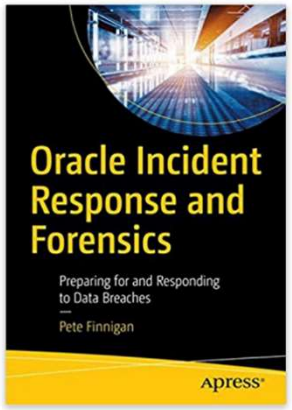
Designing a Good Database Audit Trail

Legal Notice

Designing a Good Database Audit Trail

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

PFCL
PETEFINNIGAN.COM LIMITED



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

- Introduction
- Designing an audit trail
- The basics to include
- A default audit trail and hacking
- Implement some rules
- Hack again

Section

Introduction

History of the Audit Events, Toolkit and Talks

- In 2009 piece of work to help design audit trails
  - Site had limited staff, little time to design, deploy, maintain any audit trails
  - I came up with some simple ideas, proof of concepts – to package up audit trails for them; inc policy based audit, IPS and simple firewall
  - They spent limited time to deploy a useful audit trail
- Similar piece of work in 2011 where limited team needed to deploy audit
- 2012 to 2015 extended the toolkit
- I wrote a presentation back in 2012 and presented it just once at a SIG on practical audit trails where I mentioned this toolkit for the first time
- This then became the basis of a one day class on the same subject
- Reworked that presentation in UKOUG 2015 conference
- Customer in 2016 needed an audit trail to deploy quickly
- Deployed now to customers in UK, Ireland and Germany
- **The ideas of audit design came from these pieces of work and talks**

PFCL
PETEFINNIGAN.COM LIMITED

Some People Think They Have Designed Audit

- I see sites with some audit settings
- This is not a WELL DESIGNED audit trail
- Usually these random set of parameters in the Oracle database will not catch a good range of events that could be an attack
- Some sites have application audit and OS audit
- BUT worse; lots of sites have no audit at the database engine level
- If there is a breach a lack of audit makes forensic response very difficult

Design

- Before we get started implementing
- **Design** must be the first step
- The final chosen solution implements the design
- Therefore until you know the design you cannot specify the right solution – **right?**
- The solution could be “free” or “commercial” solution or even a combination of both
- Often people buy third party products and implement out of the box with no internal requirements!

PFCL
PETEFINNIGAN.COM LIMITED

The Question

- What should I include in a basic audit trail?
  - The answer must be useful information; **to who?**
  - Should be well designed
  - Should be structured
  - Actually capture something?
    - What? Attacks of course
    - Well maybe abuse as well by workers
      - Actions outside of authority
      - Changes with no change
      - Changes to security and audit

PFCL
PETEFINNIGAN.COM LIMITED

Section

Designing an Audit Trail

PFCL
PETEFINNIGAN.COM LIMITED

What Settings Should I Include In An Audit Trail

- I get asked this question regularly sometimes multiple times per week
- **This is the cart before the horse**
- Don’t just get peoples tips and tricks and use them
- Design the audit for yourself
- Don’t just turn on “some settings” from one document or another such as CIS
- These are not designed by you or for you or for your application or deployment

PFCL
PETEFINNIGAN.COM LIMITED

The Purpose of the (your) Audit Trail

- Detect wrong doing
- Detect activity at the databases engine level
- Audit can be at multi layers
  - The applications
    - Code writing audit
    - Before and after
    - Who/when, last/when
    - Database level audit used for applications
  - The operating system
  - The database engine
- We want to focus on the database engine as this is the part that is usually missing

PFCL
PETEFINNIGAN.COM LIMITED

Who Are We Satisfying?

- Do we satisfy auditors or compromise?
  - Risk vs cost (implement and TCO/ROI)
- Keep the raw trail or the reports?
- Trail to be kept off the server or local?
- Size of the storage required?
- Performance (depends on actions captured and design decisions)
- Re-Active “vs” Pro-Active audit

PFCL
PETEFINNIGAN.COM LIMITED

What Do I Want to Know?

- Design must be based on "**what do I want to know?**"
- Maybe also "**what do I predict I want to know?**"
- What background data must I collect
- All with the remit that collecting and saving costs money
- You must know what you want otherwise
  - The collected data will never be used
  - The data collected will not be what's needed when a breach occurs; i.e. will not show the breach
- This implies we must know what all breach types look like or we collect most things for a specific type of audit
- Or; we must focus on data first and foremost

PFCL
PETEFINNIGAN.COM LIMITED

What Do I Want To Know (2)

- The audit design must encompass all areas required to audit the action (i.e. there are multiple ways to do things in Oracle), be layered but also be simple to deploy and use, it must include
  - Management
    - Manage storage
    - Purge
    - Archive
    - Adding new users or features to the database - extendable
  - Technical solutions to capture raw data
  - Reporting to decide on issues located
  - Escalation
  - Alerts for high risk items
- **i.e. - The design process is much bigger than simply turning on settings or buying a product such as "Audit Vault and DB Firewall"**

Re-Active Audit

- **What do we mean?**
- We use the audit trail as an historic representation of what happened in the database or with the data
- We keep this audit trail "**just in-case**" an attack has occurred so that we can investigate retrospectively
- The problem is that we do not know the type of attack in advance, often the audit storage is driven by regulatory requirements or legal requirements.
- Often this type of audit will not be useful for database inc incident and forensic investigation but is required for specific log data storage and use only

Pro-Active Audit

- **What do we mean?**
- We use the audit trail to let us know in real time or semi-real time if an attack against our database or data is in progress
- We use that audit data to "**take action**" by raising alerts, escalation of the issue detected, regular reporting
- This type of audit if often based on actual attack scenarios
- We know what we want to detect before it happens
- If something different occurs then we may or may not detect it
- Both audit types (reactive/pro-active) can be combined

Create audit events based on "**I Want to know?**"

