Protect your Database with SQL Firewall in 23c

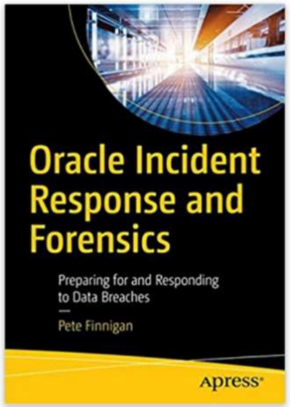
Legal Notice

Protect your Database with SQL Firewall in 23c

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

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.



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

- What is the SQL Firewall
- Why use the SQL Firewall
- Set up the data
- Set up SQL Firewall and Training
- Testing
- Hacking
- SQL Firewall Management

Section

What is the SQL Firewall?

What is the SQL Firewall

- **"The SQL Firewall blocks non-authorized SQL or PL/SQL"**
- We can expand that to **"The SQL Firewall monitors and / or blocks non-authorized SQL or PL/SQL"**
- This started as the Secerno product and became Oracles database firewall
- Now embedded in the database SQL engine in 23c

Section

Why Use the SQL Firewall?

Database Security

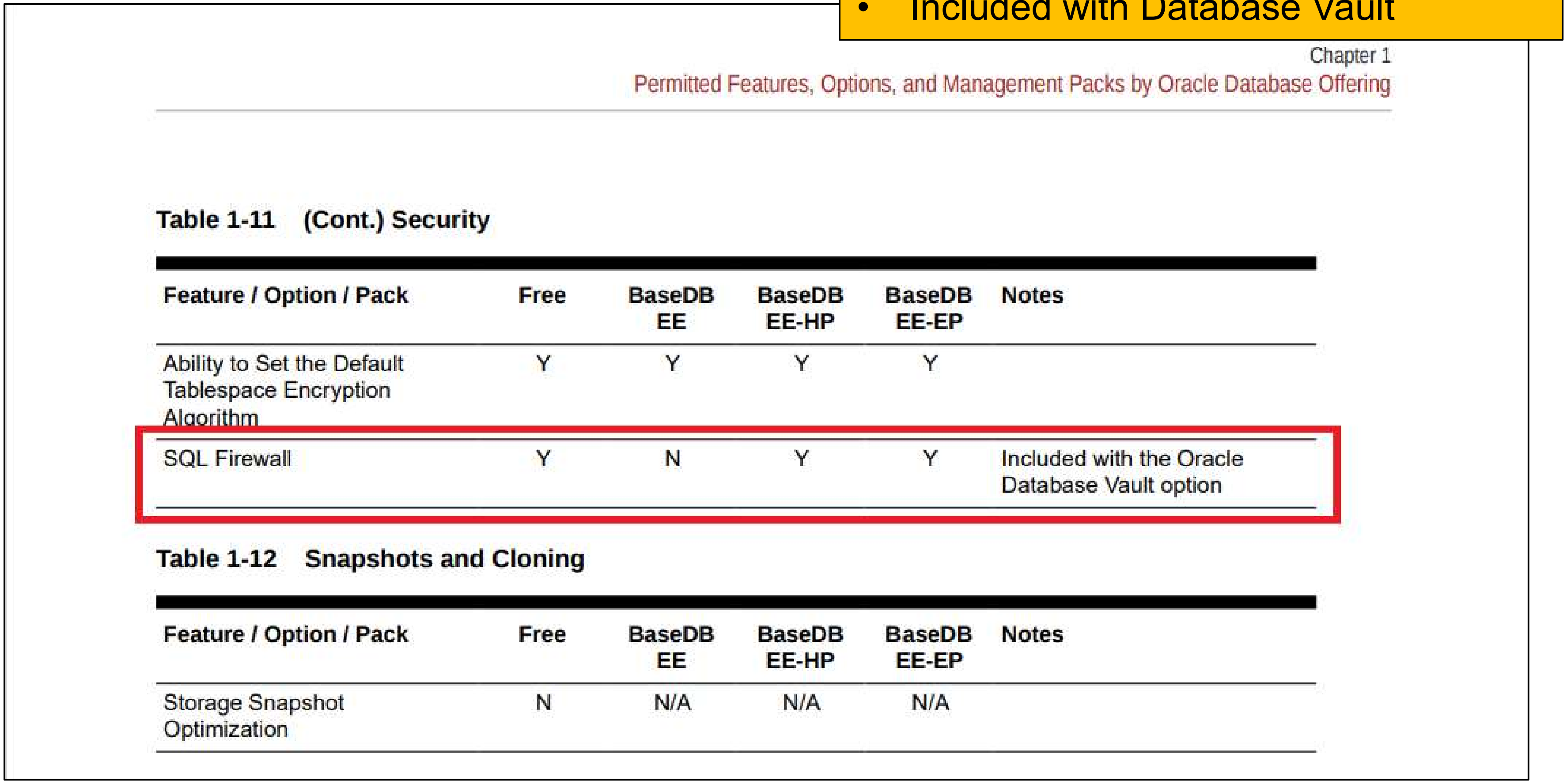
- Security patches and database Hardening
- Data security
  - Access controls
  - User controls – least rights
  - Data access controls
- Audit trails
- Secure coding
- Context based security (DV, VPD, TSDP,…)
- **Firewalls, DAM, IDS, IPS, …**

