----------|
|  | |
| Описание продукта | 4 |
| Функциональные возможности | 4 |
| Быстрый старт..... | 5 |
| Установка КриптоПро CSP..... | 5 |
| Установка Trusted TLS | 7 |
| Установка лицензионного ключа Trusted TLS..... | 7 |
| Формирование псевдоключевого файла | 8 |
| Формирование псевдоключевого файла утилитой ctgostcp_util..... | 8 |
| Создание ключевой пары и запроса на сертификат, выпуск серверного сертификата и его установка в хранилище | 9 |
| Запуск сервера | 12 |
| Проверка аутентификации | 12 |
| Проверка работы односторонней аутентификации | 12 |
| Проверка работы двусторонней аутентификации..... | 12 |
| Особенности настройки Apache HTTP Server | 13 |
| Отличия Trusted TLS версии 3.0 от 2.0 и 2.2 | 13 |
| Подключение модуля..... | 13 |
| Переменные окружения..... | 13 |
| Директива SSLCryptoDevice | 14 |
| Директива SSLCipherSuite..... | 14 |
| Директива SSLCertificateKeyFile | 14 |
| Работа с двумя серверными сертификатами | 14 |
| Использование Apache в режиме HTTP/HTTPS прокси-сервера | 14 |
| Организация двусторонней аутентификации | 16 |
| Использование переменных сервера для аутентификации в веб-приложении..... | 17 |
| Особенности настройки nginx..... | 19 |
| Директива ssl_ciphers..... | 19 |
| Директива ssl_certificate_key | 19 |
| Сервера приложений: интеграция Trusted Java с Apache HTTP Server | 19 |
| Введение..... | 19 |
| Передача сертификатов от Apache-сервера серверам приложений | 20 |
| Пересылка запросов серверу приложений от Apache-сервера | 20 |

| | |
|---|-----------|
| Apache-модуль Mod_proxy..... | 20 |
| Apache-Tomcat mod_jk connector..... | 21 |
| Oracle WebLogic WebServer Plug-Ins..... | 23 |
| IBM WebSphere 7.0 WebServer Plug-Ins..... | 23 |
| Устранение ошибок | 25 |
| Техническая поддержка..... | 26 |
| Справка о компании | 27 |

ОПИСАНИЕ ПРОДУКТА

Программный продукт **Trusted TLS 3** — это доработанный криптографический пакет OpenSSL, позволяющий использовать российские стандарты криптографической защиты информации в утилите командной строки openssl, веб-серверах Apache HTTP Server и nginx. Поддержка сертифицированных в РФ алгоритмов электронной подписи, шифрования файлов и канала передачи информации реализована посредством вызова функций криптопровайдера «КриптоПро CSP».

Trusted TLS для веб-серверов поставляется вместе с доработанными веб-серверами. В случае с Apache-серверами, изменения производились только в модуле mod_ssl. Новый модуль, созданный на базе mod_ssl, называется mod_digt_tls, и в большинстве случаев его можно подключить к веб-серверам, входящими в состав популярных дистрибутивов Linux. В отличие от Apache HTTP Server, nginx на модули не делится, поэтому Trusted TLS внедрен в его единственный исполняемый файл.

Дистрибутивы **Trusted TLS 3** для веб-серверов собраны на базе OpenSSL версии 1.0.2u и веб-серверов Apache HTTP Server 2.4.46, а также nginx 1.18.0

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

Поддержка TLS/SSL протоколов:

- версии TLS: 1.0, 1.1 и 1.2 для сертифицированных и зарубежных алгоритмов;
- поддержка шифросюит (CipherSuite): TLS_GOSTR341001_WITH_28147_CNT_IMIT (0x0081) и TLS_GOSTR341112_256_WITH_28147_CNT_IMIT (0xFF85);
- 256-битное шифрование для в соответствии с ГОСТ 28147-89;
- поддержка X.509-сертификатов с ключами по ГОСТ Р 34.10-2012 (256 и 512 бит).

