Unified Audit und SYSLOG

Curse or Blessing?

Stefan Oehrli

Stefan Oehrli – Data Platforms

stefan.oehrli@accenture.com

Tech Architecture Manager

- Since 1997 active in various IT areas
- More than 25 years of experience in Oracle databases
- Focus: Protecting data and operating databases securely
  - Security assessments and reviews
  - Database security concepts and their implementation
  - Oracle Backup & Recovery concepts and troubleshooting
  - Oracle Enterprise User and Advanced Security, DB Vault, ...
  - Oracle Directory Services
- Co-author of the book The Oracle DBA (Hanser, 2016/07)

oradba.ch

@stefanoehrli

>

DATA PLATFORMS

WHY? We are the game changer for our client's data platform projects

HOW? Maximum automation, maximum efficiency, maximum quality!

WHAT? We build innovative data platforms based on our blueprints, assets and tools.

3 key benefits

1 Architecture expertise from hands-on projects

2 Delivery of tailor-made data platforms

3 Integrated Teams - Like a Rowing team, perfect alignment and interaction.

Tools and Blueprints

Key enabler for the implementation of modern data platforms at a high speed and quality.

Continuous Optimization

Tools and Blueprints are continuously optimized to the customer and project’s needs.

Expertise

Expert group for modern data platforms from technical implementation to project management and organization

Unified Audit

How well do SYSLOG and Unified Audit get along?

1. Introduction
2. Unified Audit and Multitenant
3. Pig Bicture
4. Basic Configuration
5. Root Container (CDB$ROOT)
6. Pluggable Database (PDB)
7. Setup Example
8. Conclusion

1

Introduction

Why use SYSLOG for databases at all?

Introduction

Motivation for Auditing and SYSLOG Integration

Why Database Security at all?

- Protection of **company** and its business
- Protection of **employees**, **customers** and others
- and of course, **compliance** and **regulatory** requirements

Security measures are complex and expensive

- **Management** of security configuration e.g., Audit
- Availability of **Security Options** and **Features** (Edition, License etc.)
- **Segregation of Duties** e.g., DBAs audit themselves?
- Traceability and auditability

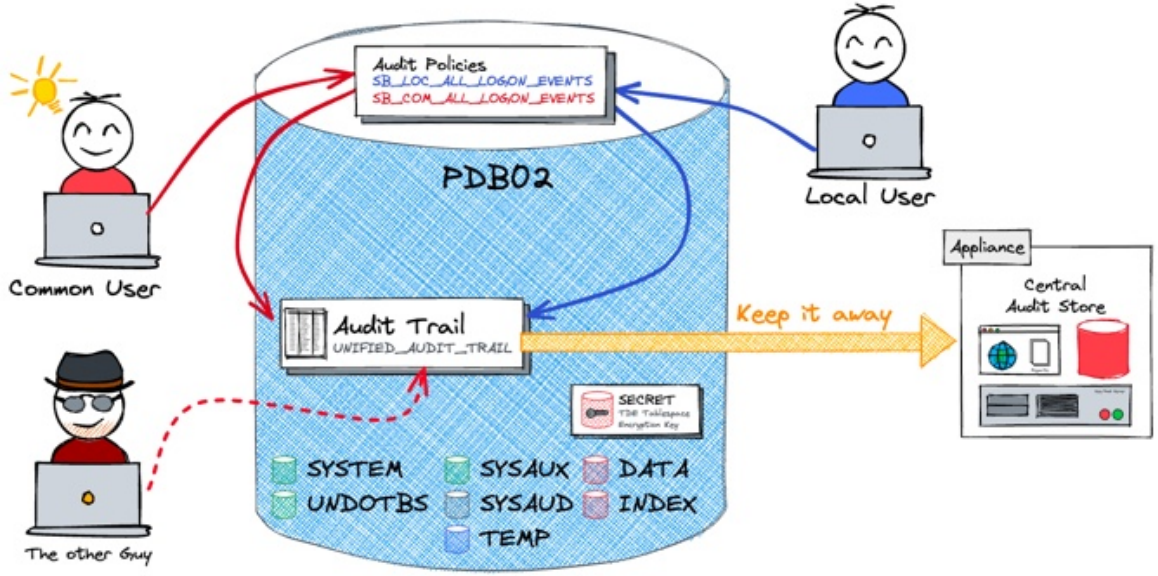
Audit and traceability is a central aspect of any security concept.

