Бумажник Oracle Wallet: использование для связи по защищенным каналам

Владимир Пржиялковский

Преподаватель технологий Oracle
prz@yandex.ru
http://open.oracle.tu2.ru/.index

Декабрь 2008 г.
Небольшая правка: январь 2009 г.

И начал меня спрашивать: «Что за человек? Откудова?» И я сказал, что с Москвы, да и подал ему господарский лист волоцкой.

Московского священника Ивана Лукьянова
хождение в Святую землю

Аннотация

Рассматривается использование электронного бумажника Oracle Wallet для установления защищенных соединений СУБД с узлами интернета по протоколу TCPS и клиентских программ с СУБД по протоколу HTTPS.

Защищенные соединения и сертификаты подлинности

Вторая активная область использования электронного бумажника Oracle – поддержка защищенных соединений между СУБД и узлами интернет. Защита соединений между участниками интернет строится на использовании шифрования передаваемых данных, обычно по схеме публичной инфраструктуры ключей (PKI), а это требует передачи от одного участника к другому шифроключа. Риск постороннего вмешательства в такую передачу значительно выше, чем при обращении СУБД к локальной файловой системе, так что кроме собственно шифроключа участники передают друг другу еще специальное подтверждение подлинности, то есть «цифровой сертификат» (*http://en.wikipedia.org/wiki/Public_key_certificate*), дающий принимающей стороне уверенность, что шифр не подложный. Бумажник Oracle способен такие сертификаты хранить.

Соединения участников сети могут быть двух типов: СУБД с другими участниками, когда она выступает в качестве клиента, и других участников (клиентских программ) с СУБД. Ниже приводятся примеры организации того и другого.

Создание бумажника с сертификатами

ПО Oracle позволяет создать пустой бумажник, и потом включить в него необходимые сертификаты подлинности. Однако программы *owm* (Wallet Manager), *orapki*, *mkstore* и *mkwallet* позволяют создать бумажник с некоторым начальным набором сертификатов. Первая из этих программ сумеет создать желаемый каталог для хранения файла бумажника сама, а остальные предполагают каталог уже имеющимся. Пример, для сервера:

```
>set WALLETLOC=C:\oracle\product\10.2.0\db_1\NETWORK\wallet
>mkdir %WALLETLOC%
>mkstore -wrl %WALLETLOC% -create
```

В диалоге потребуется указать придуманный пароль, например, по-прежнему amicus123. Заметьте, что в созданом каталоге образовались сразу два файла, что соответствует постоянно открытому бумажнику; это-то нам и требуется.

Еще пример создания:

```
>orapki wallet create -wallet %WALLETLOC% -pwd amicus123
```

Появился один файл *ewallet.p12*, но зато этой же программой можно ознакомиться с его содержимым:

```
>orapki wallet display -wallet %WALLETLOC% -pwd amicus123
```

Наконец, создать бумажник и работать с его содержимым можно через Oracle Wallet Manager (меню *Wallet* → *New*). При этом от следующего предложения придется отказаться:

В противном случае программа предложит текст заявки на получение сертификата, что пока не нужно. Свойство бумажника *Auto Login* в Wallet Manager нужно будет задать самостоятельно.

Следует обратить внимание, чтобы места нахождения файлов бумажников были доступны элементам ПО соответственно сервера и клиента.

Защищенные соединения с СУБД по протоколу SSL

ПО Oracle Advanced Security позволяет проводить авторизацию соединений клиента с сервером средствами сторонних продуктов, таких как Kerberos или RADIUS. Здесь же будет показано, как это можно сделать с использованием бумажника и техники защищенного соединения SSL (более современное название – TLS, *http://en.wikipedia.org/wiki/Transport_Layer_Security*). С SSL связано и иное решение: соединяться в Oracle без Advanced Security, но по каналу связи («туннелю») SSL в ОС. Оно здесь не рассматривается по меньшей мере потому, что лишает смысла тему настоящей статьи.