Поддерживаемые операционные системы:

- семейства Windows:
 - Windows 7/8/8.1/10/Server 2003/2008 (32-разрядные);
 - Windows 7/8/8.1/10/Server 2008/2008 R2/2012 (64-разрядные);
- совместимые с Linux Standard Base ISO/IEC 23360 (32- и 64-разрядные).

Дополнительные возможности **Trusted TLS версии 3.0**:

- совместим с «КриптоПро TLS» от компании «Крипто-Про»;
- возможность одновременной установки ГОСТ и RSA сертификатов на один виртуальный хост;
- поддерживает несколько ГОСТ-сертификатов на разных виртуальных хостах;
- аутентификация по ГОСТ-алгоритмам при проксировании;
- только в веб-серверах Apache: возможность одновременной установки ГОСТ и RSA сертификатов на одном виртуальном хосте;
- только в веб-серверах Apache: поддержка расширений сертификата Улучшенный ключ (Extended key usage) и Политики сертификата (Certificate policies) в переменных окружения сервера.

Ограничения Trusted TLS версии 3.0:

- отсутствует автоматическое скачивание Списков отозванных сертификатов;
- нельзя устанавливать пароли на ключевые контейнеры;
- не поддерживается проксирование на TLS-ресурсы, если алгоритм ключевой пары удаленного ресурса отличается от алгоритма серверного ключа на прокси-сервере.

БЫСТРЫЙ СТАРТ

Для быстрого ознакомления с продуктом Trusted TLS предлагаем воспользоваться нашими сборками веб-серверов. Их конфигурационные файлы настроены таким образом, что подготовить веб-сервер к приему TLS-соединений в режиме односторонней аутентификации можно за пять шагов:

1. установить подходящую версию СКЗИ «КриптоПро CSP»;
2. распаковать дистрибутив «Trusted TLS» в корень диска;
3. установить лицензионный ключ на «Trusted TLS»;
4. сохранить серверный сертификат рядом с конфигурационными файлами;
5. сформировать псевдоключевой файл для ключевого контейнера.

Разделы документации с описанием перечисленных выше шагов: «Установка КриптоПро CSP», «Установка Trusted TLS», «Установка лицензионного ключа Trusted TLS» и «Формирование псевдоключевого файла утилитой ctgostcp_util».

Если серверного сертификата нет, то его можно получить в тестовом Удостоверяющем центре Крипто-Про в соответствии с инструкцией «Создание ключевой пары и запроса на сертификат, выпуск серверного сертификата и его установка в хранилище».

УСТАНОВКА КРИПТОПРО CSP

Trusted TLS 3.0 совместим со следующими версиями СКЗИ «КриптоПро CSP»: 4.0 и 5.0.

Выбирать версию криптопровайдера следует исходя из операционной системы, поддержки требуемых криптографических алгоритмов и наличия сертификата ФСБ. При определении требуемой версии КриптоПро CSP удобно пользоваться таблицами на сайте разработчика: <http://www.cryptopro.ru/products/csp/compare>.

СКЗИ «КриптоПро CSP» необходимо использовать той же разрядности, что и веб-сервер. Устанавливать криптопровайдер следует в соответствии с его эксплуатационной документацией. Далее перечислены основные шаги по установке «КриптоПро CSP» для Windows и Linux.

Для Windows:

1. Установите КриптоПро CSP с помощью программы-инсталлятора. Для целей тестирования не следует устанавливать криптопровайдер в исполнении KC2, т.к. он предназначен для работы с аппаратными ДСЧ.

2. Введите лицензию на КриптоПро CSP в соответствии с документацией. Без лицензии криптопровайдер будет работать в полнофункциональном режиме в течение трех месяцев (с момента первой установки).
3. При необходимости настройте требуемые считыватели КриптоПро CSP через панель управления в соответствии с Инструкцией по использованию криптопровайдера.

