UKOUG Conference, December 5th 2007

Oracle Forensics

By

Pete Finnigan

Written Friday, 19th October 2007
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)
• What is forensics and Oracle forensics?
• In real terms what does it mean?
• What information is out there
• Are there any tools?
• The issues – audit on, audit off and more
• Where to find forensic data
• Finding evidence – correlating data
• Plan for forensic analysis – make it easy
What is Forensics?

fo·ren·sics

n. (used with a sing. verb)

1. The art or study of formal debate; argumentation.

2. The use of science and technology to investigate and establish facts in criminal or civil courts of law.

What is Oracle Forensics?

• Oracle forensics is the process by which someone (an auditor?) tries to determine when / how / why (and by who) something happened by gathering correlated and incriminating evidence.

• Oracle forensics often occurs when as an auditor I am called in to help a client discover how a breach occurred and hopefully some clue as to who did it.

• These techniques are often championed through the need to do this with no audit trail, no archive logs or worse – the success rates are dependant on how fast we can look and what is available.

• If this leads to criminal proceedings the evidence must be gathered without distortion or change to the system.
What Information Is Out There?

• 2 books – (note: neither book is available as I write this):

• Pete Finnigan (2003) - Detecting SQL Injection in Oracle - [http://www.securityfocus.com/infocus/1714](http://www.securityfocus.com/infocus/1714) some forensics ideas - mining redo, sql extraction, trace, audit


• Pete Finnigan (2004) – Oracle Forensics module – SANS training
What Information Is Out There? (2)


Are There Any Tools?

• Yes and no
• There are no specific Oracle forensics tools – Yet.
  – David is developing FEDS
• Most of the evidence can be extracted with existing tools
  – Simple SQL Queries
  – Database dumps
  – More exotic options, BBED, ORA-Dude, AUL/MyDUL
  – Connect to the SGA to read the SQL in the SGA
The Issues

- The problem when you want to investigate why is that inevitably there is no audit trail
- If audit is on, then use it. Beware of testing for altered audit trails
- If no audit and archive log is on use the changes captured
- If no audit, no archive logs then there is still hope
- Mining blocks and redo is time and error prone
- Detecting “Select” statements is harder
Where To Find Forensic Data

- TNS listener log
- Many types of trace files
- Sqlnet logs (server and clients)
- Sysdba audit logs
- Datafiles for deleted data
- Redo (and archive) logs
- SGA (v$sql etc)
- Apache access logs
Where To Find Forensic Data (2)

- v$db_object_cache
- Wrh$%% views
- Wri$ views
- Statspack views
- col_usage$
- Audit trails –
  - AUD$, FGA_LOG$
  - Application audit (who/when, triggers, other)
- Flashback, recycle bin
- More?
Looking For A Password Change

The disadvantage of the SGA is that a database restart flushes it, a shared pool flush will also remove evidence and also the data is very transient.

For a password change everything ran as SYS so other correlations are necessary to find the actual user who did it.

Views such as v$sql_bind_data and v$sql_bind_capture can sometimes reveal data.
Data Gathering From AUD$

The advantage of the audit trail is that historic data is present.
Audit trail Example

• If an audit trail exists then this can provide the best evidence
  – Check for SYS.AUD$ or core audit to OS
  – Check for SYS.FGA_LOG$
  – Check for Triggers and shadow tables
  – Test for who/when (E-Business Suite supports this)
• Don’t depend on audit though as it may have been altered!
• Detect possible data changes first
  – Look for gaps
  – Correlate the audit trail (time, rowid, session, access and change to the audit trail itself – audit on audit)
Audit Example 2

Beware of deleted rows Can you spot the issue?

<table>
<thead>
<tr>
<th>ROWID</th>
<th>USERID</th>
<th>ACTION#</th>
<th>OBJ$NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAAlUAABAABFKAA</td>
<td>SCOTT</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>AAAAlUAABAABFKAAC</td>
<td>X</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>AAAAlUAABAABFKAAD</td>
<td>SYSTEM</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>AAAAlUAABAABFKAEE</td>
<td>SYSTEM</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>AAAAlUAABAABFKAFF</td>
<td>SYSTEM</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>AAAAlUAABAABFKAAG</td>
<td>SYSTEM</td>
<td>43</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>AAAAlUAABAABFKAHH</td>
<td>X</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>AAAAlUAABAABFKAII</td>
<td>SYSTEM</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>AAAAlUAABAABFKAJJ</td>
<td>X</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>AAAAlUAABAABFKAJK</td>
<td>SYSTEM</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>AAAAlUAABAABFKAAL</td>
<td>X</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>AAAAlUAABAABFKAAN</td>
<td>SYSTEM</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>AAAAlUAABAABFKAAN</td>
<td>SYSTEM</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>AAAAlUAABAABFKAAD</td>
<td>SYSTEM</td>
<td>43</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>AAAAlUAABAABFKAAP</td>
<td>SYSTEM</td>
<td>7</td>
<td>AUD$</td>
</tr>
</tbody>
</table>

15 rows selected.
Using timestamps on the object you are investigating or in general across the database can be useful to detect change and also for correlation.
Correlation

- Use correlation in two ways
  - If you have one piece of evidence look for others with matching values (could be time, address, sql_hash, scn, xid …)
  - If you don’t know what to search for, i.e. you have been hacked but not sure how but know the time period; use the timestamp to locate all correlated evidence.
- Use timestamps on objects, redo (Log Mining) and more within the database
- Correlate time based evidence with external sources (oracle) such as listener.log, sql*net logs, sysdba trace
- Correlate user information with OS logs, client PC logs, firewalls, personal firewalls, web server logs
Great source of data.

