Как взломать парольную защиту Oracle или как ее обойти

Владимир Пржиялковский

Преподаватель технологий Oracle
prz@yandex.ru
http://www.ccas.ru/prz

ЦАРСКОСЕЛЬСКАЯ СТАТУЯ[1]

Урну с водой уронив, об утес ее дева разбила.
Дева печально сидит, праздный держа черепок.
Чудо! не сякнет вода, изливаясь из урны разбитой:
Дева над вечной струей вечно печальна сидит.

Александр Сергеевич Пушкин

Чуда не вижу я тут. Генерал-лейтенант Захаржевский,
В урне той дно просверлив, воду провел чрез нее.

Алексей Константинович Толстой

Введение

СУБД Oracle, подобно всем, реально конкурирующим с ней, является старой системой, создание которой происходило, как и продолжается ныне развитие, в рыночных условиях. В этой СУБД, как и у конкурентов, есть целый ряд конструктивных решений, принятых в свое время второпях, и со временем оказавшихся неудовлетворительными. Что-то удается усовершенствовать: например механизмы выделения динамической памяти для текущих нужд СУБД, регулирования доступа к общим ресурсам СУБД или буферизации блоков данных. Однако некоторые заложенные на ранних стадиях развития механизмы или же не удается изменить вовсе (недоразвитое понятие схемы БД) или удается, но с большим запозданием. К числу таковых относится механизм парольной защиты пользователей (user) и ролей (role). Особенности парольной защиты Oracle, способствующие несанкционированному проникновению в БД, затронуты в этой статье.

Приводимый далее материал существенно использует сведения из *http://www.red-database-security.com/*, *http://www.petefinnigan.com/*, *http://isc.sans.org/*.

Реализация парольной защиты в Oracle

Основным принятым средством аутентификации (проверки подлинности) пользователя Oracle и включаемой/выключаемой роли является указание пароля. Так, пароль указывается при выполнении соединения с СУБД (например, в SQL*Plus в команде CONNECT), в предложении SQL создании пользователя или в полном определении связи с посторонней БД (database link).

Хранение пароля

Заданный для пользователя Oracle командой CREATE/ALTER USER пароль подвергается преобразованию и попадает в словарь-справочник в виде свертки (password hash). При указании пароля в момент установления соединения с СУБД Oracle заново вычислит свертку и сравнит ее с хранимой в БД. В открытом виде пароли в БД в настоящее время не хранятся.

Основное место хранения свертки пароля – таблица словаря-справочника SYS.USER$. Над этой таблицей как базовой построена производная, SYS.DBA_USERS. Если в профиле (profile) пользователя включен параметр PASSWORD_REUSE_TIME, свертки пароля также хранятся в SYS.USER_HISTORY$. До версии 10.2 пароли пользователей в *открытом виде* хранились также в таблице SYS.LINK$.

Увидеть свертки логически можно, выдав например:

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> SELECT username, password FROM dba_users;

USERNAME                       PASSWORD
------------------------------ ------------------------------
MGMT_VIEW                      34D8B04B40368661
SYS                            8A8F025737A9097A
SYSTEM                         2D594E86F93B17A1
DBSNMP                         FFF45BB2C0C327EC
SYSMAN                         2CA614501F09FCCC
XDB                            FD6C945857807E3C
SCOTT                          F894844C34402B67
ADAM                           DC8670031DD24E45
PROF                           3D2DEE6D12BD13D2
FORD                           0805304F10B59B54
XTEST                          5E3A5B0B1B1B4755
STREAMADMIN                    77079477FD902BB1
OUTLN                          4A3BA55E08595C81
EXFSYS                         66F4EF5650C20355
WMSYS                          7C9BA362F8314299
DIP                            CE4A36B8E06CA59C
TSMSYS                         3DF26A8B17D0F29F
ANONYMOUS                      anonymous

18 rows selected.
```

Последняя строка в приведенной выдаче является иллюстрацией применения недокументированной, но широко известной возможности Oracle занести в БД на место свертки в БД непосредственное значение:

```
SQL> ALTER USER scott IDENTIFIED BY VALUES 'Это не свертка никакого пароля';

User altered.

