

UKOUG Unix SIG, January 22nd 2008

Using Oracle VPD In The Real World

By
Pete Finnigan

Written Friday, 27th December 2007

Introduction - Commercial Slide. ☹

- PeteFinnigan.com Limited
- Founded February 2003
- CEO Pete Finnigan
- Clients UK, States, Europe
- Specialists in researching and securing Oracle databases
- <http://www.petefinnigan.com>
- Consultancy and training available
- Author of Oracle security step-by-step
- Published many papers, regular speaker (UK, USA, Slovenia, Norway, more)
- Member of the Oak Table



Agenda

- What is VPD, is it used, info?
- Differences in various Oracle versions
- Securing VPD – often not considered
- Attacking VPD
- Problems – performance – design
- Conclusions

What Is VPD

- Called Virtual Private Database (VPD)
- Called Row Level Security (RLS)
 - Also DBMS_RLS controls it
- Called Fine Grained Access Control (FGAC)
- VPD includes:
 - Fine Grained Access Control
 - Application Contexts
 - Global Application Contexts

Is VPD Used In Anger?

- In my experience not much – why?
 - I have worked with a few clients to implement VPD
 - It is free with EE; not a cost option that may put people off like OLS
- Oracle are increasingly using it
 - In XDB ACL's
 - In E-Business Suite
 - As part of Database Vault and Audit Vault

Where To Find Information

- Effective Oracle database 10g security by design - ISBN-13: 978-0072231304
- RLS chapter - <http://www.devshed.com/c/a/Oracle/RowLevel-Security-with-Virtual-Private-Database/>
- Does VPD, FGA or audit really cause performance issues - <http://www.insight.co.uk/files/presentations/Does%20VPD.%20FGA%20or%20Audit%20Cause%20Performance%20Issues.pdf>
- Oracle Row Level Security - <http://www.securityfocus.com/infocus/1743>
- Row Level Security - <http://www.dbazine.com/oracle/articles/jlewis15>

VPD Through The Versions

- Row Level Security added in 8.1.5 release
- 9i adds multiple policies per table and policy groups controlled by application driving context
- 9i adds global contexts for connection pooling
- 10g adds column level policies, column masking, policy types (5) added for performance to allow caching, contexts updated to allow values to be passed to parallel slaves.
- 11g provides integration for Enterprise manager for Row Level Security Policies.

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

7

Securing VPD

- Leaking predicates
- Leaking policies
- RBAC on VPD structure / configuration
- Bypassing VPD by means of exception
- SQL Injection issues
- Direct data access

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

8

Finding the Predicate

- There are a number of possibilities to find predicates and details
 - Event 10730
 - Event 10060
 - V\$vpd_policy – no one has access by default
- Library cache dump? – if static data present can also be leaked
- SGA can be dumped for binds, SQL, optimizer and more
- Common denominator – ALTER SESSION / SYSTEM / trace (many options - http://www.petefinnigan.com/ramblings/how_to_set_trace.htm)

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

9

Create A Simple Policy

- See code <http://www.petefinnigan.com/vpd.sql>
- Create a user PXF,
 - do some grants,
 - Create a table (copy of scott.emp)
 - Create a predicate function to block “deptno != 10”
 - Create a policy on pxf.emp
 - Number of rows restricted by 3
 - Demo!