SQL Firewall is the Last Step

- We must implement all of the other layers of data security to protect data first
- SQL Firewall is the final layer on top of other data security and auditing
- We should not rely just on the SQL Firewall
- It is based on “good/bad” SQL
  - We must tell it what is good

License?

- https://docs.oracle.com/en/database/oracle/oracle-database/23/dblic/database-licensing-information-user-manual.pdf
- Not in base cloud EE
- Included with Database Vault



Chapter 1
Permitted Features, Options, and Management Packs by Oracle Database Offering

Table 1-11 (Cont.) Security

Feature / Option / Pack	Free	BaseDB EE	BaseDB EE-HP	BaseDB EE-EP	Notes
Ability to Set the Default Tablespace Encryption Algorithm	Y	Y	Y	Y	
SQL Firewall	Y	N	Y	Y	Included with the Oracle Database Vault option

Table 1-12 Snapshots and Cloning

Feature / Option / Pack	Free	BaseDB EE	BaseDB EE-HP	BaseDB EE-EP	Notes
Storage Snapshot Optimization	N	N/A	N/A	N/A	

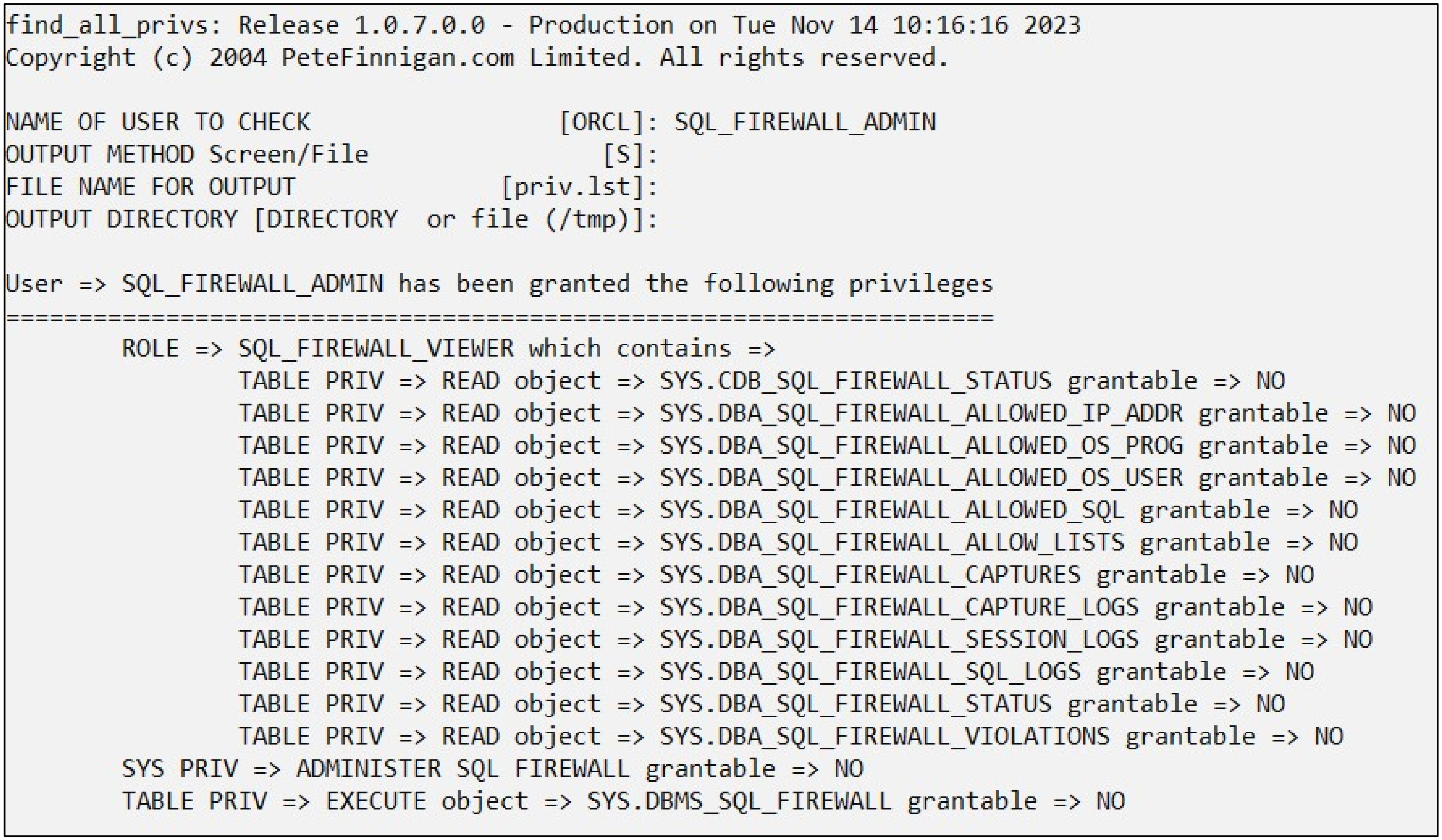
Section

Set up the Data

SQL Firewall Permissions - 1

- System Privilege
  - ADMINISTER SQL FIREWALL
- PL/SQL Package
  - DBMS_SQL_FIREWALL
- Views
  - dba_sql_firewall_violations, dba_sql_firewall_allowed_sql, ….
- Roles
  - SQL_FIREWALL_ADMIN
  - SQL_FIREWALL_VIEWER

SQL Firewall Permissions - 2



```
find_all_privs: Release 1.0.7.0.0 - Production on Tue Nov 14 10:16:16 2023
Copyright (c) 2004 PeteFinnigan.com Limited. All rights reserved.

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