SQL> SELECT username, password FROM dba_users WHERE username = 'SCOTT';

USERNAME                       PASSWORD
------------------------------ ------------------------------
SCOTT                          Это не свертка никакого пароля
```

По сути это обесценивает привилегию CREATE SESSION, если таковая у пользователя имеется (соединение все равно невозможно). Возможность занести в БД непосредственно свертку позволяет обладателю привилегии ALTER USER подменить на время пароль, чтобы за законных основаниях войти в систему под чужим именем. Однако если это пользователь SYS, замененный таким образом ему «пароль» не фиксируется в файле *PWD.ORA*, так что особой проблемы с доступностью это свойство не создает.

Если параметр СУБД O7_DICTIONARY_ACCESSIBILITY имеет значение TRUE (умолчание в версии 8), к трем (до версии 10.2 – к четырем) указаным таблицам может обратиться любой обладатель системной привилегии SELECT ANY DICTIONARY; в противном случае – только владелец SYS.

Физически свертки можно наблюдать в файлах ОС: «парольном» *PWD.ORA*; табличного пространства SYSTEM, где хранятся SYS.USER$ и SYS.USER_HISTORY$ (часто это *SYSTEM01.DBF*); полного экспорта; архивированных журналов.

Алгоритм вычисления свертки пароля

Алгоритм вычисления свертки пароля перед помещением его в словарь-справочник БД и при проверки подлинности (аутентичности) официально фирмой-изготовителем не опубликован. Тем не менее вызывающие доверие источники:

http://groups.google.com/group/comp.security.misc/msg/83ae557a977fb6ed?output=gplain
http://isc.sans.org/diary.html?storyid=793 (далее обозначаемый как [1])

сообщают о следующей последовательности действий:

К имени пользователя приклеивается справа текст пароля.
В получившейся строке буквам повышается регистр.
Символы строки переводятся в двухбайтовый формат дополнением слева нулевым значением 0x00 (для символов ASCII), и справа строка дописывается нулевыми байтами до общей длины 80.
Получившаяся строка шифруется алгоритмом DES в режиме сцепления блоков шифротекста (CBC) ключом 0x0123456789ABCDEF.
Из последнего блока результата удаляются разряды четности и полученная строка (56 разрядов) используется для нового шифрования исходной строки тем же способом.
Последний блок результата переводится в знаки шестнадцатиричной арифметики и объявляется конечным результатом – сверткой.

Особенности такого алгоритма:

Свертка не зависит от регистра букв. Например, пары SCOTT/TIGER, Scott/Tiger, scoTT/TigeR дадут одну и ту же свертку **F894844C34402B67**.
Одинаковые склеенные пары *имя_пользователя*/*пароль* дают одинаковую свертку. Например, пары SCOTT/TIGER, SCOT/TTIGER, SCOTTTIG/ER дадут одну и ту же свертку **F894844C34402B67**.
Свертка не зависит от БД. Например, где бы мы ни создавали БД Oracle, свертка для пользователя SCOTT и пароля TIGER всегда будет **F894844C34402B67**.
Используется шифрование DES.

Обход парольной защиты

Не следует забывать, что подсоединение к СУБД может быть выполнено в обход проверки подлинности паролем. В Unix доверительное подключение пользователя SYS, не требующее указания пароля, возможно при работе от имени пользователя ОС, входящего в группу ОС dba, а в Windows – входящего в группу ORA_DBA, но еще при дополнительном условии, что в файлах *sqlnet.ora* на клиентской машине и на сервере имеется значение NTS для параметра SQLNET.AUTHENTICATION_SERVICES. При заведении ПО Oracle на Windows это значение этого параметра устанавливается автоматически, что часто игнорируется начинающими администраторами Oracle на Windows и составляет одну из наиболее популярных ошибок.

Пример доверительного подключения к СУБД пользователя SYS уже встречался выше:

```
SQL> CONNECT / AS SYSDBA
```

В версии 8 и предыдущих ему полностью соответствовала запись:

```
SQL> CONNECT INTERNAL
```

Возможно беспарольное подключение и других пользователей при условии, что имена таких пользователей в Oracle соотнесены именам пользователей ОС или же употреблены в справочнике каталогов LDAP.

Устанавливаются такие свойства командами типа:

```
CREATE/ALTER USER ... IDENTIFIED EXTERNALLY ...
CREATE/ALTER USER ... IDENTIFIED GLOBALLY AS ...
```

В любом случае речь идет о передаче задачи аутентификации внешним по отношению к Oracle средам: ОС или серверу каталогов, и ответственность за несанкционированное проникновение перекладывается на них. Иногда (но не обязательно) это позволяет обеспечить даже лучшую защищенность, чем ту, что дает СУБД Oracle.

Беспарольное подключение выглядит совсем просто:

```
CONNECT /
```

или же:

```
CONNECT /@имя_соединения
```

Пользователи Oracle с подобными свойствами тоже могут обладать привилегией SELECT ANY TABLE, позволяющей читать любые свертки (с учетом оговорки, сделаной выше).

Кроме того, привилегией SELECT ANY TABLE обладают и многие пользователи-схемы, штатно включаемые в состав БД в Oracle. Если администратор не изменит исходно установленные для них пароли, возникает риск несанкционированного прочтения свертки паролей прочих пользователей. Список исходных паролей для многих штатных пользователей Oracle можно найти на *http://www.cirt.net/cgi-bin/passwd.pl?method=showven&ven=Oracle* и в *http://www.oracle.com/technology/deploy/security/pdf/twp_security_checklist_db_database.pdf*.

Взлом пароля

Подбор пароля в Oracle облегчается свойствами принятого алгоритма вычисления свертки ([1]).

А) Сведение алфавита к одним только большим буквам существенно упрощает перебор. Имея в виду 26 больших букв латинского алфавита и 10 цифр, разных паролей длиною n может быть 36^n; если же буквы могуг быть и большие, и маленькие, их полное число становится 52, и паролей может быть 62^n.

(Может показаться, что эти числа чуть преувеличены, так как Oracle не позволяет начинать пароль с цифры, однако такую проверку СУБД делает в момент установления пароля, а это легко нейтрализовать:

```
SQL> ALTER USER scott IDENTIFIED BY a;

User altered.

SQL> ALTER USER scott IDENTIFIED BY 1;
ALTER USER scott IDENTIFIED BY 1
                               *
ERROR at line 1:
ORA-00988: missing or invalid password(s)

SQL> ALTER USER scott IDENTIFIED BY 1a;
ALTER USER scott IDENTIFIED BY 1a
                               *
ERROR at line 1:
ORA-00988: missing or invalid password(s)

SQL> ALTER USER scott IDENTIFIED BY "1";

User altered.

