RISK 2008, Oslo, Norway, April 23\textsuperscript{rd} 2008

Oracle Security Auditing

By

Pete Finnigan

Written Friday, 25th January 2008
Introduction - Commercial Slide.

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

• Part 1 – Overview of database security
  – What is Oracle Security?
  – Why a database must be secured
  – How can a database be breached?

• Part 2 – Conducting a database audit
  – Planning the audit
  – Conducting an Oracle database security audit
  – Analysis

• Part 3 – The correction phase
  – What to do next
What Is Oracle Security?

• It is about creating a secure database and storing critical / valuable data securely
• To do this Oracle security is about all of these:
  – 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…
Internal Or External Attacks

- Internal attacks are shown to exceed external attacks in many recent surveys, Delloite surveys the top 100 finance institutes.
- The reality is likely to be worse as surveys do not capture all details or all companies.
- Data is often the target now not system access; this could be for identity theft to clone identities.
- 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
  - **Internal staff have access already!!**
How Easy Is It To Attack?

- Many and varied attack vectors
- 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
- More?
Example Exploit

```sql
SQL> show user
USER is "SCOTT"
SQL> @10g_exploit

<table>
<thead>
<tr>
<th>USERNAME</th>
<th>GRANTED_ROLE</th>
<th>ADM</th>
<th>DEF</th>
<th>OS_</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCOTT</td>
<td>APP_ROLE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>SCOTT</td>
<td>CONNECT</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>SCOTT</td>
<td>RESOURCE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

PL/SQL procedure successfully completed.

<table>
<thead>
<tr>
<th>USERNAME</th>
<th>GRANTED_ROLE</th>
<th>ADM</th>
<th>DEF</th>
<th>OS_</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCOTT</td>
<td>APP_ROLE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>SCOTT</td>
<td>CONNECT</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>SCOTT</td>
<td>DBA</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>SCOTT</td>
<td>RESOURCE</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>
```

http://www.milw0rm.com/exploits/4572
Example Exploit (2)

IDS and IPS evasion is a major problem for vendors 
“payloads” are infinite!
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 security 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
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:
  - 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
  - No form of IP blocking and filtering

- Do not do any of these!

11gR1 has broken this!!
Part 2 – Conducting A Database Audit

• Planning and setting up for An Audit
• Selecting a target
• Interview key staff
• Versions, patches and software
• Enumerate users and find passwords
• File system analysis
• Network analysis
• Database configuration
Planning An Audit

• Create a simple plan, include
  – The environments to test
  – The tools to use
  – Decide what to test and how “deep”
  – The results to expect
  – Looking forward
  – What are you going to do with the results?

• Don’t create “war and peace” but provide due diligence, repeatability
The Test Environment

• This is a key decision
• Which environment should be tested?
• A live production system should be chosen
• Some elements can be tested in other systems
  – i.e. a complete clone (standby / DR) can be used to assess configuration
  – The file system and networking and key elements such as passwords / users must be tested in production
• Choose carefully
There are a few standalone tools available:

- www.petefinnigan.com/find_all_privs.sql
- www.petefinnigan.com/who_has_priv.sql
- www.petefinnigan.com/who_can_access.sql
- www.petefinnigan.com/who_has_role.sql
- www.petefinnigan.com/check_parameter.sql

Hand code simple queries as well
Checklists – Basis For The Audit

- There are a number of good checklists to define what to check:
### Decide The Scope Of The Test

- **What is to be tested (what checks to use)?**
- **The checklists provide extensive lists of checks**
- **My advice: keep it simple to start with**
  - Concentrate on the “LOW FRUIT”
  - **Key issues**
    - Passwords
    - Simple configuration issues
    - RBAC issues
Questions?

