

UKOUG DBMS SIG, March 17th 2009

Using Oracle VPD In The Real World

By
Pete Finnigan

Updated Friday, 24th February 2009

Why Am I Qualified To Speak

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



Agenda

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

“Whilst VPD is a security solution, security solutions must also be secured themselves and unfortunately they also increase the attack surface”

What Is VPD – Many Names. 😊

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

“Used by Oracle themselves in Label Security (the Trusted Oracle replacement) and also Database Vault”

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

- Oracles on-line documentation
- 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.

18/03/2009

Copyright (c) 2009
PeteFinnigan.com Limited

7

Securing VPD

- Leaking predicates
- Leaking policies
- “R”ole “B”ased “AC”cess (RBAC) on VPD structure / configuration
- Bypassing VPD by means of exception
- SQL Injection issues
- Direct data access

“Remember: An important concept in using security features is to ensure that the security feature itself is also secure”

18/03/2009

Copyright (c) 2009
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)

“In Oracle security we must think in layers: the object, the meta-data, the access rights, different paths to the data or to the “right”....”

18/03/2009

Copyright (c) 2009
PeteFinnigan.com Limited

9

Create A Simple Policy

- See code <http://www.petefinnigan.com/vpd2.sql>
- Create a user PXF,
 - Grant some privileges,
 - 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

18/03/2009

Copyright (c) 2009
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 => DMVSYS (ADM = NO)
User => XDB (ADM = NO)
```

18/03/2009

Copyright (c) 2009
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 DEPTNO
COMM
-----
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

18/03/2009

Copyright (c) 2009
PeteFinnigan.com Limited

12

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
    
```

18/03/2009

Copyright (c) 2009
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.

18/03/2009

Copyright (c) 2009
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 grantees,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
    
```

18/03/2009

Copyright (c) 2009
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”

18/03/2009

Copyright (c) 2009
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.
    
```

18/03/2009

Copyright (c) 2009
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
COMM
-----
DEPTNO
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
    
```

18/03/2009

Copyright (c) 2009
PeteFinnigan.com Limited

24

Example (3)

```
Repeat 464 Times
8229D0 00080800 00080800 2359C203 4C494006 [.....PF MIT]
8229D0 05524540 52454043 48C2034B B6770753 [LER_CLERK_INS.W.]
8229D0 01011701 08C30303 08C103FF 0308082C [.....]
8229D0 040358C2 44524F46 414E4107 5453594C [P_FORD_ANALYST]
8229E0 4146C203 0C2B7707 01011013 FF1E3092 [L.C.W.]
8229E10 2C15C102 C2020800 414A0550 0553454D [.....P JAMES.]
8229E10 52454043 48C2034B B6770753 0101101C [CLERK_Mc.W....]
8229E10 04C20301 C102FF33 08082C1E 4D4FC203 [.....]
8229E10 4144410F 48083440 48083440 5245C203 [ADAMS_CLERK_INS]
8229E0 05B87707 01010117 FF0C0202 2C15C102 [M.....TURNER.S.]
8229E0 02080800 4808204E 4808204E 41530802 [.....]
8229E0 4D534540 C2034E41 7707014D 01080905 [LESMAN_Mc.W....]
8229E0 02080800 08080110 08080110 4FC20308 [.....]
8229E0 49480428 5008474E 49534552 54484544 [L_KING_PRESIDENT]
8229E0 B67707FF 0101110B 33C20101 08C102FF [M.....]
8229E0 03080800 055A48C2 544F4351 4E410754 [.....NY SCOTT_AM]
8229E0 53594C41 4C203034 B6770753 01011104 [ALYST_LC.W....]
8229E0 18C20301 15C202FF 03080800 055A48C2 [.....INS.]
8229E0 52414C43 414D074B 4547414E 4FC20352 [CLARK_MANAGER_O]
8229E0 B67707FF 01010906 15C20101 C102FF33 [M.....]
8229E0 08082C0B 634DC203 414C4205 4D07454B [M.....]
8229E0 47414E41 C203034E 7707014E 01010305 [MANAGER_OI.W..]
8229E0 C2030101 02FF331D 002C1E01 4DC20308 [.....]
8229E0 414D074B 48083440 48083440 41530802 [P MARTIN_SALESM]
8229E0 4DC2034E B6770753 01011008 0DC20101 [N_Mc.W....L.C.]
8229E0 0C203031 2C15C102 C2030800 4A08344C [P.....L.C.]
8229E0 53454E4E 4E414D07 52454741 284FC203 [ONES_MANAGER_OI]
8229E0 48087707 01011012 4C1E0203 15C102FF [M.....L.]
8229E0 03080800 041640C2 48524157 4C415308 [.....L WARD_SAL]
8229E0 414E414E 4C203034 B6770753 01011602 [ERMAN_Mc.W....]
8229E0 0DC20301 06C20323 2C15C102 C2030800 [.....]
8229E0 41054448 4E454C4C 4C415308 414D5745 [RD_ALLEN_SALESM]
```

Screens Can Break

- Certification and Support for Third party products - <http://blogs.oracle.com/schan/newsItems/departments/extendingApps/2006/05/18#a200>
- Adding 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

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...

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!

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

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