Create Audit Events

ID	Description	Category	Type	Report	Report Time
AE.1.0	Every connection to the database whether successful or not	ENGINE	COLLECT	NO	NONE
AE.1.1	Detect individuals sharing database one account	ENGINE	NORMAL	YES	SLOW
AE.1.2	Detect individuals who have access to multiple database accounts	ENGINE	NORMAL	YES	REGULAR
AE.1.3	Detect all failed logins	ENGINE	COLLECT	NO	NONE
AE.1.4	Detect a frequency of failed logins where the frequency is low (For example more than 3 per minute are detected)	ENGINE	NORMAL	YES	QUICK
AE.1.5	Detect a frequency of failed logins where the frequency is high (For example more than 50 per minute are detected). 1017, 28002 etc errors	SECURITY	ALERT	YES	IMMEDIATE
AE.1.6	Detect developer access (**note: This will be allowed in development databases**)	ENGINE	NORMAL	YES	REGULAR
AE.1.7	Capture access to dormant accounts (3 months dormant)	ENGINE	NORMAL	YES	REGULAR
AE.2.0	Capture all DDL activity in the database	ENGINE	COLLECT	NO	NONE
AE.2.1	Capture structural changes (for instance tablespaces, data files)	ENGINE	NORMAL	YES	REGULAR
AE.2.2	Detect any user changes (legitimate)	SECURITY	COLLECT	NO	NONE
AE.2.3	Detect any user changes (not legitimate)	SECURITY	ALERT	YES	IMMEDIATE
AE.2.4	Detect profile changes	SECURITY	NORMAL	YES	QUICK
AE.2.5	Detect any GRANTS for roles, system privileges or objects (not legitimate)	SECURITY	ALERT	YES	IMMEDIATE

Build On The Audit Events

- Work backwards from the events to decide what raw audit to collect
- Then how to work out if event has occurred
- Then how to report
- Then how to alert
- Then how to escalate
- When you have this “table” decide on the technical solution that can be implemented and deployed

Section

The Basics to Include

I said don't base the list on technical settings
BUT we can base the list on possible events

What To Audit (1) – Technical Level

- DBA or Power User – DBA or Developer or Power like activities
  - Use of Privilege
  - Creating Objects
  - Changing Objects
  - Changing structure
- Support like activities
  - Schema and application maintenance
  - Changes to applications
- End user activities - Use of privileges
- Connections to the database
  - Default account logon failure
  - High frequency logon failures

What To Audit (2) – Technical Level

- Audit of break glass
- Audit of context based security
- Audit of configuration
  - Database
  - Application
- Attacks
  - Genuine attacks – i.e. web based, forms based, client based SQL or PL/SQL or statement injections of SQL, PL/SQL into the application.
  - Staff access outside of their authorised realm
  - Third party access
  - CPU, 0-Day
- Escalation of rights – attack or DBA or other activities

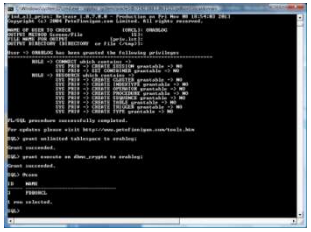
Section

Default Audit Trail and Hacking

PFCL
PETEFINNIGAN.COM LIMITED

- Oracle Linux
- Oracle SE1 Database
- Applications (Front Facing Website, back office customer processing)

