PeteFinnigan.com Limited
Oracle Security

You Don’t Have Database Vault

So, What Can You Do Instead?

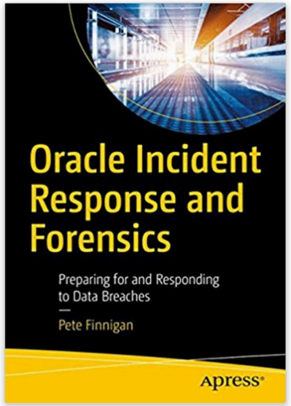
Legal Notice

Database Vault Or Not!

Published by
PeteFinnigan.com Limited
9 Beech Grove
Acomb
York
England, YO26 5LD

Limit of Liability / Disclaimer of warranty. This information contained in 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 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 material.



Oracle Incident Response and Forensics

Preparing for and Responding to Data Breaches

Pete Finnigan

Apress

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

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

What Is New In 12c In Database Vault

- Pre-installed software (DV and OLS)
- Works with Multitenant
  - DV must be enabled in the root container before a pluggable container
  - Management with common accounts or delegated or local
- Mandatory realms to protect against direct grants and object owner
  - Was possible in 11g but only with very complex rules
- Privilege analysis allows discovery of used or not used rights
- Simple basic hardening is better in 12.1.0.2 and 12.2.0.1 core database
- Shipped policies for products such as E-Business Suite and SAP and Peoplesoft
- Unified audit trail and default audit for DV and OLS

Default Basic Hardening

- When DV is installed Oracle does some basic hardening and securing automatically for you
- This is described here - https://docs.oracle.com/database/121/DVADM/dv_impact.htm#DVADM70123
- If you are in a multitenant database the hardening is applied to the root container and all pluggable containers are affected
  - If you do not want DV in a PDB and do not agree with these changes you must put them back manually; there are also issues with RAC nodes where manual hardening is needed in some cases on other nodes
- The changes include
  - Parameters changed
  - DBA, IMP_FULL_DATABASE, EXECUTE_CATALOG_ROLE, SCHEDULER_ADMIN,
  - UTL_FILE EXECUTE revoked from PUBLIC
  - ALTER / CREATE / DROP on USER / PROFILE restricted
- SYS and SYSTEM cannot change passwords anymore

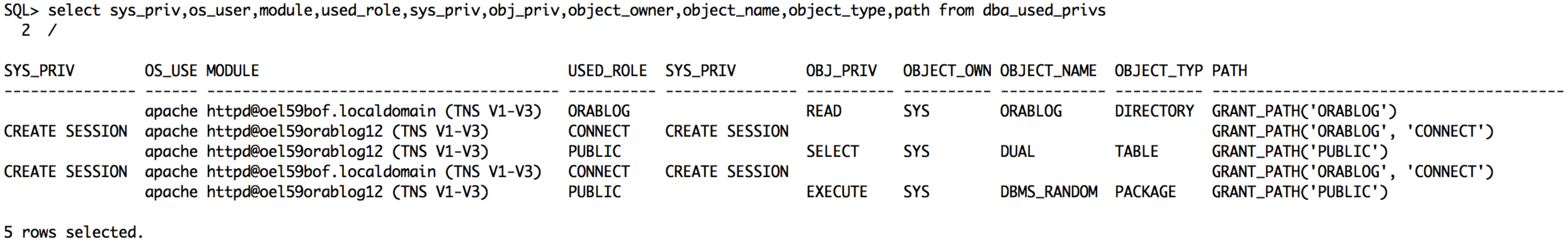
The Main DV Components

- 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)

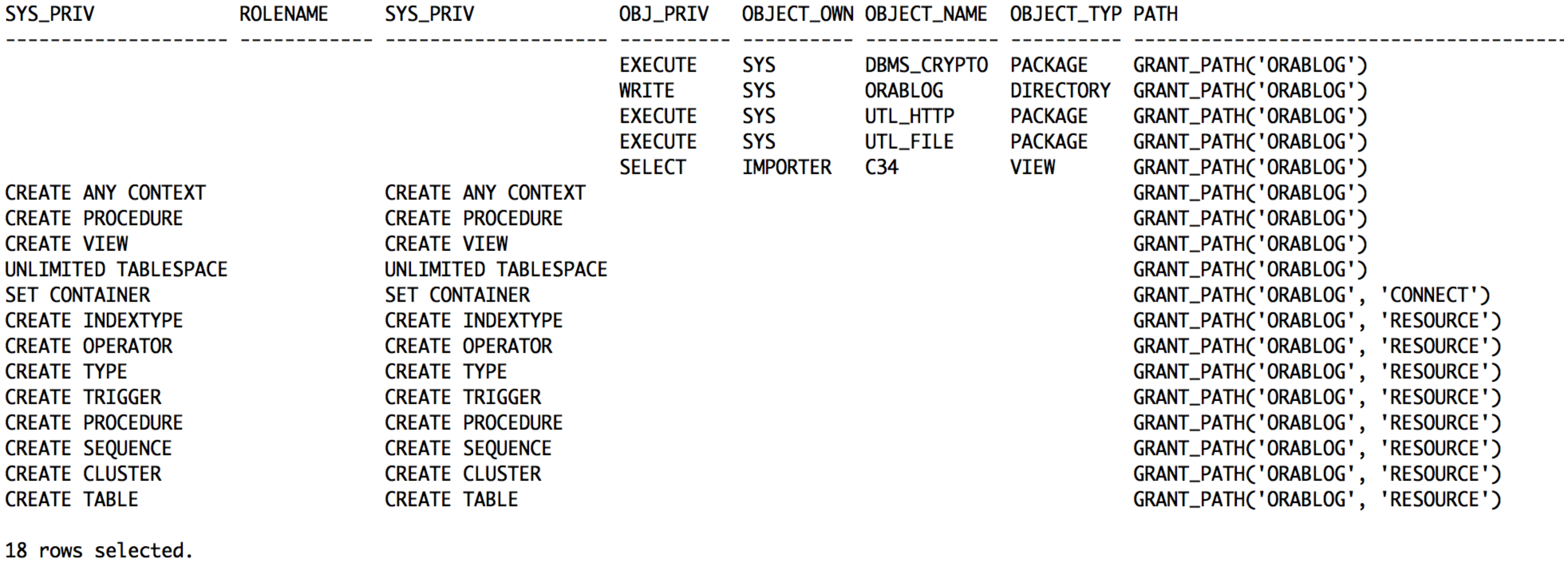
Some limits:

- DBMS_CRYPTO, UNLIMITED TABLESPACE, UTL_FILE
- Run time rights, Maintenance, Create
- Duplicate Rights not distinguished

Privilege Analysis



```
SQL> select sys_priv,os_user,module,used_role,sys_priv,obj_priv,object_owner,object_name,object_type,path from dba_used_privs
  2  /

SYS_PRIV        OS_USE MODULE                                   USED_ROLE  SYS_PRIV        OBJ_PRIV   OBJECT_OWN OBJECT_NAME  OBJECT_TYP PATH
--------------- ------ ---------------------------------------- ---------- --------------- ---------- ---------- ------------ ---------- ----------------------------------------
                apache httpd@oel59bof.localdomain (TNS V1-V3)   ORABLOG                    READ       SYS        ORABLOG      DIRECTORY  GRANT_PATH('ORABLOG')
CREATE SESSION  apache httpd@oel59orablog12 (TNS V1-V3)         CONNECT    CREATE SESSION                                                GRANT_PATH('ORABLOG', 'CONNECT')
                apache httpd@oel59orablog12 (TNS V1-V3)         PUBLIC                     SELECT     SYS        DUAL         TABLE      GRANT_PATH('PUBLIC')
CREATE SESSION  apache httpd@oel59bof.localdomain (TNS V1-V3)   CONNECT    CREATE SESSION                                                GRANT_PATH('ORABLOG', 'CONNECT')
                apache httpd@oel59orablog12 (TNS V1-V3)         PUBLIC                     EXECUTE    SYS        DBMS_RANDOM  PACKAGE    GRANT_PATH('PUBLIC')

5 rows selected.
```

