Oracle Forensics

Alexander Kornbrust

HACKTIVITY

Agenda

- Introduction
- Current Status
- Forensic Problems
- Available Tools
- New Approach
- Typical Patterns
- Summary

HACKTIVITY

About Red-Database-Security

- Founded 2004 in Germany
- Dedicated to Oracle Security
- Consulting / Training / Software
- More than 1500 security vulnerabilities found in Oracle products
- More than 2000 Oracle databases audited in 2011

Introduction

- More and more databases affected by attacks
- Database forensic is still an exotic/academic topic
- No easy to use tools available.
- Collected data is difficult to analyse

➔ This presentation will show new approaches which will make the analysis easier

Current Status – Books & Documents

- Oracle Forensics from Paul M. Wright out of stock (used copies 230 USD)
  , new books coming soon
  http://www.amazon.com/gp/product/0977671526/sr=8-2/qid=1315500507/ref=olp_product_details?ie=UTF8&me=&qid=1315500507&sr=8-2&seller=
- Oracle Forensics Series from David Litchfield
  http://www.databasesecurity.com/oracle-forensics.htm
- Several smaller documents

Available Tools for Forensic

- Logminer (free, Oracle)
- Data Unloader (most commercial, e.g. qDUL from Qualea)
- Verity Data Block Examiner, cadfile, … (free, v3rity Ltd.)
- McAfee Security Scanner for Databases (commercial, Analysis)

Traces

Different kind of traces could be used

- Files on OS level
- Results from OS Commands at OS level
- Volatile tables –only available if DB is up and running
- Temporary tables – content automatically by Oracle after a while
- Permanent tables

Find Traces (files)

- Listener.log
- Trace files
- Incident Response Files
- Alert.logs
- Data files
- SYSDBA Audit Logs
- Redo/Archive Logs
- Unix History Files
- …

Find Traces (Tables/Views)

- GV$* (Volatile, use GV$* instead of V$ to be Oracle cluster (RAC) compliant)
- WRH$* (Temporary)
- Audit Views
- USER$
- MON_MOD$ (Temporary)
- COL_USAGE$ (Temporary)
- Recycle-Bin
- …

Oracle Forensic Problems

- Still requires a deep knowledge of database architecture/design
- Requires good SQL know how (Outer-Joins are mandatory in many Selects queries, e.g. join audit&user tables)
- Requires a strong knowledge of the Oracle (and the application) repository
- Requires a strong knowledge about typical database attacks (what can be found where)
- Little to less tool support

Typical Approach for DB Forensics

- Collect traces from the file system and database
  - OS: copy files
  - DB: spool the output from SQL statements to a spool file to preserve the evidence[1]
- Copy the collected files to the examiner PC
- Analyze the collected evidence

➔ Difficult to analyze because the data type, format, dependencies is lost.

➔ Just a big text file. No query language.

[1] http://www.databasesecurity.com/dbsec/LiveResponse.pdf

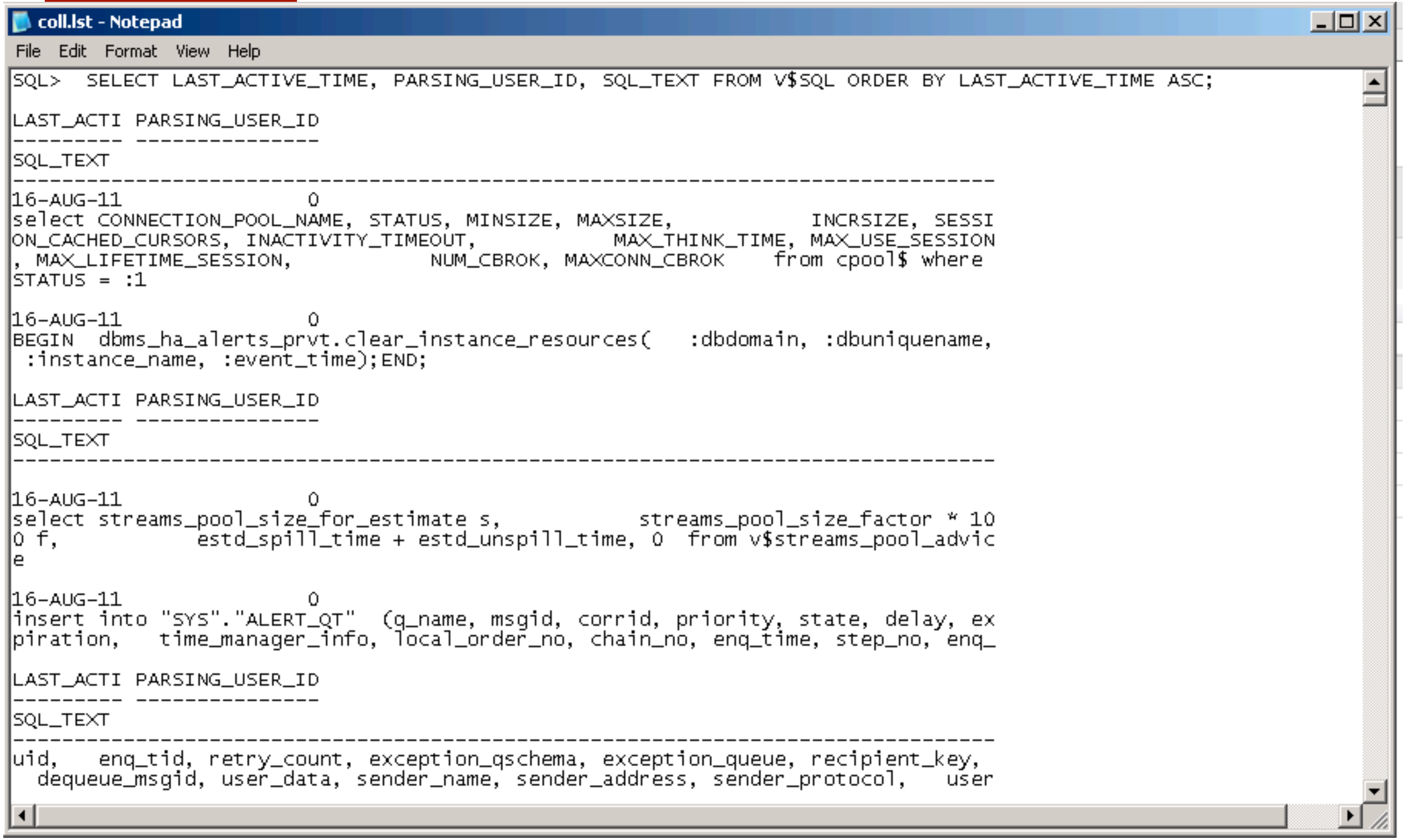
Current Approach

Victim DB

```
Sqlplus / as sysdba

SQL> spool coll.lst

SQL> SELECT LAST_ACTIVE_TIME, PARSING_USER_ID, SQL_TEXT FROM V$SQL
ORDER BY LAST_ACTIVE_TIME ASC;

SQL> SELECT ST.PARSING_SCHEMA_ID, TX.SQL_TEXT FROM WRH$_SQLSTAT ST,
WRH$_SQLTEXT TX WHERE TX.SNAP_ID = ST.SNAP_ID;

SQL> SELECT * FROM AUD$;
SQL> SELECT USER_ID, SESSION_ID, SAMPLE_TIME FROM SYS.WRH
$_ACTIVE_SESSION_HISTORY ;
SQL> SELECT SID, USER#, USERNAME, TERMINAL, OSUSER, PROGRAM,
LOGON_TIME FROM V$SESSION;

SQL> SELECT USER#, NAME, ASTATUS, PASSWORD, CTIME, PTIME, LTIME FROM
SYS.USER$ WHERE TYPE#=1;
```