• Before you start you should assess what you expect as results
• This drives two things:
  – The scale of the test
  – What you can do with the results
• It should help derive
  – What to test for
  – What to expect
• If you decide in advance it’s easier to cope with the output (example: if you do a test in isolation and find 200 issues, it’s highly unlikely anyone will deal with them)
Interview Key Staff

• Perform interviews with key staff
  – DBA
  – Security
  – Applications

• Understand
  – Policies
  – Backups
  – How different groups of staff use and access the database

• The checklists include interview questions
• Prepare an interview list to work to (see the CIS benchmark for examples -

Line up the key people in advance
Don’t base only on internal policies
Look at the installed software and features / functions in the database
Database Version

```
SQL> desc v$version
Name                     Null? Type
-----------------------------------------------------
BANNER                   N      VARCHAR2(32)

SQL> select * from v$version;
BANNER

Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
PL/SQL Release 11.1.0.6.0 - Production
CORE 11.1.0.6.0 Production
TNS for Linux: Version 11.1.0.6.0 - Production
NLSRTL Version 11.1.0.6.0 - Production
```

Ensure it’s a supported version
Patch Status

- DBA_REGISTRY_HISTROY (should work now since Jan 2006 CPU)
- Opatch –lsinventory
- Checksum packages, functions, procedures, libraries, views
  - Rorascanner has example code
  - Some Commercial tools do this
  - Problems – if PL/SQL is not updated in CPU
  - Time based approaches with last_ddl_time
- Ask the DBA we are not trying to break in
# User Enumeration

![SQL Plus output showing user enumeration results](image)