Для Linux:

В дальнейшем под обозначением `<arch>` будет подразумеваться один из следующих идентификаторов платформы:

- `ia32` - для 32-разрядных систем;
- `amd64` - для 64-разрядных систем.

1. Установите КриптоПро CSP из дистрибутива в следующем порядке:

- пакеты, устанавливаемые 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
- основные пакеты криптопровайдера (для варианта исполнения KC1, 64-разрядные):
lsb-cprocsp-base – общий пакет для 32- и 64-разрядных криптопровайдеров
lsb-cprocsp-rdr-64 – базовые считыватели
lsb-cprocsp-kc1-64 – библиотеки криптопровайдера
lsb-cprocsp-capilite-64 – библиотеки CryptoAPI 2.0 Lite и утилиты
- дополнительно может потребоваться установка вспомогательных пакетов, например:
lsb-cprocsp-rdr-pcsc – считыватели токенов и смарт-карт

Команда по установке пакетов для операционных систем с поддержкой менеджера пакетов RPM: `rpm -ivh <rpm-файл>`

На Debian/Ubuntu сначала нужно установить системные пакеты `lsb-base` `lsb-core` и `alien`, после чего устанавливать пакеты криптопровайдера: `alien -kci <rpm-файл>`

2. При необходимости укажите правильное расположение в системе файла `libcurl.so` в файле `/etc/opt/cprocsp/config.ini` (`config64.ini` для 64-разрядной сборки), строке:
`"libcurl.so" = "/usr/local/lib/libcurl.so"`
3. Введите лицензию на КриптоПро CSP 4.0 с помощью команды

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

Без установленной лицензии КриптоПро CSP три месяца с момента первой установки проработает в полнофункциональном режиме.

УСТАНОВКА TRUSTED TLS

Установку продукта «Trusted TLS» рекомендуется выполнять после установки СКЗИ «КриптоПро CSP».

Дистрибутивы Trusted TLS для веб-серверов поставляются в архивах со следующими названиями:

- TrustedTLS-3-WebServer-X.X.X-rYYY.<platform>.zip (для 32- и 64-разрядных Windows);
- TrustedTLS-3-WebServer-X.X.X-rYYY.<platform>.tar.gz (для 32- и 64-разрядных Linux)

где «WebServer-X.X.X» и «rYYY» – название, версия и номер сборки веб-сервера.

Для Windows:

В дальнейшем под значением `%TRUSTED_TLS_HOME%` будет приниматься каталог `«C:\opt\TrustedTLS-3_WebServer»`.

Для установки продукта необходимо выполнить следующие шаги:

1. Если в системе отсутствует Microsoft Visual C++ 2008 SP1 Redistributable Package, то установите его. Скачать Microsoft Visual C++ 2008 SP1 Redistributable Package можно по следующим ссылкам:
 - для 32-разрядной ОС: <http://www.microsoft.com/download/en/details.aspx?id=5582>
 - для 64-разрядной ОС: <http://www.microsoft.com/download/en/details.aspx?id=2092>
2. Распакуйте в корень диска архив дистрибутива Trusted TLS с веб-сервером.
3. Согласно инструкции ниже установите лицензионный ключ на продукт.

Для Linux:

В дальнейшем под значением `$TRUSTED_TLS_HOME` будет приниматься каталог `«/opt/TrustedTLS-3_WebServer»`.

Для установки продукта необходимо выполнить следующие шаги:

1. Распакуйте в корень файловой системы архив дистрибутива Trusted TLS с веб-сервером.
2. Согласно инструкции ниже установите лицензионный ключ на продукт.

УСТАНОВКА ЛИЦЕНЗИОННОГО КЛЮЧА TRUSTED TLS

Лицензионный ключ продукта состоит из семи групп по пять символов, разделенных дефисами. Для установки лицензионного ключа в систему необходимо запустить с правами администратора скриптовый файл:

- в Windows: `%TRUSTED_TLS_HOME%\bin\ctsamples\TrustedTLS3-reg.bat`;
- в Linux: `$TRUSTED_TLS_HOME/bin/ctsamples/TrustedTLS3-reg.sh`.

