PeteFinnigan.com Limited

Oracle Security Expertise

OUG Scotland, DBA SIG, April 30th 2008

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 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

- 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.

Cited from: The American Heritage® Dictionary of the English Language, Fourth Edition copyright ©2000 by Houghton Mifflin Company. Updated in 2003. Published by Houghton Mifflin Company. All rights reserved.

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?

- Pete Finnigan (2003) - Detecting SQL Injection in Oracle - http://www.securityfocus.com/infocus/1714 some forensics ideas - mining redo, sql extraction, trace, audit
- Pete Finnigan (2004) – Oracle Forensics module – SANS training
- Arup nanda (2005) – Mining for clues - http://www.oracle.com/technology/oramag/oracle/05-jul/o45dba.html
- Alex Gorbachev (2006) – Log Miner for forensics - http://www.pythian.com/blogs/269/oracle-logminer-helps-investigate-security-issues
- Paul Wright (2006/7) – Number of papers – http://www.oracleforensics.com + his SANS GSOC paper http://www.sans.org/reading_room/whitepapers/application/

What Information Is Out There? (2)

- David Litchfield (2007) – 6 part paper - http://www.databasesecurity.com/
- Alejandro Vargas (2007) – Log Miner 10g Implementation Example - http://static7.userland.com/oracle/gems/alejandroVargas/logminerexample.pdf
- David Litchfield (2007) – Blackhat paper - http://www.databasesecurity.com/dbsec/forensics.ppt
- 2 books – (note: one of the books is not available as I write this):
  - (2007) - Oracle Forensics: Paul Wright – ISBN-10-0977671526
  - (2008) - Oracle Forensics Analysis Using the Forensic Examiners Database Scalpel (FEDS) Tool - ISBN-10: 047019118X – *Title has changed recently*

Are There Any Tools?

- Yes and no
- There are no specific Oracle forensics tools – Yet.
  - David is developing FEDS (or whatever it will eventually be called)
- Most of the evidence can be extracted with existing tools
  - Existing OS forensics tools can be used
  - 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 (*This is one of the key tenets of forensics – validity and chain of custody*)
- 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 consuming and error prone
- Detecting “Select” statements is harder

Where To Find Forensic Data

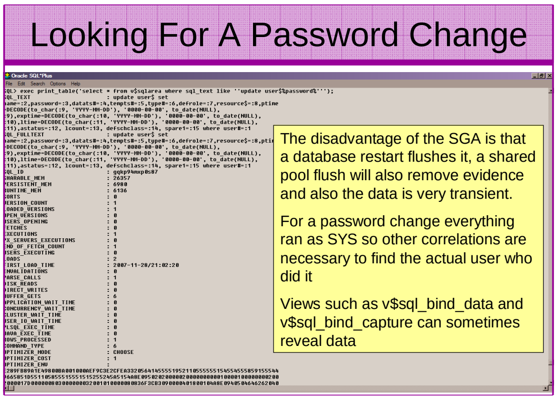
- TNS listener log
- Many types of trace files
- Sqlnet logs (server and clients)
- Sysdba audit logs
- Datafiles for deleted data
- Redo (and archive) logs
- SGA (v$sql etc)
- Apache access logs

Oracle is great at leaving a whole swathe of evidence!!

