**Краткая инструкция:**

**ИНТЕГРАЦИЯ JTLS В WAS 8.5.5.3 И НАСТРОЙКА ГОСТ TLS**

**Полная инструкция со скриншотами: Howto\_set\_gost\_tls\_WAS**

**Предполагается распространение вместе с архивом со всеми необходимыми файлами образцов и т.п.**

Поставленная **задача** - создать новую SSL конфигурацию, которая будет использовать ГОСТ TLS (сайферсюиту *TLS\_CIPHER\_2001*) на порту 9444.

***Предположим, что в виртуальной среде уже:***

1. установлена ОС RHEL 6.5 x64;
2. установлен *WAS for developers 8.5.5.3* без административной защиты, создан профайл AppSrv01, сервер server1 (порты: 9060 и 9043 - для Административной консоли, а 9080 и 9443 - для приложений);
3. установлен Крипто-Про CSP 3.6 R4 или 4.0 (в этой системе или в другой, откуда есть доступ к виртуалке – тогда, возможно, потребуется изменить настройки файрволла системы);
4. скачан последний дистрибутив JCP 2.0;
5. скачан архив с библиотеками для снятия экспортных ограничений IBM JVM (см. <https://support.cryptopro.ru/index.php?/Knowledgebase/Article/View/44/6/snjatie-ehksportnykh-ogrnichenijj>);
6. имеются файлы-образцы настройки *security\_double.xml* и *ssl.client.props* (находятся в папке *new\_confugrations/<system>* для соответствующей системы);
7. имеется тестовый сервлет TestService.war/Test для проверки соединения (находится в папке *examples/TestService/build*).

С помощью csptest в составе Крипто-Про CSP или IE (в ОС Windows) и сервлета далее будет производиться проверка https-соединения, а файлы-образцы помогут настроить ГОСТовую SSL конфигурацию WAS для нового порта 9444.

Весь процесс будет разделен на несколько частей. Предполагается, что установка всех необходимых компонентов производится под управлением учетной записи администратора.

1. ***Короткая проверка корректности установки*:**
2. запустить Административную консоль (<http://localhost:9060/ibm/console/login.do>) под учетной записью администратора;
3. перейти в «*Applications > New Application > Enterprise Application»* и установить приложение TestService.war, на последнем этапе задав его *context-root* как */test\_service*;
4. запустить пример TestService (*start up*) в списке установленных приложений. Проверить значение его *context-root* можно в «*Enterprise Applications > TestService\_war > Deployment Descriptor»*;
5. перейдя по адресам http://<server>:9080/test\_service/Test и https://<server>:9443/test\_service/Test, где *server* – адрес сервера, можно проверить работоспособность дефолтной SSL конфигурации, при этом подключение должно выполняться на алгоритме RSA. В IE можно получить ошибку сертификата – это связано с тем, что имя сервера может не соответствовать CN сертификата (для *NodeGostSSLSettings* это обычно localhost).

Далее рекомендуется настроить логирование, чтобы легче было диагностировать ошибки.

1. ***Настройка логирования*:**
2. перейти в *«Servers > Server Types > Websphere application servers > server1 > Troubleshooting > Logging and tracing > server1 > JVM Logs»* и поставить уровень *Advanced*;
3. перейти в «*Servers > Server Types > Websphere application servers > server1 > Troubleshooting»*, выбрать пункт "*Change log detail levels*", поставить \*=*fine* или даже *all* (для детального логирования) вместо текущего значения *info*.

Теперь необходимо создать копию существующей SSL конфигурации, которая называется *NodeDefaultSSLSettings*, и работать с копией. Описание дефолтной SSL конфигурации есть в файле */opt/IBM/WebSphere/AppServer/profiles/AppSrv01/config/cells/<node\_cell>/security.xml*. Создать копию можно через web-интерфейс Административной консоли.

Создание новой ГОСТовой SSL конфигурации будет поделено на несколько частей: одна часть работы будет выполнена в web-интерфейсе, другая - в консоли, а третья - в файлах security.xml и ssl.client.props.