Examiner PC

```
Notepad coll.lst
```



```
SQL>  SELECT LAST_ACTIVE_TIME, PARSING_USER_ID, SQL_TEXT FROM V$SQL ORDER BY LAST_ACTIVE_TIME ASC;

LAST_ACTI PARSING_USER_ID
--------- ---------------
SQL_TEXT
--------------------------------------------------------------------------------
16-AUG-11               0
select CONNECTION_POOL_NAME, STATUS, MINSIZE, MAXSIZE,           INCRSIZE, SESSI
ON_CACHED_CURSORS, INACTIVITY_TIMEOUT,           MAX_THINK_TIME, MAX_USE_SESSION
, MAX_LIFETIME_SESSION,           NUM_CBROK, MAXCONN_CBROK    from cpool$ where
STATUS = :1

16-AUG-11               0
BEGIN  dbms_ha_alerts_prvt.clear_instance_resources(   :dbdomain, :dbuniquename,
 :instance_name, :event_time);END;

LAST_ACTI PARSING_USER_ID
--------- ---------------
SQL_TEXT
--------------------------------------------------------------------------------

16-AUG-11               0
select streams_pool_size_for_estimate s,           streams_pool_size_factor * 10
0 f,           estd_spill_time + estd_unspill_time, 0  from v$streams_pool_advic
e

16-AUG-11               0
insert into "SYS"."ALERT_QT"  (q_name, msgid, corrid, priority, state, delay, ex
piration,   time_manager_info, local_order_no, chain_no, enq_time, step_no, enq_

LAST_ACTI PARSING_USER_ID
--------- ---------------
SQL_TEXT
--------------------------------------------------------------------------------
uid,   enq_tid, retry_count, exception_qschema, exception_queue, recipient_key,
  dequeue_msgid, user_data, sender_name, sender_address, sender_protocol,   user
```

Advanced Approach

- Same data collection approach but use external tables instead of unstructured text files
- An Oracle external table allows to preserve the entire table data including binary data, data types, …. in a binary file

➔ Requires Oracle 10.2 or higher

➔ Analysis will be much easier

➔ Much faster than normal spooling

➔ Joins and lookups between the difference collected information is still possible by using the renamed external tables

[1] http://www.databasesecurity.com/dbsec/LiveResponse.pdf

Advanced Approach

1.) Victim DB

```
• UNIX:
    • As root:collect_unix_artifacts_as_root.sh
    • As Oracle: collect_unix_artifacts_as_oracle.sh
• Oracle:
    • As SYS: collect_db_artifact_as_sys.sql
```

2.) Transfer Data to Examiner PC (+ burn to DVD)

3.) Examiner PC

```
* Create objects (prepare_examiner_db_case001.sql)
```

4.) Analyse

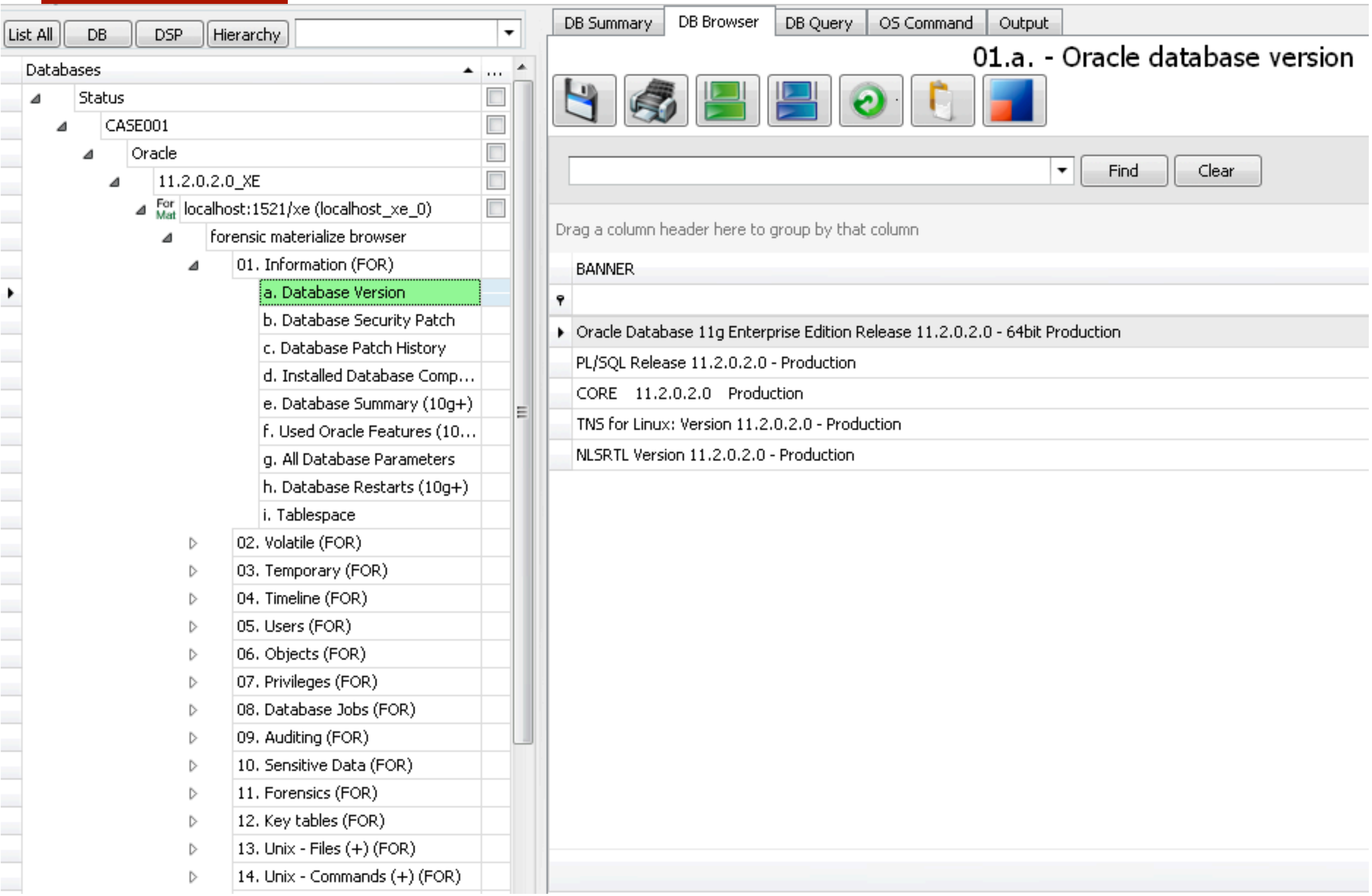
Advanced Approach II (Tables/Views)

Victim DB

```
CREATE TABLE forensicmat.ext_gvversion  ORGANIZATION
EXTERNAL( TYPE ORACLE_DATAPUMP DEFAULT DIRECTORY
data_unload_dir LOCATION ( 'ext_gvversion.dmp' ))

AS select * from gv$version;
```

Examiner PC

```
 CREATE TABLE "EXT_GVVERSION” ("INST_ID" NUMBER,
 "BANNER" VARCHAR2(80))
   ORGANIZATION EXTERNAL
     ( TYPE ORACLE_DATAPUMP
       DEFAULT DIRECTORY for_ora_ext_tables1
LOCATION
        ( 'ext_gvversion.dmp’ ) );
```