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



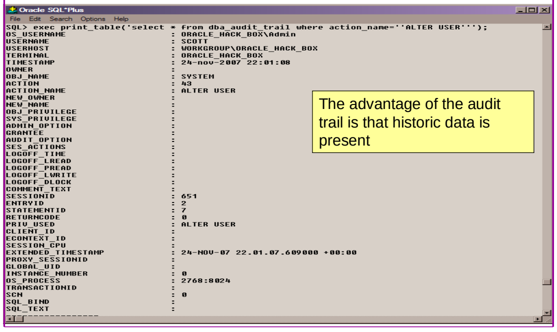
```
SQL> exec print_table('select * from v$sqlarea where sql_text like ''update user$%password%''');
SQL_TEXT                      : update user$ set
name=:2,password=:3,datats#=:4,tempts#=:5,type#=:6,defrole=:7,resource$=:8,ptime
=DECODE(to_char(:9, 'YYYY-MM-DD'), '0000-00-00', to_date(NULL),
:9),exptime=DECODE(to_char(:10, 'YYYY-MM-DD'), '0000-00-00', to_date(NULL),
:10),ltime=DECODE(to_char(:11, 'YYYY-MM-DD'), '0000-00-00', to_date(NULL),
:11),astatus=:12, lcount=:13, defschclass=:14, spare1=:15 where user#=:1
SQL_FULLTEXT                  : update user$ set
name=:2,password=:3,datats#=:4,tempts#=:5,type#=:6,defrole=:7,resource$=:8,ptim
=DECODE(to_char(:9, 'YYYY-MM-DD'), '0000-00-00', to_date(NULL),
:9),exptime=DECODE(to_char(:10, 'YYYY-MM-DD'), '0000-00-00', to_date(NULL),
:10),ltime=DECODE(to_char(:11, 'YYYY-MM-DD'), '0000-00-00', to_date(NULL),
:11),astatus=:12, lcount=:13, defschclass=:14, spare1=:15 where user#=:1
SQL_ID                        : gqkp94mxp0s87
SHARABLE_MEM                  : 26357
PERSISTENT_MEM                : 6980
RUNTIME_MEM                   : 6136
SORTS                         : 0
VERSION_COUNT                 : 1
LOADED_VERSIONS               : 1
OPEN_VERSIONS                 : 0
USERS_OPENING                 : 0
FETCHES                       : 0
EXECUTIONS                    : 1
PX_SERVERS_EXECUTIONS         : 0
END_OF_FETCH_COUNT            : 1
USERS_EXECUTING               : 0
LOADS                         : 2
FIRST_LOAD_TIME               : 2007-11-28/21:02:20
INVALIDATIONS                 : 0
PARSE_CALLS                   : 1
DISK_READS                    : 0
DIRECT_WRITES                 : 0
BUFFER_GETS                   : 6
APPLICATION_WAIT_TIME         : 0
CONCURRENCY_WAIT_TIME         : 0
CLUSTER_WAIT_TIME             : 0
USER_IO_WAIT_TIME             : 0
PLSQL_EXEC_TIME               : 0
JAVA_EXEC_TIME                : 0
ROWS_PROCESSED                : 1
COMMAND_TYPE                  : 6
OPTIMIZER_MODE                : CHOOSE
OPTIMIZER_COST                : 1
OPTIMIZER_ENV                 :
E289FB89A1E49800BA001000AEF9C3E2CFEA332056414555519521105555551545545558591555 44
9665851D5511058555155515152552 45A5154A8E0950202000002000000000010000100000000200
2000017D000000803000000032001010000080836F3CB30900004018001 04A8E0940504646262040
```

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$



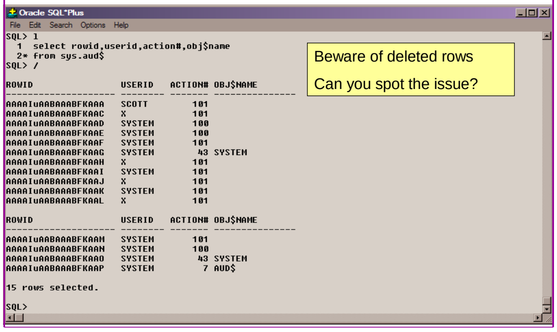
```
SQL> exec print_table('select * from dba_audit_trail where action_name=''ALTER USER''');
OS_USERNAME                   : ORACLE_HACK_BOX\Admin
USERNAME                      : SCOTT
USERHOST                      : WORKGROUP\ORACLE_HACK_BOX
TERMINAL                      : ORACLE_HACK_BOX
TIMESTAMP                     : 24-nov-2007 22:01:08
OWNER                         :
OBJ_NAME                      : SYSTEM
ACTION                        : 43
ACTION_NAME                   : ALTER USER
NEW_OWNER                     :
NEW_NAME                      :
OBJ_PRIVILEGE                 :
SYS_PRIVILEGE                 :
ADMIN_OPTION                  :
GRANTEE                       :
AUDIT_OPTION                  :
SES_ACTIONS                   :
LOGOFF_TIME                   :
LOGOFF_LREAD                  :
LOGOFF_PREAD                  :
LOGOFF_LWRITE                 :
LOGOFF_DLOCK                  :
COMMENT_TEXT                  :
SESSIONID                     : 651
ENTRYID                       : 2
STATEMENTID                   : 7
RETURNCODE                    : 0
PRIV_USED                     : ALTER USER
CLIENT_ID                     :
ECONTEXT_ID                   :
SESSION_CPU                   :
EXTENDED_TIMESTAMP            : 24-NOV-07 22.01.07.609000 +00:00
PROXY_SESSIONID               :
GLOBAL_UID                    :
INSTANCE_NUMBER               : 0
OS_PROCESS                    : 2768:8024
TRANSACTIONID                 :
SCN                           : 0
SQL_BIND                      :
SQL_TEXT                      :
-----------------
```

