Sentrigo Webinar, March 28th 2008

# Oracle Security MasterClass

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
- 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

- Introduction
- Demonstration of how to hack Oracle
- Summary of the issues found during the demo
- Why a database must be secured
- Basic Oracle security tenets
- Conclusions

# Demonstration

- Hacking an Oracle database to “steal”

# The Issues

- Access is available to the database
- Credentials are guessable
- Default accounts have access to critical data
- Critical data is easy to find
- Poor, weak encryption and protection used
- This is reality, this is what Oracle database security REALLY looks like!!

# 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

# 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 direct access
- 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 - 25 million child benefit identities lost on two discs (not stolen but lost)
- Scarborough & Tweed SQL Injection - http://doj.nh.gov/consumer/pdf/ScarboroughTweed.pdf
- 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
- ?

# Stay Ahead Of The Hackers

- When deciding what to security audit and how to security 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
  - NIST – http://nvd.nist.gov
  - 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
  - CIS Benchmark - http://www.cisecurity.org/bench_oracle.html
  - SANS S.C.O.R.E - http://www.sans.org/score/oraclechecklist.php
  - Oracle's own checklist - http://www.oracle.com/technology/deploy/security/pdf/twp_security_checklist_db_database_20071108.pdf
  - DoD STIG - http://iase.disa.mil/stigs/stig/database-stig-v8r1.zip
- Websites – petefinnigan.com, cqure, RDS, Argeniss, databasesecurity.com

# A Poll

- Please join in and answer the Poll question

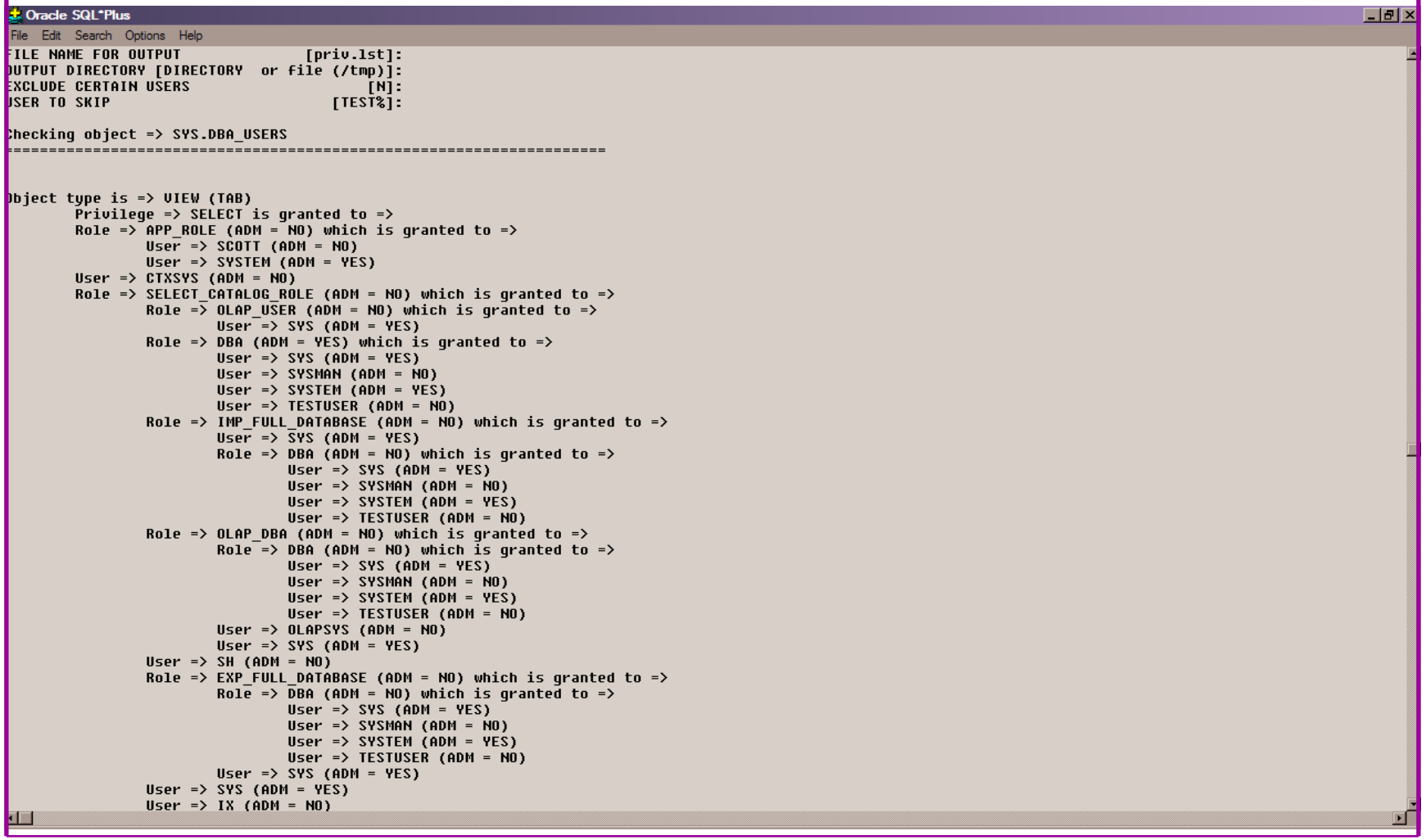
# The Access Issue

- 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

# What to Look For (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
- Tools: Use one of the free tools – CIS, OScanner, rorascanner
- 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

