PeteFinnigan.com Limited

Oracle Security Expertise

OUGF Conference 2009, May 14th 2009

Oracle Security Masterclass

By

Pete Finnigan

Updated Wednesday, 12th May 2009

Why Am I Qualified To Speak

- 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, Iceland, Finland, Germany and more)
- Member of the Oak Table Network

Agenda

- Part 1 - Background
  - Oracle security information
  - How databases can be breached
  - Tools used to audit a database
- Part 2 - Detailed investigations
  - User details and tips
  - Credit Cards – Data access
  - Operating system access
- Part 3 – Wrapping Up
  - Conclusions

Introduction

- I have taught masterclass's at various events for the last three years
  - [Year 1] - Overview of everything in Oracle security
  - [Year 2] - Overview of everything needed to perform an Oracle database security audit
- In this new masterclass I want to do something different
  - I want to cover some background "glue" but I also want to delve into around 4 / 5 specific areas and look in more depth.
  - The focus is "***how easy it is to steal***" [2 examples] and "***how easy it is to not secure properly***" [3 examples]
  - And; we are going to try quite a few demos!

Overview

- What do I want to achieve today
  - I want you to "grasp" some of the basic ideas behind securing an Oracle database – I will say what they are at the end BUT see if you can pick them up
- Anyone can perform a security audit of an Oracle database BUT we should get the ground rules right and really understand why to secure and how to secure
- **Ask questions any time you would like to**
- Try out some of the tools and techniques yourself later on or now if you have a local Oracle database on a laptop

What Is Oracle Security?

- Securely configuring an existing Oracle database?
- Designing a secure Oracle database system before implementation?
- Using some of the key security features
  - Audit facilities, encryption functions, RBAC, FGA, VPD…
- Oracle security is about all of these BUT
  - **It is about securely storing critical / valuable data in an Oracle database. In other words its about securing DATA not securing the software!**

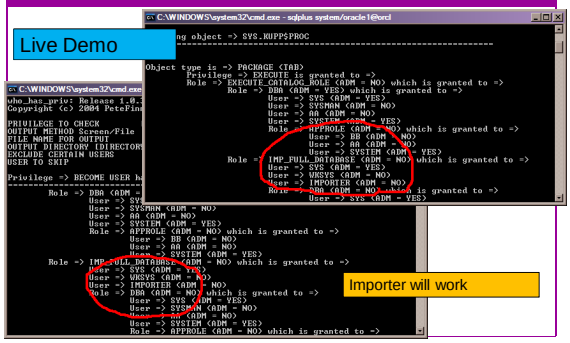
Internal Or External Attacks

- 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
- 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!!**

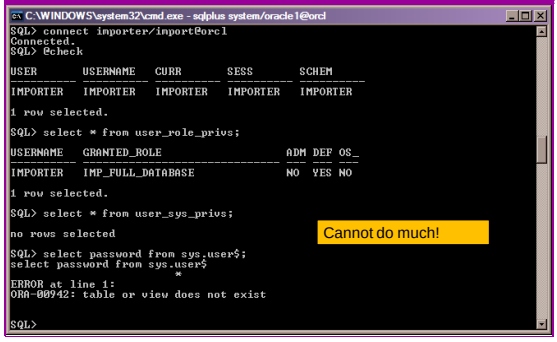
Why We Need Security

- The target is often data not the DBA role
- The exploits we are going to see first work but stealing data is much more "real"
- Its easy, not rocket science, no skill
- Real theft does not require complex techniques either
- What do you think happens in real life?
  - Exploits can be downloaded for free!
  - Stealing is easy because systems are open

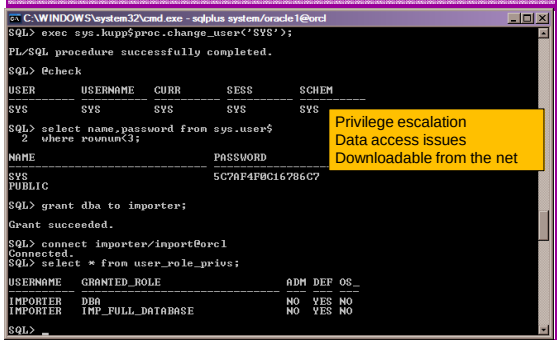
Breach 1 – Escalate Privileges



Breach 1 – Slide 2



Breach 1 – Slide 3



Breach 2 – Stealing Data

- We are now going to demonstrate a much more realistic case of simple data theft
- This is more realistic because real systems audited by us allow this to happen – indeed we know theft using techniques like this has happened