Конфигурирование Oracle Net

Процесс установки соединеий клиентов с СУБД *listener* должен быть настроен на прием заявок подключений по протоколу TCPS. Для этого в файле *listener.ora* следует пополнить описание списка портов в параметре LISTENER. По умолчанию Oracle Net использует для TCPS порт 2484; ниже эта традиция сохранена. Указание SSL_VERSION в файлах ниже приводится для определенности и не имеет принципиального значения.

Файлы сервера

Примерный фрагмент из *listener.ora* может выглядеть так:

```
...
LISTENER =
  ( DESCRIPTION_LIST =
    ( DESCRIPTION =
      ( ADDRESS = ( PROTOCOL = TCP )( HOST = oraserver )( PORT = 1521 ) )
      ( ADDRESS = ( PROTOCOL = TCPS )( HOST = oraserver )( PORT = 2484 ) )
    )
  )

SSL_CLIENT_AUTHENTICATION = FALSE

SQLNET.AUTHENTICATION_SERVICES = ( TCPS, NTS )

WALLET_LOCATION =
  ( SOURCE =
    ( METHOD = FILE )
    ( METHOD_DATA =
```

```
      ( DIRECTORY = C:\oracle\product\10.2.0\db_1\NETWORK\wallet )
    )
  )
...
```

После подобной правки файла *listener.ora* программу *listener* стоит перезапустить.

Правка файла *sqlnet.ora* может выглядеть примерно так:

```
...
SQLNET.AUTHENTICATION_SERVICES = ( TCPS, NTS )

SSL_CLIENT_AUTHENTICATION = FALSE

SSL_VERSION = 3.0

WALLET_LOCATION =
  ( SOURCE =
    ( METHOD = FILE )
    ( METHOD_DATA =
      ( DIRECTORY = C:\oracle\product\10.2.0\db_1\NETWORK\wallet )
    )
  )
...
```

Файлы клиента

Правка файла *tnsnames.ora* может выглядеть примерно так:

```
...
ORCL_SSL =
  ( DESCRIPTION =
    ( ADDRESS = ( PROTOCOL = TCPS )( HOST = oraserver )( PORT = 2484 ) )
    ( CONNECT_DATA =
      ( SERVICE_NAME = orcl )
    )
  )
...
```

(Предполагается, что имя службы БД – orcl).

Правка файла *sqlnet.ora* может выглядеть примерно так:

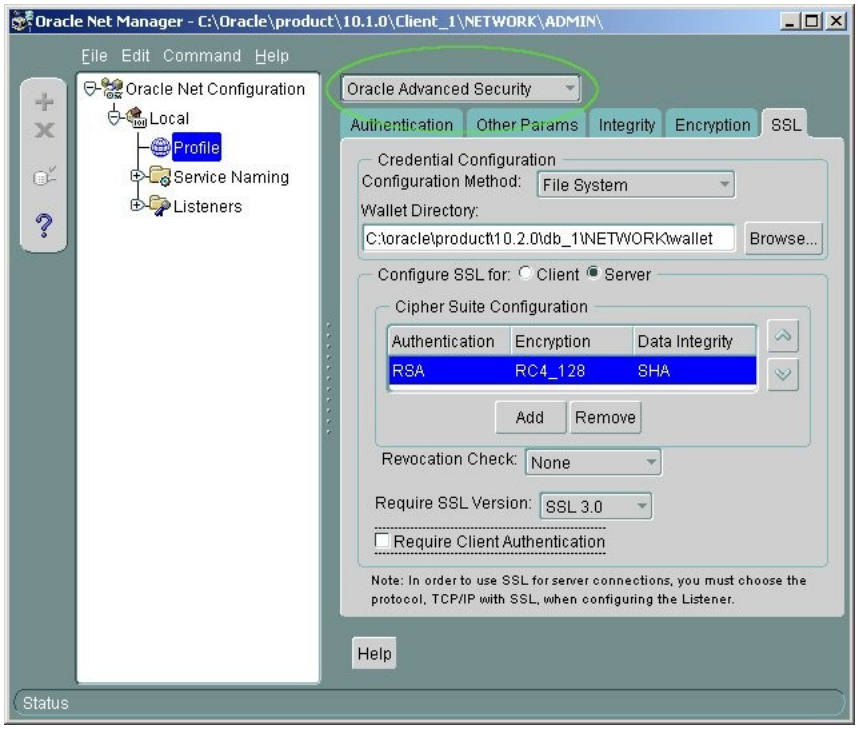
```
...
NAMES.DIRECTORY_PATH = ( TNSNAMES )

SSL_VERSION = 3.0

WALLET_LOCATION =
  ( SOURCE =
    ( METHOD = FILE )
    ( METHOD_DATA =
      ( DIRECTORY = C:\oracle\product\10.2.0\client_1\NETWORK\wallet )
    )
  )
...
```