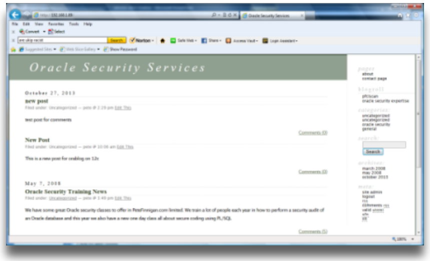


```
SYS_PRIV             ROLENAME     SYS_PRIV             OBJ_PRIV   OBJECT_OWN OBJECT_NAME  OBJECT_TYP PATH
-------------------- ------------ -------------------- ---------- ---------- ------------ ---------- ----------------------------------------
                                                       EXECUTE    SYS        DBMS_CRYPTO  PACKAGE    GRANT_PATH('ORABLOG')
                                                       WRITE      SYS        ORABLOG      DIRECTORY  GRANT_PATH('ORABLOG')
                                                       EXECUTE    SYS        UTL_HTTP     PACKAGE    GRANT_PATH('ORABLOG')
                                                       EXECUTE    SYS        UTL_FILE     PACKAGE    GRANT_PATH('ORABLOG')
                                                       SELECT     IMPORTER   C34          VIEW       GRANT_PATH('ORABLOG')
CREATE ANY CONTEXT                CREATE ANY CONTEXT                                                 GRANT_PATH('ORABLOG')
CREATE PROCEDURE                  CREATE PROCEDURE                                                   GRANT_PATH('ORABLOG')
CREATE VIEW                       CREATE VIEW                                                        GRANT_PATH('ORABLOG')
UNLIMITED TABLESPACE              UNLIMITED TABLESPACE                                               GRANT_PATH('ORABLOG')
SET CONTAINER                     SET CONTAINER                                                      GRANT_PATH('ORABLOG', 'CONNECT')
CREATE INDEXTYPE                  CREATE INDEXTYPE                                                   GRANT_PATH('ORABLOG', 'RESOURCE')
CREATE OPERATOR                   CREATE OPERATOR                                                    GRANT_PATH('ORABLOG', 'RESOURCE')
CREATE TYPE                       CREATE TYPE                                                        GRANT_PATH('ORABLOG', 'RESOURCE')
CREATE TRIGGER                    CREATE TRIGGER                                                     GRANT_PATH('ORABLOG', 'RESOURCE')
CREATE PROCEDURE                  CREATE PROCEDURE                                                   GRANT_PATH('ORABLOG', 'RESOURCE')
CREATE SEQUENCE                   CREATE SEQUENCE                                                    GRANT_PATH('ORABLOG', 'RESOURCE')
CREATE CLUSTER                    CREATE CLUSTER                                                     GRANT_PATH('ORABLOG', 'RESOURCE')
CREATE TABLE                      CREATE TABLE                                                       GRANT_PATH('ORABLOG', 'RESOURCE')

18 rows selected.
```

PeteFinnigan.com Limited
Oracle Security

- Oracle Linux
- Oracle Database
- Applications (Front Facing Website, back office customer processing)

My Sample Application Architecture



Oracle Security Services

October 27, 2013

new post

Filed under: Uncategorized — pete @ 2:29 pm Edit This

test post for comments

Comments (0)

New Post

Filed under: Uncategorized — pete @ 10:06 am Edit This

This is a new post for orablog on 12c

Comments (0)

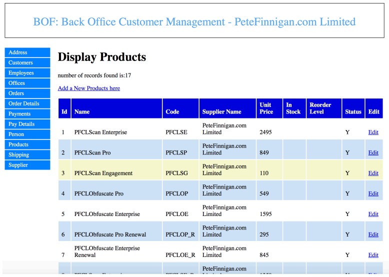
May 7, 2008

Oracle Security Training News

Filed under: Uncategorized — pete @ 1:49 pm Edit This

We have some great Oracle security classes to offer in PeteFinnigan.com limited. We train a lot of people each year in how to perform a security audit of an Oracle database and this year we also have a new one day class all about secure coding using PL/SQL.

Comments (5)



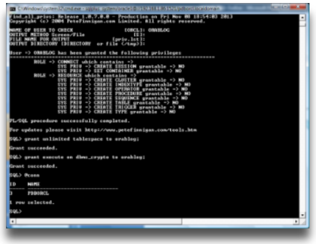
BOF: Back Office Customer Management - PeteFinnigan.com Limited