My Sample Application Architecture



Server 1 – Linux 5.9

Oracle database schemas

Orablog Schema

Power Users

End Users

Server 3 – Linux 5.9

Apache / PHP / OCI8

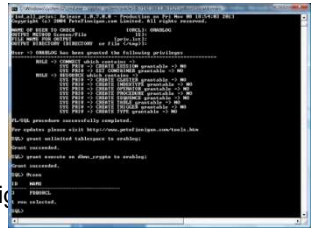
BOF: Back Office web based Application

Server 2 – Linux 5.9

Apache / PHP / OCI8

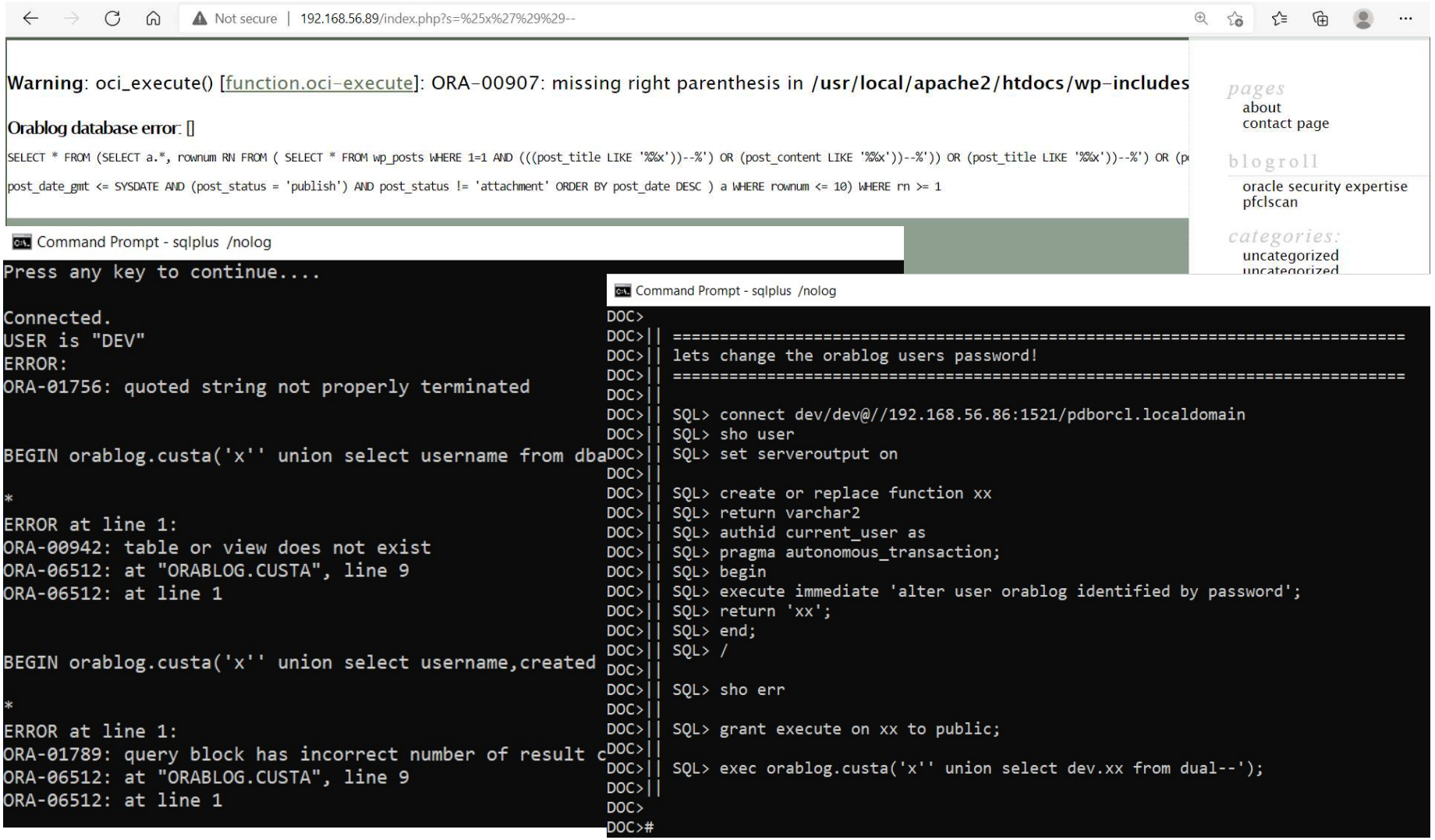
Orablog: Web Based CMS Application

DBA, DEV and Application Support Users



PFCL
PETEFINNIGAN.COM LIMITED

Some of the Hacks



Conclusions From The Hacks

- We are able to hack the database in a number of ways and as a number of “actors” -
- **From the web application**
  - We can access data processed only by back end users
  - We can change passwords, turn off audit
  - We can extract program code and create procedures
- **As a DBA**
  - We can access any data and change any functionality
- **As a power user (developer)**
  - We can exploit the database to the same level as an end user
- In all cases we use shared accounts and responsibility

Audit Report – AUD$

- Ignore Logon / logoff
- CUSTA and CREDIT_CARD are accessed normally as well as attack – is it an attack?
- No SQL Injection caught
- No Password change or AUDIT Delete
- No 942, 1031

Command Prompt - sqlplus /nolog

```
C       162530 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 08.25.24.771857 AM +01:00  EXECUTE PROCEDURE            0 ORABLOG    CUSTA
C       162530 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 08.25.24.786508 AM +01:00  EXECUTE PROCEDURE            0 ORABLOG    CUSTA
C       162530 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 08.25.27.107701 AM +01:00  EXECUTE PROCEDURE            0 ORABLOG    CUSTA
C       162530 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 08.25.27.197493 AM +01:00  EXECUTE PROCEDURE            0 ORABLOG    CUSTA
C       162530 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 08.25.30.041001 AM +01:00  EXECUTE PROCEDURE            0 ORABLOG    CUSTA
C       162531 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 08.25.34.023112 AM +01:00  EXECUTE PROCEDURE            0 ORABLOG    CUSTA
C       162532 Pete     WORKGROUP\DESKTOP-R7 DBAUSER    22-SEP-21 08.25.39.617074 AM +01:00  SESSION REC                  0 ORABLOG    CREDIT_CARD
C       162532 Pete     WORKGROUP\DESKTOP-R7 DBAUSER    22-SEP-21 08.25.39.630632 AM +01:00  SESSION REC                  0 ORABLOG    CREDIT_CARD
C       162532 Pete     WORKGROUP\DESKTOP-R7 DBAUSER    22-SEP-21 08.25.39.630717 AM +01:00  SESSION REC                  0 ORABLOG    CREDIT_CARD
C       162926 apache   oel59orablog.localdo ORABLOG    22-SEP-21 02.33.37.520009 PM +01:00  LOGON                        0
C       162928 apache   oel59orablog.localdo ORABLOG    22-SEP-21 02.34.14.918925 PM +01:00  LOGON                        0
C       162928 apache   oel59orablog.localdo ORABLOG    22-SEP-21 02.34.14.969368 PM +01:00  SESSION REC                  0 ORABLOG    CREDIT_CARD
C       162929 apache   oel59orablog.localdo ORABLOG    22-SEP-21 02.34.28.527758 PM +01:00  LOGON                        0
C       162930 apache   oel59orablog.localdo ORABLOG    22-SEP-21 02.34.38.289373 PM +01:00  LOGON                        0
C       162931 apache   oel59orablog.localdo ORABLOG    22-SEP-21 02.34.46.068759 PM +01:00  LOGON                        0
C       162933 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.46.985575 PM +01:00  LOGON                        0
C       162933 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.47.004663 PM +01:00  SESSION REC               2004 ORABLOG    CREDIT_CARD
C       162933 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.51.545630 PM +01:00  LOGOFF                       0
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.52.826167 PM +01:00  LOGON                        0
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.52.849885 PM +01:00  EXECUTE PROCEDURE            0 ORABLOG    CUSTA
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.52.851277 PM +01:00  SESSION REC               2004 ORABLOG    CREDIT_CARD
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.52.863158 PM +01:00  EXECUTE PROCEDURE            0 ORABLOG    CUSTA
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.52.864073 PM +01:00  SESSION REC               1789 ORABLOG    CREDIT_CARD
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.58.939450 PM +01:00  EXECUTE PROCEDURE            0 ORABLOG    CUSTA
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.58.943765 PM +01:00  SESSION REC                  0 ORABLOG    CREDIT_CARD
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.59.052341 PM +01:00  EXECUTE PROCEDURE            0 ORABLOG    CUSTA
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.59.054593 PM +01:00  SESSION REC                  0 ORABLOG    CREDIT_CARD
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.35.59.054681 PM +01:00  SESSION REC                  0 ORABLOG    CREDIT_CARD
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.36.00.584314 PM +01:00  EXECUTE PROCEDURE            0 ORABLOG    CUSTA
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.36.00.586784 PM +01:00  SESSION REC                  0 ORABLOG    CREDIT_CARD
C       162934 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.36.01.714810 PM +01:00  LOGOFF                       0
C       162936 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.36.02.961815 PM +01:00  LOGON                        0
C       162936 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.36.03.022558 PM +01:00  EXECUTE PROCEDURE            0 ORABLOG    CUSTA
C       162936 Pete     WORKGROUP\DESKTOP-R7 DEV        22-SEP-21 02.36.04.832193 PM +01:00  LOGOFF                       0
C       162937 Pete     WORKGROUP\DESKTOP-R7 DBAUSER    22-SEP-21 02.36.08.366763 PM +01:00  LOGON                        0
C       162937 Pete     WORKGROUP\DESKTOP-R7 DBAUSER    22-SEP-21 02.36.08.392050 PM +01:00  SESSION REC                  0 ORABLOG    CREDIT_CARD
C       162937 Pete     WORKGROUP\DESKTOP-R7 DBAUSER    22-SEP-21 02.36.08.415038 PM +01:00  SESSION REC                  0 ORABLOG    CREDIT_CARD
C       162937 Pete     WORKGROUP\DESKTOP-R7 DBAUSER    22-SEP-21 02.36.08.415122 PM +01:00  SESSION REC                  0 ORABLOG    CREDIT_CARD
C       162937 Pete     WORKGROUP\DESKTOP-R7 DBAUSER    22-SEP-21 02.36.09.233039 PM +01:00  LOGOFF                       0

58 rows selected.

SQL>
```

