**ИНТЕГРАЦИЯ Jetty 9.3.1 и JTLS 2.0**

**Предустановленные компоненты:**

1. *OS*: Windows 7 (x64)
2. *Jetty*: jetty-distribution-9.3.1.v20150714 (<http://download.eclipse.org/jetty/>) для ***localhost***
3. *Java*: JDK 1.8.0\_51
4. *Провайдеры*: JCP и JTLS версии 2.0, установленные в JDK
5. *Сняты ограничения для JCP*: <https://support.cryptopro.ru/index.php?/Knowledgebase/Article/View/44/6/snjatie-ehksportnykh-ogrnichenijj>
6. *Приложение*: тестовое приложение *TestService.war* в папке *Doc\WebServerIntegration\example\TestService\build* данного дистрибутива.
7. *Ключи сервера*: хранилище сертификатов *keystore* формата *CertStore* и ключевой контейнер *jetty* с сертификатом для аутентификации сервера формата *HDImageStore*. Пароли хранилища и серверного контейнера совпадают.
8. *Клиент (браузер) для проверки подключения к серверу*: браузер IE с установленным CSP 4.0. При требовании сервером jetty клиентского сертификата в реестр системы предварительно должен быть установлен действительный тестовый *клиентский* сертификат с привязкой к ключу и доступный пользователю браузера.
9. *Дополнительная документация*: <http://www.eclipse.org/jetty/documentation/current/>
10. Код фабрики контекста *jetty*: <http://grepcode.com/file/repo1.maven.org/maven2/org.eclipse.jetty/jetty-util/9.2.2.v20140723/org/eclipse/jetty/util/ssl/SslContextFactory.java?av=f>
11. Параметры настройки коннектора *jetty*: <https://wiki.eclipse.org/Jetty/Howto/Configure_SSL#Configuring_Jetty>
12. **Общая настройка сервера**

Скопировать тестовое приложение *TestService.war* в папку *webapps* сервера. При запуске сервера оно будет доступно автоматически. Настроить логирование в отладочных целях - создать файл *jetty-logging.properties* в папке *resources* сервера и добавить в него строки:

*# Configure Jetty for StdErrLog Logging*

*org.eclipse.jetty.util.log.class=org.eclipse.jetty.util.log.StrErrLog*

*# Overall Logging Level is INFO*

*org.eclipse.jetty.LEVEL=****DEBUG***

*# Detail Logging for WebSocket*

*org.eclipse.jetty.websocket.LEVEL=****DEBUG***

Таким образом, будет настроен детальный лог (см. <http://www.eclipse.org/jetty/documentation/current/default-logging-with-stderrlog.html>).

В файл конфигурации сервера *start.ini* в папке сервера добавить после *--module=http* строки:

*# Module: https*

*--module=https*

*# Module: logging*

*--module=logging*

Первый модуль *https* - для включения *https*, второй *logging* – для логирования. Логи будут записываться в папку *logs* сервера. Таким образом, участок *start.ini* будет выглядеть так:

*# ---------------------------------------*

*# Module: http*

*--module=http*

*# Module: https*

*--module=https*

*# Module: logging*

*--module=logging*

Далее защищенное подключение будет осуществляться по порту 8443, обычное – по порту 8080.

1. **Настройка ГОСТ TLS**

Перейти в папку *etc* сервера, открыть для редактирования файл *jetty-ssl-context.xml*. Настроить его примерно так (все его содержимое):

*<?xml version="1.0"?>*

*<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://www.eclipse.org/jetty/configure\_9\_3.dtd">*

*<!-- ============================================================= -->*

*<!-- SSL ContextFactory configuration -->*

*<!-- ============================================================= -->*

*<Configure id="sslContextFactory" class="org.eclipse.jetty.util.ssl.SslContextFactory">*

*<Set name="****KeyStorePath****"><Property name="jetty.base" default="." />/<Property name="jetty.sslContext.****keyStorePath****" deprecated="jetty.keystore" default="****etc/keystore****"/></Set>*

*<Set name="****KeyStorePassword****"><Property name="jetty.sslContext.****keyStorePassword****" deprecated="jetty.keystore.password" default="****tomcat7****"/></Set>*

*<Set name="****KeyStoreType****"><Property name="jetty.sslContext.****keyStoreType****" default="****HDImageStore****"/></Set>*

*<Set name="****KeyStoreProvider****"><Property name="jetty.sslContext.****keyStoreProvider****" default="****JCP****"/></Set>*

*<Set name="****KeyManagerPassword****"><Property name="jetty.sslContext.****keyManagerPassword****" deprecated="jetty.keymanager.password" default="****tomcat7****"/></Set>*

