Trusted Java 2.0

Описание по установке

Цифровые технологии ®

2018 г.

Настоящий документ содержит описание операций по установке и настройке криптопровайдера КриптоПро CSP и криптографической библиотеки Trusted Java. А также ее настройка совместно с Tomcat и PHP.

Руководство предназначено для администраторов системы. В нем содержится информация, необходимая для установки, настройки и эксплуатации Trusted Java, а также описание функциональных возможностей программного продукта.

По всем вопросам обращайтесь в службу технической поддержки по email support@trusted.ru

История версий

В сборке r507 доработано чтение сертификатов из хранилища. Ранее не возвращались сертификаты с алгоритмом публичного ключа ГОСТ 2012 (256 - 512). Использование:

CryptoProCSPCertStore storeCsp = new CryptoProCSPCertStore();

Collection certificates = storeCsp.getInstance("CurrentUser/ROOT", null).getAllCertificates();

В сборке r506 добавлена возможность проверки подписи с отключением/включением сортировки подписанных атрибутов (по умолчанию они не сортируются. Использование:

SignerInformation si;

...

si.setNeedSortSignedAttributes(true);

si.verify(cert, PROV_NAME))

В сборке r505 добавлена поддержка алгоритмов ГОСТ 2012 для XML.

В версии Trusted Java 2.0 (сборка r504 и выше) добавлена поддержка работы с ключами и алгоритмами ГОСТ Р 34.11-2012/ГОСТ Р 34.10-2012 (только с КриптоПро CSP 4.0).

В версии Trusted Java 2.0 изменилась система лицензирования. Теперь на серверных версиях ОС Windows для операций подписи и шифрования, а также для использования клиентских сокетов достаточно клиентской лицензии. Проверка подписи стала бесплатной, на всех платформах. Теперь полученный лицензионный ключ действителен только для одной платформы ОС и не подходит к другой. Поэтому следует пользователям, переходящим на новую версию, обновить или приобрести под другую платформу лицензионные ключи.

В версии Trusted Java 2.0 обеспечена обратная совместимость с продуктами **КриптоАРМ и КриптоАРМ ГОСТ** в плане создания и проверки ЭЦП и шифрования данных. Это, в частности, достигнуто за счет адаптации JCE провайдера для использования его в связке с библиотеками от BounceCastle: оригинальной библиотеки JCE-провайдера bcprov-jdk15-145.jar и пропатченной bcmail-библиотеки bcmail-jdk15-145-tj20-rbbb.jar, ссылки на которые можно получить в центре загрузки. В результате появилась возможность создавать и проверять заверяющие подписи.
С появлением нового проекта стандарта *в версии Trusted Java 2.0* введена поддержка нового пространства имен при создании XML-подписей.

В версии Trusted Java 2.0 улучшена работа с SSL-сокетами, усовершенствованы механизмы аутентификации сервера и клиента, реализована поддержка шифросюиты (CipherSuite) 0x81 (TLS_GOSTR341001_WITH_28147_CNT_IMIT).

Версия Trusted Java 1.5.3 поддерживает работу с КриптоПро CSP 3.6.
ВНИМАНИЕ!!! Использование классов, реализующих работу по алгоритму ГОСТ Р 34.10-94, совместно с КриптоПро CSP 3.6 и выше не поддерживается.

Версия Trusted Java 1.5.2 включает в себя как совершенно новые, так и оптимизированные по сравнению с более ранними версиями, функциональные возможности:

- обеспечена совместимость JSSE с веб-сервером Apache Tomcat 5.5;
- добавлены новые классы для обеспечения совместимости XMLSig с MS XML;
- оптимизирована подпись группы файлов на одном ключе;
- добавлена возможность получения списка ключевых контейнеров, подключенных к системе.

Общие сведения

Программный продукт Trusted Java является средством криптографической защиты информации и представляет собой набор криптоалгоритмов, реализованных в соответствии с требованиями архитектур JSSE и JCE.

Провайдер JSSE обеспечивает организацию защищенного взаимодействия по протоколам SSL и TLS с использованием российских криптографических алгоритмов. Таким образом, он даёт возможность устанавливать безопасные интернет-соединения и обеспечивать защиту канала передачи данных. Кроме того, компонент включает функциональность для шифрования данных, аутентификации клиента и сервера и целостности сообщений.

Провайдер JCE является криптографическим расширением Trusted Java, т.е. позволяет реализовать российские алгоритмы криптографических преобразований для обеспечения работы с сертификатами, создания и проверки корректности электронной цифровой подписи и шифрования данных.

ПО Trusted Java обеспечивает возможность поддержки российской криптографии в соответствии с положениями российского законодательства. Для авторизации и обеспечения юридической значимости электронных документов при обмене ими между пользователями используются функции формирования и проверки электронно-цифровой подписи, для обеспечения конфиденциальности информации и контроля ее целостности – шифрование и имитозащита.

Также Trusted Java позволяет работать с сертификатами и запросами на сертификат, создавать и проверять корректность электронной цифровой подписи данных, шифровать данные, загружать Java апплеты по протоколу TLS на криптографических алгоритмах ГОСТ.

Используемые сертифицированные российские криптоалгоритмы предназначены для:

авторизации и обеспечения юридической значимости электронных документов при обмене ими между пользователями (создание и проверка ЭЦП).

обеспечения конфиденциальности и контроля целостности информации (шифрование и имитозащита).

Библиотека Trusted Java предназначена, прежде всего, для системных интеграторов и разработчиков прикладных и бизнес-систем для решения

вопросов информационной безопасности и юридической значимости электронного документооборота. При работе с банковскими клиент-серверными системами, защищенными электронными торговыми площадками и другими бизнес-приложениями, написанными на Java, требуется поддержка шифрования, электронной цифровой подписи, строгой аутентификации в соответствии с положениями российского законодательства. ПО Trusted Java позволяет использовать эти возможности.

Состав продукта

В состав программного продукта входит:

Провайдер JCE

Провайдер JSSE

Документация

- Руководство администратора
- Руководство разработчика
- Описание новинок версии

Лицензионное соглашение

Приложение "Удалить" (только для ОС Windows)

Программа управления лицензиями (только для ОС Windows)

Функциональные возможности продукта

Библиотека Trusted Java реализует в Java-приложениях сертифицированные криптографические алгоритмы, предоставляемые криптопровайдером КриптоПро CSP от компании «Крипто-ПРО».