Display Products

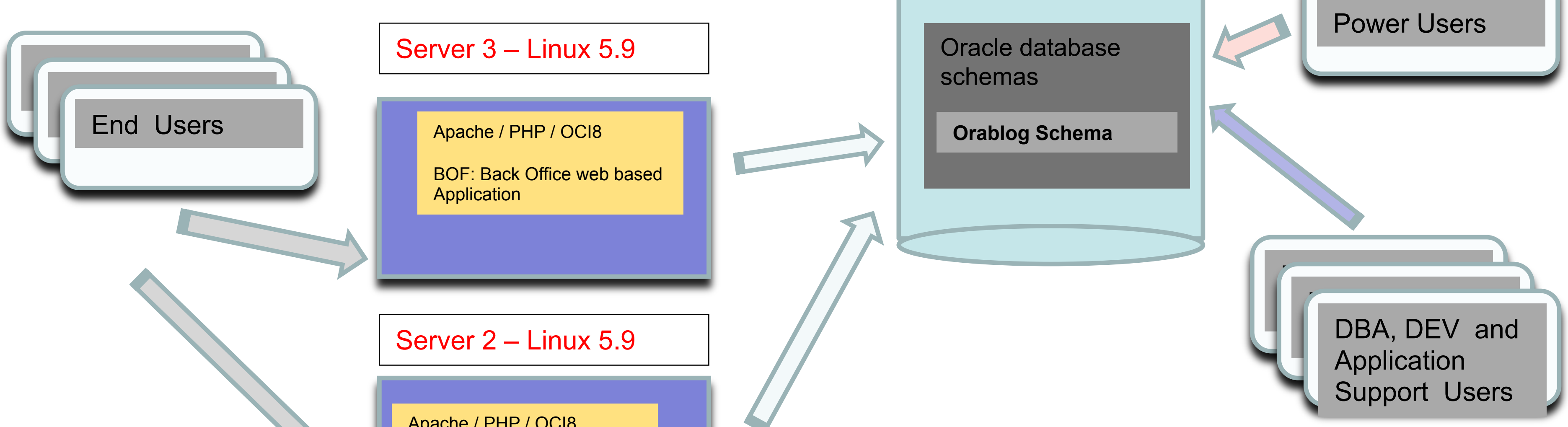
number of records found is:17

Add a New Products here

Id	Name	Code	Supplier Name	Unit Price	In Stock	Reorder Level	Status	Edit
1	PFCLScan Enterprise	PFCLSE	PeteFinnigan.com Limited	2495			Y	Edit
2	PFCLScan Pro	PFCLSP	PeteFinnigan.com Limited	849			Y	Edit
3	PFCLScan Engagement	PFCLSG	PeteFinnigan.com Limited	110			Y	Edit
4	PFCLObfuscate Pro	PFCLOP	PeteFinnigan.com Limited	549			Y	Edit
5	PFCLObfuscate Enterprise	PFCLOE	PeteFinnigan.com Limited	1595			Y	Edit
6	PFCLObfuscate Pro Renewal	PFCLOP_R	PeteFinnigan.com Limited	295			Y	Edit
7	PFCLObfuscate Enterprise Renewal	PFCLOE_R	PeteFinnigan.com Limited	845			Y	Edit
			PeteFinnigan.com					



```
find_all_privs: Release 1.0.7.0.0 - Production on Fri Nov 08 18:54:03 2013
Copyright (c) 2004 PeteFinnigan.com Limited. All rights reserved.

NAME OF USER TO CHECK                 [ORCL]: ORABLOG
OUTPUT METHOD Screen/File                [S]:
FILE NAME FOR OUTPUT              [priv.lst]:
OUTPUT DIRECTORY [DIRECTORY  or file (/tmp)]:

User => ORABLOG has been granted the following privileges
====================================================================
        ROLE => CONNECT which contains =>
                SYS PRIV => CREATE SESSION grantable => NO
                SYS PRIV => SET CONTAINER grantable => NO
        ROLE => RESOURCE which contains =>
                SYS PRIV => CREATE CLUSTER grantable => NO
                SYS PRIV => CREATE INDEXTYPE grantable => NO
                SYS PRIV => CREATE OPERATOR grantable => NO
                SYS PRIV => CREATE PROCEDURE grantable => NO
                SYS PRIV => CREATE SEQUENCE grantable => NO
                SYS PRIV => CREATE TABLE grantable => NO
                SYS PRIV => CREATE TRIGGER grantable => NO
                SYS PRIV => CREATE TYPE grantable => NO

PL/SQL procedure successfully completed.

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

SQL> grant unlimited tablespace to orablog;

Grant succeeded.

SQL> grant execute on dbms_crypto to orablog;

Grant succeeded.

SQL> @conn

ID    NAME
----- ------------------------------
3     PDBORCL

1 row selected.

SQL>
```



Server 1 – Linux 5.9

Oracle database schemas

Orablog Schema

Power Users

End Users

Server 3 – Linux 5.9

Apache / PHP / OCI8

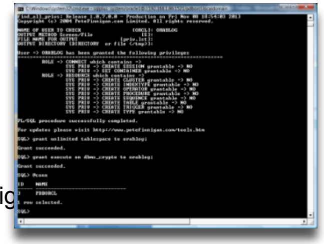
BOF: Back Office web based Application

Server 2 – Linux 5.9

Apache / PHP / OCI8

Orablog: Web Based CMS Application

DBA, DEV and Application Support Users



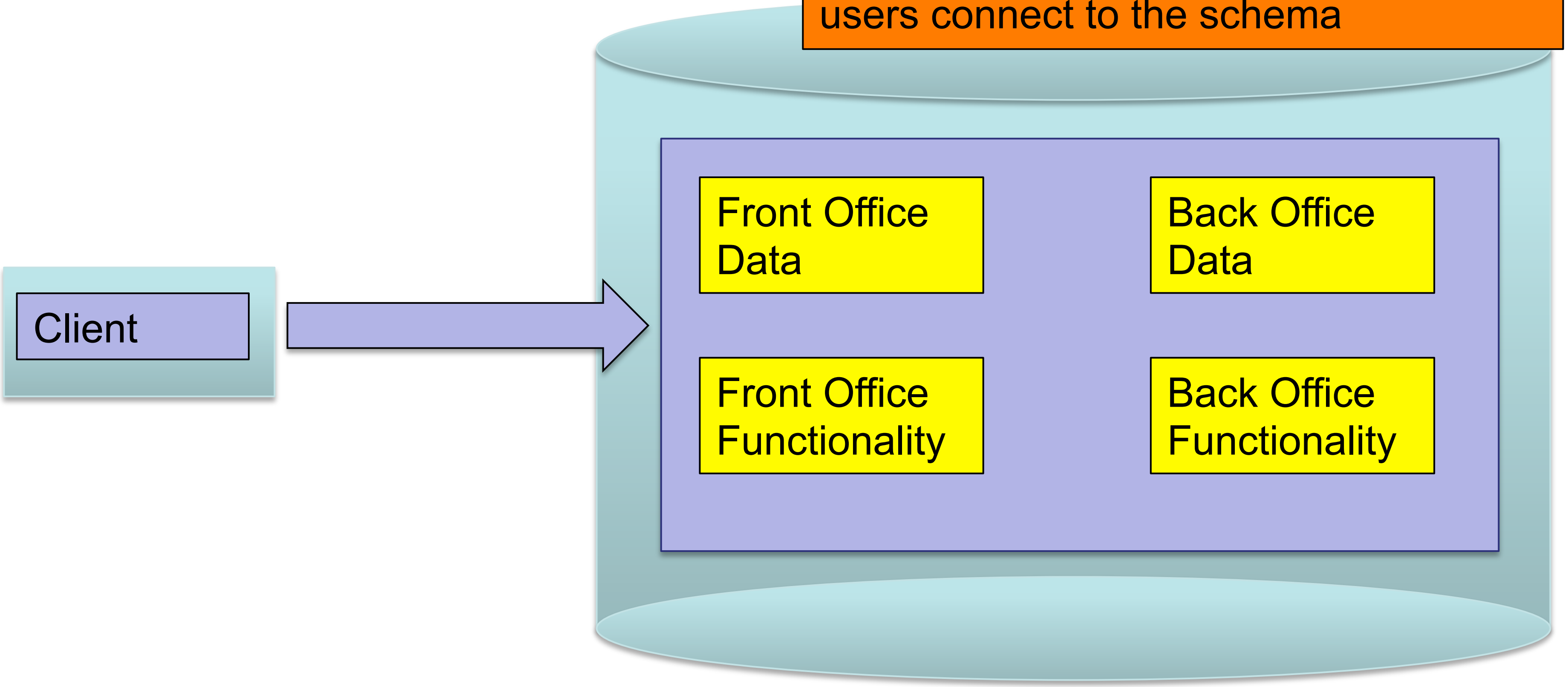
```
find_all_privs: Release 1.0.7.0.0 - Production on Fri Nov 08 18:54:03 2013
Copyright (c) 2004 PeteFinnigan.com Limited. All rights reserved.

NAME OF USER TO CHECK                 [ORCL]: ORABLOG
OUTPUT METHOD Screen/File                [S]:
FILE NAME FOR OUTPUT              [priv.lst]:
OUTPUT DIRECTORY [DIRECTORY  or file (/tmp)]:

User => ORABLOG has been granted the following privileges
====================================================================
        ROLE => CONNECT which contains =>
                SYS PRIV => CREATE SESSION grantable => NO
                SYS PRIV => SET CONTAINER grantable => NO
        ROLE => RESOURCE which contains =>
                SYS PRIV => CREATE CLUSTER grantable => NO
                SYS PRIV => CREATE INDEXTYPE grantable => NO
                SYS PRIV => CREATE OPERATOR grantable => NO
                SYS PRIV => CREATE PROCEDURE grantable => NO
                SYS PRIV => CREATE SEQUENCE grantable => NO
                SYS PRIV => CREATE TABLE grantable => NO
                SYS PRIV => CREATE TRIGGER grantable => NO
                SYS PRIV => CREATE TYPE grantable => NO

PL/SQL procedure successfully completed.

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

SQL> grant unlimited tablespace to orablog;

Grant succeeded.

SQL> grant execute on dbms_crypto to orablog;

Grant succeeded.

SQL> @conn

ID    NAME
----- ------------------------------
3     PDBORCL

1 row selected.

SQL>
```

Data Domains – BAD!!