Databases

- Status
  - CASE001
    - Oracle
      - 11.2.0.2.0_XE
        - localhost:1521/xe (localhost_xe_0)
          - forensic materialize browser
            - 01. Information (FOR)
              - a. Database Version
              - b. Database Security Patch
              - c. Database Patch History
              - d. Installed Database Comp…
              - e. Database Summary (10g+)
              - f. Used Oracle Features (10…
              - g. All Database Parameters
              - h. Database Restarts (10g+)
              - i. Tablespace
            - 02. Volatile (FOR)
            - 03. Temporary (FOR)
            - 04. Timeline (FOR)
            - 05. Users (FOR)
            - 06. Objects (FOR)
            - 07. Privileges (FOR)
            - 08. Database Jobs (FOR)
            - 09. Auditing (FOR)
            - 10. Sensitive Data (FOR)
            - 11. Forensics (FOR)
            - 12. Key tables (FOR)
            - 13. Unix - Files (+) (FOR)
            - 14. Unix - Commands (+) (FOR)

01.a. - Oracle database version

BANNER
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - 64bit Production
PL/SQL Release 11.2.0.2.0 - Production
CORE 11.2.0.2.0 Production
TNS for Linux: Version 11.2.0.2.0 - Production
NLSRTL Version 11.2.0.2.0 - Production

Advanced Approach (OS Commands)

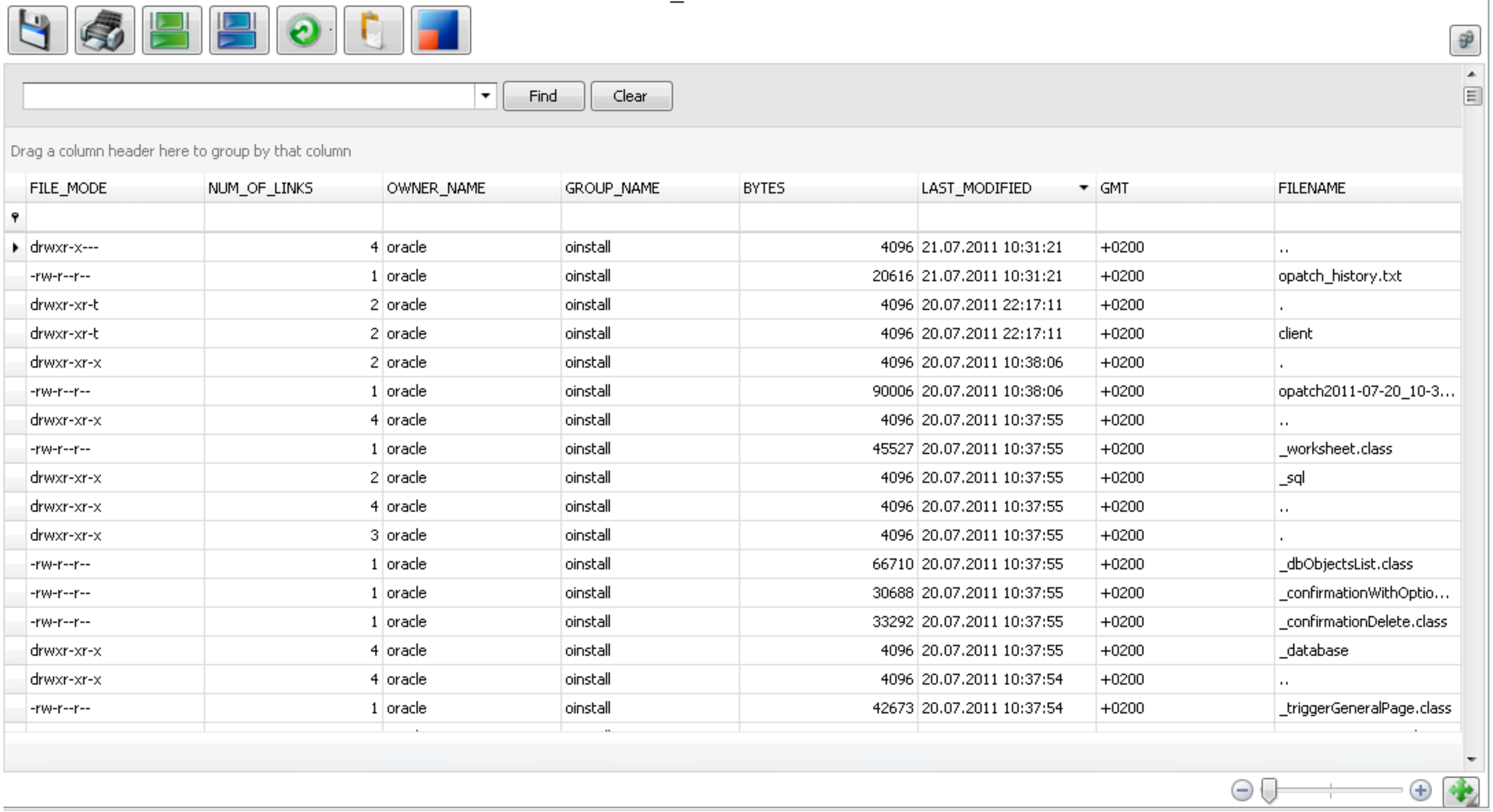
Victim DB

```
ls -laR --full-time $ORACLE_HOME | tee -a >$FORDIR/
oracle/commands/all_files.txt
```

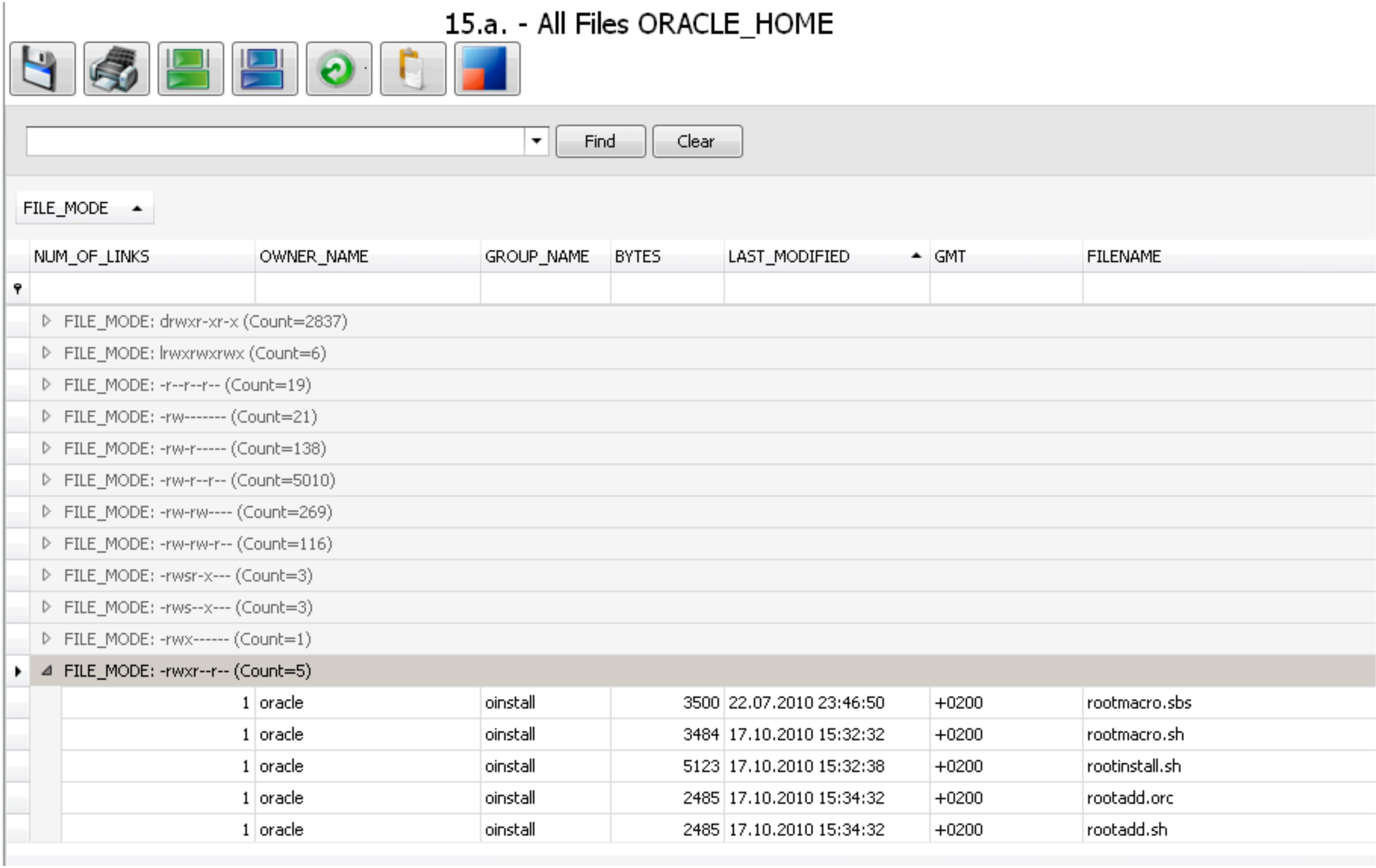
Examiner PC

```
CREATE TABLE ext_all_files
(file_mode varchar2(11), num_of_links number,
owner_name varchar2(32), group_name varchar2(32),
bytes number, file_last_mod_date varchar2(10),
file_last_mod_time varchar2(20), gmt varchar2(6),
filename varchar2(256) )
    ORGANIZATION EXTERNAL
  (   TYPE oracle_loader
   DEFAULT DIRECTORY for_ora_commands1
   ACCESS PARAMETERS
   (RECORDS DELIMITED BY NEWLINE
    FIELDS TERMINATED BY ' '
    MISSING FIELD VALUES ARE NULL  )
   LOCATION ('all_files.txt')   )
   PARALLEL 5 REJECT LIMIT UNLIMITED;
```

