Secure Your Database in Just One Day

Arup Nanda

Longtime Oracle DBA

Oracle Patch Set Updates (PSU)

What You'll Learn

- Addresses three Areas:
  - Identify and Seal Vulnerabilities
  - Build a Monitoring System
  - Enforce Change Control
- 60% of the compliance

What You'll Learn

- Each recommendation has:
  - pros, cons and impact
- Take away scripts
  - download www.proligence.com/sec_scripts.zip

Preliminaries

- Physical Security
  - Access control to the server
  - Authentication (unix userid password, etc.)
  - Surveillance and Auditing
  - OS Level Security – patches, unknown users, etc.
- Oracle specific
  - OS Vulnerabilities, including Listener
  - Database Vulnerabilities

Security Principle #1

Removal of possibility is better than strengthening it.

Security Principle #2

Recording deters theft

Protecting the Oracle Account

1

- Institute an indirect login policy
- No one logs into the Oracle software account; they will need to login to their own account, e.g. “jsmith”
- They execute commands that require Oracle software owner privilege using `sudo`

```
sudo –u oracle sqlplus / as sysdba
```

- This leaves an audit trail of actions

Listener as a Launchpad

2

- Listener is passed commands to be executed
  - Including malicious ones
- Disable Online Modification
  - `ADMIN_RESTRICTIONS_<ListenerName> = ON`
  - This will force values to be changed in `LISTENER.ORA` and then listener reloaded.

Impact on CPU Patches

- Most listener vulnerabilities are of two types
  - Buffer overflow
  - Privilege escalation
- The previous fix will prevent online modification and command executions
  - will prevent some of the buffer overflow and privilege escalations

Preventing SYSDBA

3

- Normally SYS logs in as:
  1. `sqlplus "/ as sysdba"`
  2. `sqlplus sys/<sys_password> as sysdba`
- Change SQLNET.ORA file:
  `SQLNET.AUTHENTICATION_SERVICES=(NONE)`
- After this change, the login attempt# 1 above will fail; SYS has to provide the password.

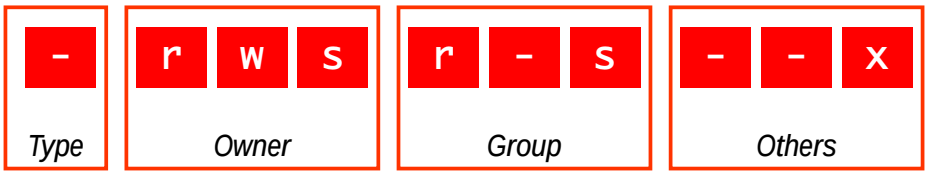
Impact on CPU Patching

- Many vulnerabilities exploit a technique known as privilege escalation
- They login as a regular user; but taking advantage of some vulnerabilities in programs such as tnslsnr and expdp, become the sysdba user
- If a password is forced, that vulnerability will be reduced
- Important: the vulnerability will not be *eliminated*; just reduced.

Permissions Issues

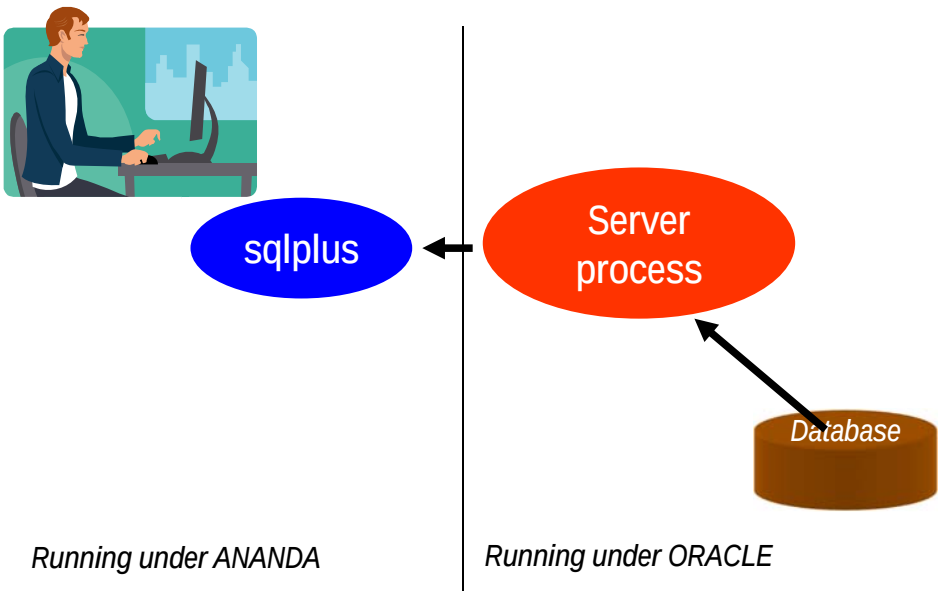
- The "oracle" executable

```
$ ls -l oracle
-rwsr-s--x 1 oracle oinstall 69344968 Jun 10 14:05 oracle
```



