

# Database Vault Without Database Vault

---

# Legal Notice

---

## Secure Your Data Or Bust

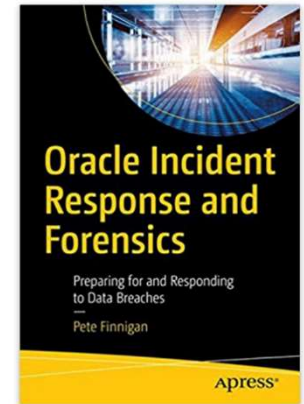
Published by  
PeteFinnigan.com Limited  
Tower Court  
3 Oakdale Road  
York  
England, YO30 4XL

Copyright © 2022 by PeteFinnigan.com Limited

No part of this publication may be stored in a retrieval system, reproduced or transmitted in any form by any means, electronic, mechanical, photocopying, scanning, recording, or otherwise except as permitted by local statutory law, without the prior written permission of the publisher. In particular this material may not be used to provide training of any type or method. This material may not be translated into any other language or used in any translated form to provide training. Requests for permission should be addressed to the above registered address of PeteFinnigan.com Limited in writing.

**Limit of Liability / Disclaimer of warranty.** This information contained in this course and this material is distributed on an “as-is” basis without warranty. Whilst every precaution has been taken in the preparation of this material, neither the author nor the publisher shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions or guidance contained within this course.

**TradeMarks.** Many of the designations used by manufacturers and resellers to distinguish their products are claimed as trademarks. Linux is a trademark of Linus Torvalds, Oracle is a trademark of Oracle Corporation. All other trademarks are the property of their respective owners. All other product names or services identified throughout the course material are used in an editorial fashion only and for the benefit of such companies with no intention of infringement of the trademark. No such use, or the use of any trade name, is intended to convey endorsement or other affiliation with this course.



## Pete Finnigan – Background, Who Am I?

---

- Oracle Security specialist and researcher
- CEO and founder of PeteFinnigan.com Limited in February 2003
- Writer of the longest running Oracle security blog
- Author of the Oracle Security step-by-step guide and “Oracle Expert Practices”, “Oracle Incident Response and Forensics” books
- Oracle ACE for security
- Member of the OakTable
- Speaker at various conferences
  - UKOUG, PSOUG, BlackHat, more..
- Published many times, see
  - <http://www.petefinnigan.com> for links
- Influenced industry standards
  - And governments



# Agenda

---

- Part 1
  - What is Database Vault?
  - Components of Database Vault?
  - Hacking with Database Vault
- Part 2
  - What can we do to simulate the features of Database Vault without Database Vault
  - What is possible for free?



**PFCL**  
PETEFINNIGAN.COM LIMITED

## What is Database Vault?

# What Is Database Vault?

---

- Declarative security framework to allow fine grained access to database objects (tables, views, procedures...) grouped into realms
- Literally unlimited context based security rules can be added to control access to any (well almost any) database object or command
- Default use is to protect against **SYSTEM ANY** privileges
- Because it is “built-in” to the database kernel it is harder to bypass
- Pre-built shipped realms protect risky parameter changes and the data dictionary and more
- Separation of duties are added by default by creation of a security administrator, user account manager and vault owner
- SYS, SYSTEM and DBA are restricted in value
- Basic default hardening done on install



**PFCL**  
PETEFINNIGAN.COM LIMITED

# Main Database Vault Components

# The Main DV Components

---

- Factors
  - Individual elements to use in rules (e.g. IP Address)
- Rules
  - True/False questions for the database
- Rule Sets
  - Groups of rules (Also results in True/False – with AND/OR)
- Realms
  - Protect database objects (**uses rules, factors**)
- Command Rules
  - Protect access to SQL commands (e.g. CONNECT) (**uses rules, factors**)
- Secure Application Roles (SAR)
  - Protective access to enable a role (**uses rules, factors**)