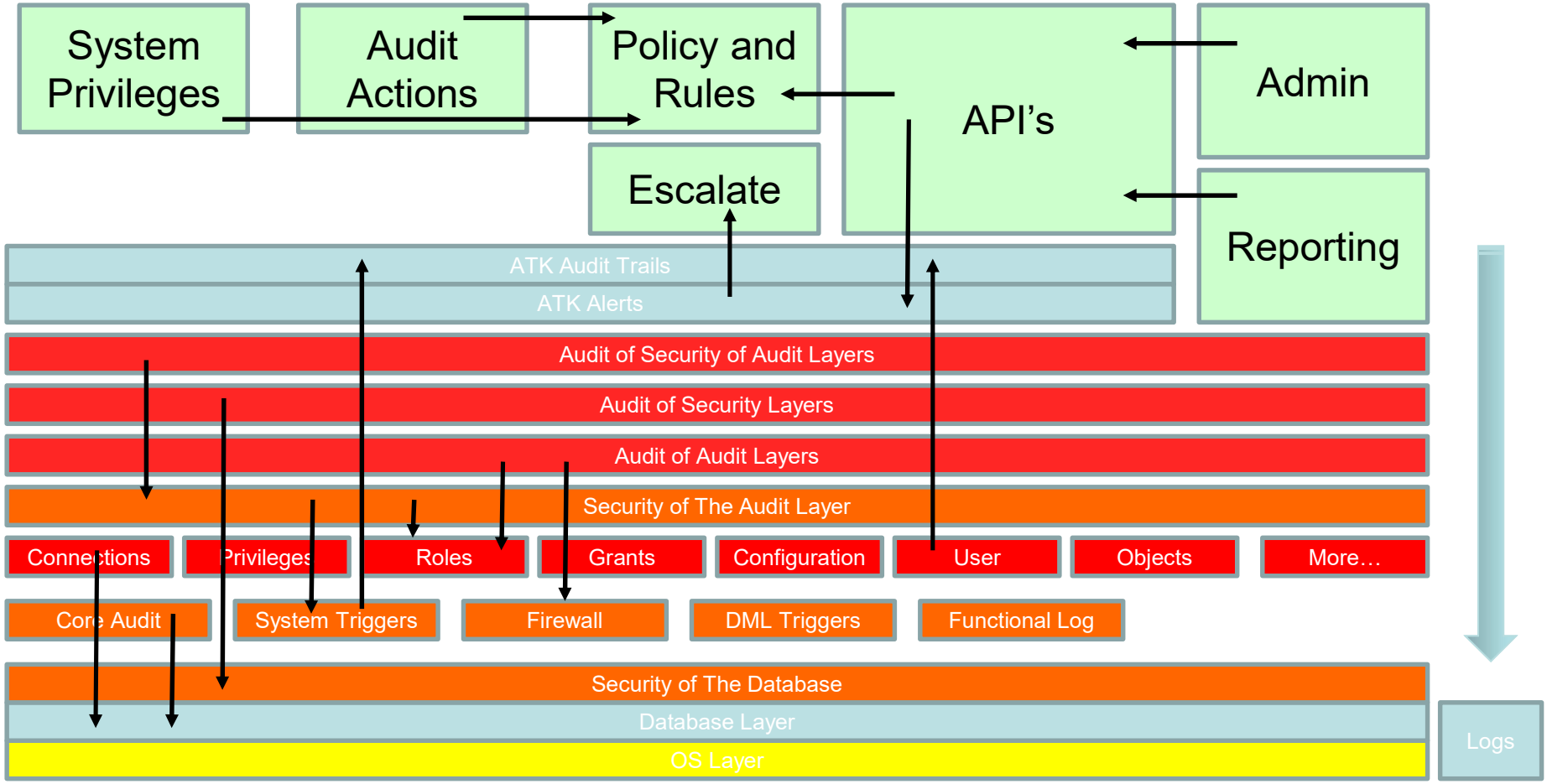
Section

Implement an Example Audit Trail

PFCL
PETEFINNIGAN.COM LIMITED

- PL/SQL and SQL based toolkit – 17k lines of code.
- We use in consulting engagements
- Audit the database engine itself