15.a. - All Files ORACLE_HOME



FILE_MODE	NUM_OF_LINKS	OWNER_NAME	GROUP_NAME	BYTES	LAST_MODIFIED	GMT	FILENAME
drwxr-x---	4	oracle	oinstall	4096	21.07.2011 10:31:21	+0200	..
-rw-r--r--	1	oracle	oinstall	20616	21.07.2011 10:31:21	+0200	opatch_history.txt
drwxr-xr-t	2	oracle	oinstall	4096	20.07.2011 22:17:11	+0200	.
drwxr-xr-t	2	oracle	oinstall	4096	20.07.2011 22:17:11	+0200	client
drwxr-xr-x	2	oracle	oinstall	4096	20.07.2011 10:38:06	+0200	.
-rw-r--r--	1	oracle	oinstall	90006	20.07.2011 10:38:06	+0200	opatch2011-07-20_10-3...
drwxr-xr-x	4	oracle	oinstall	4096	20.07.2011 10:37:55	+0200	..
-rw-r--r--	1	oracle	oinstall	45527	20.07.2011 10:37:55	+0200	_worksheet.class
drwxr-xr-x	2	oracle	oinstall	4096	20.07.2011 10:37:55	+0200	_sql
drwxr-xr-x	4	oracle	oinstall	4096	20.07.2011 10:37:55	+0200	..
drwxr-xr-x	3	oracle	oinstall	4096	20.07.2011 10:37:55	+0200	.
-rw-r--r--	1	oracle	oinstall	66710	20.07.2011 10:37:55	+0200	_dbObjectsList.class
-rw-r--r--	1	oracle	oinstall	30688	20.07.2011 10:37:55	+0200	_confirmationWithOptio...
-rw-r--r--	1	oracle	oinstall	33292	20.07.2011 10:37:55	+0200	_confirmationDelete.class
drwxr-xr-x	4	oracle	oinstall	4096	20.07.2011 10:37:55	+0200	_database
drwxr-xr-x	4	oracle	oinstall	4096	20.07.2011 10:37:54	+0200	..
-rw-r--r--	1	oracle	oinstall	42673	20.07.2011 10:37:54	+0200	_triggerGeneralPage.class



15.a. - All Files ORACLE_HOME

NUM_OF_LINKS	OWNER_NAME	GROUP_NAME	BYTES	LAST_MODIFIED ▲	GMT	FILENAME
▷ FILE_MODE: drwxr-xr-x (Count=2837)						
▷ FILE_MODE: lrwxrwxrwx (Count=6)						
▷ FILE_MODE: -r--r--r-- (Count=19)						
▷ FILE_MODE: -rw------- (Count=21)						
▷ FILE_MODE: -rw-r----- (Count=138)						
▷ FILE_MODE: -rw-r--r-- (Count=5010)						
▷ FILE_MODE: -rw-rw---- (Count=269)						
▷ FILE_MODE: -rw-rw-r-- (Count=116)						
▷ FILE_MODE: -rwsr-x--- (Count=3)						
▷ FILE_MODE: -rws--x--- (Count=3)						
▷ FILE_MODE: -rwx------ (Count=1)						
◢ FILE_MODE: -rwxr--r-- (Count=5)						
1	oracle	oinstall	3500	22.07.2010 23:46:50	+0200	rootmacro.sbs
1	oracle	oinstall	3484	17.10.2010 15:32:32	+0200	rootmacro.sh
1	oracle	oinstall	5123	17.10.2010 15:32:38	+0200	rootinstall.sh
1	oracle	oinstall	2485	17.10.2010 15:34:32	+0200	rootadd.orc
1	oracle	oinstall	2485	17.10.2010 15:34:32	+0200	rootadd.sh

Advanced Approach (OS Files)