User => SQL_FIREWALL_ADMIN has been granted the following privileges
====================================================================
        ROLE => SQL_FIREWALL_VIEWER which contains =>
                TABLE PRIV => READ object => SYS.CDB_SQL_FIREWALL_STATUS grantable => NO
                TABLE PRIV => READ object => SYS.DBA_SQL_FIREWALL_ALLOWED_IP_ADDR grantable => NO
                TABLE PRIV => READ object => SYS.DBA_SQL_FIREWALL_ALLOWED_OS_PROG grantable => NO
                TABLE PRIV => READ object => SYS.DBA_SQL_FIREWALL_ALLOWED_OS_USER grantable => NO
                TABLE PRIV => READ object => SYS.DBA_SQL_FIREWALL_ALLOWED_SQL grantable => NO
                TABLE PRIV => READ object => SYS.DBA_SQL_FIREWALL_ALLOW_LISTS grantable => NO
                TABLE PRIV => READ object => SYS.DBA_SQL_FIREWALL_CAPTURES grantable => NO
                TABLE PRIV => READ object => SYS.DBA_SQL_FIREWALL_CAPTURE_LOGS grantable => NO
                TABLE PRIV => READ object => SYS.DBA_SQL_FIREWALL_SESSION_LOGS grantable => NO
                TABLE PRIV => READ object => SYS.DBA_SQL_FIREWALL_SQL_LOGS grantable => NO
                TABLE PRIV => READ object => SYS.DBA_SQL_FIREWALL_STATUS grantable => NO
                TABLE PRIV => READ object => SYS.DBA_SQL_FIREWALL_VIOLATIONS grantable => NO
        SYS PRIV => ADMINISTER SQL FIREWALL grantable => NO
        TABLE PRIV => EXECUTE object => SYS.DBMS_SQL_FIREWALL grantable => NO
```