Breach 2 – Slide 2

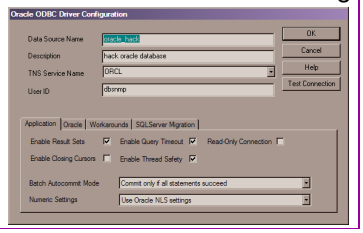
- Hacking an Oracle database to “steal”
- 15 minutes demonstration

Live Demo

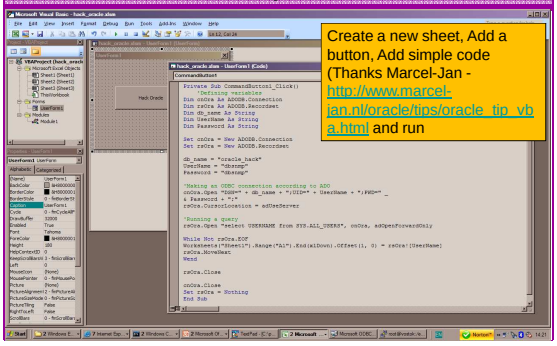
Breach Example 3 – Simple!

- Demo of connecting to the database via MS Excel
- Most sites include standard builds allowing this way in

Live Demo

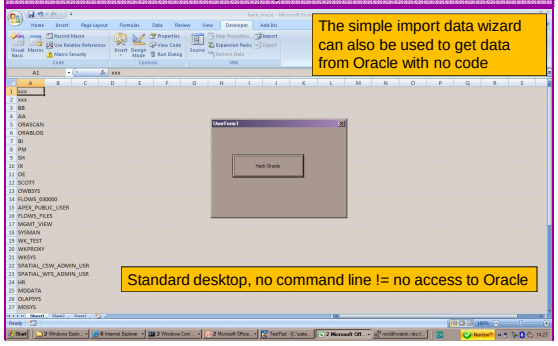


Breach Example 3 – Slide 2



Create a new sheet, Add a button, Add simple code (Thanks Marcel-Jan - http://www.marcel-jan.nl/oracle/tips/oracle_tip_vba.html and run

Breach Example 3 – Slide 3



The simple import data wizard can also be used to get data from Oracle with no code

Standard desktop, no command line != no access to Oracle

Breach 1 - Reaction

- Exploits are easy to download
  - Exploit code from sites like http://www.milw0rm.com
  - Or from papers such as http://blog.tanelpoder.com/2007/11/10/oracle-security-all-your-dbas-are-sysdbas-and-can-have-full-os-access/ - our example
- No real skill is needed (the code exists – your users do not need to write or understand it – or know Oracle)
- Insider threat

Breach 2 - Reaction

- 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!!

Breach 3 and Onwards

- You have to think like a hacker and be suspicious
- Realise the ease with which data can be stolen
- Downloaded exploits are a real issue
- Breach 3 emphasises the need to block connections to the database not developer tools such as SQL*Plus or TOAD
- Key basic issues are a problem in real life
- The threat is to all data not "**grant DBA to scott**" as often shown at conferences in examples

The Access Issue

- This is the number 1 Oracle security issue for me
- 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
- A database can only be accessed at the TNS level if there is a direct route from the user (authorised or not) and the database

11gR1 has broken this with the default sid/service name feature

Access Issue 2

- At lots of sites we audit we see:
  - Tnsnames.ora deployed to all servers and desktops
  - Tnsnames.ora with details of every database
  - access to servers is open (no IP blocking)
  - Guessable SID/Service name
  - Weak passwords
- Do not do any of these at your sites!