All data, front and back office are in the same schema; ORABLOG

All functionality for front and back office are in the same schema

The web application and back office users connect to the schema

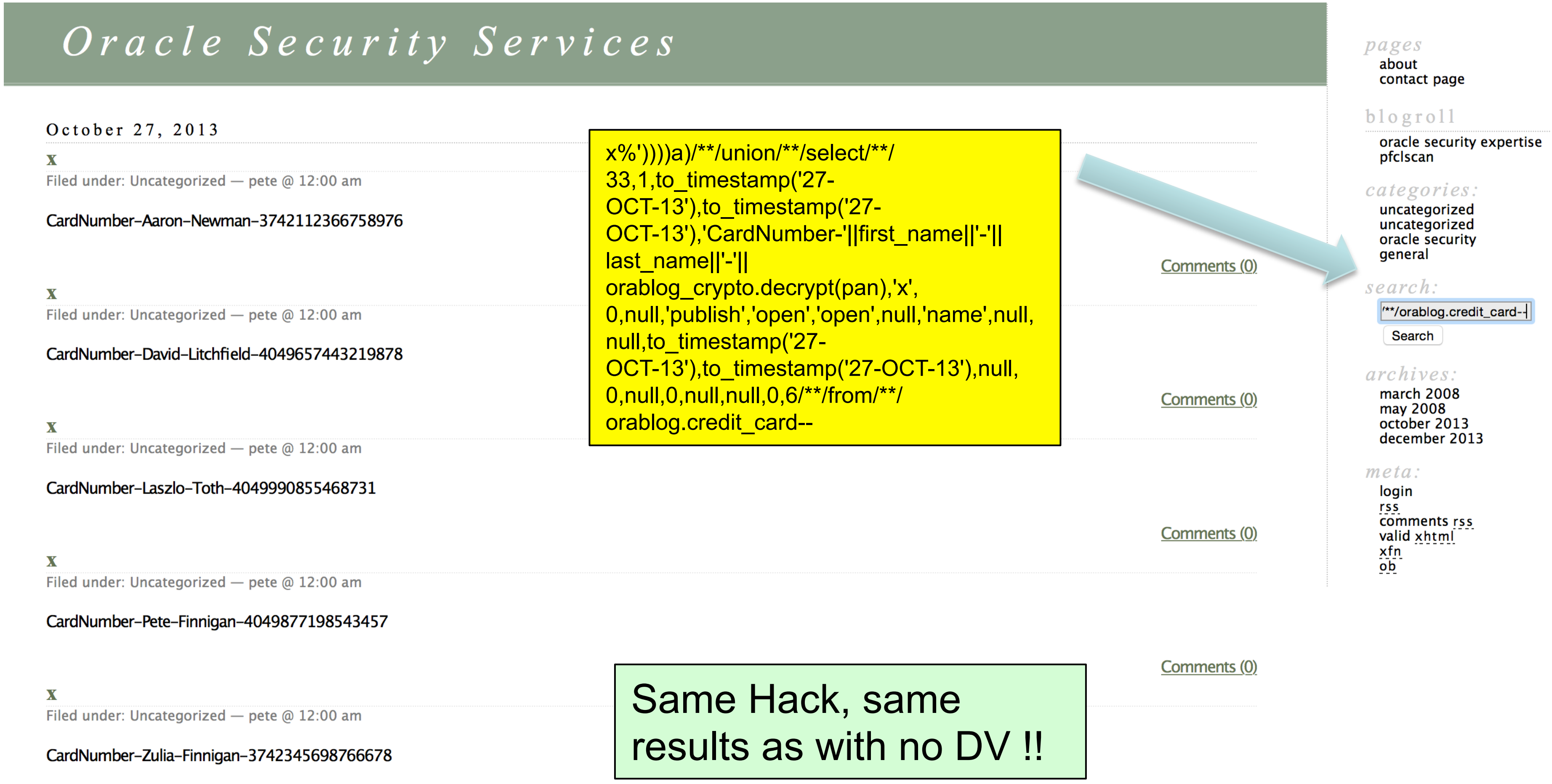


Hacking My Sample Database / Applications

- Three levels of Hacking
  - As a website un-authenticated user
  - As a database user with just CREATE SESSION
  - As a DBA

As an un-athenticated web user

Hacking The Sample Database With Realm



Oracle Security Services

October 27, 2013

x

Filed under: Uncategorized — pete @ 12:00 am

CardNumber-Aaron-Newman-3742112366758976

Comments (0)

x

Filed under: Uncategorized — pete @ 12:00 am

CardNumber-David-Litchfield-4049657443219878

Comments (0)

x

Filed under: Uncategorized — pete @ 12:00 am

CardNumber-Laszlo-Toth-4049990855468731

Comments (0)

x

Filed under: Uncategorized — pete @ 12:00 am

CardNumber-Pete-Finnigan-4049877198543457

Comments (0)

x

Filed under: Uncategorized — pete @ 12:00 am

CardNumber-Zulia-Finnigan-3742345698766678

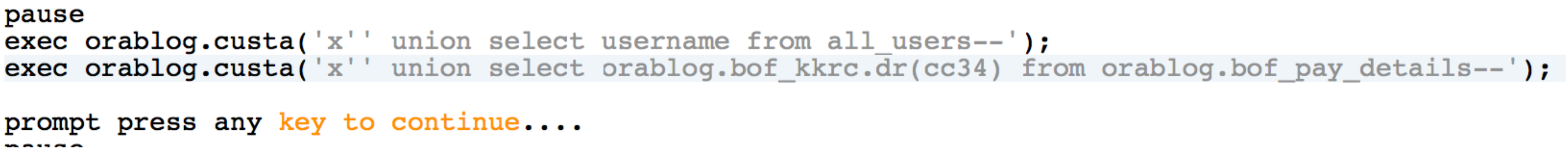
```
x%')))a)/**/union/**/select/**/
33,1,to_timestamp('27-
OCT-13'),to_timestamp('27-
OCT-13'),'CardNumber-'||first_name||'-'||
last_name||'-'||
orablog_crypto.decrypt(pan),'x',
0,null,'publish','open','open',null,'name',null,
null,to_timestamp('27-
OCT-13'),to_timestamp('27-OCT-13'),null,
0,null,0,null,null,0,6/**/from/**/
orablog.credit_card--
```

Same Hack, same results as with no DV !!