- sf_create_users.sql
- sf_create_sf.sql
- sf_run_vm.sql

Set up Test Data

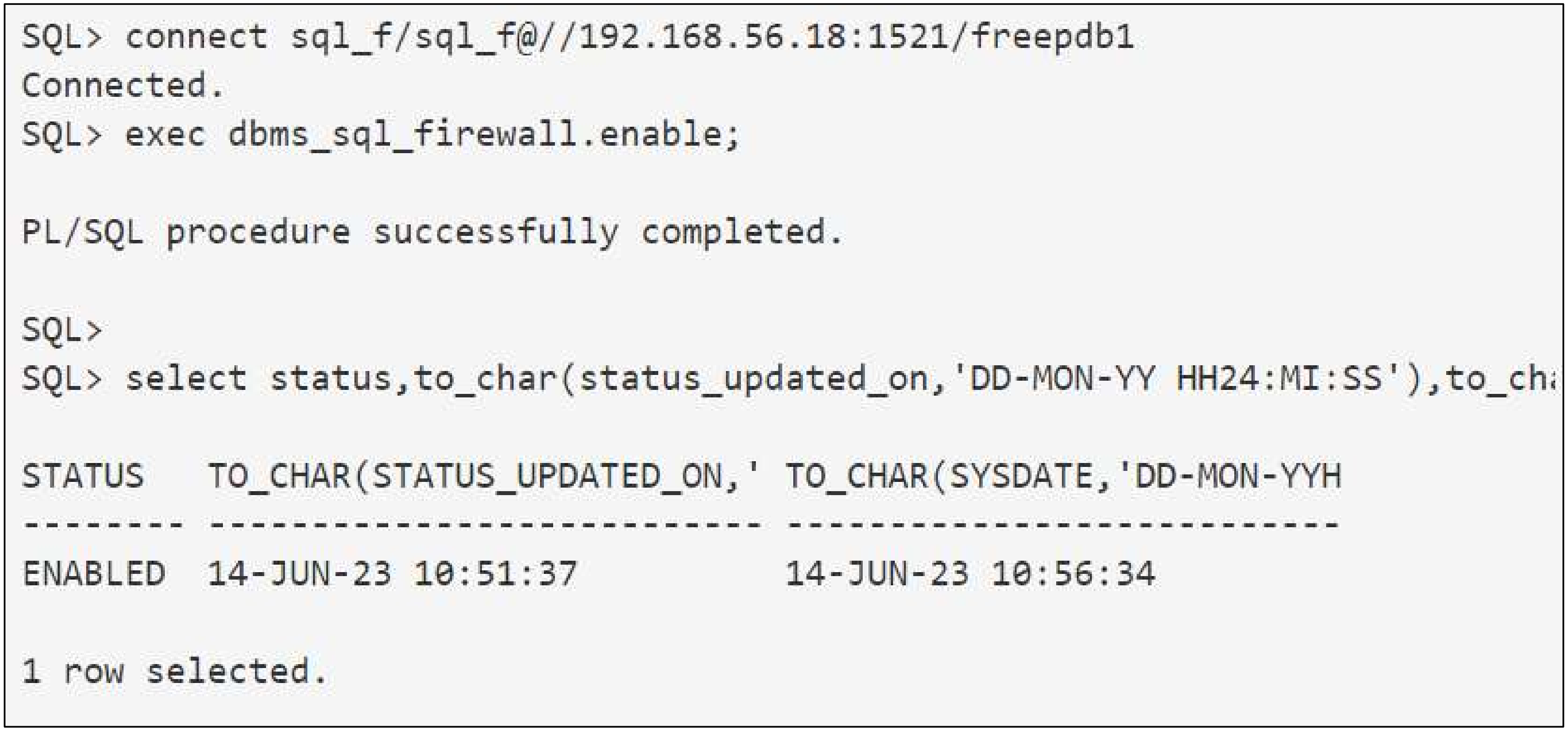
- Create a schema ORABLOG to own some tables, data and PL/SQL Code
- Create a connection user VM to access data
- Make the grants
- Create a SQL Firewall Admin user
- Run some sample SQL and PL/SQL

Section

Set up SQL Firewall and Training

- sf_enable.sql

Enable the SQL Firewall



```
SQL> connect sql_f/sql_f@//192.168.56.18:1521/freepdb1
Connected.
SQL> exec dbms_sql_firewall.enable;

PL/SQL procedure successfully completed.

SQL>
SQL> select status,to_char(status_updated_on,'DD-MON-YY HH24:MI:SS'),to_ch

STATUS   TO_CHAR(STATUS_UPDATED_ON,' TO_CHAR(SYSDATE,'DD-MON-YYH
-------- --------------------------- ---------------------------
ENABLED  14-JUN-23 10:51:37          14-JUN-23 10:56:34

1 row selected.
```

- sf_capture.sql
- sf_run_vm.sql
- sf_stop.sql
- sf_log.sql

Set up the Capture

- We need to teach the SQL Firewall what good SQL and PL/SQL looks like
- Create a capture for the user VM
- Run sample (ALL) business logic
- Turn off the capture
- Review the capture logs
- NOTE: Some items we did not do directly
- **Do not "teach" the SQL Firewall BAD SQL**