**PFCL**  
PETEFINNIGAN.COM LIMITED

## Hacking with Database Vault?

# Hacking The Sample Database With Realm

## Oracle Security Services

October 27, 2013

X  
Filed under: Uncategorized — pete @ 12:00 am

CardNumber-Aaron-Newman-3742112366758976

X  
Filed under: Uncategorized — pete @ 12:00 am

CardNumber-David-Litchfield-4049657443219878

X  
Filed under: Uncategorized — pete @ 12:00 am

CardNumber-Laszlo-Toth-4049990855468731

X  
Filed under: Uncategorized — pete @ 12:00 am

CardNumber-Pete-Finnigan-4049877198543457

X  
Filed under: Uncategorized — pete @ 12:00 am

CardNumber-Zulia-Finnigan-3742345698766678

pages  
about  
contact page

blogroll  
oracle security expertise  
pfclscan

categories:  
uncategorized  
uncategorized  
oracle security  
general

search:

archives:  
march 2008  
may 2008  
october 2013  
december 2013

meta:  
login  
rss  
comments rss  
valid xhtml  
xfn  
ob

```
x%'))))a)**/union/**/select/**/33,1,to_timest
amp('27-OCT-13'),to_timestamp('27-OCT-
13'),'CardNumber-||first_name||'-
'last_name||'-
'||orablog_crypto.decrypt(pan),'x',0,null,'publi
sh','open','open',null,'name',null,null,to_time
stamp('27-OCT-13'),to_timestamp('27-OCT-
13'),null,0,null,0,null,null,0,6/**/from/**/orabl
og.credit_card--
```

[Comments \(0\)](#)

[Comments \(0\)](#)

[Comments \(0\)](#)

[Comments \(0\)](#)

[Comments \(0\)](#)

# Hacking The Sample Database With Realm

Connect to the database as a user with just CREATE SESSION and exploit a vulnerable package (CUSTA) owned by ORABLOG and read card details

```
pause  
exec orablog.custa('x' union select username from all_users--');  
exec orablog.custa('x' union select orablog.bof_kkrc.dr(cc34) from orablog.bof_pay_details--');
```

prompt press any key to continue....

PL/SQL procedure successfully completed.

name:=[3742345698766678]

name:=[4049877198543457]

PL/SQL procedure successfully completed.

press any key to continue....

- Low privileged database user can see data in the BOF application

# Hacking The Sample Database With Realm

---

Connect as a DBA with the DBA role and simply select credit card details – no hacking needed as we use SYSTEM ANY

```
SQL> select * from orablog.bof_pay_details;  
select * from orablog.bof_pay_details  
*
```

```
ERROR at line 1:  
ORA-01031: insufficient privileges
```

```
SQL>  
SQL> prompt decrypt the cards  
decrypt the cards  
SQL> select name_on_card,orablog.bof_kkrc.dr(cc34) pan  
2 from orablog.bof_pay_details;  
from orablog.bof_pay_details  
*
```

```
ERROR at line 2:  
ORA-01031: insufficient privileges
```

- DV has some effect BUT only for SYSTEM ANY
- BUT, the DBA could find a way around this by doing the same hack as a lower privileged user in the last slide

Hmmm, the apps are now broken; we need to add ORABLOG to the realm but it defeats the object; if we hack the database again; same result

# Add A Mandatory Realm To ORABLOG Instead

```
SQL> exec dbms_macadm.delete_realm('BOF Realm');
```

PL/SQL procedure successfully completed.

```
SQL>
SQL> -- create the BOF realm
SQL> begin
2     dbms_macadm.create_realm(
3         realm_name => 'BOF Realm',
4         description => 'Protect BOF objects',
5         enabled => dbms_macutl.g_yes,
6         audit_options => dbms_macutl.g_realm_audit_fail,
7         realm_type => 1);
8 end;
9 /
```