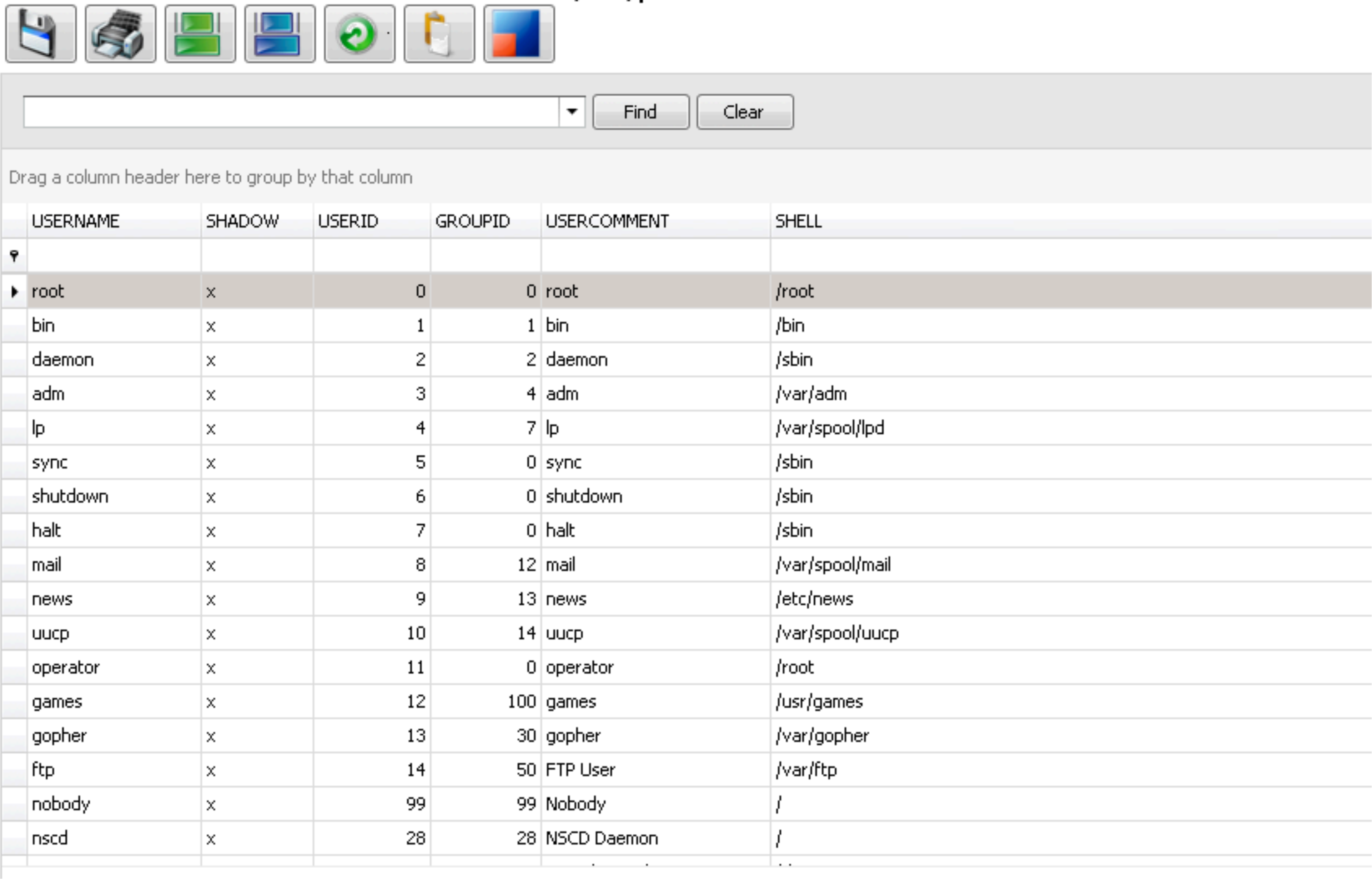
Victim DB

```
cp -p -v /etc/passwd $FORDIR/unix/files/passwd.txt
```

Examiner PC

```
CREATE TABLE ext_etc_passwd
(username varchar2(32), shadow varchar2(32),
userid number, groupid number,
usercomment varchar2(128), shell varchar2(128) )
    ORGANIZATION EXTERNAL
  (   TYPE oracle_loader
   DEFAULT DIRECTORY for_unix_files1
   ACCESS PARAMETERS
   (RECORDS DELIMITED BY NEWLINE
    FIELDS TERMINATED BY ':'
    MISSING FIELD VALUES ARE NULL   )
   LOCATION ('passwd.txt')   )
   PARALLEL 5   REJECT LIMIT UNLIMITED;
```

13.b. - /etc/passwd



USERNAME	SHADOW	USERID	GROUPID	USERCOMMENT	SHELL
root	x	0	0	root	/root
bin	x	1	1	bin	/bin
daemon	x	2	2	daemon	/sbin
adm	x	3	4	adm	/var/adm
lp	x	4	7	lp	/var/spool/lpd
sync	x	5	0	sync	/sbin
shutdown	x	6	0	shutdown	/sbin
halt	x	7	0	halt	/sbin
mail	x	8	12	mail	/var/spool/mail
news	x	9	13	news	/etc/news
uucp	x	10	14	uucp	/var/spool/uucp
operator	x	11	0	operator	/root
games	x	12	100	games	/usr/games
gopher	x	13	30	gopher	/var/gopher
ftp	x	14	50	FTP User	/var/ftp
nobody	x	99	99	Nobody	/
nscd	x	28	28	NSCD Daemon	/

Timeline Creation

- A timeline can be helpful during the analysis of forensic data
- Data from different source is displayed together
- Easy to implement

Timeline Creation

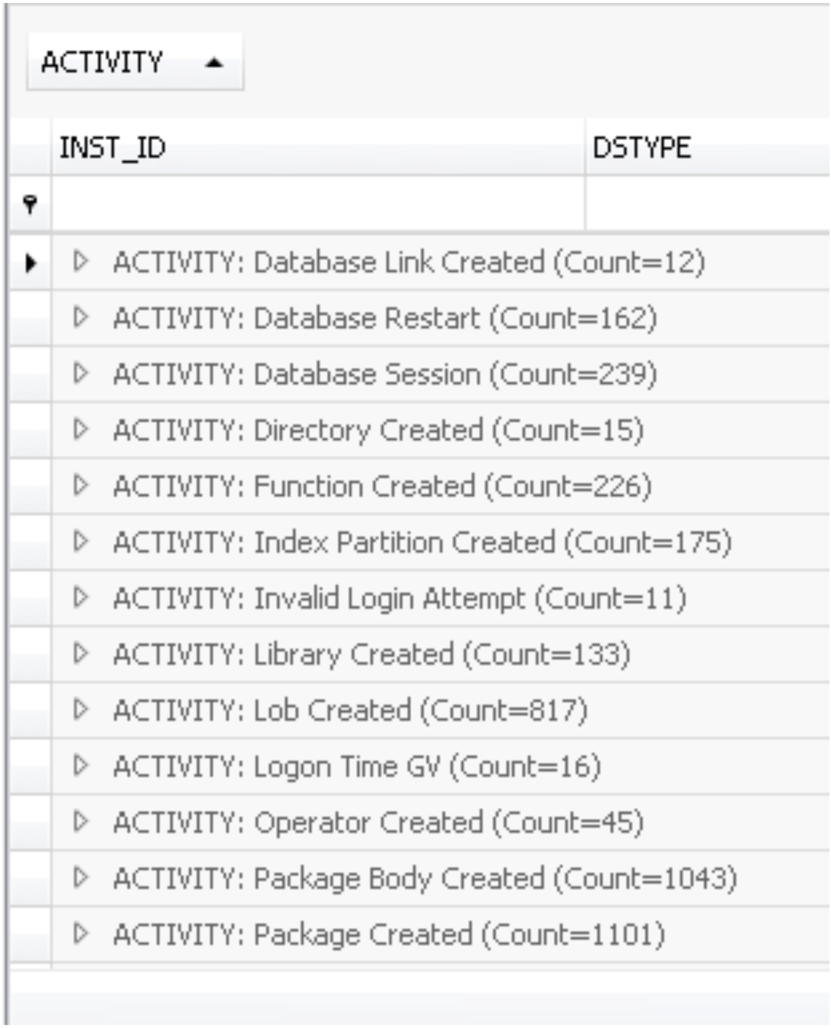
- Every information with a timestamp (e.g. User locking) will be a separate row and unified with the UNION command
  - SYS.USER$ contains different timestamps
    - CTIME – User created
    - PTIME – Password changed
    - LTIME – User locked
- A single row in SYS.USER$ will become 3 lines in the timeline table/view
- Additional information must be added from different tables/view (e.g. DB startup, auditing, ...)