<table>
<thead>
<tr>
<th>USER</th>
<th>Rol</th>
<th>Sys</th>
<th>Ob</th>
<th>Tab</th>
<th>PL</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYS</td>
<td>49</td>
<td>200</td>
<td>14</td>
<td>870</td>
<td>1327</td>
<td>OPEN</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>3</td>
<td>6</td>
<td>46</td>
<td>3</td>
<td>153</td>
<td>OPEN</td>
</tr>
<tr>
<td>QUIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>OPEN</td>
</tr>
<tr>
<td>DIP</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>OPEN</td>
</tr>
<tr>
<td>TSMSYS</td>
<td>0</td>
<td>4</td>
<td>12</td>
<td>20</td>
<td>5</td>
<td>OPEN</td>
</tr>
<tr>
<td>ORACLE_OC</td>
<td>0</td>
<td>4</td>
<td>12</td>
<td>20</td>
<td>5</td>
<td>OPEN</td>
</tr>
<tr>
<td>DBSNMP</td>
<td>1</td>
<td>7</td>
<td>52</td>
<td>43</td>
<td>133</td>
<td>OPEN</td>
</tr>
<tr>
<td>UTMSYS</td>
<td>3</td>
<td>10</td>
<td>13</td>
<td>23</td>
<td>66</td>
<td>OPEN</td>
</tr>
<tr>
<td>EXPSYS</td>
<td>1</td>
<td>12</td>
<td>68</td>
<td>10</td>
<td>10</td>
<td>OPEN</td>
</tr>
<tr>
<td>CIXSYS</td>
<td>2</td>
<td>12</td>
<td>68</td>
<td>10</td>
<td>10</td>
<td>OPEN</td>
</tr>
<tr>
<td>XDI</td>
<td>3</td>
<td>12</td>
<td>68</td>
<td>10</td>
<td>10</td>
<td>OPEN</td>
</tr>
<tr>
<td>ANONYMOUS</td>
<td>0</td>
<td>12</td>
<td>68</td>
<td>10</td>
<td>10</td>
<td>OPEN</td>
</tr>
<tr>
<td>ORDSYS</td>
<td>1</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>OPEN</td>
</tr>
<tr>
<td>ORDPLUGIN</td>
<td>0</td>
<td>12</td>
<td>68</td>
<td>10</td>
<td>10</td>
<td>OPEN</td>
</tr>
<tr>
<td>SI_INFORM</td>
<td>0</td>
<td>12</td>
<td>68</td>
<td>10</td>
<td>10</td>
<td>OPEN</td>
</tr>
<tr>
<td>MDSYS</td>
<td>2</td>
<td>18</td>
<td>30</td>
<td>108</td>
<td>239</td>
<td>OPEN</td>
</tr>
<tr>
<td>OLAFSYS</td>
<td>2</td>
<td>13</td>
<td>41</td>
<td>126</td>
<td>59</td>
<td>OPEN</td>
</tr>
<tr>
<td>MDATA</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>OPEN</td>
</tr>
<tr>
<td>SPATIAL_W</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OPEN</td>
</tr>
<tr>
<td>SPATIAL_C</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OPEN</td>
</tr>
<tr>
<td>UKOS</td>
<td>2</td>
<td>59</td>
<td>32</td>
<td>56</td>
<td>50</td>
<td>OPEN</td>
</tr>
<tr>
<td>WKPROXY</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OPEN</td>
</tr>
<tr>
<td>UNTEST</td>
<td>2</td>
<td>19</td>
<td>681</td>
<td>387</td>
<td>OPEN</td>
<td>OPEN</td>
</tr>
<tr>
<td>SVMAN</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OPEN</td>
</tr>
<tr>
<td>MGML_USER</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OPEN</td>
</tr>
<tr>
<td>FLOUS_FIL</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OPEN</td>
</tr>
<tr>
<td>FLOUS_PUBL</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>1</td>
<td>OPEN</td>
</tr>
<tr>
<td>FLOUS_030</td>
<td>3</td>
<td>28</td>
<td>98</td>
<td>212</td>
<td>73</td>
<td>OPEN</td>
</tr>
<tr>
<td>OWBYS</td>
<td>10</td>
<td>23</td>
<td>43</td>
<td>4</td>
<td>2</td>
<td>OPEN</td>
</tr>
<tr>
<td>SCOTT</td>
<td>2</td>
<td>11</td>
<td>14</td>
<td>10</td>
<td>1</td>
<td>OPEN</td>
</tr>
<tr>
<td>HH</td>
<td>12</td>
<td>14</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>OPEN</td>
</tr>
<tr>
<td>OE</td>
<td>1</td>
<td>11</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>OPEN</td>
</tr>
<tr>
<td>IX</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>OPEN</td>
</tr>
<tr>
<td>SH</td>
<td>2</td>
<td>12</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>OPEN</td>
</tr>
<tr>
<td>SM</td>
<td>2</td>
<td>12</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>OPEN</td>
</tr>
<tr>
<td>BI</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OPEN</td>
</tr>
<tr>
<td>PETE</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OPEN</td>
</tr>
<tr>
<td>BILL</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OPEN</td>
</tr>
</tbody>
</table>

---

**PL/SQL procedure successfully completed.**
Auditing Passwords

• Three types of checks (ok 4)
  – Password=username
  – Password=default password
  – Password=dictionary word
  – Password is too short

• Default check tools or password cracker?

• Password cracker
  – http://soonerorlater.hu/index.khtml?article_id=513
  – http://www.toolcrypt.org/tools/orabf/orabf-v0.7.6.zip
Password Cracker (1)