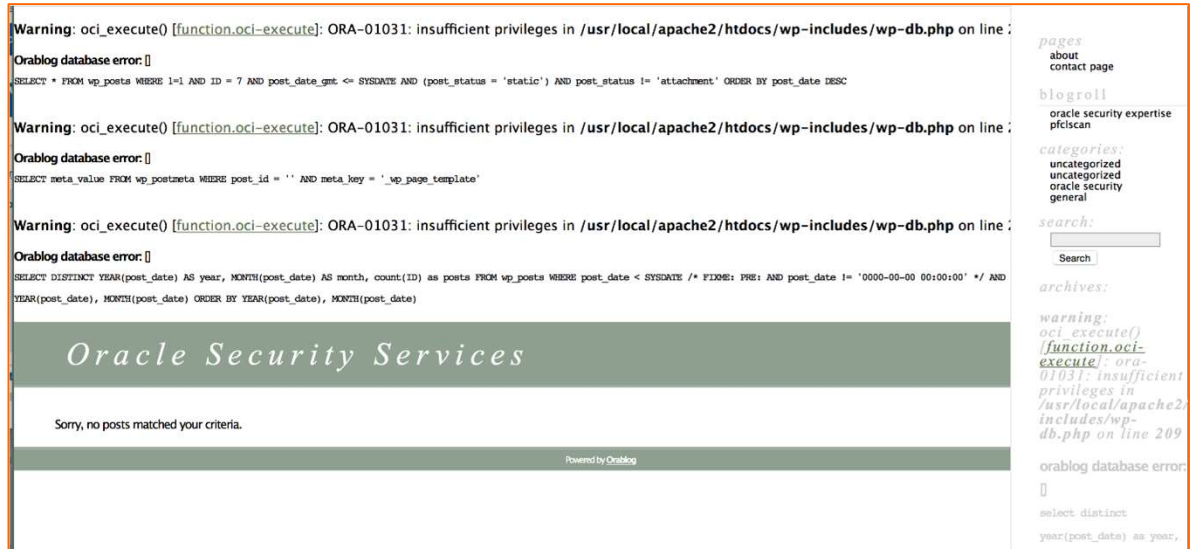
PL/SQL procedure successfully completed.

```
SQL>
SQL> -- add the objects to the realm
SQL> begin
2     dbms_macadm.add_object_to_realm(
3         realm_name => 'BOF Realm',
4         object_owner => 'ORABLOG',
5         object_type => '%',
6         object_name => '%');
7 end;
8 /
```

PL/SQL procedure successfully completed.

```
SQL> select name, realm_type from dvsys.dba_dv_realm;
```

```
NAME
-----
Oracle Database Vault
Database Vault Account Management
Oracle Enterprise Manager
Oracle Default Schema Protection Realm
Oracle System Privilege and Role Management Realm
Oracle Default Component Protection Realm
BOF Realm
```



The screenshot shows a search result on the Oracle Security Services blog. The search query was `oci_execute()`. The results show a warning: `Warning: oci_execute() [function.oci-execute]: ORA-01031: insufficient privileges in /usr/local/apache2/htdocs/wp-includes/wp-db.php on line 209`. The page also displays a search bar and a list of categories including 'uncategorized', 'oracle security', and 'general'.



**Warning:** ociexecute() [function.ociexecute]: ORA-01031: insufficient privileges in /usr/local/apache2/htdocs/bof\_address.php on line 78

**Warning:** ocifetchstatement() [function.ocifetchstatement]: ORA-24374: define not done before fetch or execute and fetch in /usr/local/apache2/htdocs/bof\_address.php on line 80

**BOF: Back Office Customer Management - PeteFinnigan.com Limited**

MANDATORY

## DV Command Rule - Results

---

```
SQL> connect orablog/orablog@//192.168.56.94:1521/dvtst.localdomain  
ERROR:  
ORA-47306: 20403: SQL*Plus not allowed for ORABLOG from the Webserver
```

```
SQL> !hostname  
oe159orablog12
```

```
SQL> connect orablog/orablog@//192.168.56.94:1521/dvtst.localdomain  
Connected.  
SQL> !hostname  
Peters-MBP
```

- The rules are not perfect as we have implemented properly only for Orablog and not BOF but BOF has no client tools installed
- The client\_program\_name is not set from the server so we have used instead Module – but it would be better to use the hash
- Implementing factors, rules, rule sets and command rules or rule sets for realms is a large task when a lot of controls are needed



## What Can We Do to Simulate DV?

# What If: No Database Vault Available?

---

- If we do not have DV or It is not possible (i.e. SE/SE1/SE2) what can we do?
  - Replicate the technical features of DV?
  - Remove as much of the “problem” as possible that is solved by Database Vault?
- Start with a good security design
  - Aim for least rights
  - Aim for lock down
  - Aim for proper data access controls
  - Add context based security without DV
- Do not use defaults
- Consider application design changes
  - Code and data access levels