Функциональные возможности	**Направления работы**	**Возможные операции**
Поддержка криптографических алгоритмов	ГОСТ 28147-89 (шифрование)	Предназначен для обеспечения конфиденциальности информации и контроля ее целостности посредством шифрования имитозащиты
	ГОСТ Р 34.11-94 (хеширование) ГОСТ Р 34.10-94 (ЭЦП, поддержка оставлена только	Предназначены для авторизации и обеспечения юридической значимости электронных документов при обмене ими между

	для совместимости с предыдущими версиями) ГОСТ Р 34.10-2001 (ЭЦП) ГОСТ Р 34.11-2012 ГОСТ Р 34.10-2012	пользователями. Это достигается благодаря использованию процедур: - создания ЭЦП - проверки ЭЦП
Работа с сертификатами и запросами на сертификат	Операции с цифровыми сертификатами	создание сертификата установка сертификата в хранилище работа со списками отзванных сертификатов (COC) поддержка CMS поддержка TSP поддержка OCSP
	Операции с запросами на сертификат	формирование запроса на сертификат
Криптографические операции	Электронная цифровая подпись (ЭЦП) данных	создание ЭЦП проверка корректности ЭЦП по сертификату создание ЭЦП в XML документах проверка ЭЦП в XML документах
	Шифрование	шифрование данных
Загрузка Java-апплетов	возможность загрузки Java-апплетов по ГОСТ TLS для Internet Explorer 6 (и выше) с установленным Sun Java Plugin 1.5 (и выше)	
Поддержка JCE и JSSE	провайдер JCE	поддержка ГОСТ-алгоритмов в протоколе на стороне Сервера поддержка двухфакторной аутентификации по ГОСТ сертификатам на стороне Клиента

		поддержка прокси на стороне Клиента
	провайдер JSSE	поддержка ГОСТ в TSP

Требования к программному обеспечению

Для установки и функционирования программного продукта Trusted Java необходимы следующие компоненты:

- Операционная система
  - Windows 2000 SP4 и выше (32|64 bit),
  - RedHat Enterprise Linux 4|5 (32|64 bit),
- Криптопровайдер КриптоПро CSP версии 4.0;
- Java 2 (Sun JRE версии 1.6 и выше);
- Дополнительное ПО
  - для Windows: библиотеки Microsoft Visual C++ 2008 SP1;
  - для unix-платформ: библиотеки GCC.

Установка ПО

В главе Установка ПО содержится информация, необходимая для установки и первоначальной настройки обязательных для корректной работы ПО составляющих: криптопровайдера КриптоПро CSP и, непосредственно, самого программного продукта Trusted Java.

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

Установка СКЗИ КриптоПро CSP

Процедура установки криптопровайдера КриптоПро CSP проходит в 3 этапа:

- ознакомление с требованиями к системе
- установка дистрибутива СКЗИ КриптоПро CSP

- изменение списка устройств хранения ключевой информации

Установка дистрибутива СКЗИ КриптоПро CSP под Windows

Установка дистрибутива должна производиться пользователем, имеющим права администратора.

Перед установкой дистрибутива ПО СКЗИ КриптоПро CSP удалите все ранее существующие версии устанавливаемого ПО.

Если модуль криптографической поддержки не удален, новая версия не будет установлена.

Для удаления ранее установленного ПО СКЗИ КриптоПро CSP используйте пункты основного меню Windows Пуск -> Панель управления -> Установка и удаление программ.

Для установки ПО запустите на исполнение файл дистрибутива. Далее следуйте стандартным инструкциям установщика.

После завершения установки дистрибутива перезагрузите компьютер.

Изменение списка устройств хранения ключевой информации под Windows

При инсталляции программного обеспечения по умолчанию устанавливаются все модули, обеспечивающие работу с различными поддерживаемыми устройствами хранения ключевой информации.

Если для работы с ПО СКЗИ необходимы дополнительные типы устройств работы с ключевыми носителями, выберите режим изменения их состава. Для этого:

Откройте панель управления компьютером, используя пункты меню Пуск -> Панель управления.

В окне панели управления выберите значок КриптоПро CSP.

В панели настройки СКЗИ КриптоПро CSP выберите закладку «Оборудование» и, нажав кнопку «Настроить считыватели...», добавьте (или удалите) из списка те устройства, которые будут (или не будут) использованы в качестве считывателей ключевой информации.

Установка дистрибутива СКЗИ КриптоПро CSP под Linux

1. Установите криптопровайдер КриптоПро CSP 4.0 из дистрибутива.
   1.1.Для систем с поддержкой RPM-пакетов установите RPM-пакеты

командой rpm -ivh <файл_пакета> в следующем порядке:

1.1.1. 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

1.1.2. дополнительно может потребоваться установка вспомогательных пакетов:

curl – библиотека для скачивания файлов по FTP, HTTP и некоторым другим протоколам

lsb-cprocsp-rdr-pcsc – считыватели токенов и смарт-карт

2. 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 4.0 будет работать 3 месяца в полнофункциональном режиме. Приобрести бессрочную лицензию можно в магазине: https://cryptoarm.ru

Установка Trusted Java под Windows

Перед установкой Trusted Java на платформе семейства Microsoft Windows желательно, чтобы установка Java Runtime Environment уже была произведена, т.к. в ней будут отражены изменения в процессе установки Trusted Java . Их можно будет увидеть в окне инсталлятора. Для установки Trusted Java выполните команду, например,

```
TrustedJava_x86_2.0.465.exe
```

Если текущая ОС поддерживает механизм контроля учетных записей пользователей (UAC), то потребуется разрешить запуск инсталлятора от имени администратора. Далее установка продолжается стандартным образом.

После установки Trusted Java может потребоваться установить в системе Microsoft Visual C++ 2008 SP1 Redistributable Package, если такового в системе нет. Для этого в основном меню Пуск\Все Программы\ Digt\Trusted Java 2.0 выбрать соответствующий пункт и пройти по основным шагам его установки.