The Core Problems

- Incorrect versions and products installed
- Unnecessary functions and features installed
- Excessive users / schemas installed
- Elevated privileges for most database accounts
- Default and insecure configurations
- Lack of audit trails in the database
- Data often held outside the database
- Evidence of ad-hoc maintenance

Configuration And Defaults

- Default database installations cause some weak configurations
- Review all
  - configuration parameters – checklists?
  - File permissions
- Some examples
  - No audit configuration by default (fixed in 10gR2 for new installs)
  - No password management (fixed in 10gR2 new installs)
- In your own applications and support accounts
  - Do not use default accounts
  - Do not use default roles including DBA
  - Do not use default passwords

Background Information

- Basic information must be to hand for familiarisation rather than actual use
- 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

Background Information 2

- Some background information we do use!
- There are a few standalone tools available
- I would start with manual queries and toolkit of simple scripts such as:
  - 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

Background Information 3

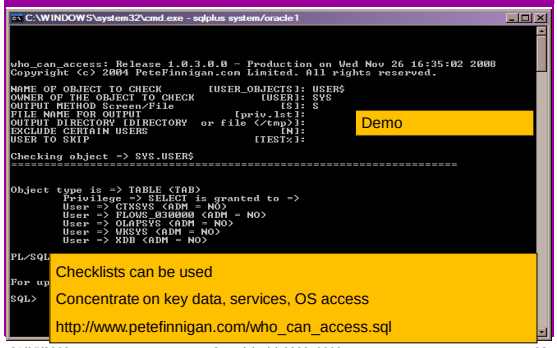
- There are a number of good checklists to define what to check:
- CIS Benchmark - http://www.cisecurity.org/bench_oracle.html
- SANS S.C.O.R.E - http://www.sans.org/score/oraclechecklist.php
- Oracle's own checklist - http://www.oracle.com/technology/deploy/security/pdf/twp_security_checklist_db_database_20071108.pdf
- DoD STIG - http://iase.disa.mil/stigs/stig/database-stig-v8r1.zip
- Oracle Database security, audit and control features – ISBN 1-893209-58-X

Exploring The Toolkit

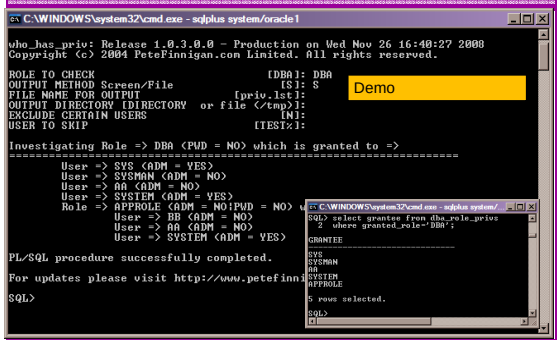
- We are going to demonstrate the 5 scripts
- Assess access to key data
- Assess who has key system privileges
- Assess who has roles
- Assess all the privileges assigned to a user
- Assess parameter settings

Live Demo

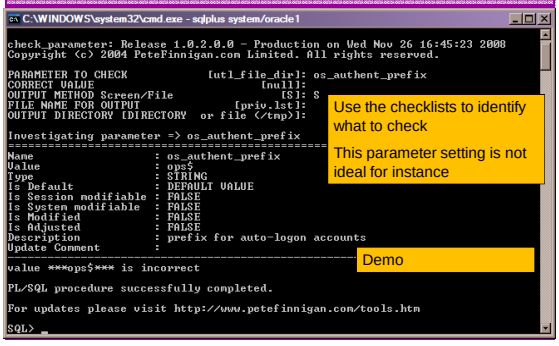
Access To Key Data (SYS.USER$)



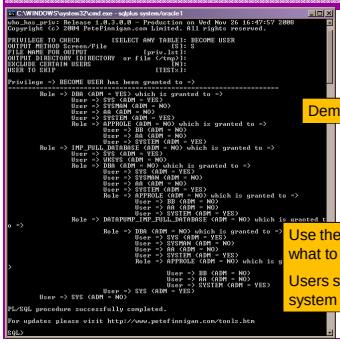
Who Has Key Roles



Check Parameters



Check System Privileges

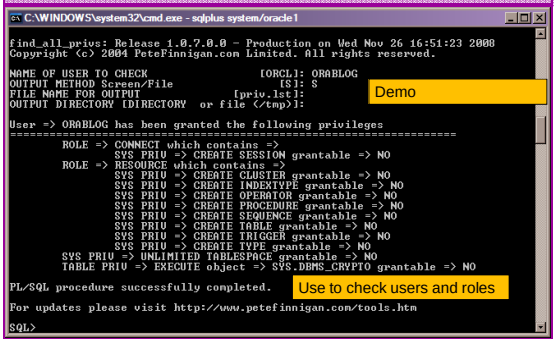


Demo

Use the checklists to identify what to check

Users should not have system privileges

Who Has What Privileges



Demo

Use to check users and roles

Auditing Oracle

- Part 2 of this masterclass
- We are going to delve into three areas of in-depth analysis of an Oracle database
- The three areas are:
  - User analysis
  - Access to key data – Credit cards example
  - Access to services – Operating system files