При этом следует иметь в виду, что данные скрипты не проверяют формат и срок действия введенного лицензионного ключа, а лишь записывают его в реестр (в Windows) или в файл (в Linux).

ФОРМИРОВАНИЕ ПСЕВДОКЛЮЧЕВОГО ФАЙЛА

В настройках любого веб-сервера вместе с серверным сертификатом необходимо указывать соответствующий ему ключевой файл. В данном разделе рассматриваются несколько сценариев формирования такого файла для ключевого контейнера «КриптоПро CSP».

Если на компьютере с веб-сервером имеется ключевой контейнер, управляемый СКЗИ «КриптоПро CSP», и соответствующий ему сертификат, то экспортируйте сертификат в файл (в формате Base64 и обязательно со строками «-----BEGIN CERTIFICATE-----» и «-----END CERTIFICATE-----»). Затем по инструкции ниже сформируйте псевдоключевой файл, который будет ссылаться на ключевой контейнер криптопровайдера.

Если ключевой контейнер «КриптоПро CSP» и соответствующий ему сертификат имеются, но не пока установлены на компьютере с веб-сервером, то сначала необходимо подключить ключ и установить сертификат в хранилище личных сертификатов, и затем по инструкции «Формирование псевдоключевого файла утилитой `ctgostcp_util`» сформировать псевдоключевой файл.

Если у Вас нет сертификата TLS-сервера, то получите его в Удостоверяющем центре. Для целей тестирования можно выпустить сертификат в тестовом УЦ компании Крипто-Про, см. инструкцию ниже.

Формирование псевдоключевого файла утилитой `ctgostcp_util`

Т.к. ключевая пара создается и используется криптопровайдером «КриптоПро CSP», то в составе Trusted TLS (в подкаталоге `bin`) содержится утилита **`ctgostcp_util`**, которая позволяет на основе существующего ключевого контейнера «КриптоПро CSP» сформировать файл-ссылку на него, а также протестировать корректность такого псевдоключевого файла.

Вариант 1: для создания псевдоключевого файла по сертификату, установленному в личном хранилище сертификатов «КриптоПро CSP», экспортируйте его из хранилища в файл `server-gost.crt` и запустите команду:

```
ctgostcp_util generate -certfile <имя_файла_сертификата> -out <имя_файла_ключа>
```

Для экспорта сертификата из хранилища Windows воспользуйтесь оснасткой Пуск – Все программы – Крипто-Про – Сертификаты.

В ОС Linux экспортировать сертификат необходимо в 2 этапа. Сначала он записывается в бинарном формате (DER), а затем полученный файл нужно преобразовать в Base64:

- 1) `/opt/cprosp/bin/<arch>/certmgr -export -dest ./server-gost.cer`
- 2) `$TRUSTED_TLS_HOME/bin/openssl.exe x509 -in ./server-gost.cer -inform DER -out ./server-gost.crt`

Вариант 2: для формирования псевдоключевого файла по имени контейнера выполните команду:

```
./ctgostcp_util generate -container <имя_контейнера> -out <имя_файла_ключа>
```

Получить список контейнеров закрытых ключей «КриптоПро CSP» можно утилитой **`csptestf`** (**`csptest.exe`** для Windows). Ниже приведены примеры ее использования:

Для Windows:

```
"%ProgramFiles%\Crypto Pro\CSP\csptest.exe" -keyset -provtype 75 -enum_containers -verifycontext -fqcn
```

Для Linux:

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

Создание ключевой пары и запроса на сертификат, выпуск серверного сертификата и его установка 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. Если Вы получили серверный сертификат с ключом на отчуждаемом носителе, то подключите его к компьютеру и сформируйте псевдоключевой файл по имени ключевого контейнера в соответствии с информацией в разделе «Формирование псевдоключевого файла».