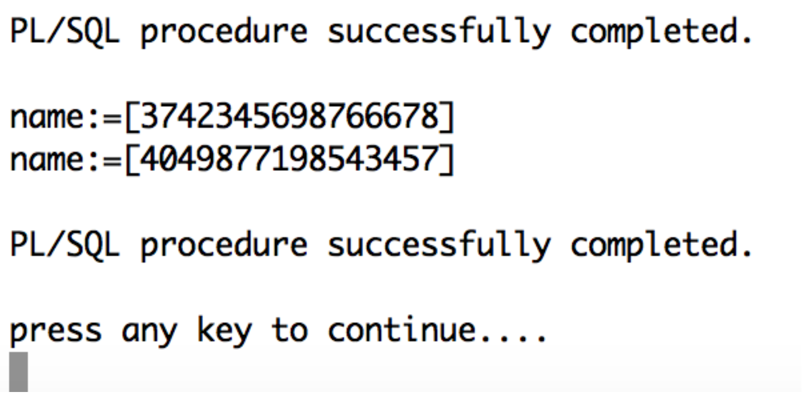
As a low power database user

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....
pause
```



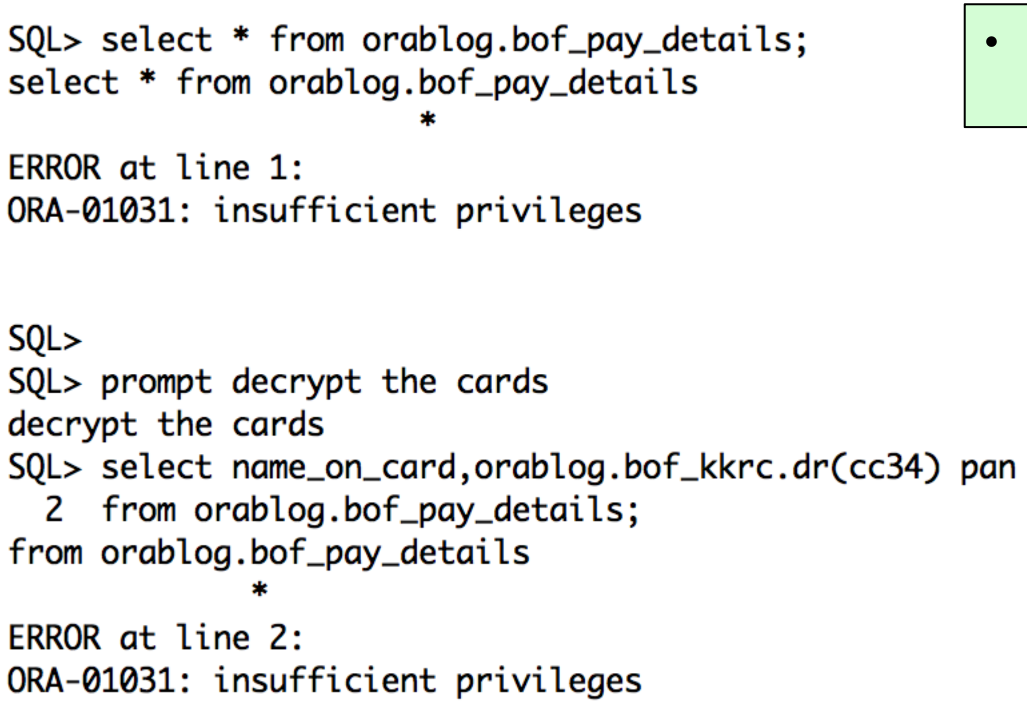
```
PL/SQL procedure successfully completed.

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

- DV has some effect BUT only for SYSTEM ANY

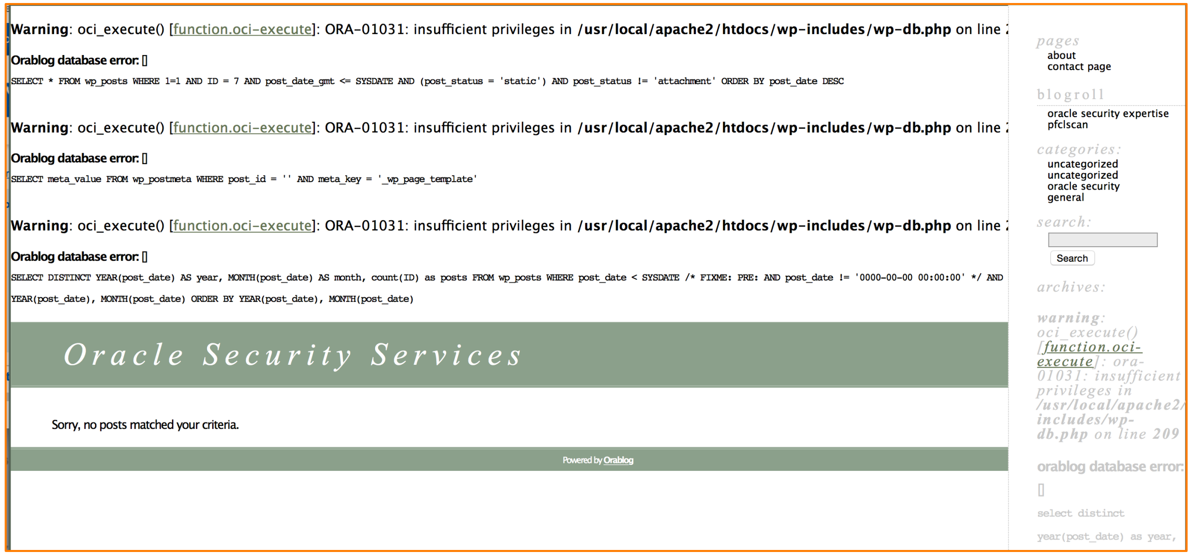


```
SQL> select * from orablog.bof_pay_details;
select * from orablog.bof_pay_details
                      *
ERROR at line 1:
ORA-01031: insufficient privileges


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
```