ananda: `sqlplus scott/tiger`

Two Task Architecture



Server Process

```
$ sqlplus scott/tiger
$ ps -aef|grep sqlplus
ananda  6339  6185  0 13:06 pts/0    00:00:00 sqlplus
$ ps -aef|grep 6339
ananda  6339  6185  0 13:06 pts/0    00:00:00 sqlplus
oracle    6340  6339  0 13:06 ?        00:00:00
  oraclePRODB1
  (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
```

Client Process

Server Process

Change Permission

4

- Remove SUID

  ```
  $ chmod 0700 $ORACLE_HOME/bin/oracle
  ```

- New Permissions

  ```
  -rwx------   1 oracle    oinstall
    248754168 Oct  8 07:11 oracle
  ```

- Test

  ```
  $ sqlplus scott/tiger
  ```

  The user will immediately get an error.

  ```
  ERROR:
  ORA-12546: TNS:permission denied
  ```

Fix

- Add in TNSNAMES.ORA

```
PRODB1 =
 (DESCRIPTION =
   (ADDRESS_LIST =
     (ADDRESS = (PROTOCOL = TCP)
     (HOST = prolin1)(PORT = 1521))
   )
   (CONNECT_DATA =
     (SERVICE_NAME = PRODB1)
   )
 )
```

- `$ sqlplus scott/tiger@prodb1`

Separate Client Oracle Home

5

- Do not use the OH of the RDBMS installation
- Install a separate OH for clients
  `/u01/app/oracle/11.2/client1`
- This allows complete separation of DB and clients
- This allows you to make the OH for DB completely invisible to non-Oracle OS user
  `$ chmod 700 $OH`

Impact on CPU Patches

- Some vulnerabilities exploit the IPC capabilities of the Oracle executables.
- Using a connection made through the listener process, you remove that capability
- As a result, vulnerabilities exploiting that capability will be reduced.

Backup Executables

6

- Presence of executable files with “0” or “O” at the end in $OH/bin

```
$ ls -l *0 *O
-rwsr-s--x  1 oracle oinstall 158490093 Sep 10  2009
  oracleO
```

- produced when you relink Oracle executables
- Same functionality as the previous ones

Other Executables

- Permissions with SetUID
- Find them:

```
find . -type f \( -perm -2000 -o -
  perm -4000 \) -exec ls -l {} \;
```

  - oracle0. chown 0600
  - oradism
  - emtgtctl2 – EM Agent. chown 0700
  - nmb – Grid Control Agent
  - nmo - Grid Control Agent
  - extjob and extjob0 – 0700

Other Executables

- DBSNMP

```
-rwsr-s---   1 root  dba 2986836 Jan 26  2005 dbsnmp
```

  - Change it.

```
chown oracle:dba dbsnmp
chmod 0700 dbsnmp
```

- lsnrctl and (lsnrctl0) and tnslsnr (and tnslsnr0)

  $ ls -l *lsnr*

```
-rwxr-x--x 1 oracle oinstall 214720 Oct 25 01:23
  lsnrctl
-rwxr-x--x 1 oracle oinstall 1118816 Oct 25 01:23
  tnslsnr
```