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

10

Example

```
who_has_priv: Release 1.0.3.0.0 - Production on Wed Jan 16 19:13:16 2008
Copyright (c) 2004 PeteFinnigan.com Limited. All rights reserved.

PRIVILEGE TO CHECK [SELECT ANY TABLE]: ALTER SESSION
OUTPUT METHOD Screen/File (S): S
-----
Privilege => ALTER SESSION has been granted to =>
*****
Role => DBA (ADM = YES) which is granted to =>
User => SYS (ADM = YES)
User => SYSMAN (ADM = NO)
User => SYSTEM (ADM = YES)
User => TESTUSER (ADM = NO)
User => SYS (ADM = NO)
User => IX (ADM = NO)
User => SH (ADM = NO)
Role => RECOVERY_CATALOG_OWNER (ADM = NO) which is granted to =>
User => SYS (ADM = YES)
User => BI (ADM = NO)
User => CTXSYS (ADM = NO)
Role => OLAP_USER (ADM = NO) which is granted to =>
User => SYS (ADM = YES)
User => SCOTT (ADM = NO)
User => HR (ADM = NO)
User => DMOSYS (ADM = NO)
User => XDB (ADM = NO)
```

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

11

Example (2)

```
SQL> alter session set sql_trace=true;
Session altered.
SQL> alter session set events '10730 trace name context forever';
Session altered.
SQL> select * from pxf.emp;
-----
EMPNO ENAME JOB MGR
-----
7369 SMITH CLERK 7902 17-DEC-80 800
20
-----
SQL> alter session set events '10730 trace name context off';
Session altered.
SQL> alter session set sql_trace=false;
Session altered.
SQL>
```

As a normal user –
SCOTT - I am able to
determine the rules
VPD imposes on me

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

12

Example (3)

```

=====
PARSING IN CURSOR #4 len=29 dep=1 uid=76 oct=47 lid=76 tim=56027100276 hv=1410654723 ad='2bafbac'
begin cco := PREDICATE(en, 'on), end;
END OF STMT
PARSE #4:c0,e=47,p=0,cr=0,cu=0,mis=0,r=0,dep=1,og=1,tim=56027100270
EXEC #4:c0,e=127,p=0,cr=0,cu=0,mis=0,r=1,dep=1,og=1,tim=56027101747
=====
Logon user: SCOTT
Table View: PXF.EMP
Policy name: PXFTST
Policy function: PXF.PREDICATE
SEL view:
SELECT EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO FROM PXF.EMP WHERE
(deptno = '10')
=====
PARSING IN CURSOR #4 len=50 dep=1 uid=54 oct=3 lid=54 tim=56027104042 hv=216005283 ad='2bafbac'
SELECT /* OPT_JOIN_SAMP */ /* ALL_ROWS FORCE WHERE CLAUSE NO_PARALLEL(SAMPLESUB)
opt_param('parallel_execution_enabled', 'false') NO_PARALLEL_INDEX(SAMPLESUB) NO_SQL_TUNE /*
NO_INDEX(C1) /*SYS_B_0) NO_INDEX(C2) /*SYS_B_1) FROM (SELECT /*+ COMPARE ANGER CLAUSE
NO_PARALLEL(EMP) FULL(EMP) NO_PARALLEL_INDEX(EMP) */ /*SYS_B_2) AS C1, CASE WHEN
EMP.DEPTNO = /*SYS_B_3) THEN /*SYS_B_4) ELSE /*SYS_B_5) END AS C2 FROM /*P*/ /*EMP) SAMPLESUB
END OF STMT
PARSE #4:c0,e=543,p=0,cr=0,cu=0,mis=1,r=0,dep=1,og=1,tim=56027104045
EXEC #4:c0,e=1306,p=0,cr=0,cu=0,mis=1,r=0,dep=1,og=1,tim=560271056137
FETCH #4:c0,e=933,p=0,cr=3,cu=0,mis=0,r=1,dep=1,og=1,tim=56027104044
STAT #4 lg=1 cnt=1 pid=0 pos=1 obj=0 op='SORT AGGREGATE (cr=3, pr=0, pw=0, tlm=940 us)'
STAT #4 lch=1 cnt=14 pids=1 pos=1 obj=54300 ops=TABLE ACCESS FULL EMP (cr=3, pr=0, pw=0, tlm=907 us)'
=====
PARSING IN CURSOR #3 len=21 dep=0 uid=54 oct=3 lid=54 tim=560271071623 hv=3160365295 ad='2bb8804'
select /* from pxf.emp
END OF STMT
PARSE #3:c=62500,e=71942,p=0,cr=4,cu=0,mis=1,r=0,dep=0,og=1,tim=560271071614
=====