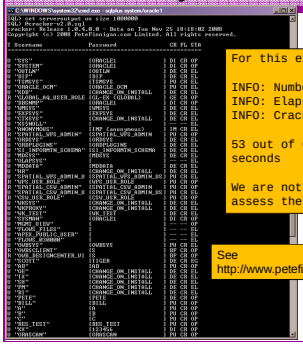
What We Are Looking For

- These three areas are going to be shown in more depth as examples of "**what to look for**"
- I want to show you the similarities in all three areas
- I want to emphasise
  - Depth
  - The focus on data
  - The focus on solution

Analysis Of Users - 1

- Four types of checks
  - Password=username
  - Password=default password
  - Password=dictionary word
  - Password is too short
- Default check tools or password cracker?
- Password cracker
  - http://www.petefinnigan.com/oracle_password_cracker.htm
  - http://soonerorlater.hu/index.khtml?article_id=513
  - http://www.red-database-security.com/software/checkpwd.html
  - http://www.toolcrypt.org/tools/orabf/orabf-v0.7.6.zip

Analysis Of Users - 2

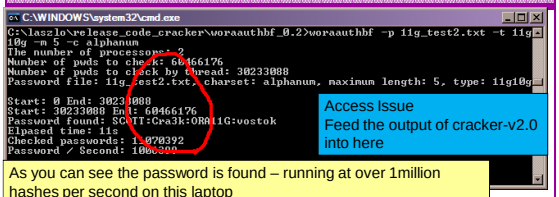


```
For this example run

INFO: Number of crack attempts = [61791]
INFO: Elapsed time = [4.36 Seconds]
INFO: Cracks per second = [14170]

53 out of 60 accounts cracked in 4.3
seconds

We are not trying to break in BUT trying to
assess the "real security level"
```

See http://www.petefinnigan.com/oracle_password_cracker.htm

Access Issue

Analysis Of Users - 3



Access Issue
Feed the output of cracker-v2.0 into here

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

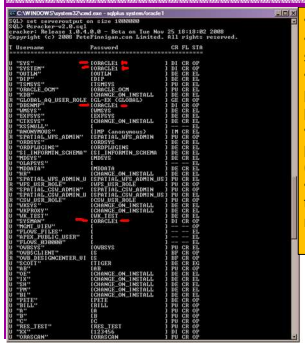
Woraauthbf can also be used to crack from authentication sessions

Woraauthbf can be used in dictionary or brute force mode

Use it to supplement the PL/SQL based cracker

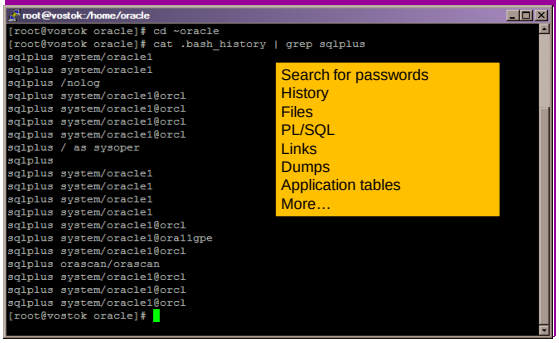
http://www.soonerorlater.hu/download/woraauthbf_src_0.22.zip
http://www.soonerorlater.hu/download/woraauthbf_0.22.zip

Analysis Of Users - 4



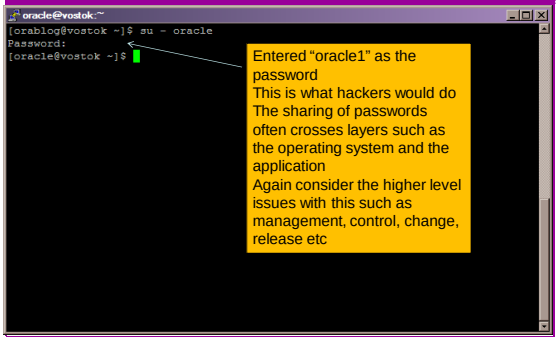
- Shared passwords are a problem
- All privileged accounts have the same password
- This often implies that the same people do one job or multiple people share passwords
- If database links exist they possibly share the same passwords (check dump files)
- Assess not just what you see BUT the implications in terms of management and administration