*<Set name="****TrustStorePath****"><Property name="jetty.base" default="." />/<Property name="jetty.sslContext.****trustStorePath****" deprecated="jetty.truststore" default="****etc/keystore****"/></Set>*

*<Set name="****TrustStorePassword****"><Property name="jetty.sslContext.****trustStorePassword****" deprecated="jetty.truststore.password" default="****tomcat7****"/></Set>*

*<Set name="****TrustStoreType****"><Property name="jetty.sslContext.****trustStoreType****" default="****CertStore****"/></Set>*

*<Set name="****TrustStoreProvider****"><Property name="jetty.sslContext.****trustStoreProvider****" default="****JCP****"/></Set>*

*<Set name="EndpointIdentificationAlgorithm"></Set>*

*<Set name="****NeedClientAuth****"><Property name="jetty.sslContext.needClientAuth" deprecated="jetty.ssl.needClientAuth" default="****true****"/></Set>*

*<Set name="WantClientAuth"><Property name="jetty.sslContext.wantClientAuth" deprecated="jetty.ssl.wantClientAuth" default="false"/></Set>*

*<Set name="****Protocol****">****GostTLS****</Set>*

*<Set name="****Provider****">****JTLS****</Set>*

*<Set name="****IncludeProtocols****">*

*<Array type="java.lang.String">*

*<Item>****TLSv1****</Item>*

*</Array>*

*</Set>*

*<Set name="****IncludeCipherSuites****">*

*<Array type="String">*

*<Item>****TLS\_CIPHER\_2001****</Item>*

*<Item>****TLS\_CIPHER\_2012****</Item>*

*</Array>*

*</Set>*

*<Set name="useCipherSuitesOrder"><Property name="jetty.sslContext.useCipherSuitesOrder" default="true"/></Set>*

*</Configure>*

**Жирным** выделены основные настройки, смысл которых понятен из названия (по аналогии с *tomcat*). *keystore* – это хранилище доверенных корневых сертификатов формата *CertStore*. Его рекомендуется поместить в папку *etc* сервера. Пароли хранилища и серверного контейнера совпадают.

Как в случае настройки сервера *tomcat*, доверенное хранилище указывается дважды. Параметр *NeedClientAuth* означает требование клиентского сертификата. Сейчас он установлен *true*. Необходимо указать протокол (*GostTLS*) и провайдер (*JTLS*) для соединения, а также поддерживаемые модулем cpSSL сайфер-сюиты (*TLS\_CIPHER\_\**) и версию *TLS* протокола (*TLSv1*).

Помимо этого, в папку пользователя (*C:/Users/<user>/AppData/Crypto Pro*) сервера *jetty* следует поместить сам ключевой контейнер, который и будет использоваться сервером. Ключевой контейнер должен быть доступен пользователю, под управлением которого запускается сервер. Сертификат в контейнере должен быть действителен и содержать OID политики использования для аутентификации сервера (расширенное использование ключа).

1. **Проверка подключения**

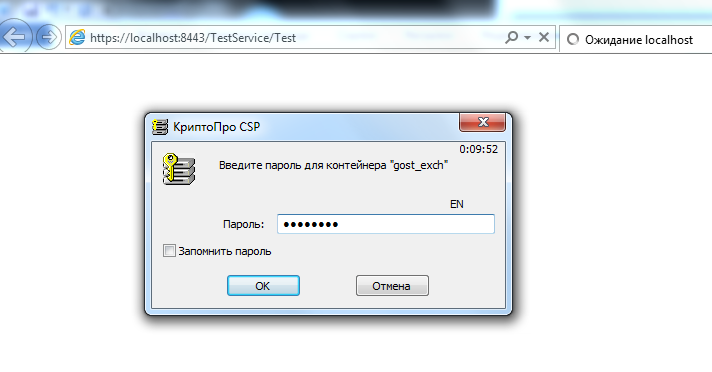
Ранее была произведена настройка с требованием сертификата клиента, поэтому после запуска сервера (в одну строку):

*C:\software\jetty-distribution-9.3.1.v20150714>"C:\Program Files\Java\jdk1.8.0\_51\bin\java"* ***-Dcom.sun.security.enableCRLDP=true*** *-jar start.jar*