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                                                                             MANDATORY
```



Warning: oci_execute() [function.oci-execute]: ORA-01031: insufficient privileges in **/usr/local/apache2/htdocs/wp-includes/wp-db.php** on line

Orablog database error: []

SELECT * FROM wp_posts WHERE 1=1 AND ID = 7 AND post_date_gmt <= SYSDATE AND (post_status = 'static') AND post_status != 'attachment' ORDER BY post_date DESC

Warning: oci_execute() [function.oci-execute]: ORA-01031: insufficient privileges in **/usr/local/apache2/htdocs/wp-includes/wp-db.php** on line

Orablog database error: []

SELECT meta_value FROM wp_postmeta WHERE post_id = '' AND meta_key = '_wp_page_template'

Warning: oci_execute() [function.oci-execute]: ORA-01031: insufficient privileges in **/usr/local/apache2/htdocs/wp-includes/wp-db.php** on line

Orablog database error: []

SELECT DISTINCT YEAR(post_date) AS year, MONTH(post_date) AS month, count(ID) as posts FROM wp_posts WHERE post_date < SYSDATE /* FIXME: PRE: AND post_date != '0000-00-00 00:00:00' */ AND YEAR(post_date), MONTH(post_date) ORDER BY YEAR(post_date), MONTH(post_date)

Oracle Security Services

Sorry, no posts matched your criteria.

Powered by Orablog

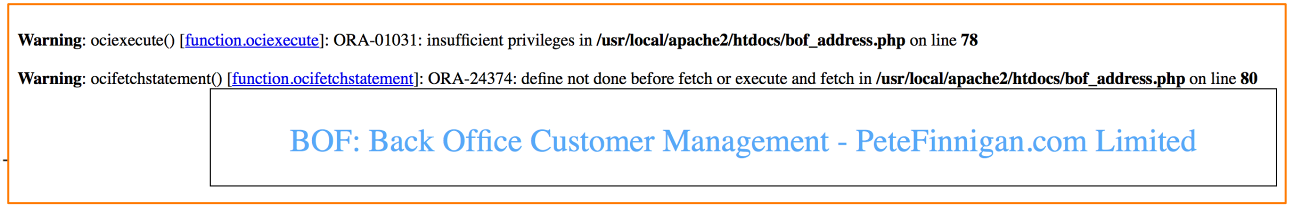
warning: oci_execute() [function.oci-execute]: ora-01031: insufficient privileges in /usr/local/apache2/ includes/wp-db.php on line 209

orablog database error:

[]

select distinct

year(post_date) as year,



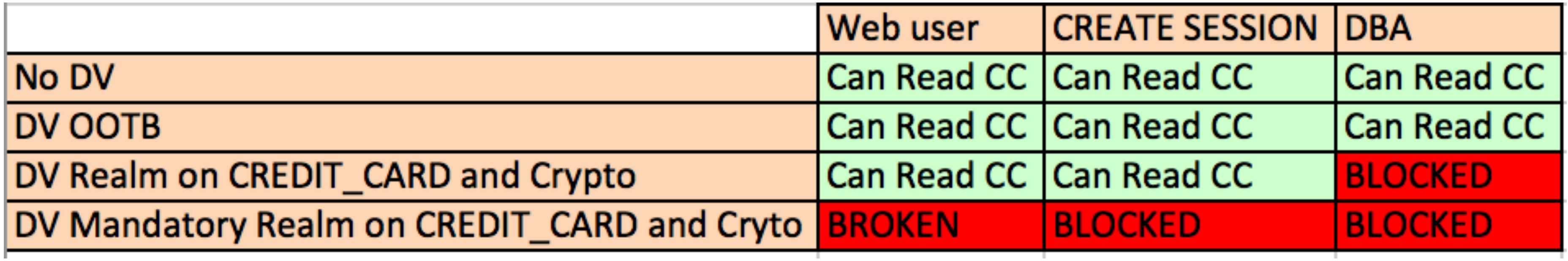
Warning: ociexecute() [function.ociexecute]: ORA-01031: insufficient privileges in **/usr/local/apache2/htdocs/bof_address.php** on line **78**

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

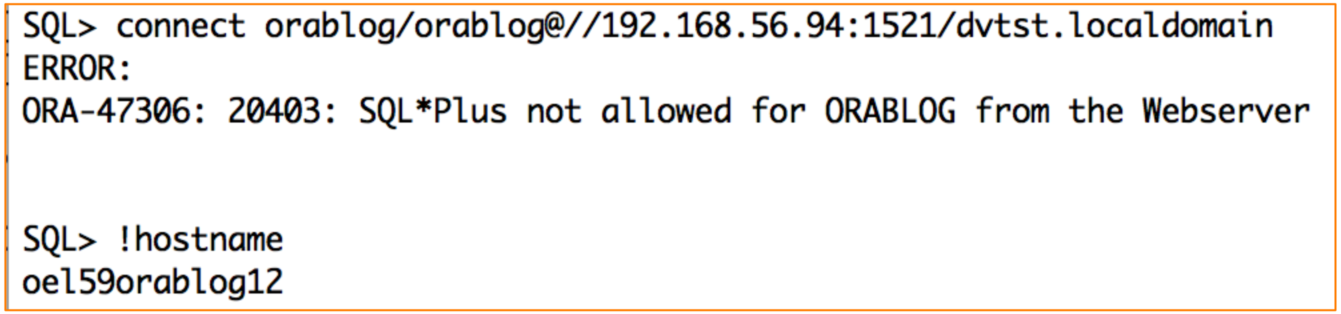
Hacking My Sample Database / Applications

- Three levels of Hacking
  - As a website un-authenticated user
  - As a database user with just CREATE SESSION
  - As a DBA
- Different Attack Types:

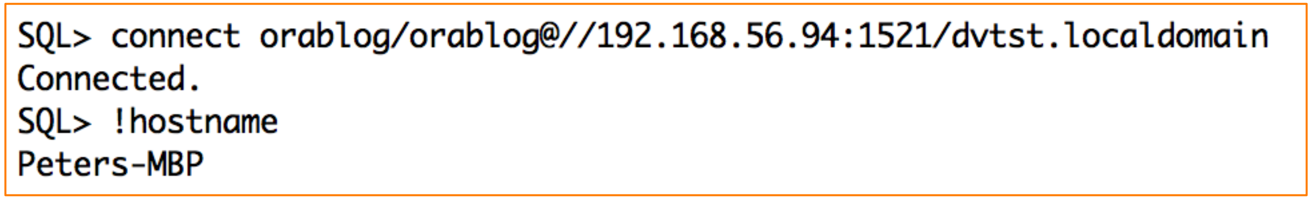


	Web user	CREATE SESSION	DBA
No DV	Can Read CC	Can Read CC	Can Read CC
DV OOTB	Can Read CC	Can Read CC	Can Read CC
DV Realm on CREDIT_CARD and Crypto	Can Read CC	Can Read CC	BLOCKED
DV Mandatory Realm on CREDIT_CARD and Cryto	BROKEN	BLOCKED	BLOCKED

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
oel59orablog12
```



```
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

Duct Tape?

- Is Database Vault really duct tape?
  - Most sites **have/use** bad data security designs; excessive rights, lack of data access controls
  - DV could be seen as duct tape to prevent these bad designs (threats) becoming risks
- At its core, DV is solving issues that could be solved differently
  - Design least rights – revoke privileges – do not use System ANY
  - SoD can be done with careful design of users and other simple protections
  - Partly issues are caused also by process; “way of working”

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?

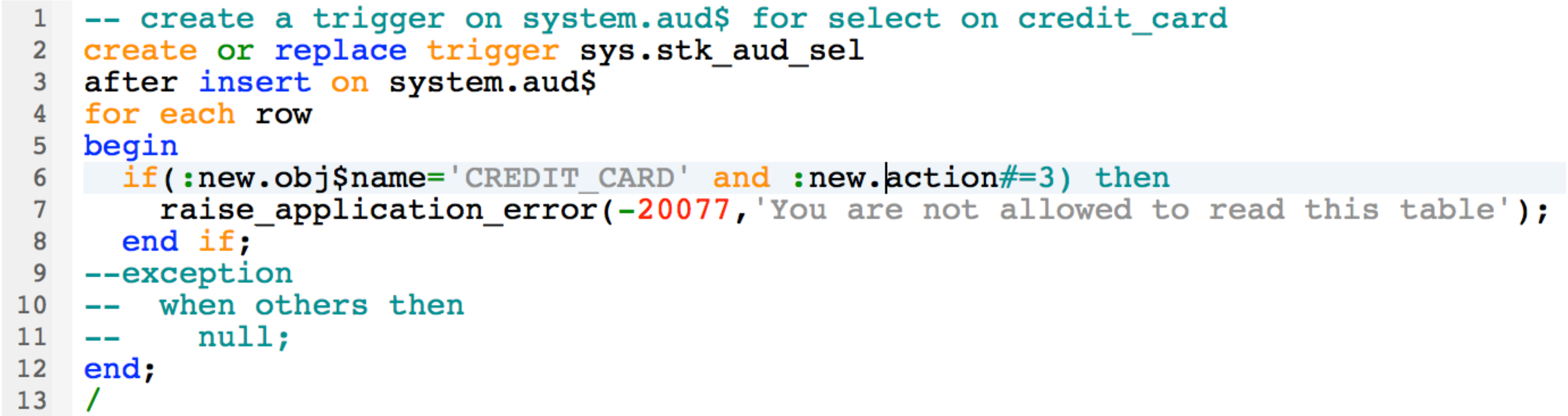
- There are a lot of features in DV that we could use: Declarative API's, factors, realms, rules, SARs, Command rules and within these protect objects, commands, SoD, parameters and much much more…
- If we focus on three simple tasks to consider for replication:
  - SET ROLE, DBMS_SESSION.SET_ROLE to 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

We Need a Select Trigger

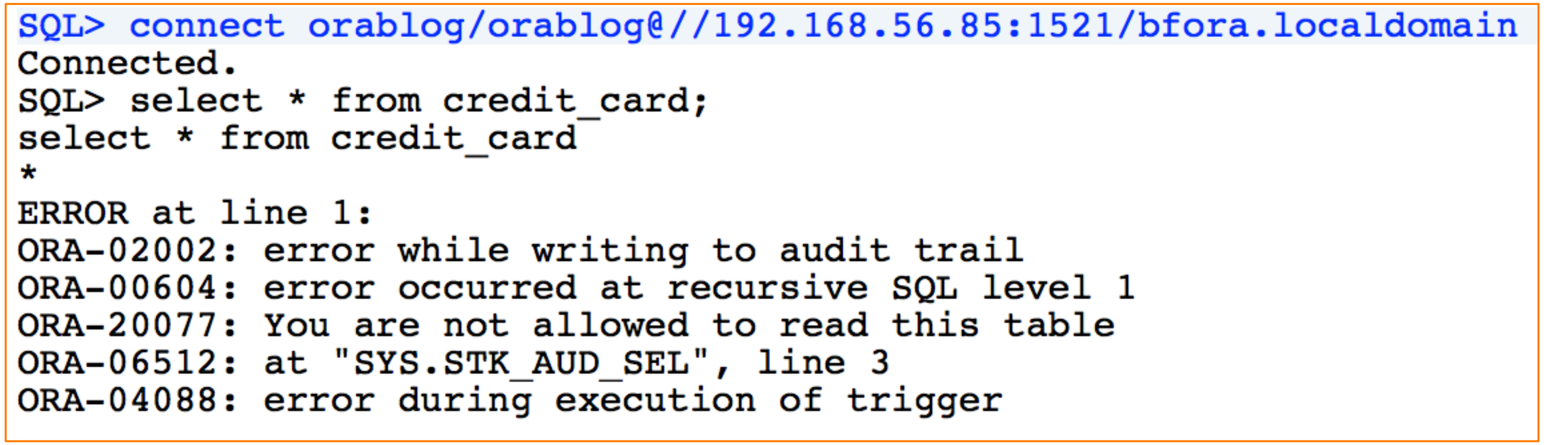
- There are limited options to capture a SELECT or SELECT ANY
  - FGA handler (needs EE so not for SE/SE1/SE2/XE)
  - Materialised View (needs views on everything)
  - VPD policy function (Again EE)
  - Trigger on AUD$
- Even more limited options for some actions such as SET ROLE or ALTER SYSTEM
  - **So we could use a trigger on AUD$**
- Note 72460.1 – This note is no longer available but talked about moving AUD$ tablespace and user and adding triggers BUT
  - This note states it is not supported to do this
  - BUT, DV install moves AUD$ to SYSTEM up to 11.2 but not 12c

Looks like a Realm!

Blocking A Select Statement

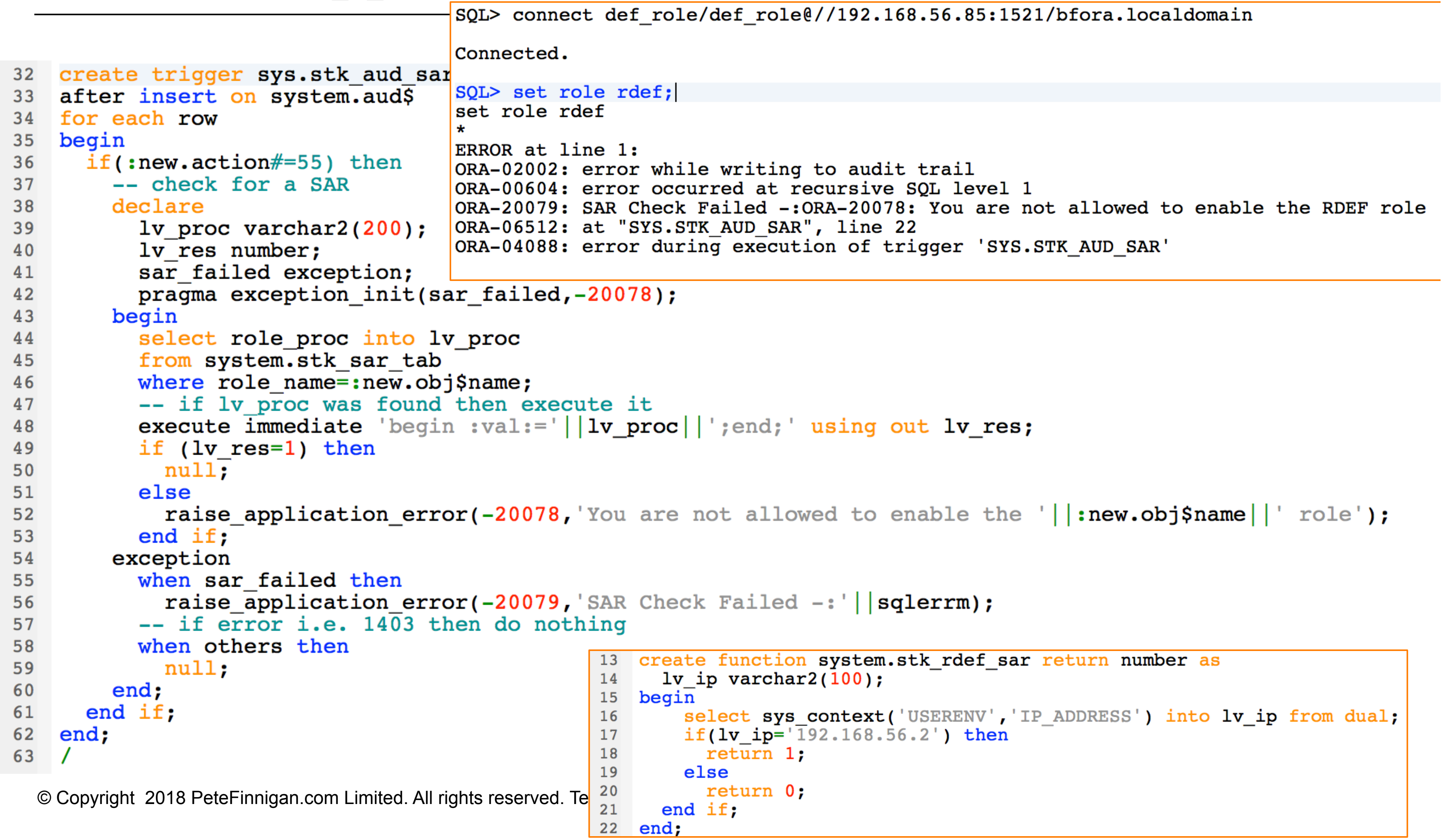