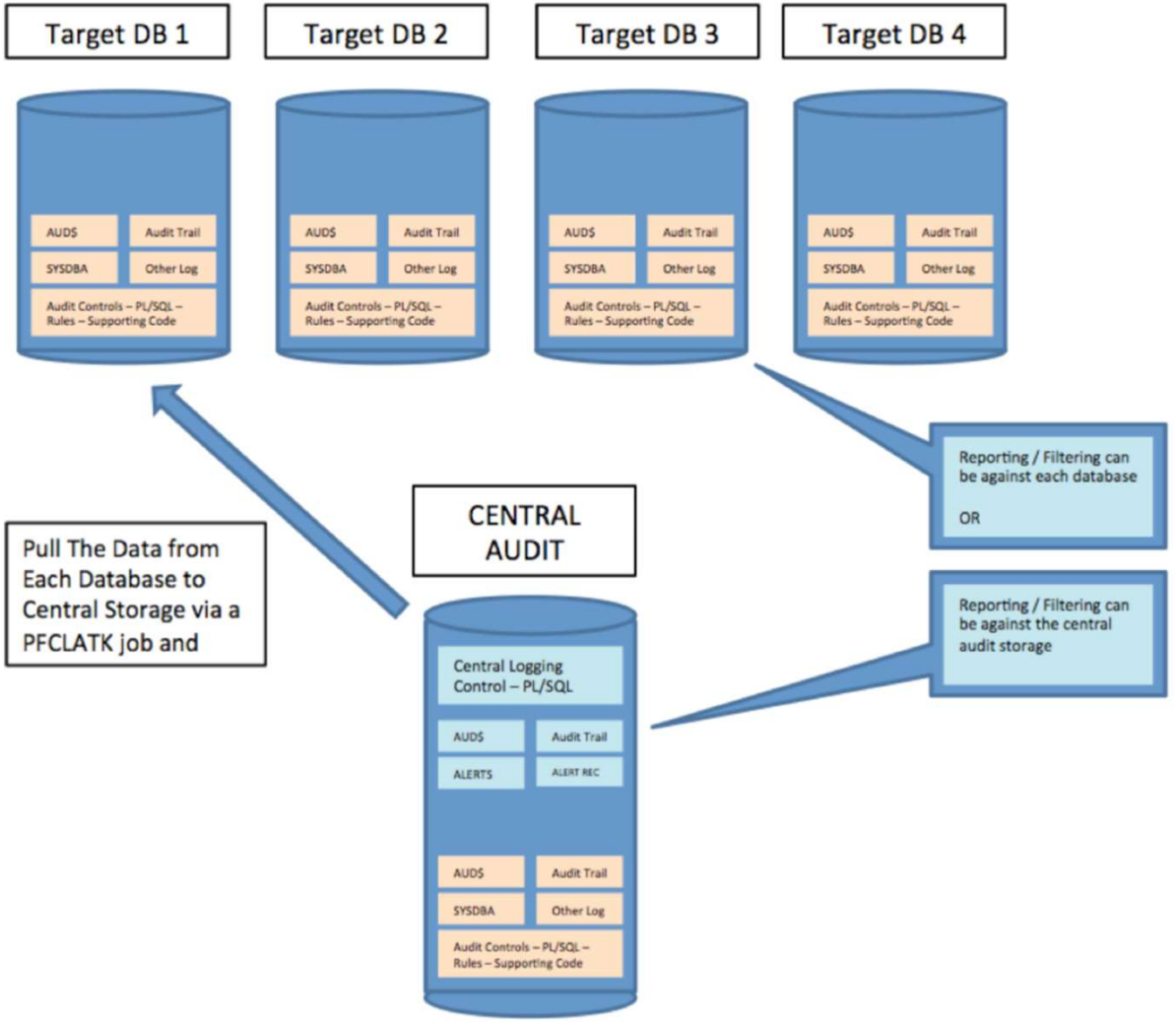
PFCLATK Block Overview



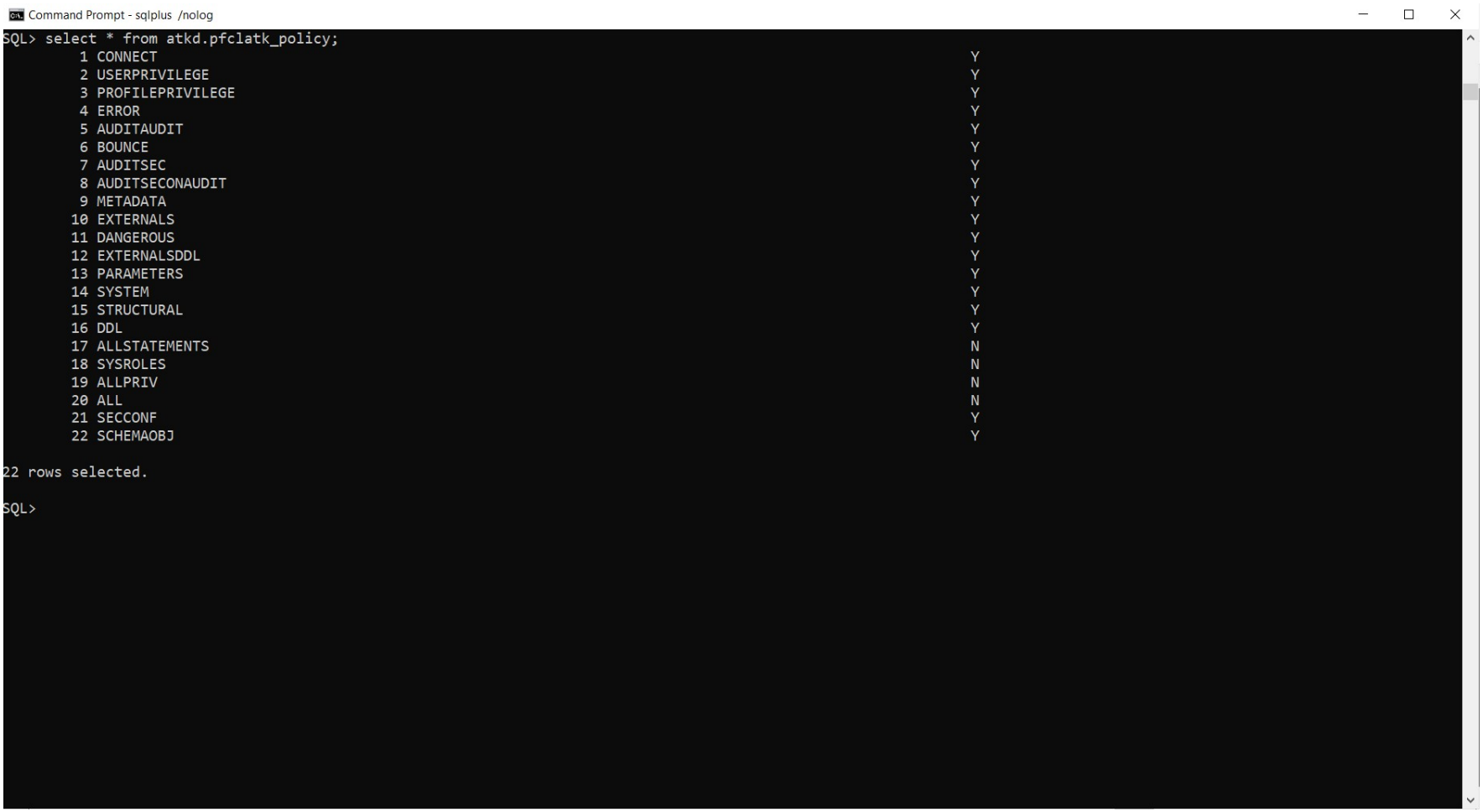
PFCL
PETEFINNIGAN.COM LIMITED

PFCLATK Architecture

- The PFCLATK toolkit is designed to be deployed to a target or central database
- When enabled simply adding target link details to the ATC database starts the PUL process automatically



