Установка Wildcard сертификата для jira + tomcat на платформе Windows 2008 R2

Оригинал статьи взят с сайта habrahabr.ru

Приветствую всех!

Работаю системным администратором, и недавно возникла казалось бы простая задача:
Установить wild certificate на jira + tomcat(windows server 2008).

Казалось бы задача тривиальная, мануалов должно быть море, но оказалось что все на так просто.

Задача: **Прикрутить сертификат к tomcat, чтобы не было ругани на невалидный сертификат.**

Информацию пришлось собирать по кусочкам с разных ресурсов, потому что единой статьи я так и не нашел, в итоге решение было собрано в единую последовательность действий, которая давала нормальный результат.
Для корректного выполнения команд понадобится дополнительный софт:
Java development kit, плагин для явы ImportKey, и openssl.
Также нам потребуется подписанный сертификат от центра сертификации (извините за тавтологию)
Я заказывал сертификат от Thawte, создавал запрос на сертификат с windows 2008 R2, iis 7
На запрос нам пришло 4 файла – сертификат на домен, два intermediate сертификата и root сертификат. Все с расширением .crt
В случае с godaddy, насколько я знаю приходит файл server.key и тогда конвертация ключа с помощью openssl не нужна.
После того как thawte прислал нам файлы я добавил на сервере откуда создавал запрос intermediate сертификаты в intermediate certification authority, root сертификат в third-party root certification authority. Сертификат на домен в personal. После чего нам нужен export доменного сертификата в формат pfx, обязательно с выгрузкой private key. После чего полученный pfx копируется на сервер с tomcat. После этого его нужно сконвертировать.
В моем примере папка для сертификатов на сервере tomcat c:\keystore\
В моем случае:

Export encrypted private key:
openssl.exe pkcs12 -in c:\keystore\forjira.pfx -nocerts -out c:\keystore\forjira.key

Export certificate:
openssl.exe pkcs12 -in c:\keystore\forjira.pfx -clcerts -nokeys -out c:\keystore\forjira.cer

Decrypt private key:
openssl rsa -in c:\keystore\forjira.key -text -out c:\keystore\rsaforjira.key

pfx c:\keystore\forjira.pfx
cer c:\keystore\forjira.cer
промежуточный key c:\keystore\forjira.key
итоговый key c:\keystore\rsaforjira.key

1. Подготавливаем ключ:
openssl pkcs8 -topk8 -nocrypt -in c:\keystore\rsaforjira.key -inform PEM -out c:\keystore\forjira.der -outform DER
2. Импортируем все сертификаты который прислал нам центр сертификации в windows в личное хранилище, уже на сервере jira.
3. Затем выгружаем в файл сертификат вида *.domain.ru в формате PKCS#7, ставим галку включить в путь все сертификаты, в файл c:\keystore\forjira.p7b

Далее нам понадобиться jdk и ImportKey.
4. импортируем с помощью утилиты ImportKey.java (компиляция > %JAVA_HOME%\bin\javac ImportKey.java)
Для меня было не очевидно что javac идет именно в JDK. И что ImportKey.java нужно скачивать отдельно. Так что учтите это.
Далее делаем команду
> %JAVA_HOME%\bin\java ImportKey c:\keystore\forjira.der c:\keystore\forjira.p7b tomcat

5. В домашней папке пользователя программа создаст хранилище:
C:\user\{пользователь}\keystore.ImportKey, Alias: tomcat Password: importkey
6. Переименовываем в .keystore(я пользовался total commander, так как windows не дает создать файл с именем начинающимся на точку), копируем в корень диска c:\. В моем случае, у меня почему-то tomcat упорно игнорировал настройку в server.xml «keystoreFile=», не знаю уж почему, но хранилище пришлось оставить на c:\.keystore, где tomcat ищет по умолчанию хранилище ключей. Далее мы правим файл server.xml, который находится в \Atlassian\JIRA\conf, где указываем путь для хранилища ключей и пароль на него:

<Connector port=«443»

maxHttpHeaderSize=«8192»
maxThreads=«150»
minSpareThreads=«25»
maxSpareThreads=«75»
enableLookups=«false»
disableUploadTimeout=«true»
acceptCount=«100»
scheme=«https» secure=«true»
clientAuth=«false» sslProtocol=«TLS»
keystoreFile=«c:\.keystore» keypass=«importkey»

В этом же файле правим:
/>
По умолчанию там стоит порт 8443.

Если вы хотите, чтобы tomcat работал только через https — открываем файл web.xml и добавляем или раскомментируем там строчки:
<security-constraint>
<web-resource-collection>
<web-resource-name>all-except-attachments</web-resource-name>
<url-pattern>*.jsp</url-pattern>
<url-pattern>*.jspa</url-pattern>
<url-pattern>/browse/*</url-pattern>
</web-resource-collection>
<user-data-constraint>
<transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
</security-constraint>

Все, после этого перезапускаем tomcat, и он начинает работать только через https, с валидным сертификатом.