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».

Для Windows:

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

Рекомендуемый способ:

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 «КС2».

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». Но вводить pin-код не следует, т.к. Trusted TLS 3.0 не поддерживает ввод пароля при запуске сервера.

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.

Альтернативные способы:

Если по каким-то причинам не удалось сгенерировать ключевой контейнер в Internet Explorer, то есть возможность это сделать с помощью утилиты csptest.exe, входящий в состав КриптоПро CSP, либо с помощью утилиты openssl.exe.

Примеры команд для csptest.exe приведены в этом же разделе, только для Linux.

Примеры команд для openssl.exe см. в файле скрипта, входящем в поставку Trusted TLS %TRUSTED_TLS_HOME%\bin\ctsamples\create-key.bat.

Для Linux:

Рекомендуемый способ:

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/cproccsp/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/cproccsp/bin/<arch>/cryptcp -creatrqst -provtype <provtype> -both -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
```

В качестве <provtype> укажите 75 (для ГОСТ 34.10-2001), 80 (для ГОСТ 34.10-2012, 256 бит) или 81 (для ГОСТ 34.10-2012, 512 бит).

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/cproccsp/bin/<arch>/csptestf -passwd -change <новый_пароль> -provtype <provtype> -  
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/cproccsp/bin/<arch>/cryptcp -instcert -provtype <provtype> /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/cproccsp/bin/<arch>/certmgr -inst -file /tmp/server-gost.cer -cont "\\\\.\\HDIMAGE\\tlsserver"
```

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

Альтернативный способ:

Сгенерировать ключевой контейнер можно также с помощью утилиты openssl.exe, входящей в поставку Trusted TLS. В скрипте \$TRUSTED_TLS_HOME/bin/ctsamples/create-key.sh есть примеры команд для выполнения данной операции.

ЗАПУСК СЕРВЕРА

Старт веб-сервера выполняется стандартным для него способом.

Внимание! Если при доступе к ключевому контейнеру «КриптоПро CSP» возникнут проблемы, то веб-сервер может запуститься без аварийного завершения, при этом ошибки будут зафиксированы в его журнале.

ПРОВЕРКА АУТЕНТИФИКАЦИИ

Проверка работы односторонней аутентификации

Перед обращением к серверу по TLS-протоколу установите в хранилище "Доверенные корневые Центры Сертификации" сертификат УЦ, которым подписан серверный сертификат.

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>.

Проверка работы двусторонней аутентификации

Настройка клиентской аутентификации описана в разделе «Организация двусторонней аутентификации».

Проверка работы клиентской 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» (Extended key usage, 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».

ОСОБЕННОСТИ НАСТРОЙКИ APACHE HTTP SERVER

Большинство настроек `mod_digt_tls` совпадает с настройками `mod_ssl` (см. официальную документацию по `mod_ssl` на веб-страницах http://httpd.apache.org/docs/2.2/mod/mod_ssl.html и http://httpd.apache.org/docs/2.4/mod/mod_ssl.html). В данном разделе перечислены директивы, требующие специфичной настройки для работы Apache HTTP Server с Trusted TLS.

Отличия Trusted TLS версии 3.0 от 2.0 и 2.2

Сертификаты УЦ не требуется размещать в хранилищах КриптоПро CSP. Вместо этого нужно обязательно включить директиву `SSLCACertificateFile` (или `SSLCACertificatePath`).

Отсутствует автоматическая загрузка списков отзыва сертификатов. Вместо этого требуется обязательное включение директивы `SSLCARevocationFile` (или `SSLCARevocationPath`).

Поведение директивы `SSLVerifyDepth` теперь соответствует официальной документации на Apache HTTP Server.

В директиве `SSLCertificateKeyFile` необходимо указывать псевдоключевой файл. Его создание описано в разделе «Формирование псевдоключевого файла».