Policies I Will Implement

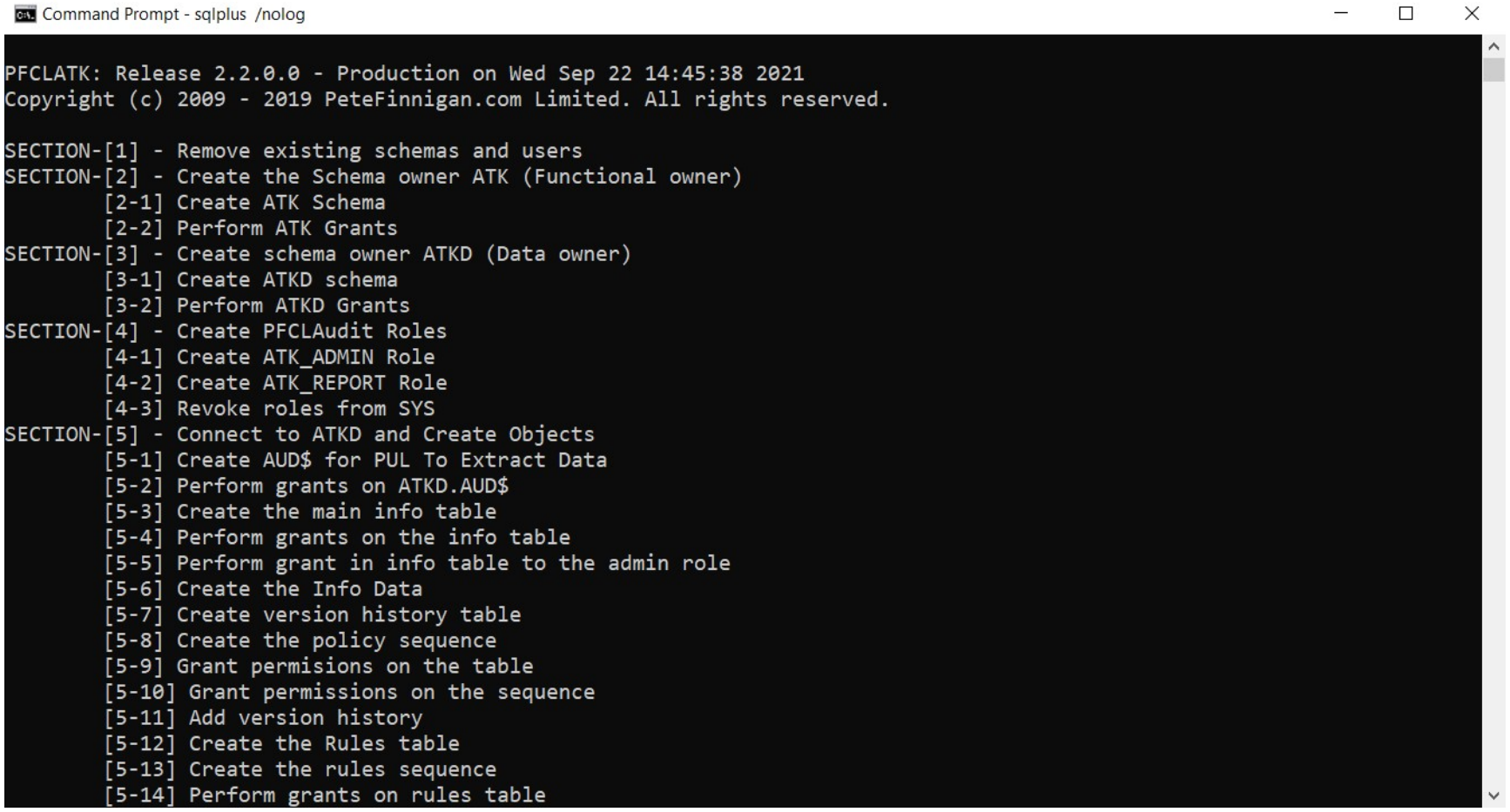


Demo: Configure and deploy

Configure and Deploy

- Edit atk.sql
  - Edit required settings needed for the toolkit
- Edit conf.sql
  - Add connection details
- Demo deployment
  - Run atk.sql