# Access To Key Data (DBA_USERS)



# Password Cracker (1)

Run in SQL*Plus

http://soonerorlater.hu/download/woraauthbf_src_0.2.zip

http://soonerorlater.hu/download/woraauthbf_0.2.zip

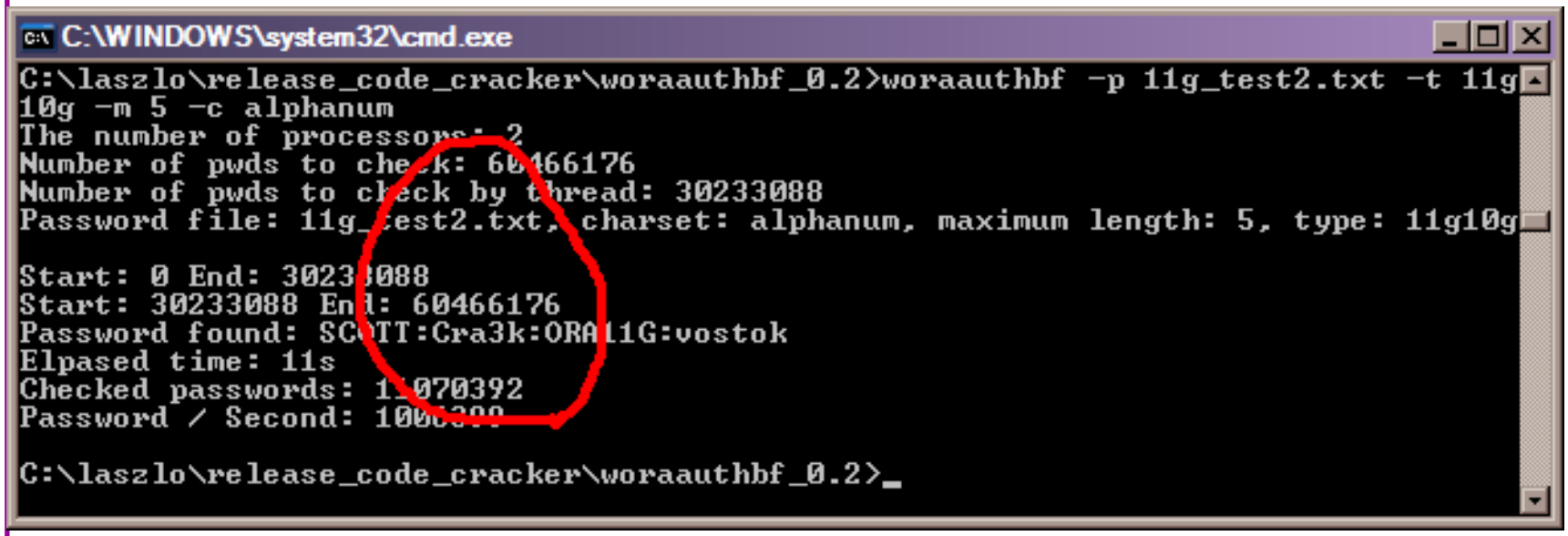
```
Select u.name||':'||u.password
   ||':'||substr(u.spare4,3,63)
   ||':'||d.name||':‘
   ||sys_context('USERENV','SERVER_HOST')||':'
from sys.user$ u, sys.V_$DATABASE d where u.type#=1;
```

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

```
SCOTT:9B5981663723A979:71C46D7FD2AB8A607A93489E899C0
   8FFDA75B147030761978E640EF57C35:ORA11G:vostok:
```

Then run the cracker

# Password Cracker (2)




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

Woraauthbf can also be used to crack from authentication sessions

Woraauthbf can be used in dictionary or brute force mode

Use it to check user=pwd and defaults

# Role Based Access (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

# Secure Listener by Default?

```
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...
```

# 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
    - DBMS_JOBS, DBMS_SCHEDULER
    - OS authenticated users under certain circumstances
- Clean up all of the
  - ad-hoc scripts
  - Maintenance evidence
  - Trace files
  - Data files, exports..
  - Audit logs….

# 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

# 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!

# 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)

# 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

# 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 to the database or cracking
- Sort out the RBAC, configuration, installed software and privileges
- Use audit / IDS / IPS solutions

# 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