Конфигурирование программой Net Manager

Выставить требуемые значения параметрам можно программой Net Manager, например:



Может случиться, что ПО Advanced Security доустанавливалось в последствии. Из-за внутренних ошибок программа Net Manager не всегда в таких случаях способна показать позицию *Oracle Advanced Security* ниспадающего меню. Выйти из положения можно, подправив строку из файла *netproperties* в каталоге *%ORACLE_HOME%\network\tools*; например заменить:

```
INSTALLEDCOMPONENTS=CLIENT,ORACLENET
```

на:

```
INSTALLEDCOMPONENTS=CLIENT,ORACLENET,ANO.
```

(ANO – сокращение старого названия нынешнего ПО Advanced Security Option).

Получение цифрового сертификата

Для установления защищенного соединения по SSL потребуется получить цифровые сертификаты подлинности: (а) «пользователя» и (б) «корневой». Сертификат пользователя будем получать на обращение к ресурсу, обозначенному следующим «различительным именем» (distinguished name, dn):

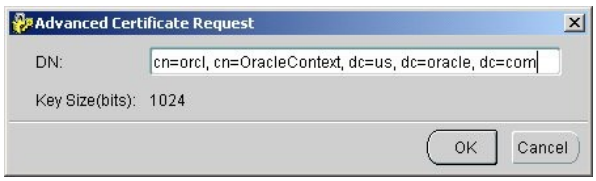
```
cn=orcl, cn=OracleContext, dc=us, dc=oracle, dc=com
```

(Предполагается, что имя экземпляра СУБД – orcl). Правила построения различительных имен многократно описаны в литературе и в интернете.

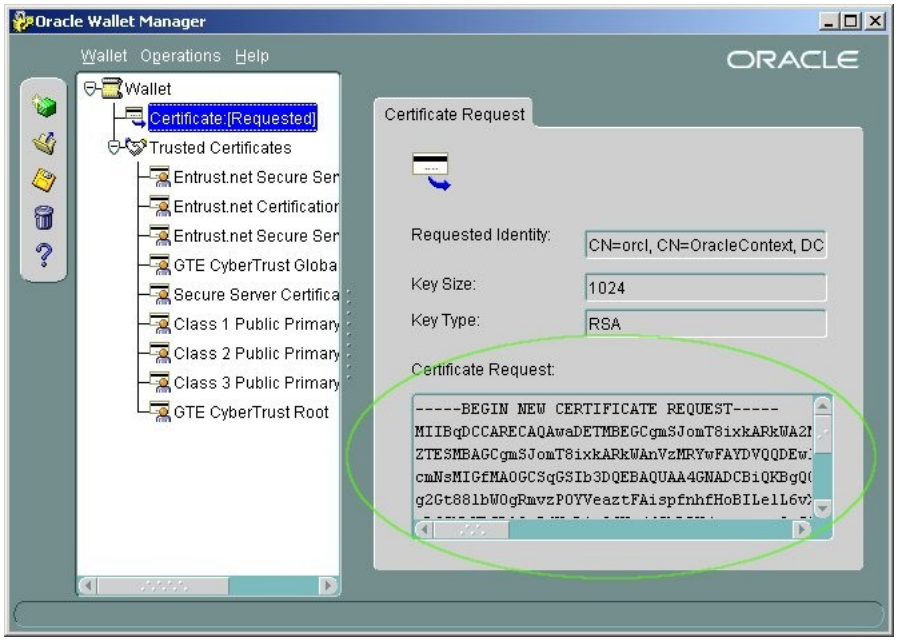
Корневой сертификат понадобится для того, чтобы подтвердить сертификат пользователя. Подтвержденя ему не требуется, так как он подтверждает сам себя (отсюда название).