Install the Audit Events and Toolkit

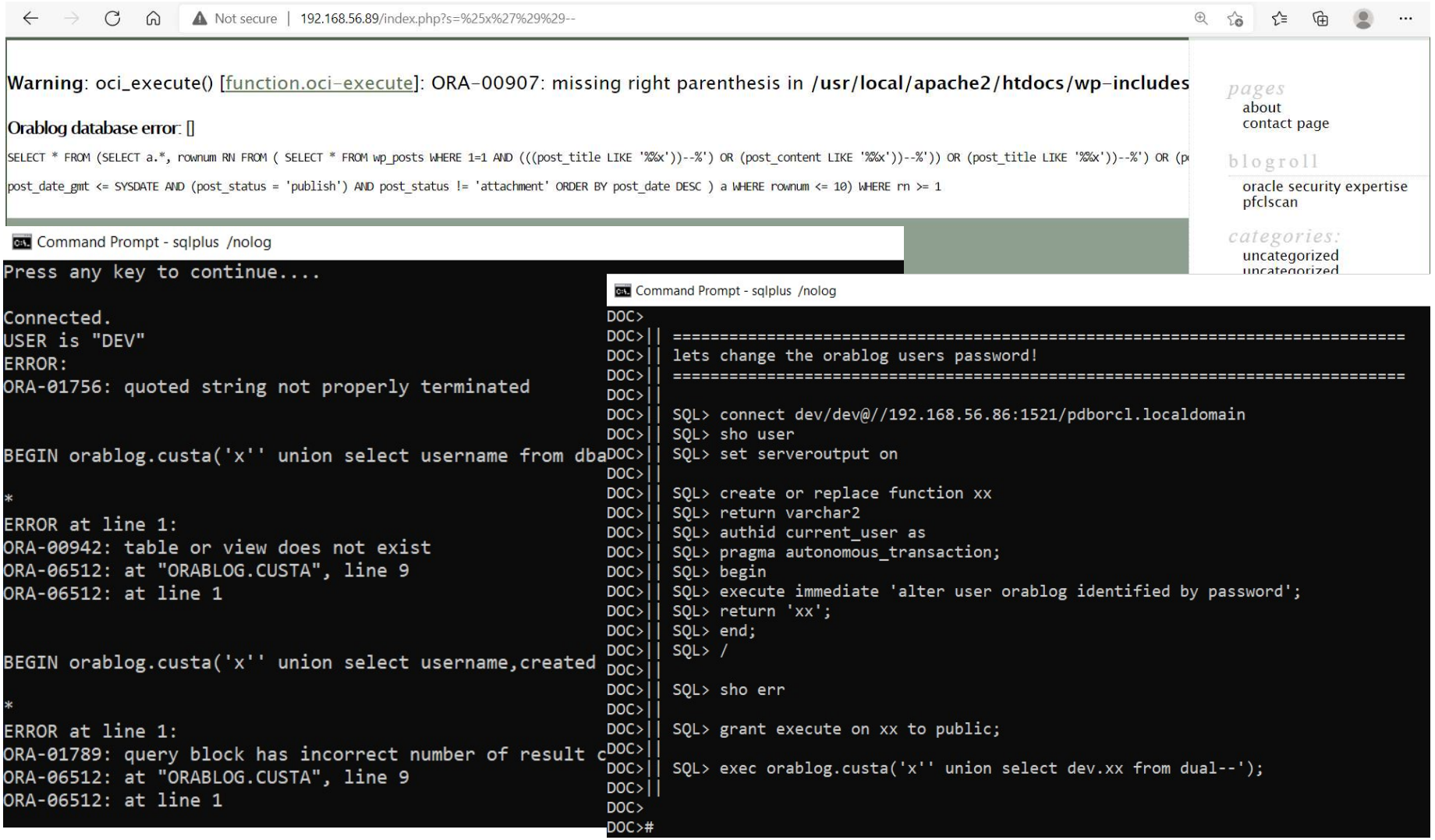


Section

Hacking Again

PFCL
PETEFINNIGAN.COM LIMITED

Some of the Hacks



PFCL
PETEFINNIGAN.COM LIMITED

Audit Report - New

- We catch the NOAUDIT
- We catch the password changes
- We catch all errors, 942, 1789, 907, 911 etc
- We catch the creation of the hack procedure
- **All events are not caught 100% but the attack was not based on the audit design!!**

