The Right Method To Secure An Oracle Database

By

Pete Finnigan

Updated Monday, 12th October 2009
Why Am I Qualified To Speak

- [http://www.petefinnigan.com](http://www.petefinnigan.com)
- First “Oracle security” blog.
- Specialists in researching and securing Oracle databases providing consultancy and training Database scanner software authors and vendors.
- Published many papers, regular speaker (UK, USA, Slovenia, Norway, Iceland, Finland and more).
- Member of the Oak Table Network.
Quick Quiz!

• How many people here know “where” their key data is held?
• How many people here understand exactly “who” can see or “modify” key data?
• How many people here understand the true “privilege model” employed to protect “key data”? 
Agenda

• Hardening databases by checklist
• Problems with checklists
• “The right method”
• Data flow
• Privilege/access assessment
• conclusions
Why We Need Security

• The target is often data not the “DBA” role
• The exploits we see on the internet work but stealing data is much more “real” and easy
• It is easy to steal, 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
Traditional Approach

• Hardening by checklist – good idea?
• A number of them available
  – SANS Step-by-step guide
  – SANS S.C.O.R.E.
  – CIS benchmark
  – DoD Stig
  – IT Governance book
  – Oracle’s own checklist

15/10/2009

Copyright (c) 2009 PeteFinnigan.com Limited
Problems With Checklists

• Not many checklists exist for Oracle databases
• Most are from same initial source or are very similar
• Some structure there but not good enough
  – “tip based rather than method based”
• Lists don’t focus on securing the data
• Difficult to implement for a large number of databases
• CIS for instance has 158 pages
Time “vs” Clever

- Time solution
  - Could spend man years on even a single database
  - Finding solutions for each issue is not as simple as applying what it says in the document
- Clever solution
  - Technical solutions need to be specified
  - Onion based approach is good
  - Basic hardening in parallel
Examples Of Problems

• Two examples:
  1) Check 3.0.2 in CIS states “all files in $ORACLE_HOME/bin directory must have privileges of 0755 or less – fine - but the solution states “chmod 0755 $ORACLE_HOME/bin/*” – is it a good idea?
  2) Solutions are not as simple as indicated. For instance fixing a weak password should also include, fix the password, management, hard coded passwords, audit, policy….
## Checklists And PII Data

### Search of the CIS benchmark - There is some mention of data BUT it is not focused

<table>
<thead>
<tr>
<th>Item #</th>
<th>Configuration Item</th>
<th>Action / Recommended Parameters</th>
<th>Rationale/Remediation</th>
<th>Windows</th>
<th>Unix</th>
<th>Level &amp; Score Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.25</td>
<td>Encryption</td>
<td>Tablespace Encryption</td>
<td>Rationale: When a table contains a large number of columns of sensitive data, it can be beneficial to encrypt an entire tablespace rather than columns. Remediation: Use tablespace encryption. Audit: None.</td>
<td>✓</td>
<td>✓</td>
<td>2 N</td>
</tr>
<tr>
<td>5.26</td>
<td>Radiuskey</td>
<td>Verify and set permissions on radius.key file</td>
<td>Rationale: File permissions must be restricted to the owner of the Oracle software and dba group. Ensure proper permissions are set on $ORACLE_HOME/network/security/radius.key. Remediation: chmod 400 $ORACLE_HOME/network/security/radius.key. Audit: ls -al $ORACLE_HOME/network/security/radius.key</td>
<td></td>
<td>✓</td>
<td>1 S</td>
</tr>
<tr>
<td>5.27</td>
<td>sqnet ora</td>
<td>SSL_CERT_REVOCATION=required</td>
<td>Rationale: Ensure revocation is required client certificate authentication can pose a threat to the integrity of the SSL channel.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

15/10/2009  
Copyright (c) 2009 PeteFinnigan.com Limited
No special data mentioned at all in the SANS SCORE
The Right Method To Secure

- Start with “the data”
- Understand “data flow” and “access”
- Understand the problem of securing “your data”
- Hardening should be part of the solution **BUT** not **THE** solution
- Checklists do not mention “your” data
Complex But Simple Solutions Needed

- Overarching solutions are needed
- Remove all types of access from the data
- Ensure only those who should see, can see the data
- Unfortunately it’s not that simple as there are:
  - Many paths to the data
  - Many copies of data
  - Data stored or in transit that is accessible
  - Data copied outside of the database
Understand Architecture

Identify each type of person and a sample account for each.
Data Access Models

Other Access to Data
OS files, SQL Text, Redo, Archive Flashback, backups, datafiles, ...