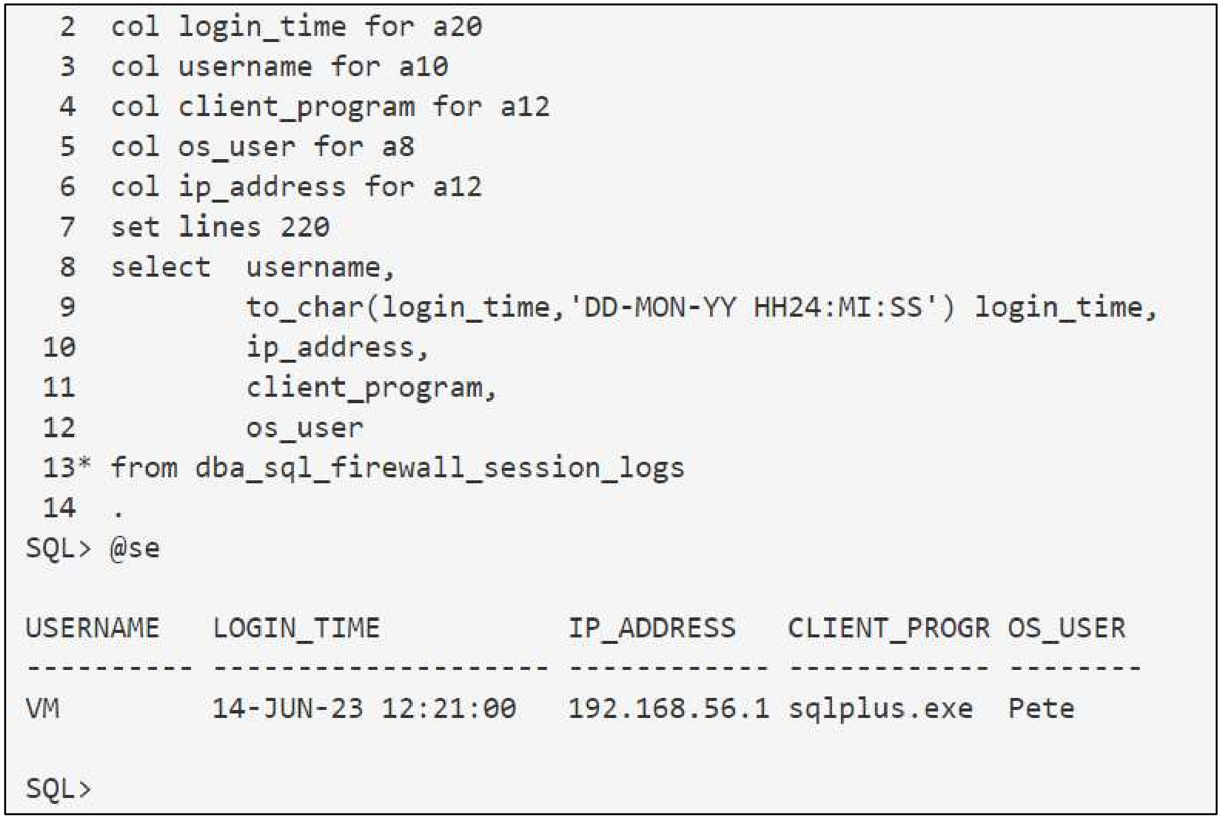
PFCL

PETEFINNIGAN.COM LIMITED

- sf_se_log.sql

Session Logs

- This shows the SQL sessions



```
  2  col login_time for a20
  3  col username for a10
  4  col client_program for a12
  5  col os_user for a8
  6  col ip_address for a12
  7  set lines 220
  8  select  username,
  9          to_char(login_time,'DD-MON-YY HH24:MI:SS') login_time,
 10          ip_address,
 11          client_program,
 12          os_user
 13* from dba_sql_firewall_session_logs
 14  .
SQL> @se

USERNAME   LOGIN_TIME           IP_ADDRESS   CLIENT_PROGR OS_USER
---------- -------------------- ------------ ------------ --------
VM         14-JUN-23 12:21:00   192.168.56.1 sqlplus.exe  Pete

SQL>
```

- sf_allow.sql

Create the Allow List

- Generate the allow list from the capture list
- Review the SQL and PL/SQL
- We can adjust the list now or in the future
  - I will not make changes for expediency
- Enable the allow list for VM
- **The SQL Firewall works on "good" SQL but we cannot operate from the reverse stand point**