Arup nanda has a great three part paper – Mining Information from the listener log –
http://www.dbazine.com/oracle/or-articles/nanda14
Tertiary Data – SYSDBA Audit

On *Nix boxes look in audit_file_dest for trace files.

Shows SYS audit if audit_sys_operations=TRUE

Connects are audited by default as sysdba
Deleted Data

- David introduced the idea of looking for deleted data in data blocks in his 6 part Oracle forensics series.
- This is not new as others more concerned with recovery, block internals, DUL like tools have found this years ago.
- The idea is being built into FEDS
- Beware:
  - This is unsupported – in terms of undefined results
  - The deleted data is transient
- Recycle bin and Flashback also good options (If available)
- As is Redo and archive logs (not transient) – again if available
Database Dumps

SQL> alter session set events 'immediate trace name library_cache level 10';
Session altered.
SQL>
Investigation Without Disturbance

• If a suspected breach has occurred
• Plan ahead
• Consider:
  – Can the results of the investigation be trusted
  – Altering the database or shutting down could remove evidence – e.g. shared pool is cleared
  – The investigation should not alter the data or create a large footprint in the database thereby changing the value of the investigation
Investigation Without Disturbance (2)

- Establish the server state – users, ports, files, dll’s, memory, system time etc
- Collect Oracle files – sysdba trace, archive logs, alert log, listener log, sqlnet logs, trace, copy data files (if possible)
- Grab the SQL from v$sql (direct SGA access is an option - http://www.petefinnigan.com/other.htm)
- Grab sys.aud$
- Grab AWR and statspack if available
- Analyse changes to users and roles and privileges
- Checksum the PL/SQL, Java, triggers, views
- Investigate
Log Miner

SQL> connect sys/change_on_install as sysdba
Connected.
SQL> select distinct member logfilename from v$logfile;

LOGFILENAME
-------------------------------
C:\ORACLE\ORADATA\ORA10GR2\REDO03.LOG
C:\ORACLE\ORADATA\ORA10GR2\REDO02.LOG
C:\ORACLE\ORADATA\ORA10GR2\REDO01.LOG

SQL> begin
  2  dbms_logmnr.add_logfile('C:\ORACLE\ORADATA\ORA10GR2\REDO03.LOG');
  3  dbms_logmnr.add_logfile('C:\ORACLE\ORADATA\ORA10GR2\REDO02.LOG');
  4  dbms_logmnr.add_logfile('C:\ORACLE\ORADATA\ORA10GR2\REDO01.LOG');
  5  end;
  6  /

SQL> begin
  2  dbms_logmnr.start_logmnr(options => dbms_logmnr.dict_from_online_catalog);
  3  end;
  4  /
Log Miner 2

```
SQL> edit aud

SQL> get aud
  1  select username,to_char(timestamp,'DD-MON-YYYY HH24:MI:SS') timestamp,
  2     seg_owner,operation/sql_undo
  3  from v$logmmr_contents
  4  where table_name='AUD$'
  5  and sql_redo like '%AAAAIuAABAAABFKKAAB%'
  6  /

USERNAME                TIMESTAMP
------------------------ ------------------------
SEG_OWNER                OPERATION
-----------------------------------------------
SQL_UNDO
-----------------------------------------------
29-NOV-2007 21:29:38

SYS
update "SYS"."AUD$" set "ACTION#" = '100', "RETURNCODE" = '0', "LOGOFF$READ" = NULL, "LOGOFF$PREAD" = NULL, "LOGOFF$WRITE" = NULL, "LOGOFF$DEAD" = NULL, "LOGOFF$TIME" = NULL, "SESSIONCPU" = NULL where "ACTION#" = '101' and "RETURNCODE" = '0' and "LOGOFF$READ" = '282' and "LOGOFF$PREAD" = '0' and "LOGOFF$WRITE" = '6' and "LOGOFF$DEAD" = '0' and "LOGOFF$TIME" = TO_DATE('29-NOV-07', 'DD-MON-RR')

USERNAME                TIMESTAMP
------------------------ ------------------------
SEG_OWNER                OPERATION
-----------------------------------------------
SQL_UNDO
-----------------------------------------------
and "SESSIONCPU" = '6' and ROWID = 'AAAAIuAABAAABFKKAAB';
```

09/12/2007

Copyright (c) 2007
PeteFinnigan.com Limited
Log Miner 3

```
SQL> col username for a8
SQL> col timestamp for a20
SQL> col seg_type_name for a8
SQL> col seg_name for a10
SQL> col sql_redo for a30 wrap
SQL> edit
Wrote file afiedt.buf

1  select username,to_char(timestamp,'DD-MON-YYYY HH24:MI:SS') timestamp,
  2     seg_type_name,seg_name,sq

3  from v$logmn
4  where operation='DDL'
5  and sql_redo like 'alter user%'
SQL> /

USERNAME    TIMESTAMP    SEG_TYPE SEG_NAME       SQL_REDO
SYSTEM  28-NOV-2007 21:02:20 USER

SQL> |
```
Build A Toolkit

- What can we build as toolkit?
- Mining blocks not ideal – time biased and not consistent – FEDS look promising BUT
- A Tool kit should / Could be methodology include:
  - A plan of actions
  - OS commands to gather files
  - SQL commands to gather details from the database
  - Dump commands
## Conclusions

- Looked at what are forensics and what are Oracle forensics?
- Looked at what information is out there
- Looked at the issues – audit on, audit off and more
- Looked at where to find forensic data
- Looked at finding evidence – correlating data
- Oracle Forensics is a new and exciting area and very current due to recent data losses
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