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! (you need to prove it is valid)
- 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

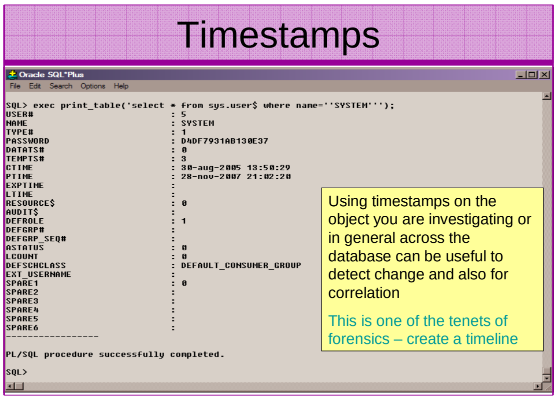


```
SQL> l
  1  select rowid,userid,action#,obj$name
  2* from sys.aud$
SQL> /

ROWID              USERID   ACTION# OBJ$NAME
------------------ -------- ------- ---------------
AAAAIuAABAAABFKAAA SCOTT        101
AAAAIuAABAAABFKAAC X            101
AAAAIuAABAAABFKAAD SYSTEM       100
AAAAIuAABAAABFKAAE SYSTEM       100
AAAAIuAABAAABFKAAF SYSTEM       101
AAAAIuAABAAABFKAAG SYSTEM        43 SYSTEM
AAAAIuAABAAABFKAAH X            101
AAAAIuAABAAABFKAAI SYSTEM       101
AAAAIuAABAAABFKAAJ X            101
AAAAIuAABAAABFKAAK SYSTEM       101
AAAAIuAABAAABFKAAL X            101

ROWID              USERID   ACTION# OBJ$NAME
------------------ -------- ------- ---------------
AAAAIuAABAAABFKAAM SYSTEM       101
AAAAIuAABAAABFKAAN SYSTEM       100
AAAAIuAABAAABFKAAO SYSTEM        43 SYSTEM
AAAAIuAABAAABFKAAP SYSTEM         7 AUD$

15 rows selected.

SQL>
```

Beware of deleted rows

Can you spot the issue?

Timestamps



```
SQL> exec print_table('select * from sys.user$ where name=''SYSTEM''');
USER#                         : 5
NAME                          : SYSTEM
TYPE#                         : 1
PASSWORD                      : D4DF7931AB130E37
DATATS#                       : 0
TEMPTS#                       : 3
CTIME                         : 30-aug-2005 13:50:29
PTIME                         : 28-nov-2007 21:02:20
EXPTIME                       :
LTIME                         :
RESOURCE$                     : 0
AUDIT$                        :
DEFROLE                       : 1
DEFGRP#                       :
DEFGRP_SEQ#                   :
ASTATUS                       : 0
LCOUNT                        : 0
DEFSCHCLASS                   : DEFAULT_CONSUMER_GROUP
EXT_USERNAME                  :
SPARE1                        : 0
SPARE2                        :
SPARE3                        :
SPARE4                        :
SPARE5                        :
SPARE6                        :
-----------------

PL/SQL procedure successfully completed.

SQL>
```

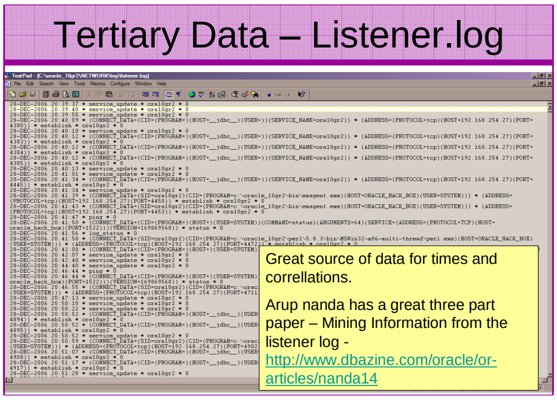
Using timestamps on the object you are investigating or in general across the database can be useful to detect change and also for correlation

This is one of the tenets of forensics – create a timeline

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, OS evidence and more
- Correlate user information with OS logs, client PC logs, firewalls, personal firewalls, web server logs