## What Do We Need To Do To Replicate DV?

---

- There are a lot of features in DV that we could use: Declarative API's, factors, realms, rules, SARs, Command rules and within these protect objects, commands, SoD, parameters and much much more...
- If we focus on three simple tasks to consider for replication:
  - SET ROLE, DBMS\_SESSION.SET\_ROLE to be able to create a SAR
  - ALTER SYSTEM to be able to detect a parameter change
  - System ANY to detect use of SELECT ANY TABLE (for instance)
- There is no way (supported) to “Trap” SET ROLE, ALTER SYSTEM or SELECT ANY TABLE
- ALTER SYSTEM is DDL But it is not trapped by a DDL trigger
- There is no simple way to detect SELECT and act upon it in real time
- Some actions can be detected such as CREATE, ALTER, DROP and most DDL
- There are many gaps in available techniques in a core database to replicate Database Vault



## Blocking A Select Statement

```
1  -- create a trigger on system.aud$ for select on credit_card
2  create or replace trigger sys.stk_aud_sel
3  after insert on system.aud$
4  for each row
5  begin
6      if (:new.obj$name='CREDIT_CARD' and :new.action#=3) then
7          raise_application_error(-20077,'You are not allowed to read this table');
8      end if;
9  --exception
10 -- when others then
11 --     null;
12 end;
13 /
```

```
SQL> connect orablog/orablog@//192.168.56.85:1521/bfora.localdomain
Connected.
SQL> select * from credit_card;
select * from credit_card
*
ERROR at line 1:
ORA-02002: error while writing to audit trail
ORA-00604: error occurred at recursive SQL level 1
ORA-20077: You are not allowed to read this table
ORA-06512: at "SYS.STK_AUD_SEL", line 3
ORA-04088: error during execution of trigger
```



# A Secure Application Role in SE

```
SQL> connect def_role/def_role@//192.168.56.85:1521/bfora.localdomain
Connected.
SQL> set role rdef;
set role rdef
*
ERROR at line 1:
ORA-02002: error while writing to audit trail
ORA-00604: error occurred at recursive SQL level 1
ORA-20079: SAR Check Failed -:ORA-20078: You are not allowed to enable the RDEF role
ORA-06512: at "SYS.STK_AUD_SAR", line 22
ORA-04088: error during execution of trigger 'SYS.STK_AUD_SAR'
```

```
32 create trigger sys.stk_aud_sar
33 after insert on system.aud$
34 for each row
35 begin
36     if (:new.action#=#55) then
37         -- check for a SAR
38         declare
39             lv_proc varchar2(200);
40             lv_res number;
41             sar_failed exception;
42             pragma exception_init(sar_failed,-20078);
43         begin
44             select role_proc into lv_proc
45             from system.stk_sar_tab
46             where role_name=:new.obj$name;
47             -- if lv_proc was found then execute it
48             execute immediate 'begin :val:= '||lv_proc||';end;' using out lv_res;
49             if (lv_res=1) then
50                 null;
51             else
52                 raise_application_error(-20078,'You are not allowed to enable the '||:new.obj$name||' role');
53             end if;
54         exception
55         when sar_failed then
56             raise_application_error(-20079,'SAR Check Failed -: '||sqlerrm);
57             -- if error i.e. 1403 then do nothing
58         when others then
59             null;
60         end;
61     end if;
62 end;
63 /
```

```
13 create function system.stk_rdef_sar return number as
14     lv_ip varchar2(100);
15 begin
16     select sys_context('USERENV','IP_ADDRESS') into lv_ip from dual;
17     if(lv_ip='192.168.56.2') then
18         return 1;
19     else
20         return 0;
21     end if;
22 end;
```



## What Can We Do for Free?

## But What Are We Really Trying to Achieve?

---

- Are we really trying to replicate DV in its technical functionality?
- Or are we really trying to replicate the results of applying DV?
- Or even do better?
- **YES, We want to replicate the results not the technical design**
- We can achieve this with:
  - Careful security design
  - Some code
  - Privilege management especially around SYS, SYSTEM, DBA...
- We can do context based security without DV
- What is the risk trying to simulate DV?
  - Should be low provided we have a good base design anyway

## Why Do We (Perceive We) Need System ANY

---

- **Needed for development/deployment of code?**
- Solutions used often is SYSTEM ANY for deployment as it is simple
- There is no grant select on orablog.tables.\* so system ANY is a good replacement BUT gives access to all data (except SYS)
- What other solutions exist:
  - Log on as the schema to deploy code
  - Use SYSTEM ANY but via a schema/protected PL/SQL API that you create – complex and hard to maintain
  - Direct grants on the schema objects but issues arise
    - How to create new objects in the same schema
    - Maintainability of rights
  - **Proxy to the schema**