```

Ora10gr2_ora_7784.trc

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

13

Leaking Policy Details

- To secure VPD all of the configuration must be secured including:
 - %_POLICY_GROUPS
 - %_POLICY_CONTEXTS
 - %_POLICIES
- Access to functions must also be protected;
 - Definitions – OBJ\$, SOURCE\$, PROCEDURE\$, ARGUMENT\$

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

14

Example

```

who_can_access: Release 1.0.3.0.0 - Production on Wed Jan 16
19:30:16 2008
Copyright (c) 2004 PeteFinnigan.com Limited. All rights reserved.

NAME OF OBJECT TO CHECK      [USER_OBJECTS]: ALL_POLICIES
OWNER OF THE OBJECT TO CHECK [USER]: SYS
OUTPUT METHOD Screen/File     [S]: S
FILE NAME FOR OUTPUT         [priv.lst]:
OUTPUT DIRECTORY [DIRECTORY or file (/tmp)]:
EXCLUDE CERTAIN USERS       [N]:
USER TO SKIP                 [TEST%]:

Checking object => SYS.ALL_POLICIES
=====
====
Object type is => VIEW (TAB)
Privilege => SELECT is granted to =>
Role => PUBLIC (ADM = NO)

```

Madness by default anyone can see what policies exist that affect them

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

15

Example(2)

```

SQL> select object_owner,object_name,policy_name,
2 pf_owner,pf_owner,function
3 from all_policies;

OBJECT_OWNER      OBJECT_NAME
-----
POLICY_NAME       PF_OWNER
-----
PF_OWNER          FUNCTION
-----
SCOTT             EMP
PXFPOL            SYS
SYS              HACK

EMP              EMP
PXFTST           PXF
PXF              PREDICATE

2 rows selected

```

As SCOTT I could find out the predicate, I can also find out the policies that affect me.

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

16

RBAC on VPD structure

- RBAC must be applied on
 - Packages – DBMS_RLS, DBMS_SESSION
 - Policies – see previous slide
 - Policy functions, structure, source code
 - Contexts, application and global
 - Supporting data – static look up data
 - System privileges used
 - Grants on access to any of the above
- Don't just rely on VPD to protect data

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

17

Bypassing VPD

- VPD configuration should be designed normally to work with users (end users / identities)
 - i.e. access to groups of data is based on actual people, this is reflected in the VPD
- This is often done in total or part using application contexts – These are tied to the session
- BUT, they must use static data, session data, application data (i.e. FND_PROFILES) to ascertain who is who
- Whilst the context is reasonably secure often the data used could be changed/bypassed/spoofed
- All of the identity must be considered and hardened

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

18

Exempt Access Policy

```

who_has_priv: Release 1.0.3.0.0 - Production on Wed Jan 16 16:26:56
2008
Copyright (c) 2004 PeteFinnigan.com Limited. All rights reserved.

PRIVILEGE TO CHECK      [SELECT ANY TABLE]: EXEMPT ACCESS POLICY
OUTPUT METHOD Screen/File      [S]: S
FILE NAME FOR OUTPUT      [priv.lst]:
OUTPUT DIRECTORY [DIRECTORY or file (/tmp)]:
EXCLUDE CERTAIN USERS      [N]:
USER TO SKIP              [TEST%]:

Privilege => EXEMPT ACCESS POLICY has been granted to =>
=====
User => X (ADM = NO)

PL/SQL procedure successfully completed.

For updates please visit http://www.peteфиннigan.com/tools.htm

SQL> http://www.peteфиннigan.com/who\_has\_priv.sql
    
```

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

19

SQL Injection

- SQL Injection could be used in a number of ways to exploit VPD:
 - Litchfield shows how to inject a call to DBMS_RLS.DROP_POLICY via XDB.XDB_PITRIG_PKG.PITRIG_DROP – see <http://www.databasesecurity.com/dbsec/ohh-defeating-vpd.pdf>
 - Many exploits from sites such as <http://milw0rm.com> can be used in the same way
 - Packages that expose VPD – see next slide
 - Applications that VPD could have components exploited – i.e. if the predicate is “constructed” using concatenation it could be exploited.