Tertiary Data – Listener.log



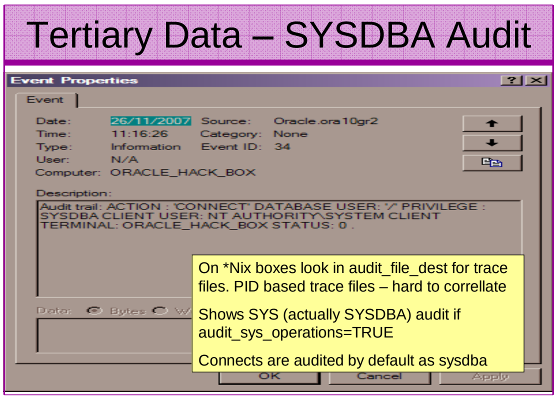
TextPad - [C:\oracle_10gr2\NETWORK\log\listener.log]
File Edit Search View Tools Macros Configure Window Help

```
28-DEC-2006 20:39:37 * service_update * ora10gr2 * 0
28-DEC-2006 20:39:40 * service_update * ora10gr2 * 0
28-DEC-2006 20:39:55 * service_update * ora10gr2 * 0
28-DEC-2006 20:40:09 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=__jdbc__)(USER=))(SERVICE_NAME=ora10gr2)) * (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.254.27)(PORT=
4380)) * establish * ora10gr2 * 0
28-DEC-2006 20:40:10 * service_update * ora10gr2 * 0
28-DEC-2006 20:40:12 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=__jdbc__)(USER=))(SERVICE_NAME=ora10gr2)) * (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.254.27)(PORT=
4383)) * establish * ora10gr2 * 0
28-DEC-2006 20:40:12 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=__jdbc__)(USER=))(SERVICE_NAME=ora10gr2)) * (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.254.27)(PORT=
4384)) * establish * ora10gr2 * 0
28-DEC-2006 20:40:12 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=__jdbc__)(USER=))(SERVICE_NAME=ora10gr2)) * (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.254.27)(PORT=
4385)) * establish * ora10gr2 * 0
28-DEC-2006 20:40:13 * service_update * ora10gr2 * 0
28-DEC-2006 20:41:01 * service_update * ora10gr2 * 0
28-DEC-2006 20:41:34 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=__jdbc__)(USER=))(SERVICE_NAME=ora10gr2)) * (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.254.27)(PORT=
4445)) * establish * ora10gr2 * 0
28-DEC-2006 20:41:34 * service_update * ora10gr2 * 0
28-DEC-2006 20:41:39 * (CONNECT_DATA=(SID=ora10gr2)(CID=(PROGRAM=c:\oracle_10gr2\bin\emagent.exe)(HOST=ORACLE_HACK_BOX)(USER=SYSTEM))) * (ADDRESS=
(PROTOCOL=tcp)(HOST=192.168.254.27)(PORT=4450)) * establish * ora10gr2 * 0
28-DEC-2006 20:41:43 * (CONNECT_DATA=(SID=ora10gr2)(CID=(PROGRAM=c:\oracle_10gr2\bin\emagent.exe)(HOST=ORACLE_HACK_BOX)(USER=SYSTEM))) * (ADDRESS=
(PROTOCOL=tcp)(HOST=192.168.254.27)(PORT=4453)) * establish * ora10gr2 * 0
28-DEC-2006 20:41:47 * ping * 0
28-DEC-2006 20:41:50 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=)(USER=SYSTEM))(COMMAND=status)(ARGUMENTS=64)(SERVICE=(ADDRESS=(PROTOCOL=TCP)(HOST=
oracle_hack_box)(PORT=1522)))(VERSION=169869568)) * status * 0
28-DEC-2006 20:41:56 * log_status * 0
28-DEC-2006 20:41:58 * (CONNECT_DATA=(SID=ora10gr2)(CID=(PROGRAM=c:\oracle_10gr2\perl\5.8.3\bin\MSWin32-x86-multi-thread\perl.exe)(HOST=ORACLE_HACK_BOX)
(USER=SYSTEM))) * (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.254.27)(PORT=4472)) * establish * ora10gr2 * 0
28-DEC-2006 20:42:00 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=)(USER=SYSTEM)
28-DEC-2006 20:42:07 * service_update * ora10gr2 * 0
28-DEC-2006 20:42:40 * service_update * ora10gr2 * 0
28-DEC-2006 20:44:40 * service_update * ora10gr2 * 0
28-DEC-2006 20:46:44 * ping * 0
28-DEC-2006 20:46:44 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=)(USER=SYSTEM)
oracle_hack_box)(PORT=1522)))(VERSION=169869568)) * status * 0
28-DEC-2006 20:46:55 * (CONNECT_DATA=(SID=ora10gr2)(CID=(PROGRAM=c:\orac
(USER=SYSTEM))) * (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.254.27)(PORT=4711
28-DEC-2006 20:47:13 * service_update * ora10gr2 * 0
28-DEC-2006 20:50:10 * service_update * ora10gr2 * 0
28-DEC-2006 20:50:22 * service_update * ora10gr2 * 0
28-DEC-2006 20:50:52 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=__jdbc__)(USER
4894)) * establish * ora10gr2 * 0
28-DEC-2006 20:50:52 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=__jdbc__)(USER
4895)) * establish * ora10gr2 * 0
28-DEC-2006 20:50:52 * service_update * ora10gr2 * 0
28-DEC-2006 20:50:59 * (CONNECT_DATA=(SID=ora10gr2)(CID=(PROGRAM=c:\orac
(USER=SYSTEM))) * (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.254.27)(PORT=4902
28-DEC-2006 20:51:07 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=__jdbc__)(USER
4908)) * establish * ora10gr2 * 0
28-DEC-2006 20:51:17 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=__jdbc__)(USER
4917)) * establish * ora10gr2 * 0
28-DEC-2006 20:51:28 * service_update * ora10gr2 * 0
```