После установки может потребоваться из подкаталога jars каталога установки скопировать jar-файлы в каталог $JRE\lib\ext. В частности, для поддержки операций с XML-файлами это будут serializer.jar, xalan.jar, xmlsec.jar. Дополнительно могут понадобиться commons-logging.jar и junit.jar.

Установка Trusted Java под Linux

Предполагается, что перед установкой Trusted Java уже была проведена установка Java Runtime Environment, например, в каталог /usr/java/jre1.6.0_17. При дальнейшем рассмотрении будем считать, что значение этого каталога присвоено переменной окружения JRE.

Данный продукт поставляется в виде следующих пакетов:

- TrustedJava-<version>-<revision>-<platform>.zip (для Windows)
- TrustedJava-<version>-<revision>-<platform>.tar.gz (для Unix-систем с КриптоПро CSP 3.6 и 3.6 R2)
- TrustedJava-<version>-R3-<revision>-<platform>.tar.gz (для Unix-систем с КриптоПро CSP 3.6 R3 и выше)
- TrustedJava-<version>-R3-<revision>-<platform>.tar.gz (для Unix-систем с КриптоПро CSP 4.0)
- trustedjava-<version>-<revision>.<platform>.rpm (для Unix-систем)

Процесс установки производится с полномочиями пользователя **root**. Нужно распаковать в корень диска содержимое дистрибутива Trusted Java:

```
tar -xf ./TrustedJava_2.0_Linux.RHEL4.i386_r455.tgz -C /
```

В каталоге $JRE/lib/i386 требуется создать символическую ссылку на файл "/opt/DIGT/Trusted Java 2.0/lib/<**arch**>/libdjcp20.so":

```
ln -s "/opt/DIGT/Trusted Java 2.0/lib/<arch>/libdjcp20.so" $JRE/lib/i386/libdjcp20.so
```

В каталоге $JRE/lib/ext требуется также создать символические ссылки на все jar-файлы в каталоге "/opt/DIGT/Trusted Java 2.0/jars":

```
ln -s "/opt/DIGT/Trusted Java 2.0/jars/commons-logging.jar" $JRE/lib/ext/commons-logging.jar

ln -s "/opt/DIGT/Trusted Java 2.0/jars/junit.jar" $JRE/lib/ext/junit.jar

ln -s "/opt/DIGT/Trusted Java 2.0/jars/trusted_java20.jar" $JRE/lib/ext/trusted_java20.jar

ln -s "/opt/DIGT/Trusted Java 2.0/jars/xalan.jar" $JRE/lib/ext/xalan.jar
```

ln -s "/opt/DIGT/Trusted Java 2.0/jars/xmlsec.jar" $JRE/lib/ext/xmlsec.jar

ln -s "/opt/DIGT/Trusted Java 2.0/jars/serializer.jar" $JRE/lib/ext/serializer.jar

Для деинсталляции продукта и возможности возврата первоначальных настроек нужно сохранить файл $JRE/lib/security/java.security в $JRE/lib/security/java.security.tj:

cp $JRE/lib/security/java.security $JRE/lib/security/java.security.tj

В файле $JRE/lib/security/java.security требуется

- Добавить строчку:

  security.provider.N=com.digt.trusted.jce.provider.DIGTProvider

  где N – первый свободный номер провайдера

- Установить параметру ssl.SocketFactory.provider значение com.digt.trusted.jsse.provider.DigtSocketFactory, изменив его или добавив строчку:

  ssl.SocketFactory.provider=com.digt.trusted.jsse.provider.DigtSocketFactory

Установка лицензии на ПО

Последним шагом по установке TrustedJava остается ввести лицензию.

Установка лицензии для Windows

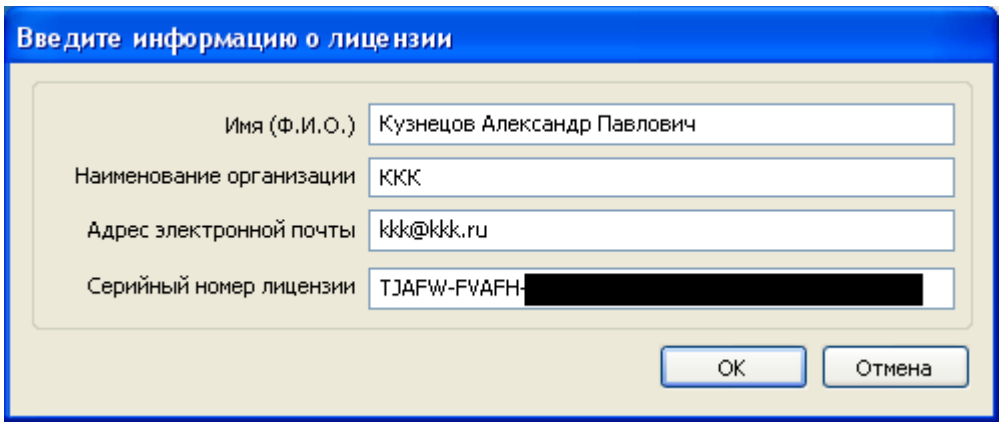
Процедура регистрации программного продукта различается в зависимости от операционной системы.

Чтобы зарегистрировать программный продукт Trusted Java в ОС Windows:

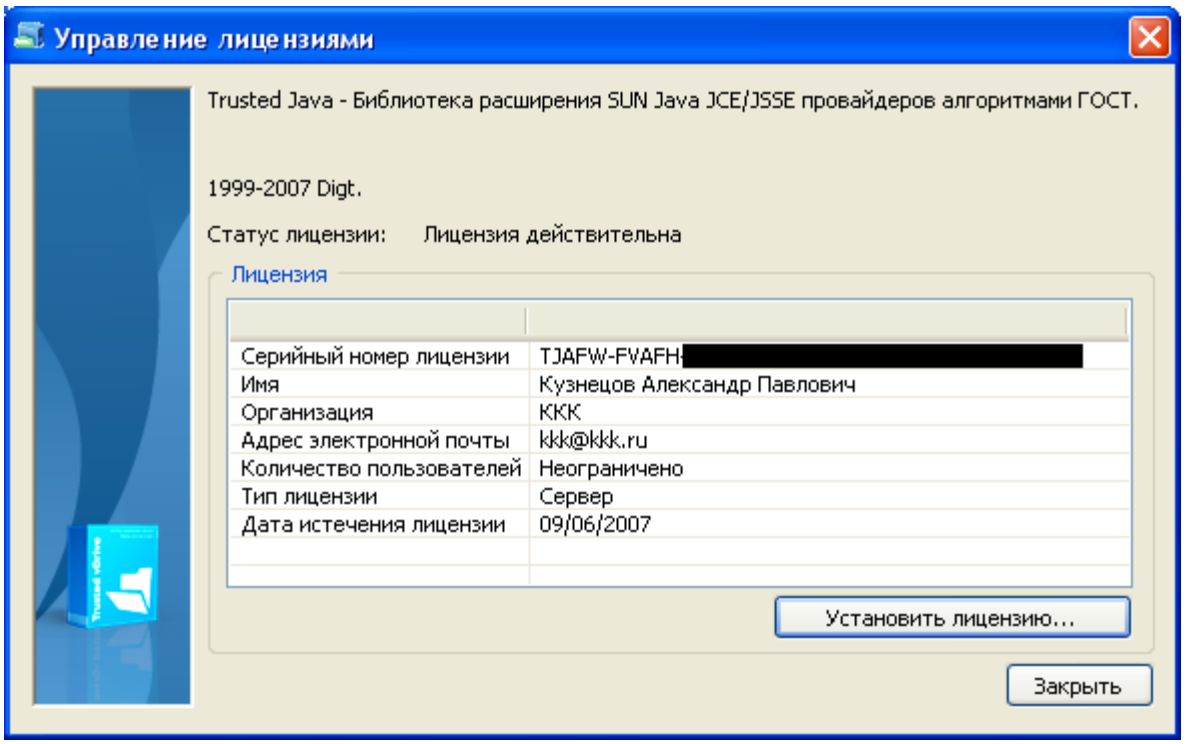
Откройте окно Управление лицензиями (Программы > Digt > Trusted Java 2.0 > Управление лицензиями).

В этом окне нажмите на кнопку «Установить лицензию…».

В открывшемся окне введите информацию о лицензии согласно выданным вам документам



При успешной регистрации ПО Trusted Java статус лицензии изменится на "Действительна" (см. в окне Управление лицензиями - > Статус лицензии):



В поле Лицензия отображается информация в соответствии с используемой лицензией на программный продукт Trusted Java:

- серийный номер лицензии
- имя администратора системы
- наименование организации, использующей данное ПО
- адрес электронной почты
- количество пользователей

- тип лицензии
- дата истечения срока действия лицензии на программный продукт

Установка лицензии для Linux

Чтобы зарегистрировать программный продукт Trusted Java в
в Unix-системах, необходимо выполнить следующие действия:

В дистрибутиве ПО существует файл license.lic, расположенный в каталоге /opt/DIGT/etc/Trusted/Java 2.0/license.lic. Откройте файл с помощью текстового редактора и впишите выданную вам лицензию в поле SerialNumber.

Проверка работоспособности ПО

Проверка работоспособности ПО под Linux

В каталоге установки /opt/DIGT/Trusted Java 2.0/tests находится архив тестов для проверки правильности установки и функционирования ПО.

Компиляцию и запуск тестов лучше провести под обычным пользователем, не root. Нужно распаковать архив

```
cd "/opt/DIGT/Trusted Java 2.0/tests"

unzip tests.zip
```

В файле build.sh нужно установить переменную JDK, значение которой должно указывать на каталог установки Java JDK, например,

```
JDK=/usr/java/jdk1.6.0_07
```

Запускаем процесс компиляции

```
./build.sh
```

Далее перед запуском тестов нужно поменять переменную JRE в файле run.sh, указав в ее значении путь до каталога установки Java JRE

```
JRE=/usr/java/jre1.6.0_17
```

Далее запустить тесты

```
./run.sh
```

Тестирование проходит достаточное количество времени и требует участия пользователя в виде нажатий на произвольные клавиши клавиатуры. Если тесты завершились примерно таким сообщением на консоли,

```
Test finished
Time: 406,109
OK (15 tests)

hKey pair is generated
Certificate is generated
X509Data element found
08.06.2010 17:07:43 org.apache.xml.security.signature.Reference verify

INFO: Verification successful for URI ""
The XML signature  is valid
```

то результат тестирования можно считать успешным.

Удаление ПО

В этой главе руководства рассматриваются вопросы удаления ПО Trusted Java.

Удаление ПО Trusted Java под Windows

Удалить ПО Trusted Java можно двумя способами:

- стандартными средствами ОС Windows
- через основное меню

Для удаления программы Trusted Java стандартными средствами ОС Windows:

Откройте Пуск - > Панель управления, выберите опцию Установка и удаление программ

В списке выберите запись Trusted Java 2.0 и нажмите на кнопку «Удалить» и подтвердите решение об удалении

Начнется процесс удаления ПО. По завершении процесса библиотека Trusted Java будет удалена с компьютера и из списка элементов Установленные программы

Для удаления программы Trusted Java с помощью основного меню выберите приложение «Удаление Trusted Java 2.0»:

Пуск > Программы > Digt > Trusted Java 2.0 > Удаление Trusted Java 2.0.

Удаление ПО Trusted Java под Linux

Процесс удаления производится с полномочиями пользователя **root**.