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

20

Ways To Access Policies

```

SQL> select owner,name,type
2 from dba_dependencies
3 where referenced_name='DBMS_RLS';

OWNER      NAME      TYPE
-----
PUBLIC     DBMS_RLS  SYNONYM
SYS        DBMS_RLS  PACKAGE BODY
SYS        LTUTIL    PACKAGE BODY
SYS        LTADM     PACKAGE BODY
XDB        DBMS_XDBZ0 PACKAGE BODY
XDB        DBMS_XDBZ0 PACKAGE BODY

SQL> select grantee,table_name from dba_tab_privs
2 where table_name in ('LTUTIL','LTADM','DBMS_XDBZ0');

GRANTEE      TABLE_NAME
-----
WMSYS        LTADM
WMSYS        LTUTIL
IMP_FULL_DATABASE LTADM
PUBLIC       DBMS_XDBZ0
    
```

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

21

Access The Data Directly

- Strings on data files
- With C or Java from the database
- Hex editors – Unix or Windows
- Block dumps – recent forensics papers cover
- Tools like bbed, CBAT, DUL like tools such as Ora*Dude and more
- Backups
- Exports
- Reports and lists of data from privileged users
- More?

Again do not consider VPD as a “be all” and “end all” – work out where the data is and how it “flows”

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

22

Example (1)

```

SQL> select distinct dbms_rowid.rowid_block_number(rowid) blk,
2 dbms_rowid.rowid_relative_fno(rowid) fno
3 from pxf.emp;

BLK      FNO
-----
420      4

1 row selected.

SQL> select file_name from dba_data_files
2 where file_id=4;

FILE_NAME
-----
C:\ORACLE\ORADATA\ORA10GR2\USERS01.DBF

1 row selected.
    
```

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

23

Example (2)

```

SQL> alter system dump datafile 4 block 420;
System altered.
SQL> connect sys/change_on_install as sysdba
Connected.
SQL> select * from pxf.emp where deptno=10;

EMPNO ENAME      JOB      MGR HIREDATE      SAL
-----
7782 CLARK      MANAGER  7839 09-JUN-81    2450
10
7839 KING      PRESIDENT 17-NOV-81    5000
10
7934 MILLER   CLERK     7782 23-JAN-82    1300
10
    
```