Run in SQL*Plus

http://soonerorlater.hu/download/worauthbf_src_0.2.zip
http://soonerorlater.hu/download/worauthbf_0.2.zip

```sql
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
Password Cracker (2)

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

Use a default password list or dictionary file.

Worauthbf can also be used to crack from authentication sessions.

Worauthbf can be used in dictionary or brute force mode.
File System Audit

- Finding passwords
- Permissions on the file system
- Suid issues
- Umask settings
- Lock down Key binaries and files
- Look for data held outside the database
- OSDBA membership
- These are a starter for 10: Much more can be done (e.g. I check for @80 separate issues at the OS level); see the checklists for ideas
Finding Passwords

This is one of the key searches
Also search the process lists
Also search history
File Permissions

Test for 777 perms
Files in ORACLE_HOME should be 750 or less
Binaries 755 or less
No one reads and follows the post installation steps
SUID and SGID

Beware of non-standard SUID binaries
Beware of “0” binaries
Change the permissions on those binaries not used
Network Audit

- Listener
  - port
  - listener name
  - service name
- Listener password or local authentication
- Admin restrictions
- Extproc and services
- Logging on
- Valid node checking
### Status of the Listener

<table>
<thead>
<tr>
<th>Alias</th>
<th>LISTENER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>TNSLSNR for Linux: Version 11.1.0.6.0 - Production</td>
</tr>
<tr>
<td>Start Date</td>
<td>31-OCT-2007 09:06:14</td>
</tr>
<tr>
<td>Uptime</td>
<td>0 days 4 hr. 56 min. 27 sec</td>
</tr>
<tr>
<td>Trace Level</td>
<td>off</td>
</tr>
<tr>
<td>Security</td>
<td>ON: Local OS Authentication</td>
</tr>
<tr>
<td>SNMP</td>
<td>OFF</td>
</tr>
<tr>
<td>Parameter File</td>
<td>/oracle/11g/network/admin/listener.ora</td>
</tr>
<tr>
<td>Log File</td>
<td>/oracle/diag/tnslsnr/vostok/listener/alert/log.xml</td>
</tr>
</tbody>
</table>

### 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 "ORA11GXDB", status READY, has 1 handler(s) for this service...
- **Service "ORA11G_XPT"** has 1 instance(s).
  - Instance "ORA11G_XPT", status READY, has 1 handler(s) for this service...

**Sidguesser can guess a SID and cannot be blocked easily**

**Duplicate services**
Listener password

Password is encrypted pre 10g
Hash can be used to log in
Check for clear text passwords or no password
Check admin_restrictions is set
Beware of default file permissions
Services

```
LSNRCTL> services
Connecting to <DESCRIPTION=<ADDRESS=<PROTOCOL=IPC><KEY=EXTPROC1>>>  
Services Summary...
Service "PLSExtProc" has 1 instance(s).
  Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0
      LOCAL SERVER
Service "ora10gr2" has 1 instance(s).
  Instance "ora10gr2", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
      LOCAL SERVER
Service "ora10gr2_KPT" has 1 instance(s).
  Instance "ora10gr2", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
      LOCAL SERVER
The command completed successfully
LSNRCTL>
```
Database Configuration Audit

• Use simple scripts or hand coded commands
• This section can only highlight; use the checklists for a complete list of things to audit
• Check profiles and profile assignment
• Check initialisation Parameters
• Privilege and role assignments
• Much more – see checklists
## Users -> Profiles

No profiles designed on this database

All accounts have same profile except one
Check Parameters

Use the checklists to identify what to check
This parameter setting is not ideal for instance
RBAC

• Review the complete RBAC model implemented
• 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 – check for open accounts
  – Reduce attack surface
Defaults

- Defaults are one of the biggest issues in Oracle
- Oracle has the most default accounts for 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
Test Users Privileges (SCOTT)

```
 find_all_prives: Release 1.0.7.0.0 - Production on Sat Nov 10 10:37:41 2007
 Copyright (c) 2004 PeteFinnigan.com Limited. All rights reserved.

 NAME OF USER TO CHECK [ORCL]: SCOTT
 OUTPUT METHOD Screen/File [S]: $  
 FILE NAME FOR OUTPUT [priv.lst]: 
 OUTPUT DIRECTORY [DIRECTORY or file (/tmp)]:

 User => SCOTT has been granted the following privileges

============================================================================

 ROLE => APP_ROLE which contains =>
    ROLE => MAN_ROLE which contains =>
      SYS_PRIV => EXECUTE ANY PROCEDURE grantable => NO
      SYS_PRIV => ALTER USER grantable => NO
      SYS_PRIV => SELECT ANY TABLE grantable => NO
      TABLE_PRIV => SELECT object => SYS.DBA_USERS grantable => NO