- Change them:

```
$ chmod 700 lsnrctl tnslsnr
$ chmod 600 lsnrctl0 tnslsnr0
```

Configuration File Perms

8

- No Oracle Configuration file should have any privilege to others

```
-rwxr-xr-x   1 oracle oinstall       779 Jun 16
  03:59 listener.ora
```

- No need to have read and execute permissions to listener.ora. Password can be made visible (older)

External Procedure

9

- Entry in listener.ora

```
(ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)
               (KEY = EXTPROC))
```

- The user executes a program *as the user oracle!*
  - Can delete data files, steals data, and so on
- Solutions:
  - Remove the lines
  - Move it to a different listener
  - Separate it to different listener.ora file

Separate Listener

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY =
    EXTPROC))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST =
    ANANDA)(
                    PORT = 1521))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL =
    IPC)(KEY=ANANDA))
      )
    )
  )
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = d:\ora9)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = ANANDA)
      (ORACLE_HOME = d:\ora9)
      (SID_NAME = ANANDA)
    )
  )
```

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS =
          (PROTOCOL = TCP)(HOST = ANANDA)(PORT =
1521))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY=ANANDA))
      )
    )
  )
LISTENER_EXTPROC =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY=EXTPROC))
      )
    )
  )
SID_LIST_LISTENER =
   (SID_LIST =
      (SID_DESC =
         (GLOBAL_DBNAME = ANANDA)
         (ORACLE_HOME = d:\ora9)
         (SID_NAME = ANANDA)
      )
   )
SID_LIST_LISTENER_EXTPROC =
   (SID_LIST =
      (SID_DESC =
         (SID_NAME = PLSExtProc)
         (ORACLE_HOME = d:\ora9)
         (PROGRAM = extproc)
      )
   )
```


Hiding Passwords

10

- `sqlplus scott/tiger @myscript`
- `sqlplus scott/$SCOTTPASS @myscript`
- Option 1:
  - `sqlplus /nolog @myscript`
  - *(Inside myscript)* `connect scott/tiger`
- Option 2:

```
sqlplus /nolog << EOF
connect scott/tiger
EOF
```

Password File

- Create a passwords file ".passwords"

```
scott tiger
arup aruppass
```

- Create a shell script ".getpass.sh"

```
fgrep $1 $HOME/tools/.passwords | cut -d
  " " -f2
```

- Use it in scripts

```
.getpass.sh scott | sqlplus –s scott
  @script.sql
```


Other Options

- Use DBMS_JOB or DBMS_SCHEDULER
  - No password is ever entered or displayed
  - Jobs start only when the database is up
- Use OPS$ Accounts

```
SQL> create user OPS$SCOTT identified externally;
$ su – scott
$ sqlplus /
```

- In RMAN scripts

```
Old: rman target=/ rcvcat=u/p@catdb
New: rman target=/
     connect catalog u/p@catdb
```

Users with Default Passwords

- About Oracle Passwords
  - PASSWORD in DBA_USERS is a hash value of the combined value of USERID and PASSWORD.
  - So even if two users have the same password, the hash value will be different.

UserID	Password	Password Hash
ABC	DEF	016811C1486D026B
ABCD	EF	016811C1486D026B

Identify Default Passwords

Create a table to hold the passwords.
Script: cr_osp_acounts.sql

```
CREATE TABLE OSP_ACCOUNTS
(
   product          VARCHAR2(30),
   security_level   NUMBER(1),
   username         VARCHAR2(30),
   password         VARCHAR2(30),
   hash_value       VARCHAR2(30),
   commentary       VARCHAR2(200)
);
```

Download and execute script
http://www.petefinnigan.com/default/osp_accounts_public.zip
Script: osp_install_data.sql

```
col password format a20
col account_status format a20
col username format a15
select o.username, o.password,
d.account_status
from dba_users d, osp_accounts o
where o.hash_value = d.password;
```

get_def_pwd.sql

In 11g you can check select * from dba_users_with_defpwd

Trim Privileges

12

- "Sweeping" Privileges
- "ANY" privileges,
  - CREATE ANY TABLE/PROCEDURE/INDEX, etc.
  - RESTRICTED SESSION
  - SELECT ANY TABLE
  - SELECT ANY DICTIONARY
  - UNLIMITED TABLESPACE
  - Script sweeping.sql