Timeline Creation

```
select 0 as inst_id, 'DBA' as dstype, 'DBA_USERS' as datasource, created as
timest, 'User Created' as activity, 'CREATED' as timestamp_name,username as
detail1, username as username, null as serial#, null as session_id  from
ext_dba_users
union all
select 0 as inst_id, 'DBA' as dstype,'DBA_USERS' as datasource, lock_date as
timest, 'User Locked' as activity, 'LOCK_DATE' as timestamp_name,username as
detail1, username as username, null as serial#, null as session_id  from
ext_dba_users where lock_date is not null
union all
select 0 as inst_id, 'DBA' as dstype,'DBA_OBJECTS' as datasource, created as
timest, 'Table Created' as activity, 'CREATED' as timestamp_name,owner||'.'||
object_name as  detail1, owner as username, null as serial#, null as session_id
from ext_dba_objects where object_type='TABLE‘
union all
select 0 as inst_id, 'DBA' as dstype,'DBA_OBJECTS' as datasource, created as
timest, 'View Created' as activity, 'CREATED' as timestamp_name,owner||'.'||
object_name as  detail1, owner as username, null as serial#, null as session_id
from ext_dba_objects where object_type='VIEW‘

...
```

Demo - Forensic

Timeline



ACTIVITY ▲

INST_ID DSTYPE

▷ ACTIVITY: Database Link Created (Count=12)

▷ ACTIVITY: Database Restart (Count=162)

▷ ACTIVITY: Database Session (Count=239)

▷ ACTIVITY: Directory Created (Count=15)

▷ ACTIVITY: Function Created (Count=226)

▷ ACTIVITY: Index Partition Created (Count=175)

▷ ACTIVITY: Invalid Login Attempt (Count=11)

▷ ACTIVITY: Library Created (Count=133)

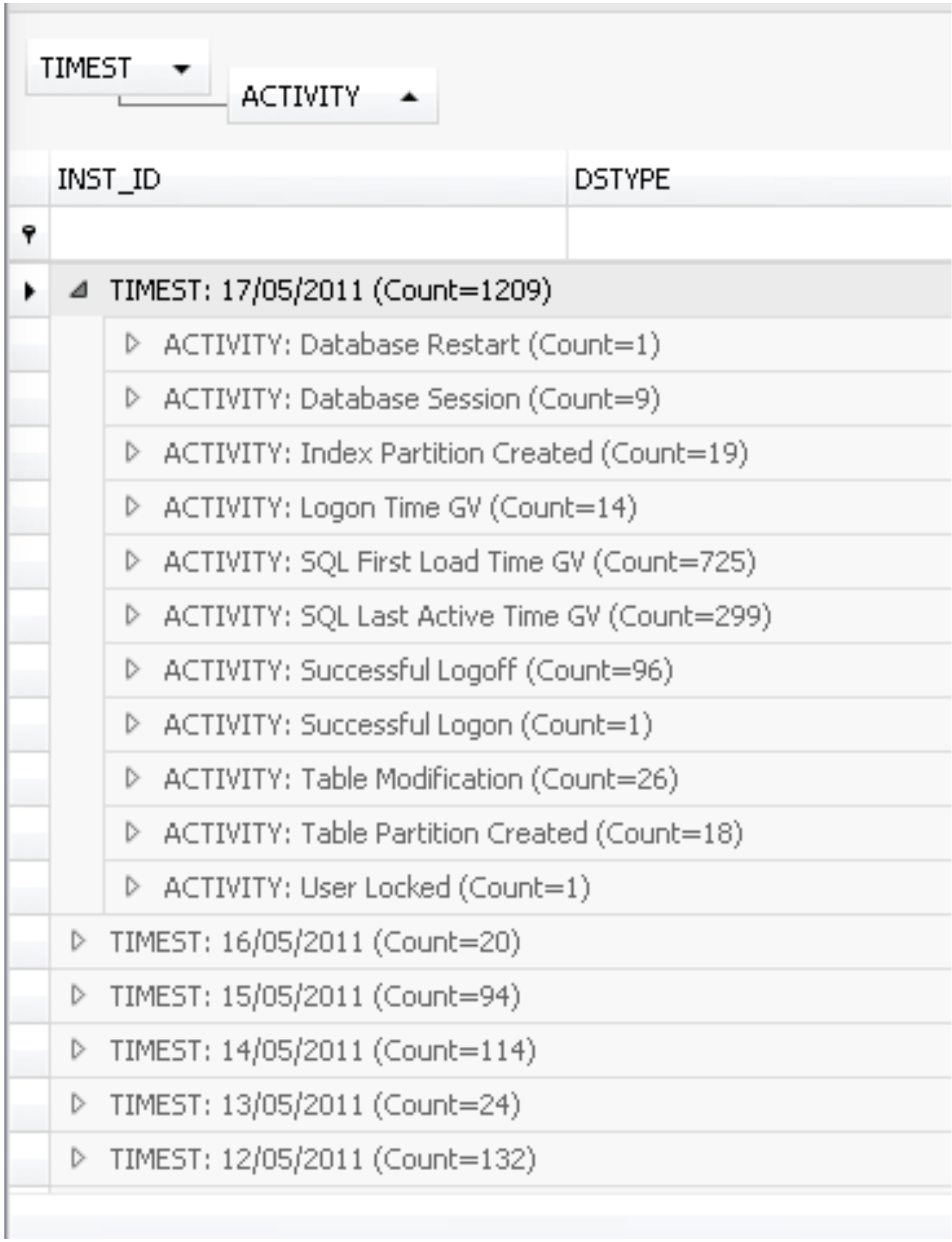
▷ ACTIVITY: Lob Created (Count=817)

▷ ACTIVITY: Logon Time GV (Count=16)

▷ ACTIVITY: Operator Created (Count=45)

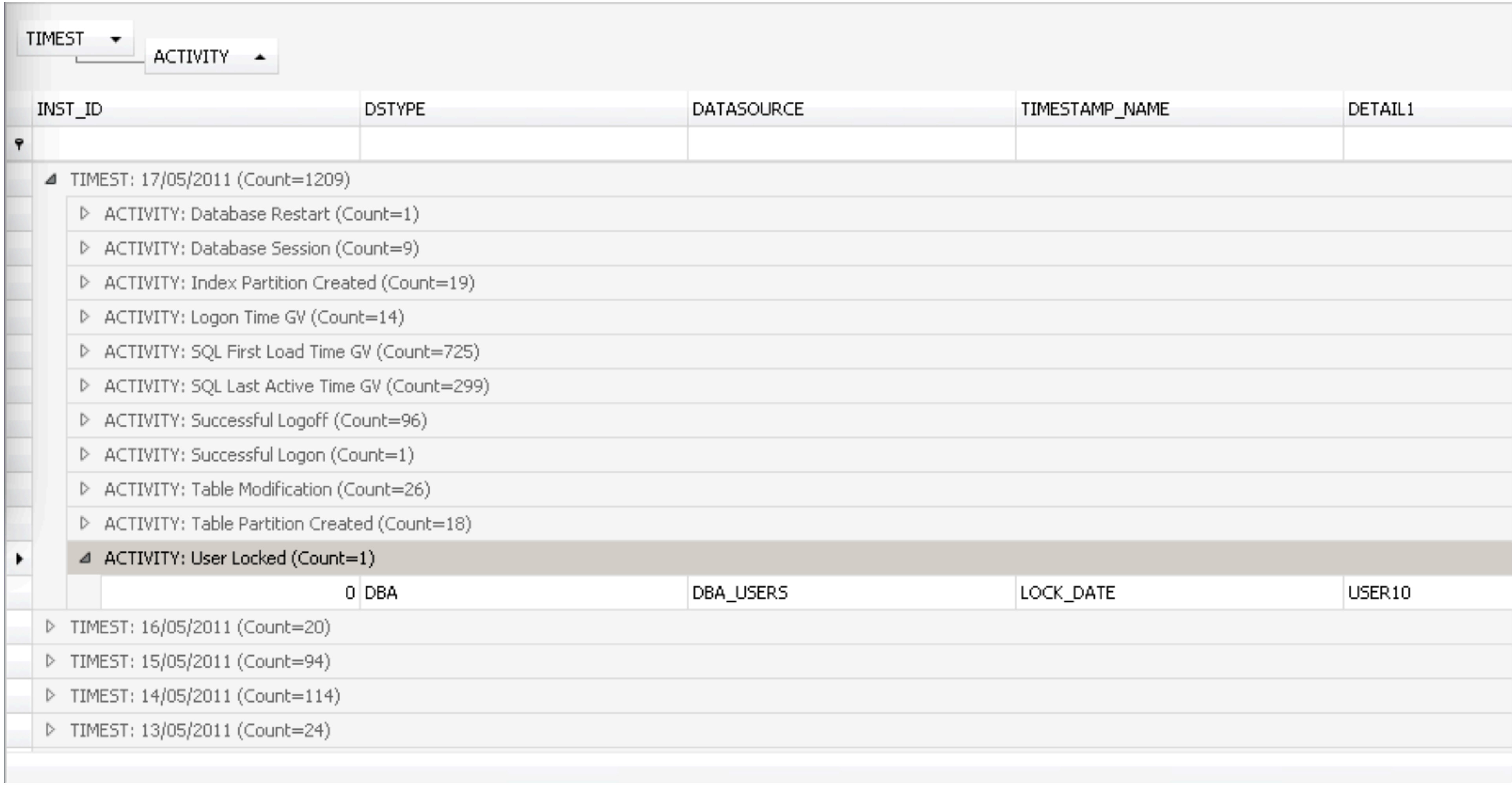
▷ ACTIVITY: Package Body Created (Count=1043)