 ROLE => CONNECT which contains =>
      SYS_PRIV => CREATE SESSION grantable => NO

 ROLE => RESOURCE which contains =>
      SYS_PRIV => CREATE CLUSTER grantable => NO
      SYS_PRIV => CREATE INDEXTYPE grantable => NO
      SYS_PRIV => CREATE OPERATOR grantable => NO
      SYS_PRIV => CREATE PROCEDURE grantable => NO
      SYS_PRIV => CREATE SEQUENCE grantable => NO
      SYS PRIV => CREATE TABLE grantable => NO
      SYS_PRIV => CREATE TRIGGER grantable => NO
      SYS_PRIV => CREATE TYPE grantable => NO

 PL/SQL procedure successfully completed.  
 For updates please visit http://www.peteFinnigan.com/tools.htm

SQL>
```
Who Has Key Roles

who has priv: Release 1.0.3.0.0 - Production on Thu Nov 22 16:00:18 2007
Copyright (c) 2004 PeteFinnigan.com Limited. All rights reserved.

ROLE TO CHECK [DBA]: DBA
OUTPUT METHOD Screen/File [$]: $  
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 -> SMSHAN (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> |
Access To Key Data (DBA_USERS)

Object type is -> VIEW (TAB)
Privilege -> SELECT is granted to ->
  Role -> APP_ROLE (ADM = NO) which is granted to ->
    User -> SCOTT (ADM = NO)
  User -> SYSTEM (ADM = YES)
User -> DBA (ADM = NO)
Role -> SELECT_DATABASE_ROLE (ADM = NO) which is granted to ->
  Role -> OLAP_ROLE (ADM = NO) which is granted to ->
    User -> SYS (ADM = YES)
    Role -> DBA (ADM = NO) which is granted to ->
      User -> SYS (ADM = YES)
      User -> SYSTEM (ADM = YES)
      User -> TESTUSER (ADM = NO)
Role -> TMP_FULL_DATABASE (ADM = NO) which is granted to ->
  User -> SYS (ADM = YES)
    Role -> DBA (ADM = NO) which is granted to ->
      User -> SYS (ADM = YES)
      User -> SYSTEM (ADM = YES)
      User -> TESTUSER (ADM = NO)
    Role -> OLAP_ROLE (ADM = NO) which is granted to ->
      User -> SYS (ADM = YES)
      User -> SYSTEM (ADM = YES)
      User -> TESTUSER (ADM = NO)
    User -> DBA (ADM = NO)
User -> SM (ADM = NO)
Role -> EXP_FULL_DATABASE (ADM = NO) which is granted to ->
  Role -> DBA (ADM = NO) which is granted to ->
    User -> SYS (ADM = YES)
    User -> SYSTEM (ADM = YES)
    User -> TESTUSER (ADM = NO)
    User -> SYS (ADM = YES)
    User -> IX (ADM = NO)
Key System Privileges

Note the problem of multiple-inheritance of privileges.
Stage 3 - What To Do Next?

• Write up the audit formally
• Prioritise the findings – Severity 1 – 3?
• Use internal policies to help define
• Other platforms can help (e.g. use your OS experience if you have it)
• Assess risk
Next Step - Create A Policy

- Perform an Oracle database audit
- Define what the key/critical issues are
- Determine / decide what to fix
- Include best practice
- Work on a top 20 basis and cycle (This is effective for new hardening)
- Create a baseline standard
  - A document
  - Scripts – maybe for BMC
  - Commercial tool such as AppDetective
Decide What To Fix

• Perform a risk assessment
• 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
Conclusions

• We didn’t mention CPU’s – Apply them – they are only part of the problem
• Think like a hacker
• Get the basics right first –
  – 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
• Use a top 10 approach in fixing, 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