```
-- create a trigger on system.aud$ for select on credit_card
create or replace trigger sys.stk_aud_sel
after insert on system.aud$
for each row
begin
  if(:new.obj$name='CREDIT_CARD' and :new.action#=3) then
    raise_application_error(-20077,'You are not allowed to read this table');
  end if;
--exception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



```
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'
```

```
create trigger sys.stk_aud_sar
after insert on system.aud$
for each row
begin
  if(:new.action#=55) then
    -- check for a SAR
    declare
      lv_proc varchar2(200);
      lv_res number;
      sar_failed exception;
      pragma exception_init(sar_failed,-20078);
    begin
      select role_proc into lv_proc
      from system.stk_sar_tab
      where role_name=:new.obj$name;
      -- if lv_proc was found then execute it
      execute immediate 'begin :val:='||lv_proc||';end;' using out lv_res;
      if (lv_res=1) then
        null;
      else
        raise_application_error(-20078,'You are not allowed to enable the '||:new.obj$name||' role');
      end if;
    exception
      when sar_failed then
        raise_application_error(-20079,'SAR Check Failed -:'||sqlerrm);
      -- if error i.e. 1403 then do nothing
      when others then
        null;
    end;
  end if;
end;
/
```

```
create function system.stk_rdef_sar return number as
  lv_ip varchar2(100);
begin
    select sys_context('USERENV','IP_ADDRESS') into lv_ip from dual;
    if(lv_ip='192.168.56.2') then
      return 1;
    else
      return 0;
  end if;
end;
```

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

Base: Good Security Design

- DV needs a good security base to start with
  - So does non DV, whether DV is eventually used or not
- This should include:
  - Data domains
  - Separation of function from data
  - Separate critical data from non
  - Separate critical function from non
- Least User rights
- Data access controls
- Hardening and patching

Demo!