Подготовка заявки на сертификат пользователя

Получение сертификата пользователя производится по подготовленной заявке (request). Подготовить заявку можно с помощью Wallet Manager, отдельно для бумажников на сервере и на клиенте. Обратиться к форме составления заявки можно через меню *Operations* → *Add Certificate Request*. В форме *Create Certificate Request* следует выбрать *Key Size* = 1024 и нажать кнопку *Advanced*. В появившуюся форму нужно внести различительное имя и нажать *OK*:



Если теперь «встать» на ветку *Certificate* ***(Requested)*** дерева объектов бумажника, справа увидим текст заявки на сертификат:



При желании можно с пощью меню запомнить текст заявки во внешнем файле.

Получение сертификатов

Получить сертификат можно разными способами:

- от уполномоченных центров сертификации (certificate authority, CA), выдающих сертификаты за плату;
- с помощью бесплатных программ получения сертификатов, например *openSSL*;
- от собственного уполномоченного центра в рамках Identity Management Infrastructure в составе Oracle Application Server.

Последний вариант выглядит естественным при работе с ПО Oracle, однако, как и второй, требует определенного разбирательства, отвлекающего от основного изложения. Простым выходом может оказаться получение пробного бесплатного сертификата с трехнедельным сроком годности от фирмы thawte.

На месте *www.thawte.com* в web нужно найти страницу «21-Day Free Trial SSL Certificate from thawte» и заполнить поля своими данными. Далее на странице «21-Day Free SSL Trial Certificate» нужно проставить отметку SSL123 Certificate (All servers) и в поле для заявки внести текст, полученный от Wallet Manager. Нажав *next*, получим в ответ текст сертификата. Пока его нужно запомнить в каком-нибудь файле, например *userCert.txt*. Такой файл получаем отдельно для бумажника на сервере и на клиенте.

Корневой сертификат можно получить по адресу *https://www.thawte.com/roots/*. Скачав файл *thawte-roots.zip*, нужно извлечь из него текстовое или двоичное представление корневого сертификата в каталоге *Thawte Test Roots*.

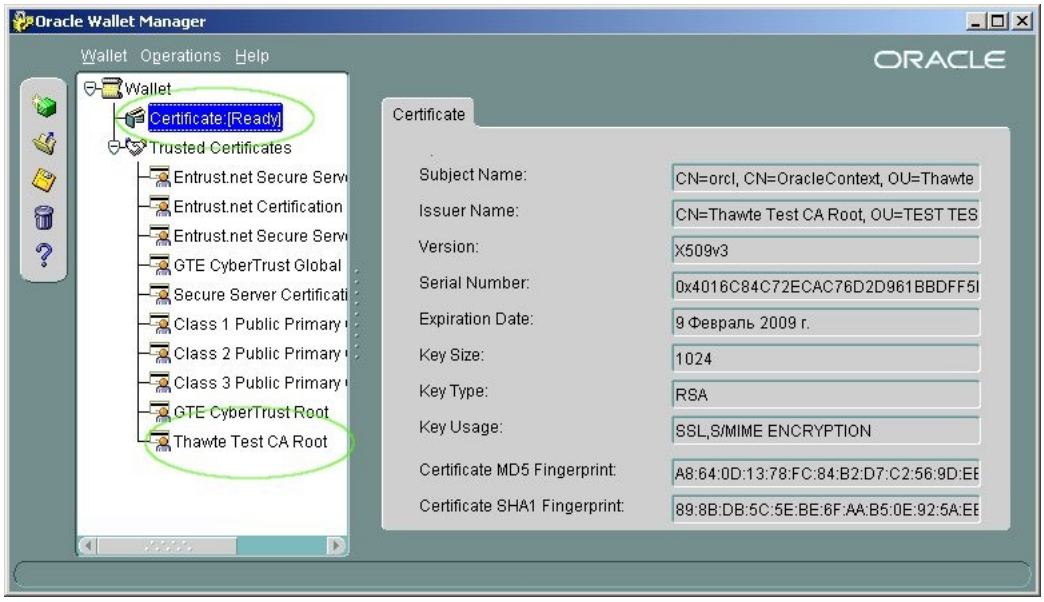
Импорт сертификатов в бумажник

Нагляднее импортировать сертификаты опять-таки в Oracle Wallet Manager, хотя делать это можно и программами из командной строки.