Analysis Of Users - 5



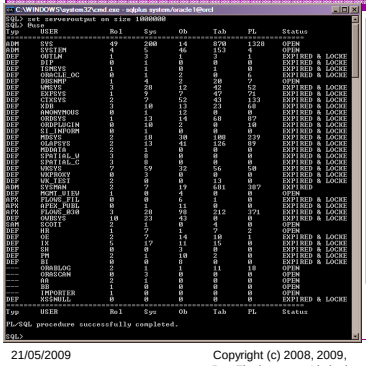
Search for passwords
History
Files
PL/SQL
Links
Dumps
Application tables
More…

Analysis Of Users - 6



Entered "oracle1" as the password
This is what hackers would do
The sharing of passwords often crosses layers such as the operating system and the application
Again consider the higher level issues with this such as management, control, change, release etc

Analysis Of Users - 7

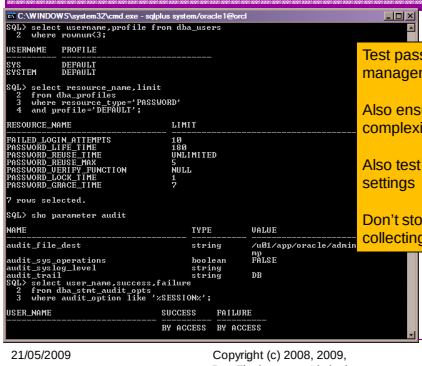


Analyse users into 2 groups

Seek to reduce the accounts (features) installed as default schemas – i.e. OEM, Intelligent agent, DIP, Samples

Analyse accounts created by "you". Assess these in terms of what should exist

Analysis Of Users - 8



Test password management

Also ensure that a complexity function exists

Also test current audit settings

Don't stop at just collecting audit data

Analysis Of Users - 9

- Fixing something as simple as a weak password is not simple!
- Passwords must be cracked regularly
- Passwords must be strengthened
- Password management must be enabled
- Password hashes must be secured
- Throttling enabled
- Audit must be enabled for connections (don't forget sysdba)

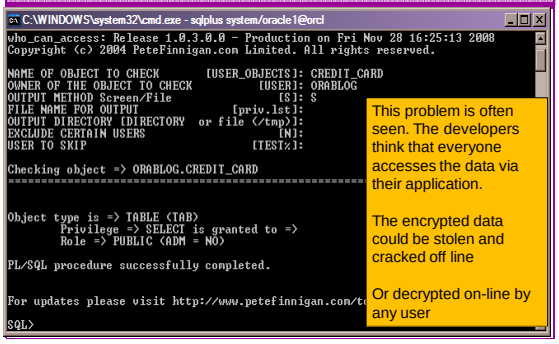
Analysis Of Users - 10

- Accounts in the database installed as defaults must be reduced
- All accounts must be analysed to assess that they conform to the "***least privilege principal***"
- All accounts must be used for one purpose
- All accounts must be linked to a person or business owner (person as well)
- Jobs that require storage of passwords must be secured (to not store)

Securing Data

- We are going to investigate in depth the issues around our credit card table seen earlier
- Remember we were able to
  - Find the table
  - Read the table
  - Decrypt the PAN easily
- Even these issues are only the "***tip of the iceberg***" though!
- Lets dig deeper

Securing Data - 2

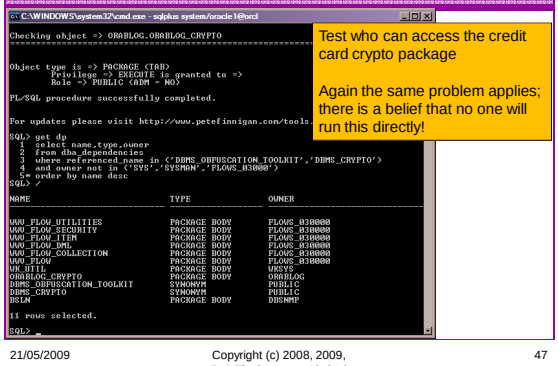


This problem is often seen. The developers think that everyone accesses the data via their application.