Нужно вернуть первоначальные настройки Java, записанные в файле $JRE/lib/security/java.security:

```
cp $JRE/lib/security/java.security.tj $JRE/lib/security/java.security
```

Если файл $JRE/lib/security/java.security.tj не существует, то в файле $JRE/lib/security/java.security нужно

- удалить строчку

  ```
  security.provider.N=com.digt.trusted.jce.provider.DIGTProvider
  ```

  где N – номер провайдера

- Установить параметру ssl.SocketFactory.provider требуемое (первоначальное) значение, изменив его или удалив строчку

  ```
  ssl.SocketFactory.provider=com.digt.trusted.jsse.provider.DigtSocketFactory
  ```

В каталоге $JRE/lib/ext требуется удалить символические ссылки на все jar-файлы в каталоге "/opt/DIGT/Trusted Java 2.0/jars":

```
rm -f $JRE/lib/ext/trusted_java20.jar

rm -f $JRE/lib/ext/commons-logging.jar

rm -f $JRE/lib/ext/junit.jar

rm -f $JRE/lib/ext/serializer.jar

rm -f $JRE/lib/ext/xalan.jar

rm -f $JRE/lib/ext/xmlsec.jar
```

В каталоге $JRE/lib/<**arch**> требуется удалить символическую ссылку на файл "/opt/DIGT/Trusted Java 2.0/lib/<**arch**>/libdjcp20.so":

```
rm -f $JRE/lib/<arch>/libdjcp20.so
```

Остается только удалить распакованный дистрибутив продукта:

```
rm -Rf "/opt/DIGT/Trusted Java 2.0"
```

и каталог с лицензией на продукт:

```
rm -Rf "/opt/DIGT/etc/Trusted/Java 2.0"
```

Если установка продукта производилась из rpm-пакета, то деинсталлировать продукт можно штатным образом, используя команду

```
rpm -e trustedjava
```

Настройка связки Trusted Java и Tomcat под Linux

Процесс установки и настройки описывается на примере Tomcat версий 5.5.28 и 6.0.20. Также проверялась совместимость с Tomcat 7.0.42. Номер версии, присутствующий в значениях каталогов, файлов, переменных окружения, можно успешно изменять, если не сказано иного. Отличия версии 6.0.20 описаны отдельно.

Требования к составу TLS-сертификатов

Для настройки SSL/TLS-шифрования для сервера потребуется X.509-сертификат, удовлетворяющий следующим требованиям:

- В имени владельца (т.е. в атрибуте «Субъект» / «Subject») сертификата элемент Common Name (CN) должен содержать конкретное доменное (DNS) имя сервера, на котором развертывается TLS-сервер, например, «webportal.yourcompany.ru», либо маску, например, «*.yourcompany.ru». Также, одно или несколько таких значений могут содержаться в расширении «Дополнительное имя субъекта» / «Subject Alternative Name».
- 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».

Если необходимо, чтобы пользователи тоже аутентифицировались по сертификатам, то их сертификаты должны соответствовать следующим критериям:

- В расширении «Улучшенный ключ» / «Enhanced Key Usage» (EKU) должен присутствовать объектный идентификатор 1.3.6.1.5.5.7.3.2, обозначающий вариант использования сертификата «Проверка подлинности клиента» / «Client Authentication».
- Поддерживаются сертификаты без шифрования, то есть у которых в расширении «Использование ключа» / «Key Usage» (KU) указаны только «Цифровая подпись, Неотрекаемость» / «Digital Signature, Non-Repudiation».

Установка Tomcat под Linux

С правами администратора

- создайте пользователя и одноименную группу tomcat, которым будет позволено запускать приложение Tomcat

  ```
  groupadd tomcat
  useradd -g tomcat tomcat
  ```

- задайте пароль пользователю tomcat

  ```
  passwd tomcat
  ```

- распакуйте дистрибутив Tomcat в каталог /opt, например

  ```
  tar -xf ./apache-tomcat-5.5.28.tar.gz -C /opt
  ```

- присвойте соответствующие права на каталог с веб-сервером для полного доступа к его файлам настроек и логам пользователю tomcat, например:

  ```
  chown -R tomcat:tomcat /opt/apache-tomcat-5.5.28
  ```

Запуск и остановка Tomcat под Linux

Войдите в систему пользователем tomcat, задайте переменные окружения CATALINA_HOME и JAVA_HOME, которые соответственно указывают на каталог установки Tomcat и JRE, например:

```
export CATALINA_HOME=/opt/apache-tomcat-5.5.28

export JAVA_HOME=/usr/java/jre1.6.0_17
```

Теперь запустите Tomcat:

```
$CATALINA_HOME/bin/startup.sh
```

Для остановки Tomcat выполните:

```
$CATALINA_HOME/bin/shutdown.sh
```

Настройка SSL в Tomcat под Linux

Настройка RSA шифрования в Tomcat под Linux

В данном разделе приводятся выдержки из документации на Tomcat со страниц http://tomcat.apache.org/tomcat-5.5-doc/ssl-howto.html и http://tomcat.apache.org/tomcat-6.0-doc/ssi-howto.html.

Под пользователем tomcat создайте новое хранилище ключей (certificate keystore), содержащее один самоподписанный сертификат, используя команду:

```
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA
```

и укажите пароль для доступа к ключу, желательно, "changeit". Дополнительно потребуется указать данные для сертификата – компанию, контактное лицо и т.д. При выполнении этой команды создается хранилище ключей в файле .keystore домашнего каталога пользователя.

Поддержка SSL в Tomcat реализуется двумя способами:

- JSSE реализация, предоставляемая частью Java Runtime (начиная с версии 1.4)
- APR реализация, которая включена по умолчанию в версии 5.5.28 и основана на OpenSSL

Дальнейшие действия по настройке приводятся для JSSE-реализации.

Для указания использования JSSE-реализации SSL, нужно раскомментировать узел с портом 8443 и схемой https в файле $CATALINA_HOME/conf/server.xml.

Для версии 5.5.28 раскомментировать строки:

```
<Connector port="8443" maxHttpHeaderSize="8192"
        maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
        enableLookups="false" disableUploadTimeout="true"
        acceptCount="100" scheme="https" secure="true"
        clientAuth="false" sslProtocol="TLS" />
```

и изменить первую строку на:

<Connector **protocol="org.apache.coyote.http11.Http11Protocol"** port="8443" maxHttpHeaderSize="8192"