Command Prompt - sqlplus /nolog

```
                                                                                                                                                      RE

DD-BE  22-SEP-21 03.10.52.713992 PM +01:00 DEV       grant execute on xx                   DEV        WORKGROUP\DESKTOP-R  Pete       163002 192.168.56.1
AUDIT  22-SEP-21 03.10.52.716622 PM +01:00 DEV       DEV.XX              GRANT OBJECT                 WORKGROUP\DESKTOP-R7 Pete       163002
AUDIT  22-SEP-21 03.10.52.728008 PM +01:00 DEV       ORABLOG.CUSTA       EXECUTE PROCEDURE            WORKGROUP\DESKTOP-R7 Pete       163002
AUDIT  22-SEP-21 03.10.52.729926 PM +01:00 DEV       ORABLOG.CREDIT_CARD SESSION REC                  WORKGROUP\DESKTOP-R7 Pete       163002
NO-BE  22-SEP-21 03.10.52.732449 PM +01:00 DEV       noaudit select on o                   DEV        WORKGROUP\DESKTOP-R  Pete       163002 192.168.56.1
DD-BE  22-SEP-21 03.10.52.733946 PM +01:00 DEV       noaudit select on o                   DEV        WORKGROUP\DESKTOP-R  Pete       163002 192.168.56.1
AUDIT  22-SEP-21 03.10.52.736187 PM +01:00 DEV       ORABLOG.CREDIT_CARD NOAUDIT OBJECT               WORKGROUP\DESKTOP-R7 Pete       163002
LOGOF  22-SEP-21 03.10.54.609230 PM +01:00 DEV                                             DEV        WORKGROUP\DESKTOP-R  Pete       163002 192.168.56.1
AUDIT  22-SEP-21 03.10.54.612925 PM +01:00 DEV       .                   LOGOFF                       WORKGROUP\DESKTOP-R7 Pete       163002
LOGON  22-SEP-21 03.10.54.661608 PM +01:00 SYS                                             SYS        WORKGROUP\DESKTOP-R  Pete   4294967295 192.168.56.1
LOGOF  22-SEP-21 03.11.00.140215 PM +01:00 SYS                                             SYS        WORKGROUP\DESKTOP-R  Pete   4294967295 192.168.56.1
AUDIT  22-SEP-21 03.11.00.196948 PM +01:00 DEV       .                   LOGON                        WORKGROUP\DESKTOP-R7 Pete       163003 Authenticated b CREATE S
ESSION
                                                                                                                                      y: DATABASE; Cl
                                                                                                                                      ient address: (
                                                                                                                                      ADDRESS=(PROTOC
                                                                                                                                      OL=tcp)(HOST=19
                                                                                                                                      2.168.56.1)(POR
                                                                                                                                      T=1043))

LOGON  22-SEP-21 03.11.00.198875 PM +01:00 DEV                                             DEV        WORKGROUP\DESKTOP-R  Pete       163003 192.168.56.1
AUDIT  22-SEP-21 03.11.00.200549 PM +01:00 DEV       SYS.DBMS_DEBUG_JDWP EXECUTE PROCEDURE            WORKGROUP\DESKTOP-R7 Pete       163003
AUDIT  22-SEP-21 03.11.00.201956 PM +01:00 DEV       SYS.AUD$            SELECT                       WORKGROUP\DESKTOP-R7 Pete       163003
DD-BE  22-SEP-21 03.11.00.213281 PM +01:00 DEV       create or replace f                   DEV        WORKGROUP\DESKTOP-R  Pete       163003 192.168.56.1
AUDIT  22-SEP-21 03.11.00.222751 PM +01:00 DEV       DEV.XX              CREATE FUNCTION              WORKGROUP\DESKTOP-R7 Pete       163003                 CREATE P
ROCEDU
                                                                                                                                                      RE

DD-BE  22-SEP-21 03.11.00.241730 PM +01:00 DEV       grant execute on xx                   DEV        WORKGROUP\DESKTOP-R  Pete       163003 192.168.56.1
AUDIT  22-SEP-21 03.11.00.242732 PM +01:00 DEV       SYS.DBMS_DEBUG_JDWP EXECUTE PROCEDURE            WORKGROUP\DESKTOP-R7 Pete       163003
AUDIT  22-SEP-21 03.11.00.245078 PM +01:00 DEV       DEV.XX              GRANT OBJECT                 WORKGROUP\DESKTOP-R7 Pete       163003
AUDIT  22-SEP-21 03.11.00.255571 PM +01:00 DEV       ORABLOG.CUSTA       EXECUTE PROCEDURE            WORKGROUP\DESKTOP-R7 Pete       163003
DD-BE  22-SEP-21 03.11.00.260781 PM +01:00 DEV       alter user orablog                    DEV        WORKGROUP\DESKTOP-R  Pete       163003 192.168.56.1
AUDIT  22-SEP-21 03.11.00.264281 PM +01:00 DEV       .ORABLOG            ALTER USER                   WORKGROUP\DESKTOP-R7 Pete       163003
LOGOF  22-SEP-21 03.11.02.915059 PM +01:00 DEV                                             DEV        WORKGROUP\DESKTOP-R  Pete       163003 192.168.56.1
AUDIT  22-SEP-21 03.11.02.918414 PM +01:00 DEV       .                   LOGOFF                       WORKGROUP\DESKTOP-R7 Pete       163003
LOGON  22-SEP-21 03.11.02.969985 PM +01:00 SYS                                             SYS        WORKGROUP\DESKTOP-R  Pete   4294967295 192.168.56.1
AU-BE  22-SEP-21 03.11.02.986466 PM +01:00 SYS       audit select on ora                   SYS        WORKGROUP\DESKTOP-R  Pete   4294967295 192.168.56.1
DD-BE  22-SEP-21 03.11.02.988030 PM +01:00 SYS       audit select on ora                   SYS        WORKGROUP\DESKTOP-R  Pete   4294967295 192.168.56.1
DD-BE  22-SEP-21 03.11.02.995716 PM +01:00 SYS       alter user orablog                    SYS        WORKGROUP\DESKTOP-R  Pete   4294967295 192.168.56.1
LOGON  22-SEP-21 03.11.14.072448 PM +01:00 SYS                                             SYS        oel1124.localdomain  oracle     163004
```

Conclusions

- Design first
- Create the events
- Decide what technical solution to use
- Decide what raw audit to collect
- Decide how to detect that an audit event occurred
- **Audit the database engine**

Designing a Good Database Audit Trail