Поменялись идентификаторы ГОСТовых шифросюит, новые значения см. в разделе «Директива `SSLCipherSuite`».

Подключение модуля

Файл модуля Trusted TLS называется `mod_digt_tls.so`, поэтому для его подключения добавьте в конфигурационный файл строку

```
LoadModule ssl_module modules/mod_digt_tls.so
```

Переменные окружения

В `mod_digt_tls` добавлены переменные сервера для следующих расширений сертификатов:

- Улучшенный ключ (Extended key usage, EKU) – `SSL_CLIENT_EKU` и `SSL_SERVER_EKU`;
- Политики сертификата (Certificate policies) – `SSL_CLIENT_POLICIES` и `SSL_SERVER_POLICIES`.

Значения данных переменных окружения состоят из объектных идентификаторов, разделенных косой чертой. В начале и в конце содержится еще по косой черте.

Если в сертификате отсутствует расширение «Улучшенный ключ», то в качестве значения возвращается строка «`SSL_EKU_ANY`».

Пример использования `SSL_CLIENT_EKU` в `SSLRequire`:

```
SSLRequire % {SSL_CLIENT_EKU} =~ m/\x2F1\3\6\1\5\5\7\3\2\x2F/
```

Директива SSLCryptoDevice

Для поддержки «КриптоПро CSP» данную директиву необходимо включить со следующим значением:

```
SSLCryptoDevice ctgostcp
```

Директива SSLCipherSuite

В дополнение к зарубежным алгоритмам в данной директиве поддерживаются следующие значения:

- GOST2001-GOST89-GOST89 – является идентификатором шифросюиты 0x0081 (TLS_GOSTR341001_WITH_28147_CNT_IMIT);
- GOST2012-GOST89-GOST89 – является идентификатором шифросюиты 0xFF85 (TLS_GOSTR341112_256_WITH_28147_CNT_IMIT);
- kGOST – обозначает группу шифросюит с обменом ключами по ГОСТ-схемам, в которую входят обе вышеуказанные шифросюиты.

Директива SSLCertificateKeyFile

Данная директива должна указывать на ключевой файл, который следует формировать по инструкции из раздела «Формирование псевдоключевого файла».

Работа с двумя серверными сертификатами

Apache HTTP Server с 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).

При выборе шифросюиты (алгоритма шифрования соединения) будут учитываться шифросюиты, посылаемые клиентским программным обеспечением. Если браузер поддерживает ГОСТ TLS, например, Internet Explorer с установленным КриптоПро CSP, и директива *SSLHonorCipherOrder* установлена в *off*, то выбор алгоритма шифрования будет осуществляться исходя из предпочтений браузера. Если *SSLHonorCipherOrder* включена, то шифросюита будет вычислена по правилу из директивы *SSLCipherSuite*.

Использование Apache в режиме HTTP/HTTPS прокси-сервера

Apache HTTP Server с 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).