Вначале импортируется корневой сертификат. Используется меню *Operations* → *Import* ***Trusted*** *Certificate*.

Во вторую очередь импортируется сертификат пользователя посредством меню *Operations* → *Import* ***User*** *Certificate*. Оба действия допускают указание как ссылки на файл, так и живого текста.

В результате ветка *Certificate (Requested)* объектов бумажника получит вид *Certificate (**Ready**)*:



Сертификаты импортируются отдельно в бумажники сервера и клиента.

Установление защищенного соединения

На клиенте набираем:

```
>sqlplus scott/tiger@orcl_ssl
```

Убедиться, что действительно произошло соединение по SSL, можно по файлам протокола (журнала) соединений *listener.log*. Если же включить трассировку сеанса (параметр TRACE_LEVEL_CLIENT в файле *sqlnet.ora*), увидим в файле трассировки примерно следующий фрагмент:

```
...
[10-DEC-2008 20:13:11:828] nzos_Trace_Negotiated_Cipher: The Final
Negotiated SSL Cipher Suite is: SSL_RSA_WITH_3DES_EDE_CBC_SHA
[10-DEC-2008 20:13:11:828] ntzdosecneg: SSL handshake done
...
```

Обращение в интернет по протоколу https

Наличие бумажника с необходимым сертификатом позволяет устанавливать соединения СУБД с узлами сети по защищенному протоколу HTTPS с помощью пакета UTL_HTTP. Для примера рассмотрим адрес *https://www.thawte.com*. Чтобы обратиться к нему из программы потребуется занести в бумажник сертификат. Сертификат можно извлечь из полученного ранее файла *thawte-roots.zip*, из каталога *Thawte Server Roots*. С равным успехом для импорта в бумажник годятся и двоичный файл *ThawtePremiumServerCA.cer* и содержимое текстового *ThawtePremiumServerCA_b64.txt*. Импортируем сертификат как и выше, с помощью Wallet Manager и меню *Operations* → *Import Trusted Certificate*.

Подготовим текст программы для SQL*Plus:

```
SET SERVEROUTPUT ON

DECLARE
req         UTL_HTTP.REQ;
```

```
resp       UTL_HTTP.RESP;
walletdir  VARCHAR2 ( 100 )
           := 'file:C:\oracle\product\10.2.0\db_1\NETWORK\wallet';
walletpass VARCHAR2 ( 100 ) := 'amicus123';
targeturl  VARCHAR2 ( 100 ) := 'https://www.thawte.com';

BEGIN
UTL_HTTP.SET_WALLET            ( walletdir, walletpass );
req  := UTL_HTTP.BEGIN_REQUEST ( targeturl );
resp := UTL_HTTP.GET_RESPONSE  ( req );
UTL_HTTP.END_RESPONSE          ( resp );

DBMS_OUTPUT.PUT_LINE ( 'HTTP response status code: ' || resp.status_code );
DBMS_OUTPUT.PUT_LINE ( 'HTTP response reason: '      ||
resp.reason_phrase );
END;
/
```

При наличии приближенного сервера (proxy) потребуется связаться через него (UTL_HTTP.SET_PROXY).

Прогон текста должен дать результат:

```
HTTP response status code: 200
HTTP response reason: OK
```

Упражнение. Убрать из текста программы обращение к бумажнику (SET_WALLET) и проверить доступ к адресам: (а) *https://www.thawte.com*, (б) *http://www.thawte.com*.

В созданом средствами Oracle бумажнике имеется несколько предустановленных сертификатов. Один из них готов для установления прямого защищенного соединения по протоколу HTTPS с узлом по адресу *https://www.entrust.net/*.