The encrypted data could be stolen and cracked off line

Or decrypted on-line by any user

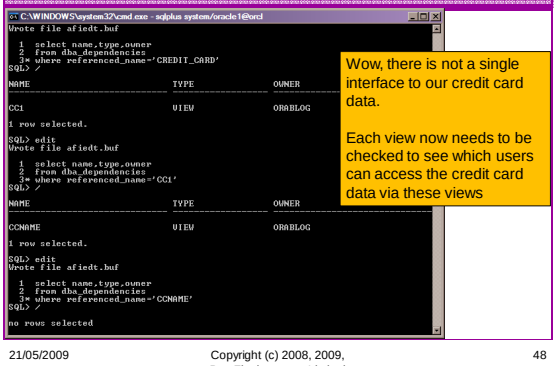
Securing Data - 3



Test who can access the credit card crypto package

Again the same problem applies; there is a belief that no one will run this directly!

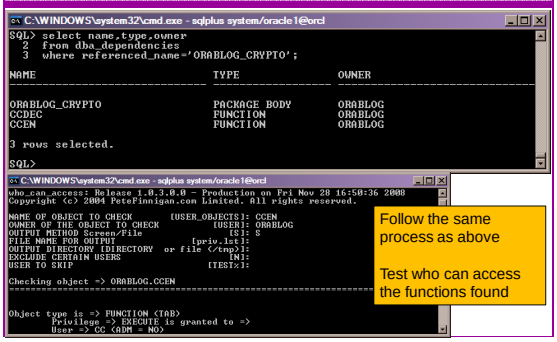
Securing Data - 4



Wow, there is not a single interface to our credit card data.

Each view now needs to be checked to see which users can access the credit card data via these views

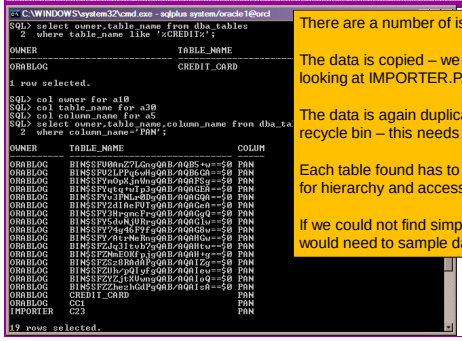
Securing Data - 5



Follow the same process as above

Test who can access the functions found

Securing Data - 6



There are a number of issues here

The data is copied – we can check by looking at IMPORTER.PAN

The data is again duplicated in the recycle bin – this needs to be handled

Each table found has to be checked for hierarchy and access

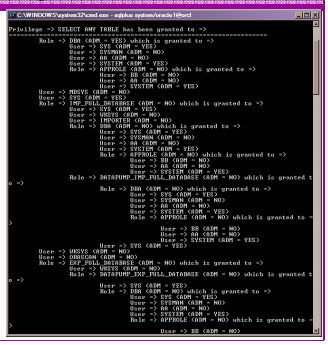
If we could not find simply as here we would need to sample data

Securing Data - 7

Sweeping privileges are still dangerous for our data – o7_dictionary_accessibility prevents some hacks but does not stop sweeping data access

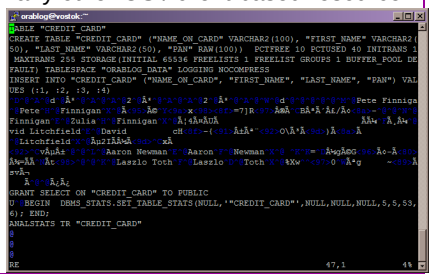
Remember there are other privileges; INSERT, UPDATE, DELETE…

Remember other privileges still that would allow data theft; TRIGGERS, EXECUTE PROCEDURE…

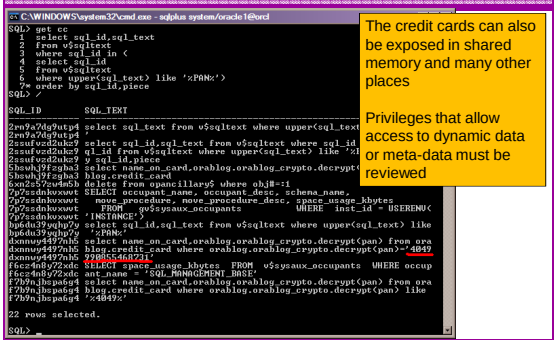


Securing Data - 8

- The credit card data can be exposed via export, list files or any other OS / client based resource