Privileges

View

API

View

API

Database

Data Table (Copy)

Sweeping Privileges

Privileges

Privileges

Copyright (c) 2009 PeteFinnigan.com Limited
Data Access Is Not “Flat”

- Data model is not flat – remove the blinkers
- Access rights are also not flat
- Data is often replicated
  - In other tables – in interfaces – flexfields …
  - Indexes
  - Shared memory
  - Data files
  - Operating system
  - Many more…
### How / Who

- The data must be identified (found?)
- The access paths must be found
- The “people” – real people identified
- Map to these to database user accounts
- Assess who can access data and how
- Only now can we hope to secure data
Securing Data

• We are going to investigate in depth the issues around a simple credit card table

• We need to
  – find the credit card details table
  – Find duplicate copies of credit card data
  – Assess who can access all of the data
  – Look for other places the data exists
  – More…

• 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.

Look for the credit cards.
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!

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>OWNER</th>
</tr>
</thead>
<tbody>
<tr>
<td>WWW_FLOW_UTILITIES</td>
<td>PACKAGE BODY</td>
<td>FLOWS_030000</td>
</tr>
<tr>
<td>WWW_FLOW_SECURITY</td>
<td>PACKAGE BODY</td>
<td>FLOWS_030000</td>
</tr>
<tr>
<td>WWW_FLOW_ITEM</td>
<td>PACKAGE BODY</td>
<td>FLOWS_030000</td>
</tr>
<tr>
<td>WWW_FLOW_DML</td>
<td>PACKAGE BODY</td>
<td>FLOWS_030000</td>
</tr>
<tr>
<td>WWW_FLOW_COLLECTION</td>
<td>PACKAGE BODY</td>
<td>FLOWS_030000</td>
</tr>
<tr>
<td>WWW_FLOW</td>
<td>PACKAGE BODY</td>
<td>FLOWS_030000</td>
</tr>
<tr>
<td>W2_UTIL</td>
<td>PACKAGE BODY</td>
<td>W2SYS</td>
</tr>
<tr>
<td>ORAEBLOG_CRYPTO</td>
<td>PACKAGE BODY</td>
<td>ORAEBLOG</td>
</tr>
<tr>
<td>DBMS_OBFUSCATION_TOOLKIT</td>
<td>SYNONYM</td>
<td>PUBLIC</td>
</tr>
<tr>
<td>DBMS_CRYPTO</td>
<td>SYNONYM</td>
<td>PUBLIC</td>
</tr>
<tr>
<td>DBLN</td>
<td>PACKAGE BODY</td>
<td>DBSNMP</td>
</tr>
</tbody>
</table>

11 rows selected.
Securing Data

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.
Follow the same process as above

Test who can access the functions found
Securing Data

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 using simple ideas as here we would need to sample data or use specific algorithms
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

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

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 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"
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](http://www.petefinnigan.com/oracle_password_cracker.htm)

This is called the “Access Issue”
User Password Analysis

- 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 also the implications in terms of management and administration
- This is an example of just one issue
Rounding Up

- A simple picture is built of all access to the key data
- All users are assessed and mapped to the data access
- Solutions are very specific but generally
  - Reduce default accounts
  - Reduce access to data
  - Remove duplicate privileges
  - Simplify privilege and access models
  - Generalise
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
• Oracle security is complex though because we must consider “where” the “data” is and “who” can access it and “how”
• Often there are “layers” and “duplication”
• Careful detailed work is often needed
Quick Quiz – Again!

• How many people know “where” their key data is held?
• How many people understand exactly “who” can see or “modify” key data?
• How many people understand the true “privilege model” employed to protect “key data”?
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