▷ ACTIVITY: Package Created (Count=1101)



```
TIMEST: 17/05/2011 (Count=1209)
    ACTIVITY: Database Restart (Count=1)
    ACTIVITY: Database Session (Count=9)
    ACTIVITY: Index Partition Created (Count=19)
    ACTIVITY: Logon Time GV (Count=14)
    ACTIVITY: SQL First Load Time GV (Count=725)
    ACTIVITY: SQL Last Active Time GV (Count=299)
    ACTIVITY: Successful Logoff (Count=96)
    ACTIVITY: Successful Logon (Count=1)
    ACTIVITY: Table Modification (Count=26)
    ACTIVITY: Table Partition Created (Count=18)
    ACTIVITY: User Locked (Count=1)
TIMEST: 16/05/2011 (Count=20)
TIMEST: 15/05/2011 (Count=94)
TIMEST: 14/05/2011 (Count=114)
TIMEST: 13/05/2011 (Count=24)
TIMEST: 12/05/2011 (Count=132)
```

Timeline



```
TIMEST ▾
    ACTIVITY ▴

INST_ID        DSTYPE        DATASOURCE        TIMESTAMP_NAME        DETAIL1

◢ TIMEST: 17/05/2011 (Count=1209)
    ▷ ACTIVITY: Database Restart (Count=1)
    ▷ ACTIVITY: Database Session (Count=9)
    ▷ ACTIVITY: Index Partition Created (Count=19)
    ▷ ACTIVITY: Logon Time GV (Count=14)
    ▷ ACTIVITY: SQL First Load Time GV (Count=725)
    ▷ ACTIVITY: SQL Last Active Time GV (Count=299)
    ▷ ACTIVITY: Successful Logoff (Count=96)
    ▷ ACTIVITY: Successful Logon (Count=1)
    ▷ ACTIVITY: Table Modification (Count=26)
    ▷ ACTIVITY: Table Partition Created (Count=18)
▸   ◢ ACTIVITY: User Locked (Count=1)
                     0 DBA           DBA_USERS         LOCK_DATE             USER10
▷ TIMEST: 16/05/2011 (Count=20)
▷ TIMEST: 15/05/2011 (Count=94)
▷ TIMEST: 14/05/2011 (Count=114)
▷ TIMEST: 13/05/2011 (Count=24)
```

Typical Tables and Pattern

- The following slides contain typical database objects (like sys.user$) and common attack traces which can be found in these objects.
- Data from audit.logs (disabled in most cases in the real world) is not covered in this presentation
- Files (like listener.log) are skipped to save some time.

Tables

- Audit-Tables / Audit-Logs
- sys.user$
- sys.wrh$_active_session_history
- sys.wrh$_sqltext
- sys.mon_mods$

Tables – sys.user$

- Interesting Columns
  - lcount
    - Number of invalid login attempts
    - Resetted after successful login
    - Maximum number dependent from the profile setting
  - ltime (Lock-Time)
    - Lock time of the account

Tables – sys.user$

- Typical attack patterns - lcount
  - Multiple accounts have a lcount > 0
    ➔ Someone tries to guess user accounts without locking them
  - Agent Accounts (e.g. Tivoli) have an lcount> 0 & lcount < max from Profile
    ➔ Someone tries to guess the password of an agent account. Lcount of agent accounts is normally 0 or max Profile
  - Big lcount value (e.g. 30.000)
    ➔ Bruteforce attack using a tool or someone forgot to change the client side password of an agent.

Tables – sys.user$

- Typical attack patterns- ltime
  - Multiple accounts with similar ltime
    ➔ Someone tried to guess user accounts but the accounts were locked.

Tables – sys.wrh$_ active_session_history

- Interesting Columns
  - program
    - Used Program
  - Module
    - Used module name
  - Machine (since 11.2)
    - What user was coming from what machine ➔ Important for password changes

- Warning!. The data from sys.wrh $active_session_history is not always reliable. Sometimes 0 (=SYS) is used even if the connect was not done by SYS.

Tables – sys.wrh$_ active_session_history

- Typical attack patterns
  - Program
    - Unwanted/unauthorized programs
    - Export utilities
  - Module
    - Program and Module do not match (e.g. oracle.exe & „TOAD 10.3.0.1“ ➔ renamed tool to bypass login trigger
  - Machine
    - Login from unusual machine
    - Combination User & Machine

Tables – sys.wrh$_ active_session_history (11.2)

```
select  program, username, machine, count(*) as cnt

from sys.wrh$_active_session_history w, dba_users d

where w.user_id=d.user_id (+)

and (lower(program) not like '%oracle%(%)%')

group by program, username, machine
```

Tables – sys.wrh$_ active_session_history

```
select  program, username, count(*) as cnt

from sys.wrh$_active_session_history w, dba_users d

where w.user_id=d.user_id (+)

and (lower(program) not like '%oracle%(%)%')

group by program, username
```

Tables – sys.wrh$_sqltext

- Interesting Columns
  - sqltext
    - SQL Statement of a user session

Tables – sys.wrh$_sqltext

- Typical attack patterns
  - sqltext
    - Suspicious SQL statements (Insert/Update/Delete/Select)

Tables – sys.mon_mods$

- Interesting Columns
  - Inserts
  - Updates
  - Deletes

Tables – sys.mon_mods$

- Typical attack patterns
  - obj#
    - Suspicious Statements (Insert/Update/Delete/Select)
  - Inserts
    - Insert in critical tables (Privileges, ...)
  - Updates
    - Update of log entries (e.g. AUD$, custom Log-Tables, ...)
    - Update of critical data
    - High value of update values on SYS.USER$ can be an indication of brute force attacks (high lcount value)
  - Deletes
    - Delete of log entries (e.g. AUD$, custom Log-Tables, ...)