## Context: View Based Security

---

- We can create VIEW BASED security to limit access to read data
    - A PL/SQL function allows tests to be made to check whether access is allowed or not
    - We could also check in this PL/SQL whether the privilege used is SELECT ANY by checking the users actual rights
    - This can block some ANY privileges
  - **BUT system ANY for select can access the base table.**
- Solution:**
- Revoke system ANY except for sys
  - Block SYSDBA access – The first versions of DV did this



## Context: DML Based Security

---

- We can apply the same “Realm” type ideas to block DML
- This cannot be overridden as this is added to the base table and this is not view based
- Again we could check for System ANY in the PL/SQL code by looking at the callers rights
- We can also make a mandatory realm – in part at least with context based code



## Separation of Duties (SoD)

---

- Separation of Duties does not need DV to enforce it
- Even with DV real people and database accounts need to be designed and a SoD matrix created to ensure separation exists for all interactive users
- Identify and make decisions on separation
  - Account Manager, Audit Trail Admin, Security Admin, Audit Viewer
- All of these can be implemented with design, least privilege
- Custom DBA role should be created
- SYSTEM should be locked, SYS should be blocked out as SYSDBA
- Reduce, remove SYSTEM ANY
- Use technical solutions to enforce security – DDL, ALTER... type system triggers
- Accountability and audit are needed

## Context: Blocking Parameter Changes

---

- Limit ALTER SYSTEM
- Audit use of ALTER SYSTEM
- Limit even from the DBA (should have custom role anyway and limited rights – NOT DBA, SYSDBA)
- Release SYS when needed but audit use of account
- Triggers on database start and stop to detect that a parameter has changed whilst database is up? – put it back?
- We could also protect spfile with chattr to make the file immutable but only on Linux



## Command Rule: Block SQL\*Plus - Webserver

```
133     program,  
134     os_user  
135 ) values (lv_username,lv_ip_address,lv_program,lv_os_user);  
136 commit;  
137  
138 if(lv_ip_address not in('192.168.56.91','192.168.56.89','192.168.56.1','192.168.56.85','192.168.56.90')) then  
139     -- the IP address is not allowed  
140  
141     insert into stk_login_error (login_date,error_line) values (sysdate,1);  
142     commit;  
143     RAISE_APPLICATION_ERROR(-20070,'NOT AUTHORISED FROM THIS HOST');  
144  
145 else  
146     -- test for web server and not apache and not httpd  
147     if( (lv_ip_address in('192.168.56.89')) and  
148         (upper(lv_program)<>'HTTPD@OEL59ORABLOG.LOCALDOMAIN (TNS V1-V3)') and  
149         (upper(lv_os_user)<>'APACHE')) then  
150  
151         -- web server and not httpd and not apache OS user  
152         insert into stk_login_error (login_date,error_line) values (sysdate,2);  
153         commit;  
154         RAISE_APPLICATION_ERROR(-20071,'NOT AUTHORISED WITH THESE DETAILS');  
155     else  
156         -- we must be on the admin PC or the actual database server  
157         insert into stk_login_error (login_date,error_line) values (sysdate,3);  
158         commit;  
159     end if;  
160 end if;  
161 -- record that we got here  
162 insert into stk_login_error (login_date,error_line) values (sysdate,4);  
163 commit;  
164 exception  
165 when others then  
166     insert into stk_login_error (login_date,error_line) values (sysdate,5);  
167     commit;  
168     RAISE_APPLICATION_ERROR(-20073,sqlerrm);  
169 --  
170 end login_dba;  
171 /
```

- We can perfectly replicate the protection we had in DV with a logon trigger
- We can also use valid node checking but this is not granular
- In this example the httpd still works but SQL\*Plus from the webserver is blocked

# Conclusions

---

- Good security design is needed from the start
- Good lock down is needed from the start
- Don't use SYSTEM ANY
  - Don't use SYS, SYSTEM and DBA
  - Make changes via proxy to the schema
  - Do not allow DBAs to look at data
- Database Vault is Duct Tape if you do not take care to lock down and secure your data first
- Even if you use DV it must be added on top of good secure design
- So we **MUST ALWAYS DESIGN SECURITY FIRST** before using additional tools such as DV or not with SE
- DV is built-in so harder to bypass

# Questions?

---

Any Final Questions?

# Database Vault Without Database Vault

---