Great source of data for times and correllations.

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



Date: 26/11/2007 Source: Oracle.ora10gr2
Time: 11:16:26 Category: None
Type: Information Event ID: 34
User: N/A
Computer: ORACLE_HACK_BOX

Description:
Audit trail: ACTION : 'CONNECT' DATABASE USER: '/' PRIVILEGE : SYSDBA CLIENT USER: NT AUTHORITY\SYSTEM CLIENT TERMINAL: ORACLE_HACK_BOX STATUS: 0 .

Data: ● Bytes ○ W

On *Nix boxes look in audit_file_dest for trace files. PID based trace files – hard to correllate

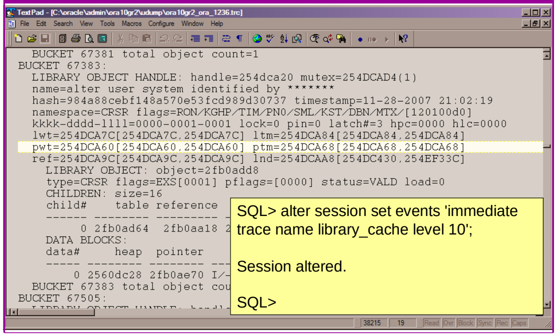
Shows SYS (actually SYSDBA) audit if audit_sys_operations=TRUE

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 (we believe)
- Beware:
  - This is unsupported – in terms of undefined results
  - The deleted data is transient
- Recycle bin and Flashback also good options (If available)
- As are Redo and archive logs (not transient) – again if available
- Tools like BBED could be used or hex editors

Database Dumps


TextPad - [C:\oracle\admin\ora10gr2\udump\ora10gr2_ora_1236.trc]
File Edit Search View Tools Macros Configure Window Help
38215 19 Read Ovr Block Sync Rec Caps

```
  BUCKET 67381 total object count=1
BUCKET 67383:
  LIBRARY OBJECT HANDLE: handle=254dca20 mutex=254DCAD4(1)
  name=alter user system identified by *******
  hash=984a88cebf148a570e53fcd989d30737 timestamp=11-28-2007 21:02:19
  namespace=CRSR flags=RON/KGHP/TIM/PN0/SML/KST/DBN/MTX/[120100d0]
  kkkk-dddd-llll=0000-0001-0001 lock=0 pin=0 latch#=3 hpc=0000 hlc=0000
  lwt=254DCA7C[254DCA7C,254DCA7C] ltm=254DCA84[254DCA84,254DCA84]
  pwt=254DCA60[254DCA60,254DCA60] ptm=254DCA68[254DCA68,254DCA68]
  ref=254DCA9C[254DCA9C,254DCA9C] lnd=254DCAA8[254DC430,254EF33C]
    LIBRARY OBJECT: object=2fb0add8
    type=CRSR flags=EXS[0001] pflags=[0000] status=VALD load=0
    CHILDREN: size=16
    child#    table reference
    ------ -------- ---------
         0 2fb0ad64  2fb0aa18 2
    DATA BLOCKS:
    data#     heap  pointer
    ----- -------- -------- ---
        0 2560dc28 2fb0ae70 I/-
  BUCKET 67383 total object cou
BUCKET 67505:
```

```
SQL> alter session set events 'immediate trace name library_cache level 10';

Session altered.

SQL>
```