Seemingly Innocuous Privileges

13

- SCOTT needs to use these statements in a regular day's work:
  - `alter session set query_rewrite_enabled = true`
  - `alter session set optimizer_mode = …`
  - `alter session set sort_area_size = …`
- Does SCOTT need ALTER SESSION privilege?
- NO! Alter Session System Privilege
  - is *not* required to change session params
  - Only required for I/O operations, e.g. trace file
  - Script – alter_sess_grantees.sql

Other Dangerous Privs

- Create ANY Directory
  - can create a directory on any directory owned by Oracle user, incl. datafiles.
- Create ANY Trigger
  - can create triggers on any schema to capture sensitive data during insert/update
- Create Database Link

Dangerous Supplied Packages

14

- UTL_TCP
  - Main attack vehicle for the “Voyager” worm!
- DBMS_SCHEDULER
  - Can cause DoS attacks by calling the executables
- DBMS_JAVA
  - Can cause system hijacking by calling java programs to execute with oracle’s OS privs
- UTL_FILE
  - Can open/close files, even if controlled.
- DBMS_ASSERT
  - Can be used by hackers to make a user the DBA

Access Control for Packages

- In 11g, there is a fine grained access control list defined for these packages

```
begin
dbms_network_acl_admin.create_acl (
acl             => 'utlpkg.xml',
description     => 'Normal Access',
principal       => 'CONNECT',
is_grant        => TRUE,
privilege       => 'connect',
start_date      => null,
end_date        => null
);
end;
```

```
begin
dbms_network_acl_admin.add_privilege (
acl             => 'utlpkg.xml',
principal       => 'SCOTT',
is_grant        => TRUE,
privilege       => 'connect',
start_date      => null,
end_date        => null);
end;
```

```
begin
dbms_network_acl_admin.assign_acl (
acl         => 'utlpkg.xml',
host        => 'www.proligence.com',
lower_port  => 22,
upper_port  => 55);
end;
```


UTL_FILE_DIR

15

- Is it set to "*"?
  - Then someone can write a PL/SQL program to read (and *WRITE*!) *any* file owned by oracle, including data files, archived log files, etc.
- Use DIRECTORY objects, instead.

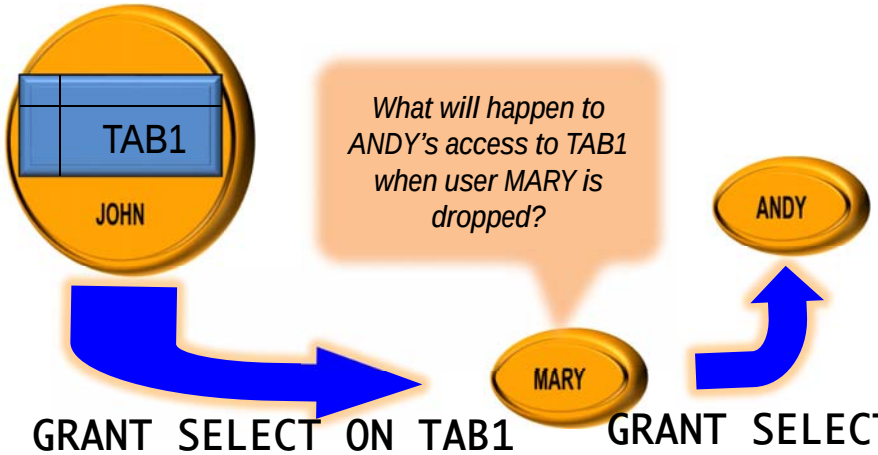
```
SQL> create directory MYDIR as '/u10/mydir';
utl_file.fopen ('MYDIR','myfile.txt','W')
```

- Revoke CREATE ANY DIRECTORY from PUBLIC
- Log Miner Dictionary File creation still needs this!

```
utl_file_dir = '/tmp'
```

- Database restart required.