1. ***Создание ГОСТовой SSL конфигурации NodeGostSSLSettings:***

3.1. *Создание копии NodeDefaultSSLSettings:*

3.1.1. перейти в «*Security > SSL certificate and key management > SSL configurations»* и нажать "*New*". Ввести имя новой конфигурации *NodeGostSSLSettings* и сохранить изменения. Фактически *NodeGostSSLSettings* станет точной копией *NodeDefaultSSLSettings* (с теми же *KeyStore* и *trustStore*);

3.1.2. перейти в *NodeGostSSLSettings*, затем в «*Quality of protection (QoP) settings»*, здесь задать протокол ***TLSv1*** вместо *SSL\_TLS* и вписать провайдер ***JTLS*** вместо *IBMJSSE*. Идентификация клиента - по требованию, но ее можно задать и позднее в security.xml;

3.1.3. остановить сервер, очистить логи (*/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs*);

3.2. *Снятие экспортных ограничений IBM JVM (/opt/IBM/WebSphere/AppServer/java/jre):*

3.2.1. Процедура детально описана здесь: <https://support.cryptopro.ru/index.php?/Knowledgebase/Article/View/44/6/snjatie-ehksportnykh-ogrnichenijj>. Все действия сводятся к распаковке и замене файлов;

3.3. *Настройка JCP и JTLS версии 2.0:*

3.3.1. внимательно ознакомиться с руководством администратора и инструкциями по использованию JCP и JTLS в части их установки/удаления, создания ключевых контейнеров и хранилищ сертификатов, а также настройки JTLS сервера;

3.3.2. под управлением учетной записи администратора установить JCP и JTLS;

3.3.2. запустить контрольную панель JCP под учетной записью администратора, перейти на закладку "*Настройки сервера*" и отключить проверку цепочки сертификатов на отзыв;

3.3.3. перейти на закладку "*Оборудование*" и указать фиксированную папку для контейнеров формата HDImageStore - например, */opt/keys*. Таким образом, независимо от пользователя сервером будет получен доступ к контейнерам. Если же не менять эту настройку, то поиск контейнеров будет производиться по умолчанию в папке */var/opt/cprocsp/keys/<username>,* что означает необходимость настраивать и запускать WAS под пользователем *username*, решать, под каким пользователем ставить JCP/JTLS, определять, каким компонентам системы требуются дополнительные права;

3.3.4. перейти в панели JCP на закладку "*Хранилища ключей и сертификатов*";

3.3.5. создать серверный ключевой контейнер (например, *ibmsrv*) с паролем WebAS и политикой сертификата "*серверная аутентификация*", ключом обмена (тип *exchange*), в CN имени сертификата указать ip сервера или его имя, чтобы позднее избежать ошибки сертификата в IE или ошибки *CERT\_E\_CN\_NO\_MATCH* в csptest;

3.3.6. сохранить запрос на сертификат в кодировке base64, затем выпустить сертификат в тестовом УЦ <https://cryptopro.ru/certsrv> и добавить его в созданный контейнер;

3.3.7. аналогично создать клиентский ключевой контейнер (например, *ibm\_client*) с любым паролем и политикой сертификата "*клиентская аутентификация*", ключом обмена (*exchange*). Он потребуется при тестировании подключения с клиентской аутентификацией;

3.3.8. на закладке "*Хранилища ключей и сертификатов*" создать хранилище доверенных сертификатов, например, */opt/IBM/WebSphere/AppServer/etc/ibm.store*, с паролем WebAS. Положить в хранилище корневой сертификат клиентского сертификата, т.е. корневой сертификат тестового УЦ <https://cryptopro.ru/certsrv>;

3.4. *Ручное редактирование security.xml и ssl.client.props:*

3.4.1. *Настройка /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/config/cells/<node\_cell>/security.xml:*

3.4.1.1. настройку лучше производить путем построчного сравнения двух файлов - исходного серверного *security.xml* и примера-образца *security\_double.xml* и вставки нужных строк в первый документ. Предварительно необходимо забекапить исходную версию *security.xml*. Преимущественно затронуты узлы *repertoire, trustManagers, keyManagers, keyStores*;