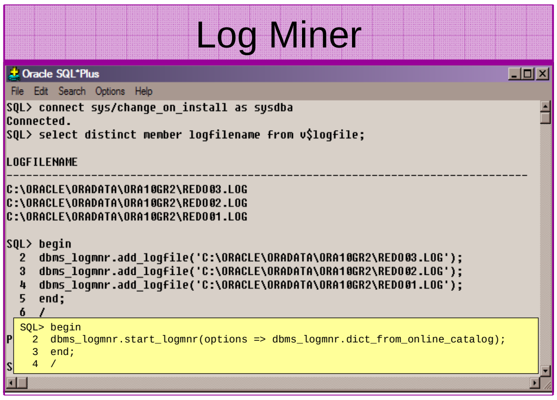
Investigation Without Disturbance

- If a suspected breach has occurred
- Plan ahead – i.e. don't blunder in
- 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 foot print 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$sql (direct SGA access is an option - http://www.petefinnigan.com/other.htm)
- Grab SYS.AUD$ and SYS.FGA_LOG$
- Grab AWR and statspack if available
- Analyse changes to users and roles and privileges
- Checksum the PL/SQL, Java, triggers, views
- Investigate

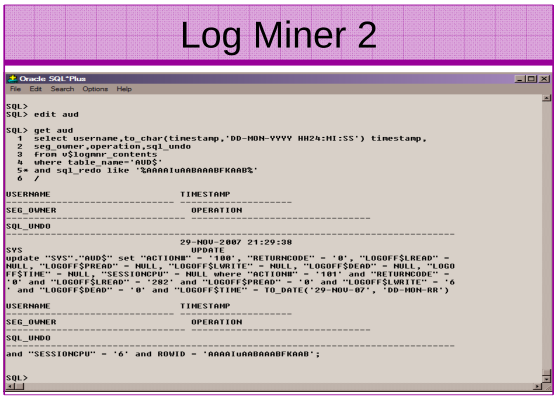
Log Miner



```
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  /
P
S
```

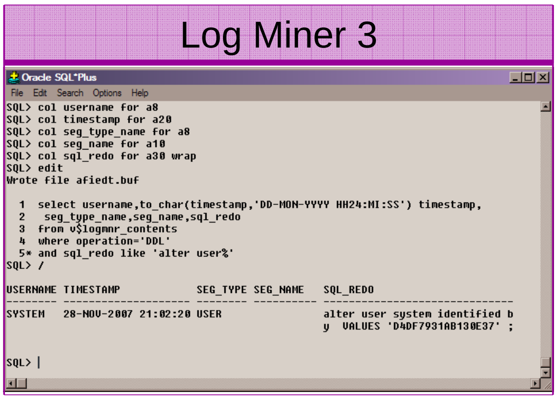
```
SQL> begin
  2  dbms_logmnr.start_logmnr(options => dbms_logmnr.dict_from_online_catalog);
  3  end;
  4  /
```

Log Miner 2



```
SQL>
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$logmnr_contents
  4  where table_name='AUD$'
  5* and sql_redo like '%AAAAIuAABAAABFKAAB%'
  6  /

USERNAME                       TIMESTAMP
------------------------------ --------------------
SEG_OWNER                        OPERATION
-------------------------------- --------------------------------
SQL_UNDO
--------------------------------------------------------------------------------
                               29-NOV-2007 21:29:38
SYS                              UPDATE
update "SYS"."AUD$" set "ACTION#" = '100', "RETURNCODE" = '0', "LOGOFF$LREAD" =
NULL, "LOGOFF$PREAD" = NULL, "LOGOFF$LWRITE" = NULL, "LOGOFF$DEAD" = NULL, "LOGO
FF$TIME" = NULL, "SESSIONCPU" = NULL where "ACTION#" = '101' and "RETURNCODE" =
'0' and "LOGOFF$LREAD" = '282' and "LOGOFF$PREAD" = '0' and "LOGOFF$LWRITE" = '6
' and "LOGOFF$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 = 'AAAAIuAABAAABFKAAB';


SQL>
```

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,sql_redo
  3  from v$logmnr_contents
  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                alter user system identified b
                                                  y  VALUES 'D4DF7931AB130E37' ;


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

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