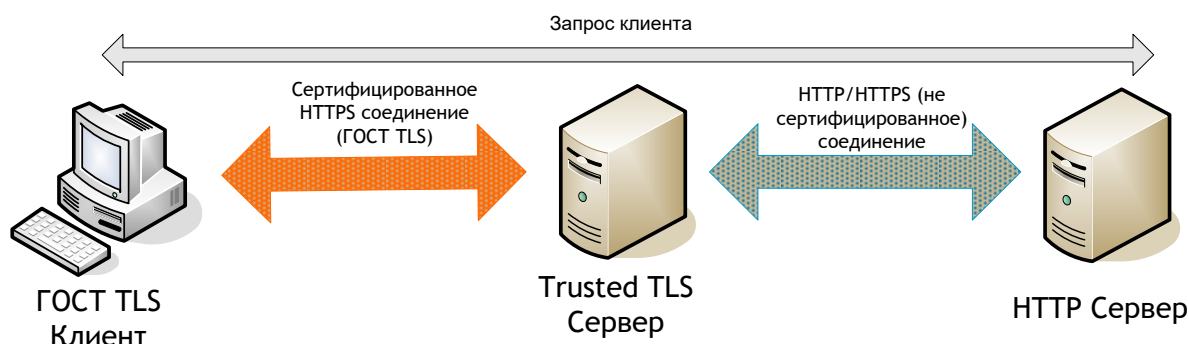


Рис. 1

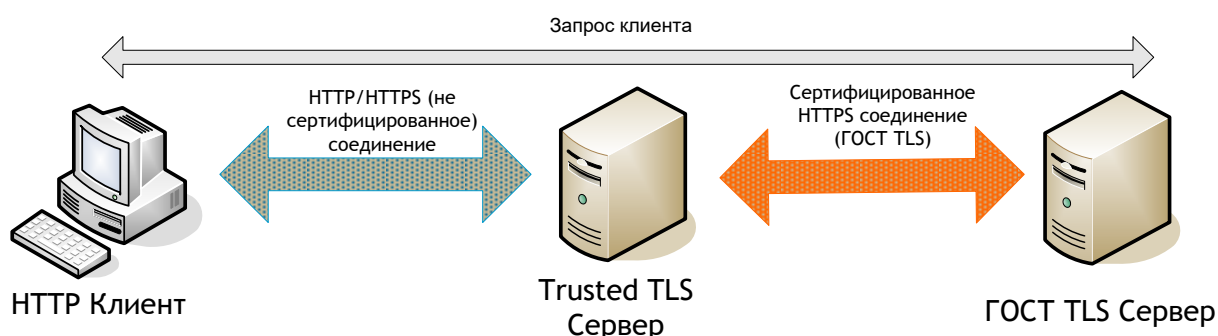


Рис. 2

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`.

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/server-proxy-gost.pem
```

где **server-proxy-gost.pem** - это файл с одним или несколькими клиентскими сертификатами и ключами в формате Base64:

```
-----BEGIN CERTIFICATE-----
MIIDkDCCAnigAwIBAgIKN7ozGAABAAAkzANBgkqhkiG9w0BAQUFADBGMRUwEwYK
/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 PRIVATE KEY-----
MDsCAQAwCgYGGKoUDAgITBQAEKjAoDBxcXC5ccmVnaXN0cnlzdGVzdHRsczMwLXVz
ZXIxoAMBAQChAwIBAQ==
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIBXzCCAQwCCQDa+k13w0caADAKBgYqhQMCAgMFADA2MQswCQYDVQQGEwJSVT
EY
...
LTvJQ+wwMD8oC2cA9+Cpx39cHQ==
-----END CERTIFICATE-----
```

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

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-аутентификацией».

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-пароллю к авторизации по сертификатам или одновременного использования обеих схем для разных категорий пользователей.

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`.

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} eq "Москва" \
    and %{SSL_CLIENT_EKU} =~ m/\x2F1\3\6\1\5\5\7\3\2\x2F/ \
    and %{SSL_CLIENT_EKU} =~ m/\x2F1\2\643\2\2\34\./
</Location>
```

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 "РѣPsCГPePIP°"
```

Вместо «`%{SSL_CLIENT_S_DN_L} eq "Москва"`» можно использовать следующее регулярное выражение:

```
%{SSL_CLIENT_S_DN_L} =~ m/\A\xD0\x9C\xD0\xBE\xD1\x81\xD0\xBA\xD0\xB2\xD0\xB0/Z/
```

Последовательность `\x2F` в строках с `%{SSL_CLIENT_EKU}` обозначает косую черту, которая разделяет и обрамляет объектные идентификаторы.

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

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 поддерживаются переменные `SSL_CLIENT_EKU` и `SSL_CLIENT_POLICIES`, описание которой приводится в разделе «Переменные окружения».

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.

ОСОБЕННОСТИ НАСТРОЙКИ NGINX

Большинство настроек совпадает с настройками ngx_http_ssl_module (см. официальную документацию http://nginx.org/ru/docs/http/ngx_http_ssl_module.html). В данном разделе перечислены директивы, требующие специфичной настройки для работы nginx с Trusted TLS.

Директива ssl_ciphers

В дополнение к зарубежным алгоритмам в данной директиве поддерживаются следующие значения:

- GOST2001-GOST89-GOST89 – является идентификатором шифросюиты 0x0081 (TLS_GOSTR341001_WITH_28147_CNT_IMIT);
- GOST2012-GOST89-GOST89 – является идентификатором шифросюиты 0xFF85 (TLS_GOSTR341112_256_WITH_28147_CNT_IMIT);
- kGOST – обозначает группу шифросюит с обменом ключами по ГОСТ-схемам, в которую входят обе вышеуказанные шифросюиты.

Директива ssl_certificate_key

Данная директива должна указывать на ключевой файл, который следует формировать по инструкции из раздела «Формирование псевдоключевого файла».

СЕРВЕРА ПРИЛОЖЕНИЙ: ИНТЕГРАЦИЯ TRUSTED JAVA С APACHE HTTP SERVER

Введение

В предлагаемом ниже материале рассматриваются методики построения защищенного канала до серверов приложений (Tomcat, IBM Websphere, Oracle Weblogic) на базе Apache HTTP Server 2.2 с Trusted TLS 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 связей:

- Apache 2.2 + Trusted TLS 3 + CSP 3.6/3.9/4.0 + Tomcat 5.5/6.0/7.0 + Trusted Java 2.0

- Apache 2.2 + Trusted TLS 3 + CSP 3.6/3.9/4.0 + IBM Websphere 6.1/7.0 + Trusted Java 2.0
- Apache 2.2 + Trusted TLS 3 + CSP 3.6/3.9/4.0 + Oracle WebLogic Server 10.3.2 + Trusted Java 2.0

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** или 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 соответствии 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 АJP-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" protocol="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 веб-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  
    #WLTmpDir      "/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, который могут возникнуть на этапе развертывания или в процессе использования «Trusted TLS 3» с веб-серверами Apache HTTP 2.4 и nginx 1.18, 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 |
|---|--|
| Ошибки при старте веб-сервера: | |
| "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 или Используется сборка Trusted TLS, не совместимая с установленной версией КриптоПро CSP (информация о соответствии приведена в разделе «Установка Trusted TLS»). |
| 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 Удостоверяющего центра, выпустившего серверный сертификат. |
| 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 «Целостность этого сертификата не гарантирована. Возможно, он поврежден или изменен. | Установите на клиентском рабочем месте тот сертификат УЦ, которым был подписан серверный сертификат. Или установите все сертификаты Уполномоченных лиц данного УЦ. |
| Не стартует сервер на 64-битной ОС семейства Unix. | Попробуйте перед запуском сервера выполнить команды (<arch> замените на соответствующий каталог): LD_LIBRARY_PATH= /opt/cprosp/lib/<arch>:\$LD_LIBRARY_PATH export LD_LIBRARY_PATH LD_PRELOAD=/opt/cprosp/lib/<arch>/libcapi20.so export LD_PRELOAD |
| Ошибки, записываемые в лог в процессе работы Apache HTTP Server: | |
| [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. |

| | |
|--|---|
| [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. |
|--|---|

Если в данном разделе нет информации, необходимой для решения Вашей проблемы, то поищите его в документации к модулю mod_ssl (для Apache) или nginx, либо обратитесь в нашу службу поддержки.

ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

ООО «Цифровые технологии»

Почтовый адрес 424033, Россия, Республика Марий Эл, г. Йошкар-Ола, а/я 67

Телефон/факс +7 (8362) 33-70-50

Веб-сайт <http://www.trusted.ru/support/>

Электронная почта:

- общие вопросы info@trusted.ru

- технические вопросы support@trusted.ru

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
- Федеральным казначейством
- Федеральной налоговой службой
- МВД
- и другими