- The web and normal user fail
- The DBA still works
- Fix? Revoke ANY from the Orablog DBA role

Hack The Locked Down System

- This is the same database and applications setup as was used in the DV examples
- Except:
    - The database, OS and Network are locked down
    - The data design has changed to secure the data from the connected user
    - The application code is still vulnerable
- Lets try the same hacks as before

As an example of one DV feature

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**

Cont’d

- Two types of rights via SYSTEM ANY
  - Object change/create (CREATE ANY PROCEDURE)
  - Data access and data change (SELECT ANY TABLE)
- Should the release person be able to change data?
  - No, BUT maybe release require data changes
  - Reading data – probably not
- In general
  - Interactive users should not have SYSTEM ANY
  - Schemas should not have SYSTEM ANY
  - A DBA can work around not having SYSTEM ANY
  - Core accounts such as SYS, SYSTEM, DBA – **don’t use**

Demo – dacc.sql

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

Demo – context_dml.sql

Context: DML Based Security

- This is a simple demo to show that 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

Demo – context_code.sql

Context: Code Based Security

- We also do not need DV to add context based security to PL/SQL code
- DV has the advantage that it is declarative and does not need code to be hand written
- BUT we can still add context based checks to our code where needed
- This example shows that we can limit a function that gets an encryption key from storage to only be called from its protective API
- In a real system we would also obfuscate and protect the PL/SQL
- When you implement DV a lot of work is still needed anyway

Separation of Duties (SoD)

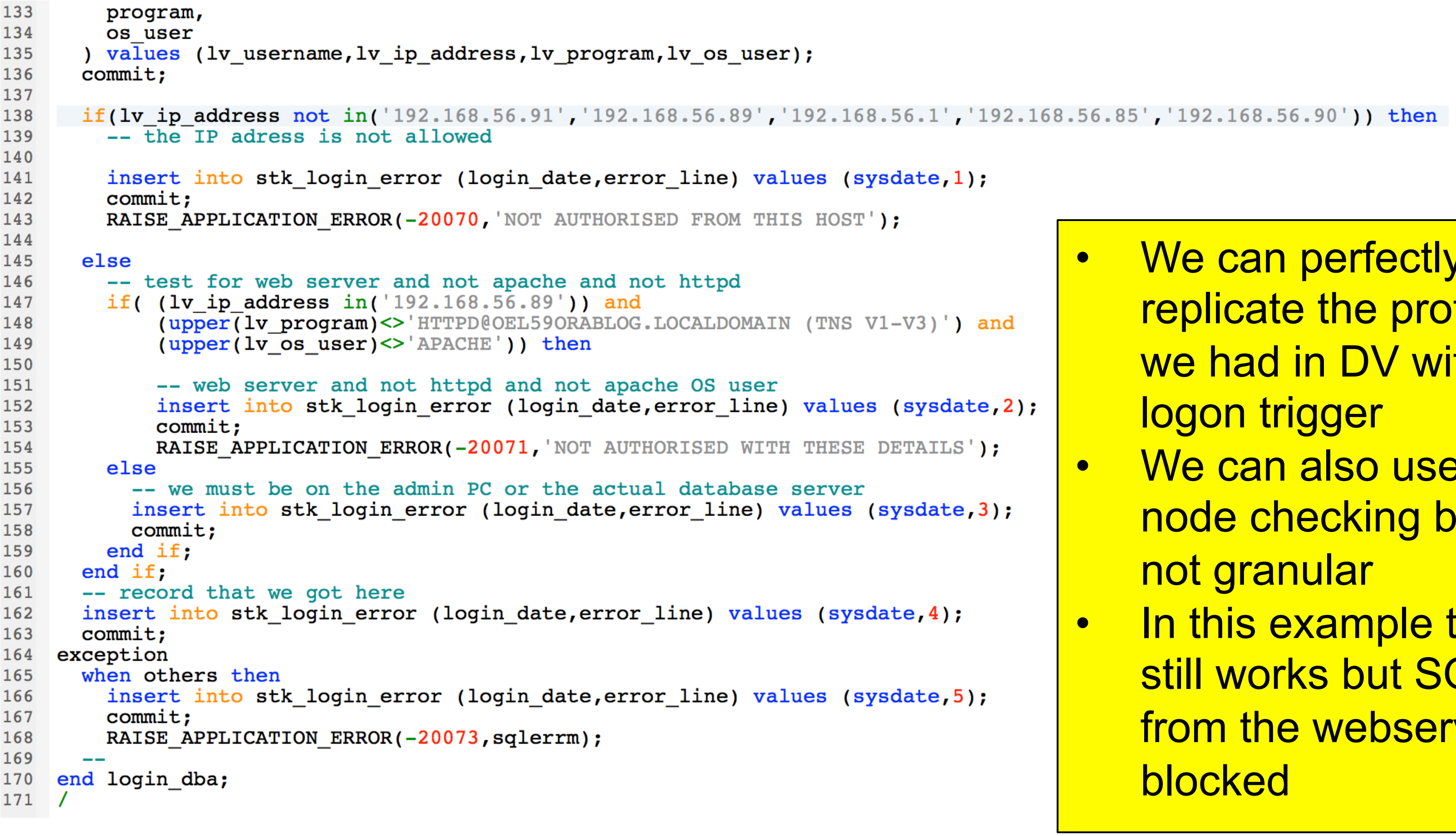
- Separation of Duties does not need DV to enforce it
- Even with DV real people and database accounts need to be designed and a SoD matrix created to ensure separation exists for all interactive users
- Identify and make decisions on separation
  - Account Manager, Audit Trail Admin, Security Admin, 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

Demo – connect to the database as ORABLOG from the web server

Command Rule: Block SQL*Plus - Webserver



```
    program,
    os_user
  ) values (lv_username,lv_ip_address,lv_program,lv_os_user);
  commit;

  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
    -- the IP adress is not allowed

    insert into stk_login_error (login_date,error_line) values (sysdate,1);
    commit;
    RAISE_APPLICATION_ERROR(-20070,'NOT AUTHORISED FROM THIS HOST');

  else
    -- test for web server and not apache and not httpd
    if( (lv_ip_address in('192.168.56.89')) and
        (upper(lv_program)<>'HTTPD@OEL59ORABLOG.LOCALDOMAIN (TNS V1-V3)') and
        (upper(lv_os_user)<>'APACHE')) then

        -- web server and not httpd and not apache OS user
        insert into stk_login_error (login_date,error_line) values (sysdate,2);
        commit;
        RAISE_APPLICATION_ERROR(-20071,'NOT AUTHORISED WITH THESE DETAILS');
    else
      -- we must be on the admin PC or the actual database server
      insert into stk_login_error (login_date,error_line) values (sysdate,3);
      commit;
    end if;
  end if;
  -- record that we got here
  insert into stk_login_error (login_date,error_line) values (sysdate,4);
  commit;
exception
  when others then
    insert into stk_login_error (login_date,error_line) values (sysdate,5);
    commit;
    RAISE_APPLICATION_ERROR(-20073,sqlerrm);
  --
end login_dba;
/
```

- 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

Privilege Analysis

- This is the simplest to replicate outside of DV
- This is because DV really uses audit or an internal version of it for Privilege Analysis
- We can use audit to establish what privileges are used
- We need to analyse the context first
  - If Roles – list all rights per role
  - If context – list all rights for the context
  - Use a version of find_all_privs that creates a row of data for each right
- Enable audit for all rights relative to the context
  - Generate audit commands from the table or policy for PFCLATK
- Create two views (used,unused) based on the audit trail and also the privileges stored and also the context
- Or do a paper based review of audit vs find_all_privs.sql

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?

You Don’t Have Database Vault

So, What Can You Do Instead?