Tables – sys.mon_mods$

```
select u.name as owner,o.name as table_name, m.inserts,
m.updates, m.deletes, m.timestamp
from sys.mon_mods$ m, sys.user$ u, sys.obj$ o
where o.obj#=m.obj# and u.user#=o.owner#
```

Database Blocks

- Contain data from tables
- Contain deleted/updated data as well

Database Blocks

```
SQL> conn sig/sig
Connected.

SQL> create table password (name varchar2(20),
password varchar2(20));
Table created.

SQL> insert into password values
('Alex','Supersecret1');
1 row created.

SQL> insert into password values ('Anna','Password1');
1 row created.

SQL> insert into password values
('Anton','Pr0d@adm1n');
1 row created.

SQL> commit;
Commit complete.
```

Database Blocks

```
SQL> select distinct dbms_rowid.rowid_block_number(rowid) from password;

DBMS_ROWID.ROWID_BLOCK_NUMBER(ROWID)
------------------------------------
                               57170

SQL> select tablespace_name from user_segments where segment_name in
('PASSWORD'
);

TABLESPACE_NAME
------------------------------
SYSTEM

SQL> select file_id from dba_data_files where tablespace_name='SYSTEM';

   FILE_ID
----------
         1
         9
SQL> alter system dump datafile 1 block 57170;

System altered.
```

Database Blocks

```
4715170 4B1AC506 0D481B50 6D6B3234 68776477  [...KP.H.42kmwdwh]
4715180 70347237 04C10277 C0000201 8D000DA3  [7r4pw...........]
4715190 4B1AC506 0D481B50 6D6B3234 68776477  [...KP.H.42kmwdwh]
47151A0 70347237 03C10277 C0000201 8C000DA3  [7r4pw...........]
47151B0 4B1AC506 0D481B50 6D6B3234 68776477  [...KP.H.42kmwdwh]
47151C0 02012C37 746E4105 500A6E6F 40643072  [7,...Anton.Pr0d@]
47151D0 316D6461 02012C6E 6E6E4104 61500961  [adm1n,...Anna.Pa]
47151E0 6F777373 2C316472 41040201 0C78656C  [ssword1,...Alex.]
47151F0 65707553 63657372 31746572 B0FF0601  [Supersecret1....]
```

Database Blocks

```
SQL> update password set password='HappyHacker' where
name='Anna';

1 row updated.

SQL> commit;

Commit complete.

SQL> alter system dump datafile 1 block 57170;

System altered.
```

Database Blocks

```
4715170 4B1AC506 0D481B50 6D6B3234 68776477  [...KP.H.42kmwdwh]
4715180 70347237 04C10277 C0000201 8D000DA3  [7r4pw...........]
4715190 4B1AC506 0D481B50 6D6B3234 68776477  [...KP.H.42kmwdwh]
47151A0 70347237 03C10277 C0000201 02022CA3  [7r4pw........,..]
47151B0 6E6E4104 61480B61 48797070 656B6361  [.Anna.HappyHacke]
47151C0 02002C72 746E4105 500A6E6F 40643072  [r,...Anton.Pr0d@]
47151D0 316D6461 02022C6E 6E6E4104 61500961  [adm1n,...Anna.Pa]
47151E0 6F777373 2C316472 41040200 0C78656C  [ssword1,...Alex.]
47151F0 65707553 63657372 31746572 B1EB0603  [Supersecret1....]
```

Database Blocks (Anonymisation)

```
SQL> update password set password='xxx' ;

3 rows updated.

SQL> commit;

Commit complete.

SQL> alter system dump datafile 1 block 57170;

System altered.
```

Database Blocks

```
4715170 4B1AC506 0D481B50 6D6B3234 68776477  [...KP.H.42kmwdwh]
4715180 70347237 04C10277 0502012C 6F746E41  [7r4pw...,...Anto]
4715190 7878036E 02012C78 6E6E4104 78780361  [n.xxx,...Anna.xx]
47151A0 02012C78 656C4104 78780378 02012C78  [x,...Alex.xxx,..]
47151B0 6E6E4104 61480B61 48797070 656B6361  [.Anna.HappyHacke]
47151C0 02012C72 746E4105 500A6E6F 40643072  [r,...Anton.Pr0d@]
47151D0 316D6461 02022C6E 6E6E4104 61500961  [adm1n,...Anna.Pa]
47151E0 6F777373 2C316472 41040201 0C78656C  [ssword1,...Alex.]
47151F0 65707553 63657372 31746572 B2230607  [Supersecret1..#.]
```

Pattern – Privilege Escalation

- Privilege escalation often uses stored procedures as helper function for privilege escalation
- Additional entries in DBA_ROLE_PRIVS, DBA_TAB_PRIVS, DBA_SYS_PRIVS
- Probably deleted entries in SYS.SYSAUTH$ / SYS.OBJAUTH$ / (visible in data blocks)

Pattern – Run OS Commands

- DBA_EXTERNAL_TABLES: External Table with preprocessor (column ACCESS_PARAMETERS)
- DBA_JAVA_POLICY: new entries
- DBA_LIBRARIES: new entries
- CTXSYS.CTX_PREFERENCE_VALUES: Oracle Text user filter , e.g. PRV_ATTRIBUTE=oratclsh.exe

Pattern – Backdoors

- Various places depending from the used backdoor
  - SYS.USER$
  - Oracle Password File
  - Logon trigger
  - Privileges (e.g. grant execute on SYS.DBMS_STREAMS_RPC to public)
  - ...

Pattern – Manipulated Audit/ Log Tables

- Update Log data: Modified ora_rowscn
- Delete Log data: Gaps in rowid
- Entries in SYS.MON_MODS$

Pattern – Data Export

- Attackers often export the database (or parts of it) using the official export utilities.
- These traces can be easily found in the
  - Listener.log
  - sys.wrh$_ active_session_history (requires special license)

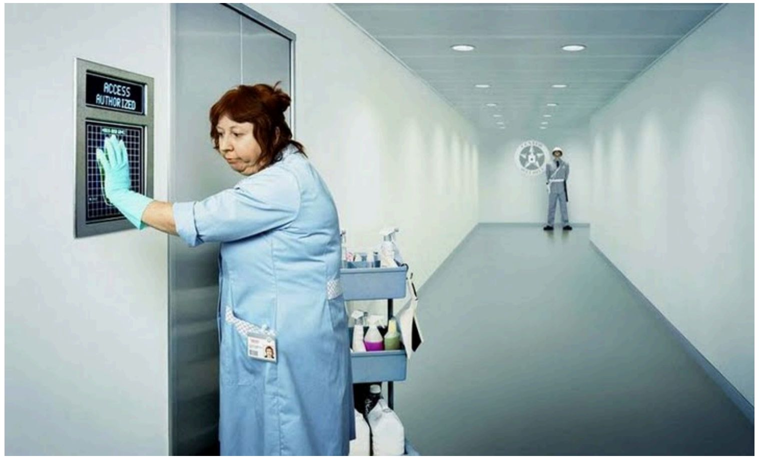
Pattern – oradebug

- Details of this attacks will be shown by Laszlo Toth talk “Almost invisible cloak in Oracle databases” at Hacktivity (15:10-15:55)
- Oradebug commands are recorded in the trace files and sometimes incident response files (if oradebug causes an Oracle error (e.g. ORA-07445))
- Tracefiles can easily be removed on OS level

Summary

- More convenient tools for databases forensics needed to allow non-databases (security) experts to find traces.
- Atomization for multiple databases needed
- Top down approaches are often easier to understand than bottom up approaches

Thank you



- Contact:

Red-Database-Security GmbH

Bliesstr. 16

D-.66538 Neunkirchen

Germany