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

24

Example (3)

```
Repeat 464 Times
8229DC0 00000000 00000000 2359C203 4C494006 [.....PF MIT]
8229DD0 05524540 52454C43 4EC2034B B6770753 [LER CLERK INS W ]
8229DE0 01011701 08C30303 08C103FF 0308082C [.....]
8229DF0 040358C2 44524F46 414E4107 5453534C [ P FORD ANALYST]
8229E00 4146C203 0C2B7707 01011013 FF1E0302 [ L C W ]
8229E10 2C15C102 C2020800 414A0550 0553454D [.....P JAMES ]
8229E20 52454C43 4EC2034B B6770753 0101101C [CLERK Mo W.....]
8229E30 04C20301 C102FF33 09092C1E 4D4FC203 [.....] OM]
8229E40 4144410F 4305344D 4B53494C 5245C203 [.....]
8229E50 05B87707 01010117 FF0C0202 2C15C102 [ M.....TURNER SA]
8229E60 02000000 54000000 44524F46 414E4107 [.....]
8229E70 4D53454C C2034E41 7707014D 01090905 [LESMAN Mo W.....]
8229E80 02000000 08080110 010C1FC1 4FC20308 [.....]
8229E90 49480428 5009474E 49534552 54484544 [ ( KING PRESIDENT]
8229EA0 B677075F 0101110B 33C20101 08C102FF [ M.....]
8229EB0 0308082C 055A48C2 544F4351 4E410754 [.....NY SCOTT AN]
8229EC0 53594C41 4C203034 B6770753 01011104 [ALYST LC W.....]
8229ED0 18C40301 15C102FF 0308082C 055A48C2 [.....NS ]
8229EE0 52414C43 414D074B 4547414E 4FC20352 [CLARK MANAGER O]
8229EF0 B677075F 01010906 15C10101 C102FF33 [ M.....]
8229F00 08092C0B 634DC203 414C4205 4D07454B [ M.....]
8229F10 47414E41 C2034945 77072048 01010305 [ANAGER OI W.....]
8229F20 C2030101 02FF331D 002C1FC1 4DC20308 [.....] MT]
8229F30 414D074B 4E494945 4C415308 414E4107 [? MARTIN SALESMAN]
8229F40 4DC2034E B6770753 01011C09 0DC20101 [N Mo W.....LC ]
8229F50 0FC20323 2C15C102 4C2E0300 4A09494C [ P.....LC ]
8229F60 53454E4F 4E414D07 52454974 284FC203 [ONES MANAGER OI]
8229F70 48577707 01011013 4C2E0303 15C102FF [ M.....L ]
8229F80 0308082C 04164C2C 44524F46 414E4107 [.....L WARD SAL]
8229F90 414E4107 4C2E0303 B6770753 01011602 [ERMAN Mo W.....]
8229FA0 04C20301 06C20323 2C15C102 C2030800 [.....]
8229FB0 41054448 4E454C4C 4C415308 414E4107 [RD ALLEN SALESMAN]
```

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

25

Screens Can Break

- Certification and Support for Third party products - <http://blogs.oracle.com/schan/newsItems/departments/extendingApps/2006/05/18#a200>
- Using VPD can break existing applications and other modules
- E-Business Suite screens have been seen to break because VPD is enabled
- There is often a fear with VPD implementers that they are not supported if VPD breaks something
- You can get into a complex support / certification saga
- If Oracle can reproduce – even if you let support have your code or an example with the same problem Oracle can help look at the issue

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

26

Layered Approach

- VPD must be part of a layered approach to securing data in an Oracle database
- RBAC on
 - Data
 - Security measures and policies
- Encryption for critical data
- Hardening must be done
- VPD as part of an overall solution
- Network security
- Audit trails
- More...

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

27

Performance

- VPD is often perceived as being bad due to perceived optimizer changes – aim to not excessively change the optimizer
- Often runs faster when VPD is enabled – less rows returned!
- Don't use excessive code in predicates i.e. select from dual or worse big tables
- Use indexes on the predicate columns
- Use static data if at all possible
- Use static policies if possible
- Keep the policy functions as simple as possible – good design is king!

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

28

Cached Policies and sys_context

- Another lesson learned was to pass back `sys_context('...', '...')` rather than resolve the `sys_context` in the policy function
- 5 types of caching can be used:
 - Static – execute once, store predicate in SGA
 - Shared_static – cache predicate across multiple objects using same policy
 - Context_sensitive – use for connection pooling, server executes policy function on statement execution if a context change detected
 - Shared_context_sensitive – as above; shared across multiple objects; same policy
 - Dynamic – no caching executed every time

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

29

Design It First

- One of the key lessons I have learned with VPD is to design carefully first. Include:
 - Business rules first (who/what/when)
 - Identify the data to be protected
 - Simplicity is the key – keep the rules / policies very simple (as simple as possible)
 - Work out the identities, the rules for all access, the default state,
 - Then design the contexts, predicates
 - Test – create boundary tests as well

16/01/2008

Copyright (c) 2008
PeteFinnigan.com Limited

30