Для версии 6.0.20 достаточно только раскомментировать строки:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
        maxThreads="150" scheme="https" secure="true"
        clientAuth="false" sslProtocol="TLS" />
```

Опция «protocol» здесь уже установлена.

После перезапуска Tomcat должна быть доступна страница

https://servername:8443/

Настройка ГОСТ-шифрования в Tomcat под Linux

НАСТРОЙКИ КОННЕКТОРА TOMCAT

Для использования защищенного соединения по протоколу GOST TLS нужно указать тип протокола “GostTLS” в описании коннектора Tomcat.

Для этого нужно в файле $CATALINA_HOME/conf/server.xml в узле с портом 8443 и схемой https заменить sslProtocol="TLS" на sslProtocol="GostTLS".

Пример для версии 5.5.28:

<Connector **protocol="org.apache.coyote.http11.Http11Protocol"** port="8443" maxHttpHeaderSize="8192"

maxThreads="150" minSpareThreads="25" maxSpareThreads="75"

enableLookups="false" disableUploadTimeout="true"

acceptCount="100" scheme="https" secure="true"

clientAuth="false" sslProtocol="GostTLS" />

Пример для версии 6.0.20:

<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"

maxThreads="150" scheme="https" secure="true"

clientAuth="false" **sslProtocol="GostTLS"** />

Убедитесь, что при установке TrustedJava в файле $JRE/lib/security/java.security параметру ssl.SocketFactory.provider было присвоено значение com.digt.trusted.jsse.provider.DigtSocketFactory.

Установка серверного сертификата под Linux

Чтобы стартовать GOST TLS на ГОСТ сертификатах, необходимо на стороне сервера иметь ключевой контейнер и соответствующий ему сертификат в файле в формате "Base-64 encoded X.509":

- Сертификат должен иметь привязку к секретному ключу пользователя (т.е. установлен в хранилище "my" пользователя tomcat).
- Сертификаты УЦ, его подписавшие, должны находиться в хранилище "root" Unix платформы

Для выполнения этих требований необходимо выполнить следующие шаги:

1. Генерация запроса на сертификат (ГОСТ 34.10-2001) для веб-сервера
2. Получение сертификата УЦ и веб-сервера
3. Установка сертификата УЦ
4. Установка сертификата веб-сервера
5. Установка системных переменных окружения Java

Генерация запроса на сертификат (ГОСТ 34.10-2001)

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 генерации запроса на сертификат необходимо выполнить под учетной записью пользователя tomcat (от имени которого запускается сервер) следующую 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"
~/tlssreq.csr
```

Параметры -cont и -dn нужно изменить в соответствии с вашими данными. Параметр -cont должен быть уникальным для каждого нового запроса, в примере он создаст ключевой контейнер в хранилище CSP на файловой системе, если вам нужно сгенерировать ключ на токене или дискете, измените его согласно руководству по 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>
```

Получение сертификатов УЦ и веб-сервера

Сгенерированный файл запроса ~/tlssreq.csr следует обработать в Удостоверяющем центре. При тестировании можно использовать тестовый УЦ компании КриптоПро: http://www.cryptopro.ru/certsrv/certrqxt.asp. На данной странице в поле «Base-64-шифрованный запрос сертификата (CMC или PKCS #10 или PKCS #7)» нужно вставить содержимое файла запроса и продолжить работу, нажав на кнопку «Выдать >». Затем следует подтвердить передачу данных по незашифрованному каналу, в следующем окне обязательно выбрать «**Base64-шифрование**» и выполнить пункт «Загрузить

сертификат». Полученный в УЦ сертификат веб-сервера сохраните на сервере в файл ~/servergost.cer.

Для получения сертификата УЦ нужно зайти на страницу http://www.cryptopro.ru/certsrv, выбрать пункт «Получить сертификат Удостоверяющего Центра или действующий список отозванных сертификатов» и продолжить работу, нажав на кнопку «Дальше >> >». В следующем окне следует выбрать «Base64-шифрование» и выполнить пункт «Загрузка сертификата ЦС». Сохраните сертификат УЦ на сервере в файл ~/caservergost.cer.

Установка сертификата УЦ

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 ~tomcat/caservergost.cer -cert
```

Если команда установки сертификата выполнилась с кодом ошибки 0x00000000, значит, сертификат УЦ успешно установлен.

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 ~tomcat/intermediategost.cer -cert
```

Установка сертификата веб-сервера

Для установки в хранилище my пользователя tomcat сертификата веб-сервера, полученного на предыдущем этапе (**обязательно в формате "Base64"**), необходимо от его имени выполнить команду

```
/opt/cprocsp/bin/<arch>/cryptcp -instcert -provtype 75 ~/servergost.cer
```

и в появившемся пронумерованном списке контейнеров указать номер соответствующего контейнера и пароль 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 ~/servergost.cer -cont "\\\\.\\HDIMAGE\\tlsserver"

Если одна из предыдущих команд установки сертификата выполнилась с кодом ошибки 0x00000000 или сообщением «Certificate is installed.», значит, серверный сертификат установлен в личное хранилище текущего пользователя со ссылкой на закрытый ключ.

Проверить наличие установленного серверного сертификата в личном ("my") хранилище сертификатов пользователя tomcat можно, выполнив следующую команду:

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

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

PrivateKey Link: Yes. Container: <имя_контейнера>

Установка системных переменных окружения Java

Скопируйте сертификат веб-сервера в каталог /opt/apache-tomcat-5.5.28/conf:

cp ~/servergost.cer /opt/apache-tomcat-5.5.28/conf

Для указания серверного сертификата и пароля к ключу соответствующего контейнера приватного ключа используются следующие параметры системного окружения Java:

com.digt.trusted.jsse.server.certFile

com.digt.trusted.jsse.server.keyPasswd

В связи с этим нужно добавить в начале файла $CATALINA_HOME/bin/startup.sh следующую строку

export JAVA_OPTS="$JAVA_OPTS -Dcom.digt.trusted.jsse.server.certFile=/opt/apache-tomcat-5.5.28/conf/servergost.cer -Dcom.digt.trusted.jsse.server.keyPasswd=*******"

Замените «*» в пароле на конкретное его значение.

Java Bridge: Интеграция Trusted Java и PHP

Настройка доступа к Trusted Java из PHP под APACHE

Вопрос установки http-сервера APACHE, PHP и модуля PHP для APACHE здесь не рассматривается. Установка Trusted Java и JRE описана выше. Далее описывается процесс настройки доступа из модуля PHP для APACHE к Java и, как следствие, к Trusted Java.

Сначала требуется получить файлы со страницы PHP/Java Bridge (требующие Java 1.6 и выше) и разместить их в каталоге, например, /opt/JavaBridge:

- /opt/JavaBridge/java/Java.inc
- /opt/JavaBridge/ext/JavaBridge.jar
- /opt/JavaBridge/ext/php-script.jar
- /opt/JavaBridge/ext/php-servlet.jar

Создайте в каталоге /opt/JavaBridge файл PHPJavaBridge_start.sh:

```
#!/bin/sh
java -Dphp.java.bridge.daemon="true" -jar /opt/JavaBridge/ext/JavaBridge.jar SERVLET_LOCAL:8080 3 JavaBridge.log
```

Перед его запуском в режиме «демона» необходимо убедиться, что при запуске java запускается установленная ранее версия JRE. Настроить использование требуемой JRE с использованием alternatives возможно, например, следующим образом:

```
alternatives --install /usr/bin/java java /usr/java/jre1.6.0_17/bin/java 10000
```

Теперь можно выполнить файл /opt/JavaBridge/ PHPJavaBridge_start.sh.

Откройте www-каталог http-сервера, например, /var/www/html и создайте в нем символическую ссылку JavaBridge на каталог /opt/JavaBridge. Положите в него тестовый файл SignAndVerify.php из архива с тестами, поставляемыми в составе дистрибутива Trusted Java. Запустите http-сервер. Проверить работу связки Trusted Java + PHP Apache + PHP Java Bridge можно обратившись из Internet Explorer по ссылке http://servername/SignAndVerify.php. В данном примере реализовано формирование ЭЦП из браузера с использованием ПО КриптоАРМ, а ее проверка на стороне сервера – с использованием Trusted Java.

Настройка доступа к Trusted Java из PHP под Tomcat

Требуется получить файл JavaBridge.war, который появляется после установки rpm-пакета, для дальнейшего его включения в сервер Tomcat6. Для создания rpm-пакета PHP Java Bridge необходимо наличие следующих компонент:

- Java JDK версии 1.4.2 или выше (рекомендуется версия 6)
- PhpDocumentor версии 1.4.2 или выше
- Ant версии 1.7.1 или выше
- Apache HTTP Server
- При использовании SELinux:
  selinux-policy – конфигуратор политик,
  selinux-policy-devel – пакет разработки,
  policycoreutils – утилиты для работы с политиками,
  checkpolicy – компилятор политик,
  coreutils – набор основных утилит GNU.

Для установки rpm-пакета PHP Java Bridge потребуются следующие компоненты:

- PHP версии 5.1.2 или выше (рекомендуется версия 5.3 или выше)
- Tomcat версии 6.x

Далее потребуется настроить установку пакетов, например, через jpackage-util (Jpackage Project), подключив, например, к yum репозитарий jpackage50.repo. Таким образом можно будет установить Ant, Tomcat6.

Rpm-пакет PhpDocumentor можно получить, например, с сайта RPM pbone.net. Там же могут быть получены связанные с ним пакеты.

Скачайте архив php-java-bridge_6.1.2.1.tar.gz, разместите в каталоге пользователя root, например, /root/soft.

Для сборки rpm-пакета выполните команду

```
rpmbuild -tb php-java-bridge_6.1.2.1.tar.gz
```

Если появляется сообщение

ошибка: синтаксическая ошибка в выражении
ошибка: /usr/src/redhat/SPECS/**php-java-bridge.spec**:**56**: parseExpressionBoolean код возврата: -1
ошибка: Пакет не имеет %description: php-java-bridge

то создайте и исполните файл /root/soft/deploy.sh:

```
#!/bin/sh
cd /root/soft
tar -zxf php-java-bridge_6.1.2.1.tar.gz *.spec
mv php-java-bridge-6.1.2.1/php-java-bridge.spec .
rm -R php-java-bridge-6.1.2.1
cp php-java-bridge_6.1.2.1.tar.gz /usr/src/redhat/SOURCES/
```

Далее в полученном файле /root/soft/php-java-bridge.spec закомментируйте строки 56-57:

```
…
#%if (%{PHP_MINOR_VERSION} == 2 && %{PHP_RELEASE_VERSION} > 6) || (%{PHP_MINOR_VERSION} > 2)

#%endif
…
```

и выполните команду:

```
rpmbuild -bb /root/soft/php-java-bridge.spec
```

Полученные rpm-файлы можно забрать из каталога /usr/src/redhat/RPMS/i386.

Установите полученный пакет PHP Java Bridge, выполнив команду

```
rpm -i /usr/src/redhat/RPMS/i386/php-java-bridge-6.1.2.1-1.i386.rpm
```

В приведенной команде может использоваться опция --nodeps для случая, когда Tomcat6 был установлен, но не через механизм rpm-пакетов. В этом случае требуется вручную довести установку PHP Java Bridge для связки с Tomcat6. Скопируйте файл JavaBridge.war в подкаталог webapps каталога установки Tomcat6:

```
cp /var/lib/tomcat6/webapps/JavaBridge.war /opt/apache-tomcat-6.0.20/webapps
```

Перезапустите Tomcat6:

```
service tomcat6 restart
```

Теперь должна быть доступна страница с тестами

http://servername:8080/JavaBridge.

Сервера приложений: интеграция Trusted Java и TrustedTLS

Введение

В предлагаемом ниже материале рассматриваются методики построения защищенного канала до серверов приложений (Tomcat, IBM Websphere, Oracle Weblogic) на базе продукта TrustedTLS (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 + Trusted TLS 2.2 + CSP 3.6/3.6 R2 + Tomcat 5.5/6.0/7.0 + Trusted Java
- Apache + Trusted TLS 2.2 + CSP 3.6/3.6 R2 + IBM Websphere 6.1/7.0 + Trusted Java
- Apache + Trusted TLS 2.2 + CSP 3.6/3.6 R2 + 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.

Предполагается, что на Apache-сервере уже настроен модуль mod_digt_tls.so (продукт TrustedTLS) 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"
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 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 поцесс 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**.

Часто задаваемые вопросы

- при встраивании Java-библиотеки:

Описание ошибки	Рекомендации по устранению
Java сообщает, что не найден провайдер DIGT	Добавьте в начало кода строку: Security.addProvider(new DIGTProvider())
Ошибка: java.lang.ClassNotFoundException: com.digt.trusted.jce.provider.DIGTProvider	Поместите trusted_java20.jar в каталог <jre>/lib/ext.
Ошибка: java.lang.UnsatisfiedLinkError: no djcpNNN in java.library.path	Java не может найти библиотеку. Поместите ее по одному из путей переменных окружения %PATH% либо укажите путь к ней с помощью параметра java машины -Djava.library.path.
Ошибка java.lang.UnsatisfiedLinkError: libdjcp20.so: undefined symbol: CertFreeCertificateContext	Перед запуском JVM добавьте в переменную окружения LD_PRELOAD путь к библиотеке libcapi20.so.3 из КриптоПро CSP, например: export LD_PRELOAD="/opt/cprocsp/lib/<**arch**>/libcapi20.so.3.0.0:$LD_PRELOAD"

- при использовании Trusted Java с Tomcat:

Описание ошибки	Рекомендации по устранению
При запуске Tomcat в его лог-файле catalina.out присутствуют строки SEVERE: Failed to load keystore type JKS with path C:\Documents and Settings\<username>/.keystore due	Сформируйте keystore с RSA-сертификатом в соответствии с описанием в разделе «Настройка RSA шифрования в Tomcat под Linux».

to C:\Documents and Settings\<username>\.keystore (The system cannot find the file specified) java.io.FileNotFoundException: C:\Documents and Settings\<username>\.keystore (The system cannot find the file specified) или (для локализованной ОС): SEVERE: Failed to load keystore type JKS with path C:\Users\<username>/.keystore due to C:\Users\<username>\.keystore (Не удается найти указанный файл) java.io.FileNotFoundException: C:\Users\<username>\.keystore (Не удается найти указанный файл)	
При запуске Tomcat в его лог-файле catalina.out присутствует строка java.io.IOException: GostTLS SSLContext not available	Пропишите DIGTProvider в java.security, т.е. добавьте строку вида: security.provider.NN=com.digt.trusted.jce.provider.DIGTProvider Также убедитесь, что в каталоге <jre>/lib/ext присутствует файл trusted_java20.jar.
Не открывается страница https://servername:8443 и в лог-файле сервера catalina.out присутствуют строки INFO: Initializing Coyote HTTP/1.1 on http-8080 11.06.2010 12:38:43 com.digt.trusted.jsse.provider.DigtSSLContext getDefaultContext SEVERE: **Error setting certificate**. 11.06.2010 12:38:43 org.apache.coyote.http11.Http11BaseProtocol init SEVERE: Error initializing endpoint	Проверьте правильность указания пути к файлу сертификата веб-сервера в переменной окружения JAVA_OPTS. **Внимание!** При объявлении JAVA_OPTS под Windows параметры certFile и keyPasswd нужно обрамлять кавычками по отдельности. Если сертификат сохранен в кодировке Base64, то удостоверьтесь в наличии Base64-заголовков (-----BEGIN

	CERTIFICATE----- и -----END CERTIFICATE-----). Если сертификат сервера квалифицированный и установленная сборка Trusted Java ниже 490, то обновите Trusted Java.
Не открывается страница https://servername:8443 и в лог-файле сервера catalina.out присутствуют строки INFO: Server startup in 633 ms Password:Password:11.06.2010 4:29:33 org.apache.tomcat.util.net.PoolTcpEndpoint acceptSocket SEVERE: Endpoint ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=8443] ignored exception: java.io.IOException: error:1408B06B:SSL routines:**SSL3_GET_CLIENT_KEY_EXCHANGE:bad decompression**	Проверьте правильность указания пароля к ключевому контейнеру в переменной окружения JAVA_OPTS, а также наличие обрамляющий кавычек «…» при ее описании.
Не открывается страница https://servername:8443 и в лог-файле сервера catalina.out присутствуют строки INFO: Server startup in 564 ms 11.06.2010 12:32:51 org.apache.tomcat.util.net.PoolTcpEndpoint acceptSocket SEVERE: Endpoint ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=8443] ignored exception: java.io.IOException: error:1409A044:SSL routines:**SSL3_SEND_SERVER_CERTIFICATE:internal error**	Проверьте правильность указания пути к файлу сертификата веб-сервера в переменной окружения JAVA_OPTS, а также наличие обрамляющих кавычек «…» при ее описании.
Не открывается страница https://servername:8443 и в лог-файле	В Trusted Java не поддерживаются NIO-сокеты.

сервера catalina.out присутствуют строки java.lang.Exception: Connector attribute SSLCertificateFile must be defined when using SSL with APR или org.apache.tomcat.util.net.**NioEndpoint setSocketOptions** SEVERE: java.lang.NullPointerException	Замените в конфигурации значение параметра protocol с «HTTP/1.1» или «org.apache.coyote.http11.Http11NioProtocol» на «org.apache.coyote.http11.Http11Protocol».

- не нашли описания своей проблемы?

Проверьте, не истек ли период использования Крипто Про CSP. В противном случае обратитесь в техническую поддержку (см. контакты ниже) и опишите сложившуюся ситуацию с примерами появившихся ошибок.

Техническая поддержка

По вопросам технической поддержки ПО Trusted Java обращайтесь:
- по электронной почте: support@trusted.ru
- по телефону: 8 (8362) 33-70-50
- по адресу: 424033, Россия, РМЭ, г. Йошкар-Ола, ул. Петрова, д.1, а/я 67.

Систематическое техническое сопровождение организаций по вопросам работы с ПО Trusted Java осуществляется при условии оплаты стоимости годового пакета технических услуг. В течение этого срока разработчик бесплатно поставляет новые реализации продукта в рамках приобретенной версии. Новые версии продукта поставляются в соответствии с прайс-листом.

О компании-разработчике

Компания «Цифровые технологии» – российский разработчик и поставщик программного обеспечения в области защиты информации, систем электронного документооборота и хранения данных. Основной сферой деятельности компании является разработка, внедрение и поддержка криптографических продуктов и решений для государственных и коммерческих структур.

Телефон: (8362) 33-70-50
Сайт: http://www.trusted.ru
E-mail: info@trusted.ru