3.4.1.2. после выполнения необходимых поправок следует проверить, что в отредактированном security.xml у конфигурации *NodeGostSSLSettings* в параметре *clientAuthentication* установлено значение *true*, т. к. данный пример предполагает настройку клиентской аутентификации. Предполагается также, что у новой SSL конфигурации будут новые *keyStores, keyManagers и trustManagers*, которые ссылаются на хранилище */opt/IBM/WebSphere/AppServer/etc/ibm.store*, но выборка ключевого серверного контейнера будет осуществляться по паролю и типу хранилища;

3.4.2. *Настройка /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/properties/ssl.client.props:*

3.4.2.1. в файл *ssl.client.props* нужно внести несколько изменений, которые отражены в приложенном примере-образце *ssl.client.props*. Эти настройки касаются дефолтной конфигурации *DefaultSSLSettings*. Значение поля *keyStoreClientAlias* необязательно должно строго соответствовать алиасу ключа.

Следует упомянуть, что пароль WebAS - это пароль WAS по умолчанию, потому будем использовать его, чтобы не менять пароли в настройках хранилищ сервера.

На стадии handshake сервер произведет поиск серверного ключевого контейнера, опираясь на установенный пароль (WebAS) и тип контейнера (*HDImageStore*), а при построении цепочки сертификатов клиента (в случае включенной клиентской аутентификации) будет использоваться хранилище ibm.store.

После того, как проделаны все настройки из пп.2-3, необходимо завершить конфигурирование, создав порт 9444 и настроив транспортную цепочку.

1. ***Настройка порта 9444 с ГОСТ TLS:***

4.1. запустить сервер;

4.2. если возникли проблемы сразу при запуске, то следует обратиться к логам (*/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1/trace.log*). Скорее всего, ошибки в security.xml;

4.3. запустить Административную консоль под учетной записью администратора;

4.4. перейти в «*Servers > Server Types > Websphere application servers > Ports»* и добавить новый порт 9444 с псевдонимом GostPort (*User-defined port*) для хоста \*. Для него потребуется настроить *associated transport chain;*

4.5. в разделе «*Application servers > server1 > Web container transport chains»* добавить новый элемент *transport chain* с именем GostSecure для только что добавленного порта GostPort (9444). Ассоциировать новую *chain* нужно с *WebContainer-Secure* шаблоном и портом GostPort (9444). Если возникнут проблемы с отображением мастера создания новой *transport chain* после нажатия на кнопку «*New*», то нужно перезапустить сервер и снова вернуться к этому пункту;

4.6. вернуться в GostSecure, «провалившись» в «*Application servers > server1 > Web container transport chains > GostSecure > SSL inbound channel (SSL\_4)»*, указать явно параметр *endpoint (select SSL Configuration)* как *NodeGostSSLSettings*, то есть данный порт будет защищаться ГОСТ TLS;

4.7. раскрыть раздел «*Environment»* в Административной консоли, перейти в «*Virtual Hosts > default\_host > Host Aliases»*, добавить в *Host Aliases* новый алиас \* с портом 9444 (иначе сервер будет возвращать код 404 при обращении к тестовому сервлету */test\_service/Test*);

4.8. перезапустить сервер, предварительно очистив папку с логами (*/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs*).

Теперь осталось только выполнить проверку соединений. Предполагается, что на стороне клиента уже установен созданный раннее ключевой контейнер *ibm\_client* (см. п. 3.3.7).

При переходе по ссылке https://<server>:9443/test\_service/Test должна отобразиться страница с упоминанием алгоритма RSA (ошибка сертификата в браузере или *CERT\_E\_CN\_NO\_MATCH* в csptest означает, что не выполнена проверка *<servername|ip>==<CN>*, но устранить ошибку в csptest можно с помощью флага *-nocheck*), а при переходе по ссылке https://<server>:9444/test\_service/Test должен быть получен запрос клиентского сертификата, после ввода пароля на ключ - страница с изображением протокола *TLS\_CIPHER\_2001* и имени сертификата клиента.