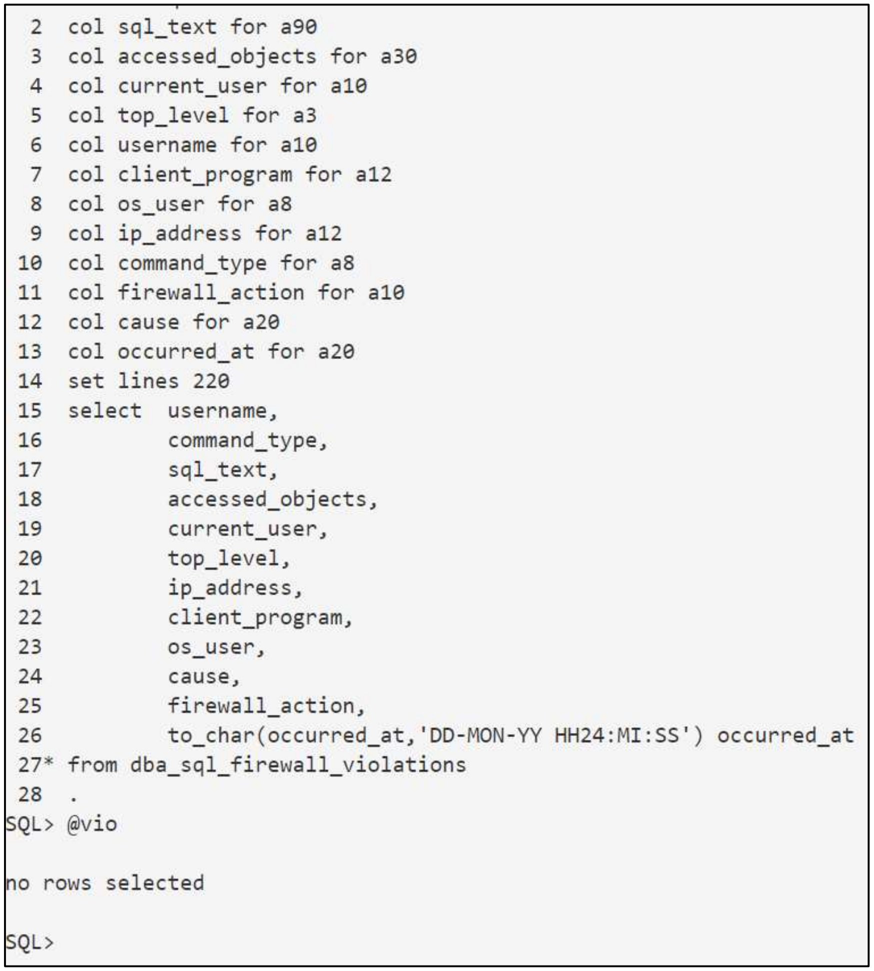
Section

Testing

- sf_vio.sql

Check for Violations

- Check for none



```
col sql_text for a90
col accessed_objects for a30
col current_user for a10
col top_level for a3
col username for a10
col client_program for a12
col os_user for a8
col ip_address for a12
col command_type for a8
col firewall_action for a10
col cause for a20
col occurred_at for a20
set lines 220
select  username,
        command_type,
        sql_text,
        accessed_objects,
        current_user,
        top_level,
        ip_address,
        client_program,
        os_user,
        cause,
        firewall_action,
        to_char(occurred_at,'DD-MON-YY HH24:MI:SS') occurred_at
from dba_sql_firewall_violations
.
SQL> @vio

no rows selected

SQL>
```

- sf_run_vm.sql

Testing

- Run normal business actions
- Test the application works
- Test that no SQL Firewall violations are found
- Adjust the rules, contexts if necessary

Section

Hacking

- sf_run.sql
- sf_vio.sql

Try and Abuse the Database

- Try an INSERT statement that is not allowed by the SQL Firewall
- Try an UPDATE not allowed by database permissions
- Try a SELECT not allowed by permissions
- The INSERT is blocked by the firewall but the other two return database errors as normal

- sf_hack.sql
- sf_hack1.sql
- sf_vio.sql

Hack The Database

- Test some SQL injection to access tables and views not allowed by the firewall but allowed for ORABLOG and not VM
- Show direct access to the same tables / views as VM directly
- The SQL Injection is not blocked
- The direct view access is
- To block SQL Injection we need to relearn with not TOP LEVEL ONLY

- sf_syn.sql
- sf_view.sql
- sf_desc.sql
- sf_vio.sql

More Testing

- Test access to the same data as VM via a synonym
- Test access to the same data via a view
- Test creation of a view
- Test describe of a table allowed by the firewall