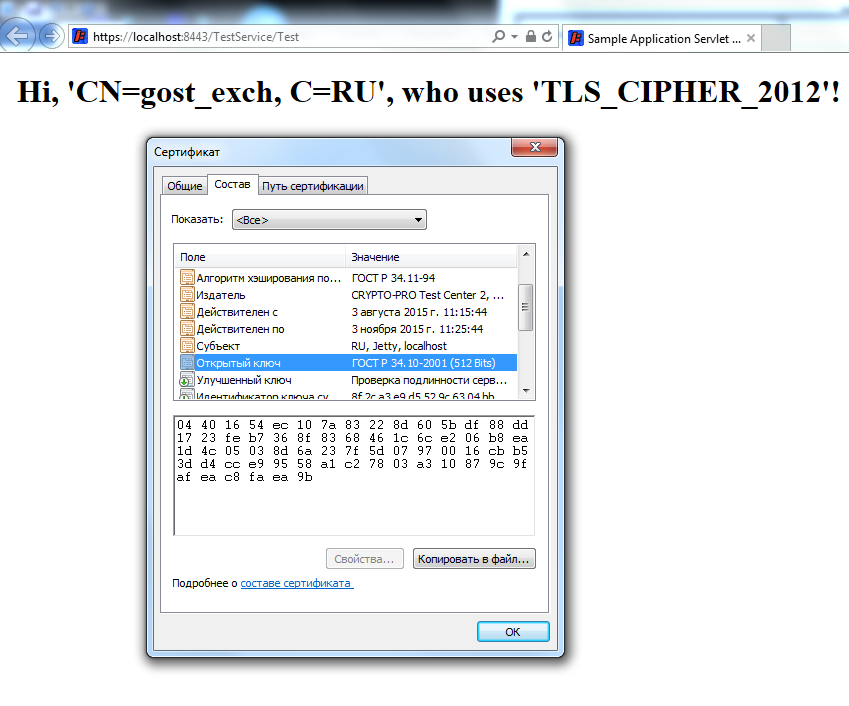
и перехода в IE по адресу:

[*https://localhost:8443/TestService/Test*](https://localhost:8443/TestService/Test)

будет предложено выбрать сертификат пользователя, и после ввода пароля к контейнеру пользователя:



откроется страница установленного ранее тестового сервиса (из списка сайфер-сюит сервером была выбрана *TLS\_CIPHER\_2012*):



Чтобы *IE* не сообщал об ошибке перехода на не доверенный сайт, CN имя в сертификате соответствует имени хоста, т.е. *localhost*.

Если список сайфер-сюит *IncludeCipherSuites* в конфиге *jetty-ssl-context.xml* сократить до одной записи - *TLS\_CIPHER\_2001*, то выбрана будет эта единственная сайфер-сюита.

Следует обратить особое внимание, что запуск сервера выполняется с указанием параметра *com.sun.security.enableCRLDP=true*. Это сделано по той причине, что в *cpSSL* по умолчанию включена проверка цепочки сертификатов клиента по *CRLDP*. Без этого параметра в логах будет ошибка:

*GeneralSecurityException: [PKIX] Online verification of certificate status is enabled but com.sun.security.enableCRLDP=false, com.ibm.security.enableCRLDP=false*

Если нужно отключить проверку цепочки, то следует запустить панель *JCP*, перейти на закладку *«Настройки сервера» («Server settings»)* и убрать галочку *«Проверка статуса…».*

После того, как сервер был успешно настроен, детальное логирование можно отключить (п. 1).

**Справочная информация по коннектору**:

*Other properties that you can set for the SslContextFactory include:*

*certAlias–Alias of a certificate to use.*

*keyStoreType–Default value: "JKS."*

*keyStoreProvider–Default is the SunJSSE provider.*

*trustStoreType–Default value: "JKS".*

*trustStoreProvider–Default is the SunJSSE provider.*

*sslKeyManagerFactoryAlgorithm–Set to the value of the "ssl.KeyManagerFactory.algorithm" system property. If there is no such property, the default is "SunX509."*

*sslTrustManagerFactoryAlgorithm–set to the value of the "ssl.TrustManagerFactory.algorithm" system property. If there is no such property, the default is "SunX509."*

*secureRandomAlgorithm–Default value is null.*

*protocol–Default value is "TLS."*

*provider–Default is the first provider that supports that protocol.*

*includeCipherSuites–See How to configure SSL Cipher Suites.*

*excludeCipherSuites–See How to configure SSL Cipher Suites.*

*needClientAuth–Default is false*

*wantClientAuth–Defaults is false.*

*validateCerts–Default is false.*

*allowRenegotiate–Default is false.*

*crlPath–Path to certificate revocation list file for SSL certificate validation.*

*maxCertPathLengh–Maximum allowed number of intermediate certificates, default is -1 (unlimited).*

If there is no value for the truststore, the system uses the keystore value. You can obfuscate passwords by using the Jetty password utility (<https://wiki.eclipse.org/Jetty/Howto/Secure_Passwords>).