Indirect Grants




```
GRANT SELECT ON TAB1
TO MARY
WITH GRANT OPTION;
```

```
GRANT SELECT ON
JOHN.TAB1
TO ANDY;
```

script ind_demo.sql

Effect of Indirect Grants

- Different Syntax for Different Privileges
  - System Privileges

```
grant create trigger to mary with admin
  option;
```

  - Object Privlileges

```
grant select on tab1 to mary with grant
  option;
```

- If mary grants these two privileges to andy, and then mary is dropped, andy will:
  - Lose the object privileges
    - In 11.2 will retain it
  - Retain the system privilege

Identify Indirect Grants

- Use script indirect_grants.sql

```
select grantee, privilege, owner,
  table_name
from dba_tab_privs
where grantor != owner;
```


Identifying Grantable Grants

17

Script grantable_privs_obj.sql

```
select grantee, owner, table_name, privilege,
  grantor
from dba_tab_privs
where grantable = 'YES'
and grantee != 'SYS';
```

Script grantable_privs_sys.sql

```
select grantee, privilege
from dba_sys_privs
where admin_option = 'YES'
and grantee not in ('SYS','DBA')
order by 1,2;
```

Simple Audit

- Set the database parameter AUDIT_TRAIL to DB_EXTENDED or at least DB
- Objective:
  - Which user connected, OS User
  - Other details – terminal, (dis)connection time, etc.
- Auditing is expensive; so start small: `audit session`

Reporting

- Use this for reporting

```
select
    to_char(timestamp,'mm/dd/yy hh24:mi:ss') li,
    username,
    os_username,
    userhost,
    terminal,
    to_char(logoff_time,'mm/dd/yy hh24:mi:ss') lo
from dba_audit_trail
where logoff_time is not null;
```

- Shows who, OS user, terminal, time of login and logout

Simple_audit.sql

Use of Simple Auditing

- Build a profile of database access
  - Which users connect, how often
  - Where they connect from, how frequently
  - How many app servers are present
  - Who is a heavy-hitter
- Prepare a Baseline
- Check regularly against the baseline to see patterns

Identify Access Violations

- Who tried but was not successful

```
select username, os_username, terminal, userhost,
to_char(timestamp,'mm/dd/yy hh24:mi:ss') logon_ts
from dba_audit_trail
where returncode = 1017;
```

Unsucc.sql

- Was someone trying to "guess" userids?

```
select username from dba_audit_trail
where returncode = 1017
minus
select username from dba_users;
```

Wrong.sql

Fringe Benefits

- CPU and IO Usage
  - Useful for Resource Manager/Profiles
  - Diagnosis of past performance issues
  - Capacity Planning

```
select username, to_char(logoff_time,'mm/dd') ts,
   count(1) cnt,
   sum(session_cpu) sum_cpu,
   avg(session_cpu) avg_cpu,
   min(session_cpu) min_cpu,
   max(session_cpu) max_cpu
from dba_audit_trail
group by username, to_char(logoff_time,'mm/dd')
order by username, to_char(logoff_time,'mm/dd')
```

Audcpu.sql

Simple Auditing

```
audit session
audit not exists
audit alter system
audit database link
audit directory
audit grant directory
audit index
audit materialized view
audit outline
audit procedure
audit grant procedure
audit profile
audit public database link
audit role
audit sequence
audit alter sequence
audit grant sequence
audit public synonym
audit synonym
audit system audit
audit system grant
audit table
audit alter table
audit grant table
audit tablespace
audit trigger
audit type
audit grant type
audit user
audit view
```

Security Principle #1
Removal of possibility is better than strengthening it.

Listener hardening
Removal of perms and privs

Security Principle #2
Recording deters theft

Auditing

Plan

- Make listener changes
- Reload listener to take effect
- Make all non-required binary changes
- Make all binary permission changes
- Make the changes to the INIT.ORA params
- Recycle the database
- Enable Auditing
- Remove Sweeping Privileges
- Remove Execute Privileges from PUBLIC

Thank You!

Download Scripts: `proligence.com/pres/auoug16`
Blog: `arup.blogspot.com`
Tweeter: `@ArupNanda` Facebook: `fb.com/ArupKNanda`