Section

Additions

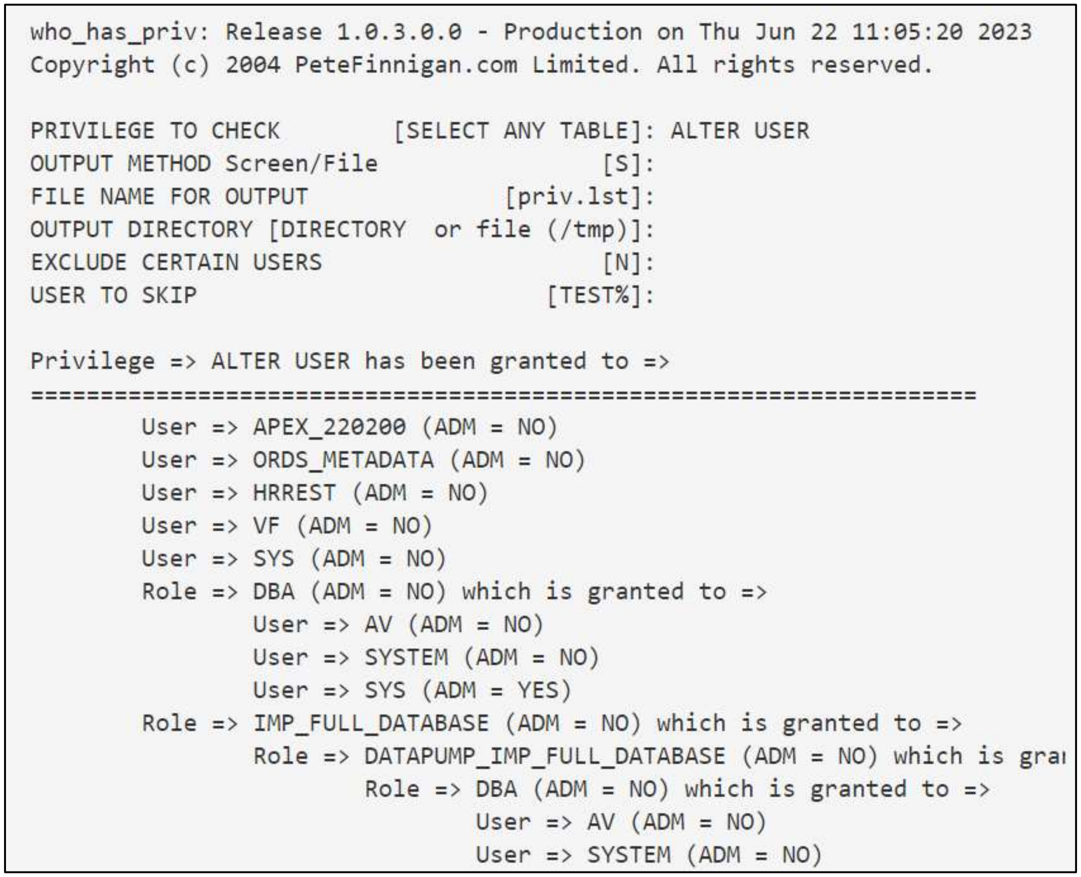
- sf_proxy.sql

Proxy

- I have long advocated the use of proxy to access a schema for maintenance
- The database knows who you are BUT you can be the schema/user in all other respects
- Proxy works with the SQL Firewall
- We create a connect user and grant access through VM

Proxy Issue

- If we have access to ALTER USER we can bypass the SQL Firewall
- So any users with IMP_FULL_DATABASE or APEX_220200 can access data or function protected by SQL Firewall by allowing …GRANT CONNECT THROUGH…



```
who_has_priv: Release 1.0.3.0.0 - Production on Thu Jun 22 11:05:20 2023
Copyright (c) 2004 PeteFinnigan.com Limited. All rights reserved.

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

Privilege => ALTER USER has been granted to =>
====================================================================
        User => APEX_220200 (ADM = NO)
        User => ORDS_METADATA (ADM = NO)
        User => HRREST (ADM = NO)
        User => VF (ADM = NO)
        User => SYS (ADM = NO)
        Role => DBA (ADM = NO) which is granted to =>
                User => AV (ADM = NO)
                User => SYSTEM (ADM = NO)
                User => SYS (ADM = YES)
        Role => IMP_FULL_DATABASE (ADM = NO) which is granted to =>
                Role => DATAPUMP_IMP_FULL_DATABASE (ADM = NO) which is gra
                        Role => DBA (ADM = NO) which is granted to =>
                                User => AV (ADM = NO)
                                User => SYSTEM (ADM = NO)
```

Section

SQL Firewall Management

Manage the SQL Firewall

- Rules / settings can be changed after learning / creation
  - Rules removed, New rules, Add more context – or remove
  - Add more users
- Clear the logs
- Connect to unified audit
  - two new columns FW_ACTION_NAME and FW_RETURN_CODE
  - New COMPONENT clause “SQL Firewall”
  - No direct link between UNIFIED_AUDIT_TRAIL and SQL Firewall views
- Deep level needed to catch SQL injection