Securing Data - 9



The credit cards can also be exposed in shared memory and many other places

Privileges that allow access to dynamic data or meta-data must be reviewed

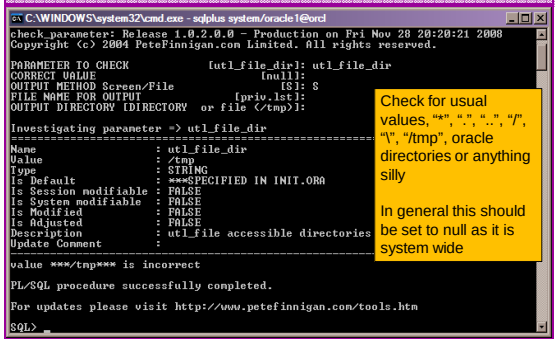
Securing Data - 10

- Securing data is not complex but we must take care of all access paths to the data
- We must consider the hierarchy
- We must consider sweeping privileges
- We must consider data leakage
- We must consider data replication
- There is more…unfortunately…
- In summary securing specific data ("***any data***") is first about knowing where that data is and who can access it and how it "***flows through the system***"

Access To The Server - 1

- We are now going to investigate in depth the issues around accessing the operating system
- We should now be ready for "***layers***" and "***hierarchy***" being evident in this investigation
- We will look at the common interfaces and common procedures

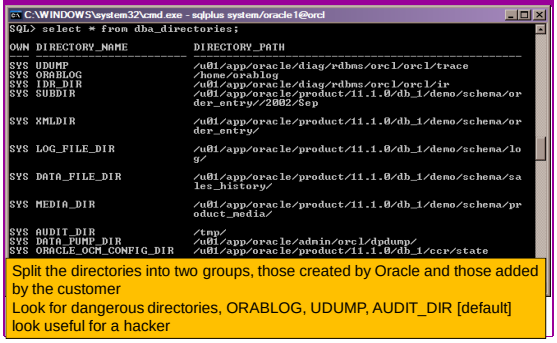
Access To The Server - 2



Check for usual values, "*", ".", "..", "/", "\", "/tmp", oracle directories or anything silly

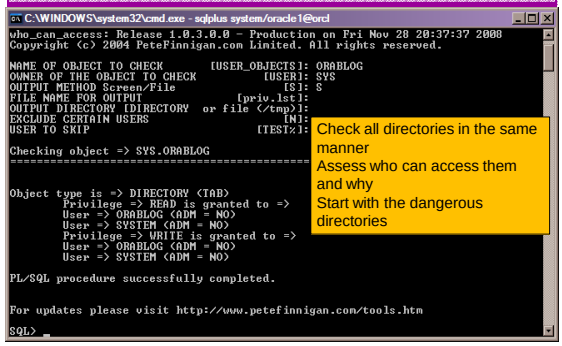
In general this should be set to null as it is system wide

Access To The Server - 3



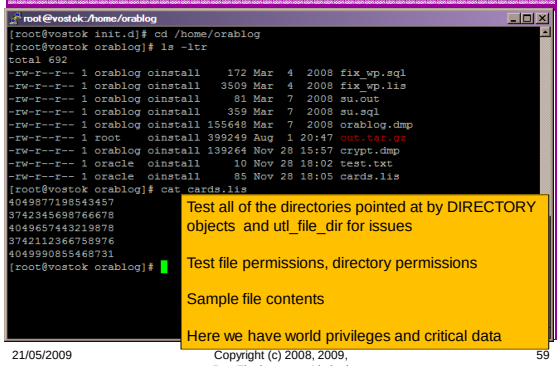
Split the directories into two groups, those created by Oracle and those added by the customer
Look for dangerous directories, ORABLOG, UDUMP, AUDIT_DIR [default] look useful for a hacker

Access To The Server - 4



Check all directories in the same manner
Assess who can access them and why
Start with the dangerous directories

Access To The Server - 5



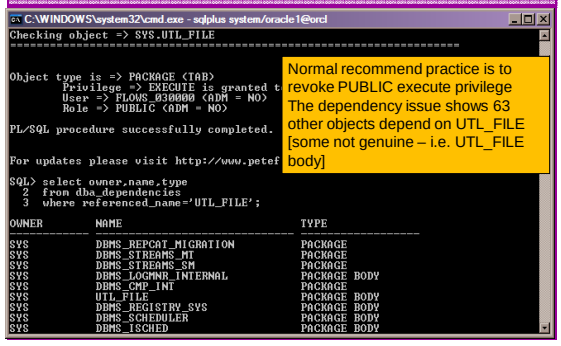
Test all of the directories pointed at by DIRECTORY objects and utl_file_dir for issues