SQL> CONNECT scott/1
Connected.
```

Но даже если бы такое ограничение существовало, оно бы не делало погоды в сокращении объемов перебора).

Б) Знание свертки и имени пользователя позволяет сократить перебор вариантов.

В) Свертка вычисляется только на основе имени и пароля, так что сам подбор можно осуществлять в собственной базе, «на стороне», не оставляя следов в исходной базе и не испытывая проблем соединения с ней.

Г) Хотя сложность взлома шифрования DES достаточно велика, по нынешним меркам этот алгоритм уже не считается достаточно стойким (*http://en.wikipedia.org/wiki/Data_Encryption_Standard*).

Сам подбор возможен как на основе списков наиболее употребимых паролей, так и грубым перебором.

На *http://www.red-database-security.com/software/checkpwd.html* приводится пример программы, подбирающей пароль перебором, отталкиваясь от известного имени пользователя и известной свертки. Еще одна ссылка на подобную программу приведена в [1]: это программа RainbowCrack (*http://www.antsight.com/zsl/rainbowcrack/*). Приведенное время распознавания 8-символьного пароля для пользователя SYSTEM последней программой примерно 4 минуты; тем не менее оригинальная программа потребовала корректировки. Есть и другие подобные программы.

Ответ фирмы Oracle на заявления о слабости парольной защиты

В ответ на опубликование 18 октября 2005 года [1] фирма Oracle 10 ноября того же года опубликовала Note: 340240.1 на *metalink.oracle.com*.

Фирма рекомендует использовать управление паролями с помощью профилей, в частности, часто менять пароль и выбирать пароли не короче 12 символов.

Пример функции проверки выставляемых паролей давно имеется в штатной поставке ПО Oracle в файле *utlpwdmg.sql*. Пример употребления может выглядеть так:

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> @?/rdbms/admin/utlpwdmg

Function created.

Profile altered.

SQL> ALTER USER scott IDENTIFIED BY tiger;
ALTER USER scott IDENTIFIED BY tiger
*
ERROR at line 1:
ORA-28003: password verification for the specified password failed
ORA-20003: Password should contain at least one digit, one character and one
punctuation
SQL> ALTER USER scott IDENTIFIED BY tiger_1234567;

User altered.

SQL> SELECT * FROM user_history$;

     USER# PASSWORD                       PASSWORD_
---------- ------------------------------ ---------
        38 F1A76B5340C01290               25-APR-07
```

(Сценарий *utlpwdmg.sql* не только заводит функцию SYS.VERIFY_FUNCTION проверки выбираемого пользователем пароля, но еще и определяет парольные параметры профиля DEFAULT, в частности устанавливает значение для PASSWORD_REUSE_TIME. Чтобы отменить их действие, потребуется выставить командой ALTER PROFILE default … значения парольных параметров в UNLIMITED).

Во-вторых, фирма рекомендует защищать все файлы, где может оказаться значение сверток паролей (см. выше).

В-третьих, фирма советует защищать передачу данных по Oracle Net, и в-четвертых – полагаться на внешние системы аутентификации («беспарольное», с точки зрения СУБД, подключение, см. выше).

В этом же пояснении фирмы приводится ссылка на находящийся в открытом доступе документ *http://www.oracle.com/technology/deploy/security/pdf/twp_security_checklist_db_database.pdf* с названием «Oracle Database Security Checklist», говорящим за себя. Документ уже более поздний: датирован январем 2007 года; знакомство с ним систематизирует многое из рассмотренного выше.

Неизменным пока остается самое уязвимое место в парольной защите Oracle: алгоритм вычисления свертки. Вероятное решение этой проблемы – дождаться версии 11 СУБД Oracle. По неофициальным сведениям в этой версии будет-таки введено различие больших и малых букв в пароле и алгоритм DES заменен на более современный: SHA-1 либо AES (*http://www.petefinnigan.com/weblog/archives/00000976.htm*, *http://www.red-database-security.com/whitepaper/oracle_passwords.html*). Обработка паролей в версиях вплоть до 10.2, вероятно, меняться не будет.

1Оба стихотворения маститых поэтов, написанные с разрывом в несколько десятков лет, посвящены одной и той же статуе, доныне сидящей в парке близ Екатерининского дворца в Царском Селе, расположенном неподалеку от кажется большого города, но без названия, вместо которого на карте находится другой город, всего за 300 последних лет пять раз принимавший четыре разных исторических названия, два из которых к настоящему моменту русские, а два – немецкие. Одна из рекомендаций, приводимых в статье – почаще и разнообразнее менять пароли пользователей, используя в том числе оба регистра букв и спецсимволы.