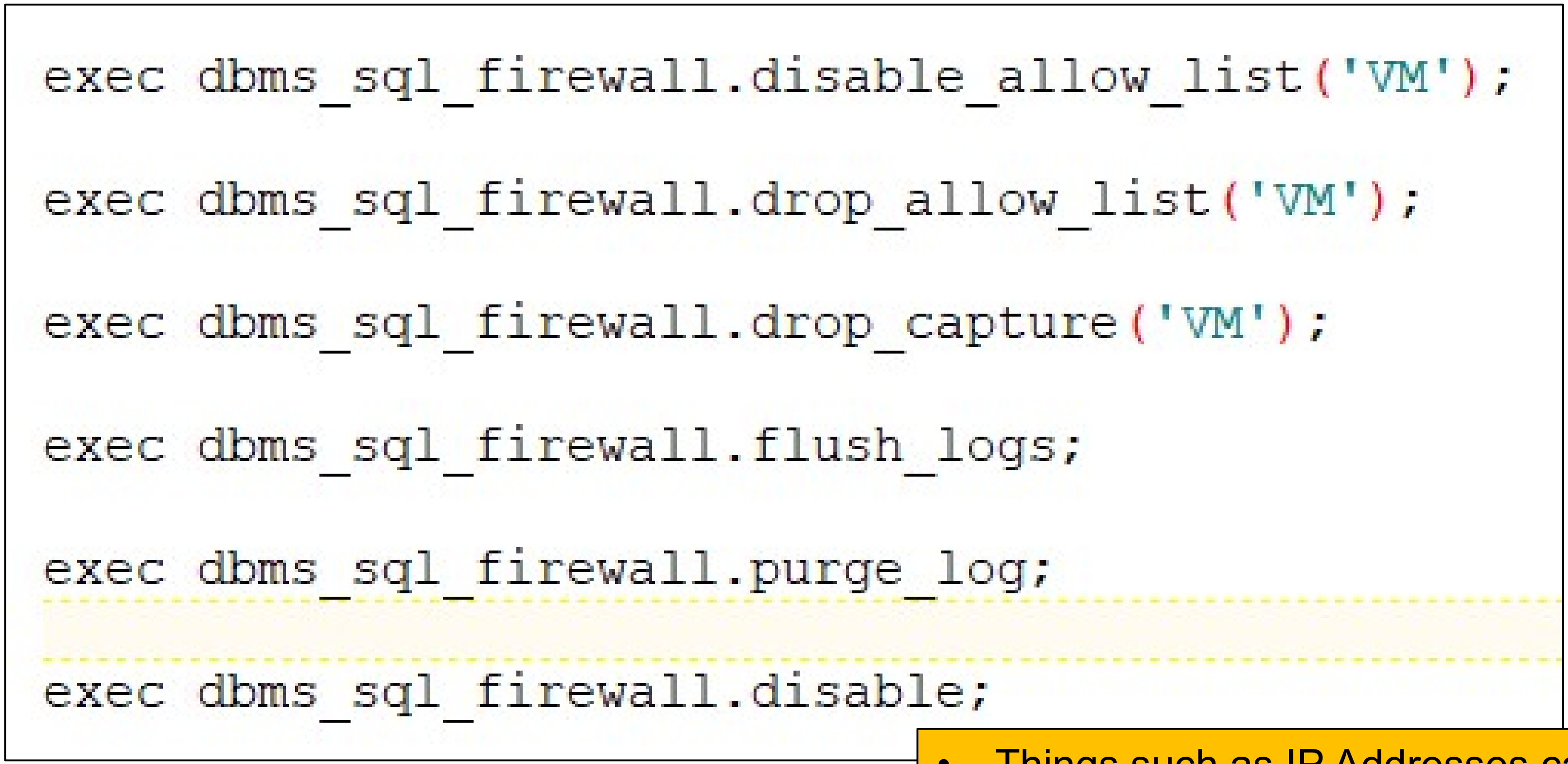
- sf_status.sql

Checking the SQL Firewall Status

- We can query all of the SQL Firewall views to check the status of the firewall, captures, allows and logs

SQL Firewall Management

- sf_dis.sql
- sf_drop_users.sql



```
exec dbms_sql_firewall.disable_allow_list('VM');

exec dbms_sql_firewall.drop_allow_list('VM');

exec dbms_sql_firewall.drop_capture('VM');

exec dbms_sql_firewall.flush_logs;

exec dbms_sql_firewall.purge_log;

exec dbms_sql_firewall.disable;
```

- Things such as IP Addresses cannot be removed if the firewall is disabled
- Disabling the firewall doesn’t remove anything

Conclusions

- Complex
- Doesn’t look like its free for lower versions
- Its very specific to users, context and SQL
- Do not train hacking
- Do not use instead of data security
- All actions must be learned – i.e. known in advance
- Will need a lot of maintenance
- Allow / **disallow**

Questions

?

If Anyone has questions, please ask now or catch me during the event!!

Protect your Database with SQL Firewall in 23c