Test file permissions, directory permissions

Sample file contents

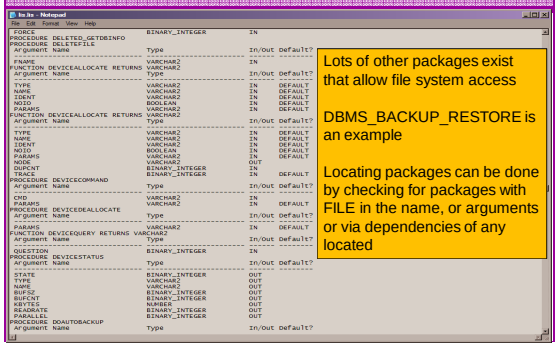
Here we have world privileges and critical data

Access To The Server - 6



Normal recommend practice is to revoke PUBLIC execute privilege
The dependency issue shows 63 other objects depend on UTL_FILE [some not genuine – i.e. UTL_FILE body]

Access To The Server - 7



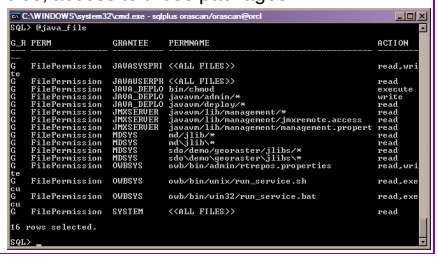
Lots of other packages exist that allow file system access

DBMS_BACKUP_RESTORE is an example

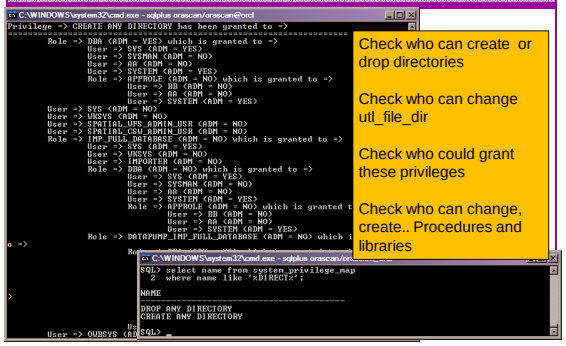
Locating packages can be done by checking for packages with FILE in the name, or arguments or via dependencies of any located

Access To The Server - 8

- Java – find file access permissions
- Locate all packages that use the privileges, check dependencies, access to those packages…



Access To The Server - 9



Check who can create or drop directories

Check who can change utl_file_dir

Check who could grant these privileges

Check who can change, create.. Procedures and libraries

Access To The Server - 10

- Securing access to the operating system is not complex but as with the data analysis there are many components, layers, hierarchy and duplication in paths
- We must understand all interfaces to the operating system
- We must understand all API's exposing these interfaces
- We must understand the privileges that allow access to the operating system
- A pattern is emerging in terms of components we must secure in Oracle

Layers, Hierarchy, Complexity

- Each of the three examples has
  - Layers of complexity
  - Multiple requirements for one area - Users
  - Multiple paths to data
  - Multiple copies of data
  - Multiple pieces of the puzzle involved with operating system objects
  - Multiple paths to the operating system
- See the pattern now?

Looking Back And Forward

- As an example passwords are easy to audit but hard to fix
- As an example user privileges are hard to audit fully and also hard to fix
- Investigating other areas? – use same ideas and techniques to ensure complete solutions
- Think about all components – use simple tools

Conclusions

- There are a few important lessons we must learn to secure data held in an Oracle database
  - We must secure the "**data**" not the software (quite obviously we MUST secure the software to achieve "**data**" security)
  - We must start with the "**data**" not the software
  - We must understand who/how/why/when "**data**" could be stolen
  - This may involve traditional downloadable exploits, it may not!

Conclusions (2)

- Oracle security is not rocket science
- Oracle security is complex though because we must consider "**where**" the "**data**" is and "**who**" can access it and "**how**"
- Looking for problems is often much easier than the solutions – remember passwords
- Often there are "**layers**" and "**duplication**"
- Careful detailed work is needed

PeteFinnigan.com Limited

Oracle Security Expertise

Any Questions?

PeteFinnigan.com Limited

Oracle Security Expertise

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