Why SYSLOG at all?

Centralized and decentralized storage of audit data

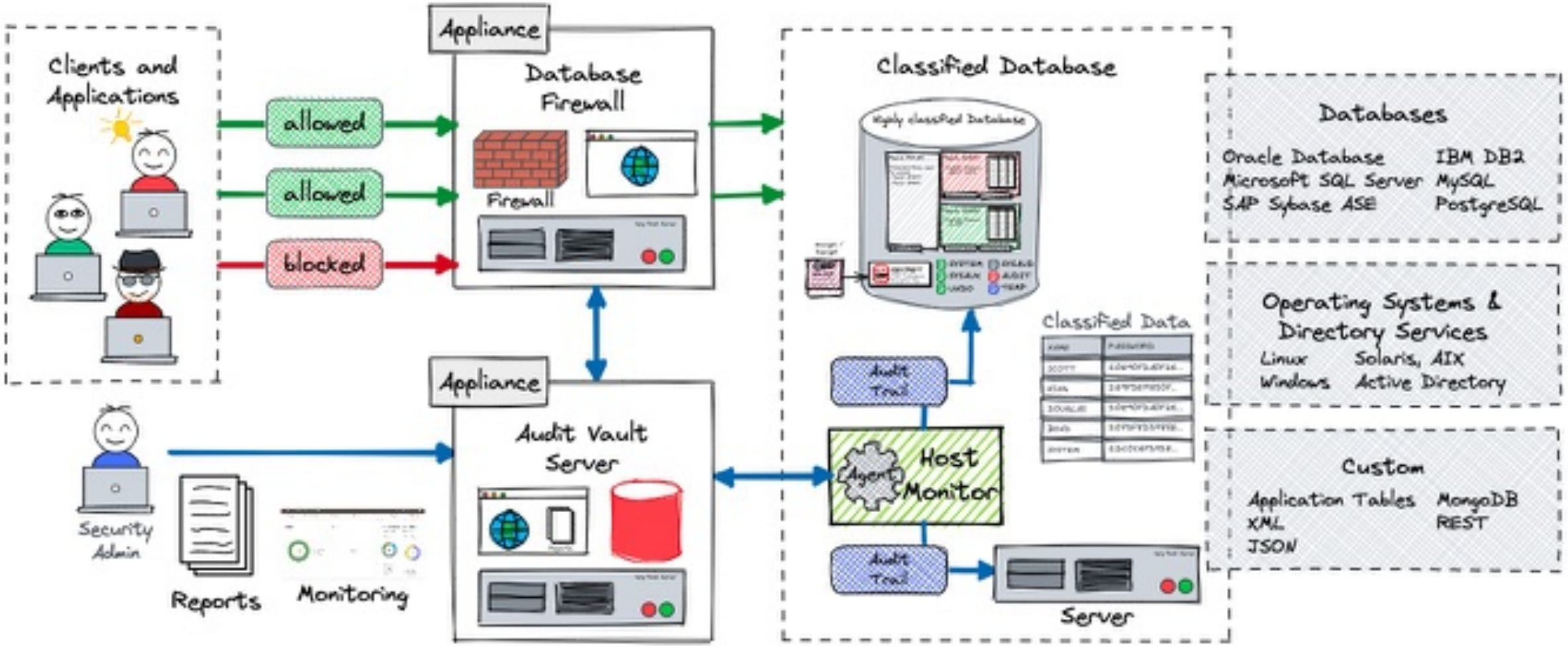
There are a couple of reasons to **not keep** audit data locally

- Risk of **tampering** of audit data
- Revision and **compliance** requirements
- Cost of the **storage** space
- Unnecessary **operating expenses** e.g. database performance, backup etc. audit trail is always part of the production
- **Central analysis** and alerting across multiple databases or systems



Centralliezed Audit Data

Oracle AVDF, the solution from Oracle…



Centralliezed Audit Data

Custom Solutions, what ever you like to build…

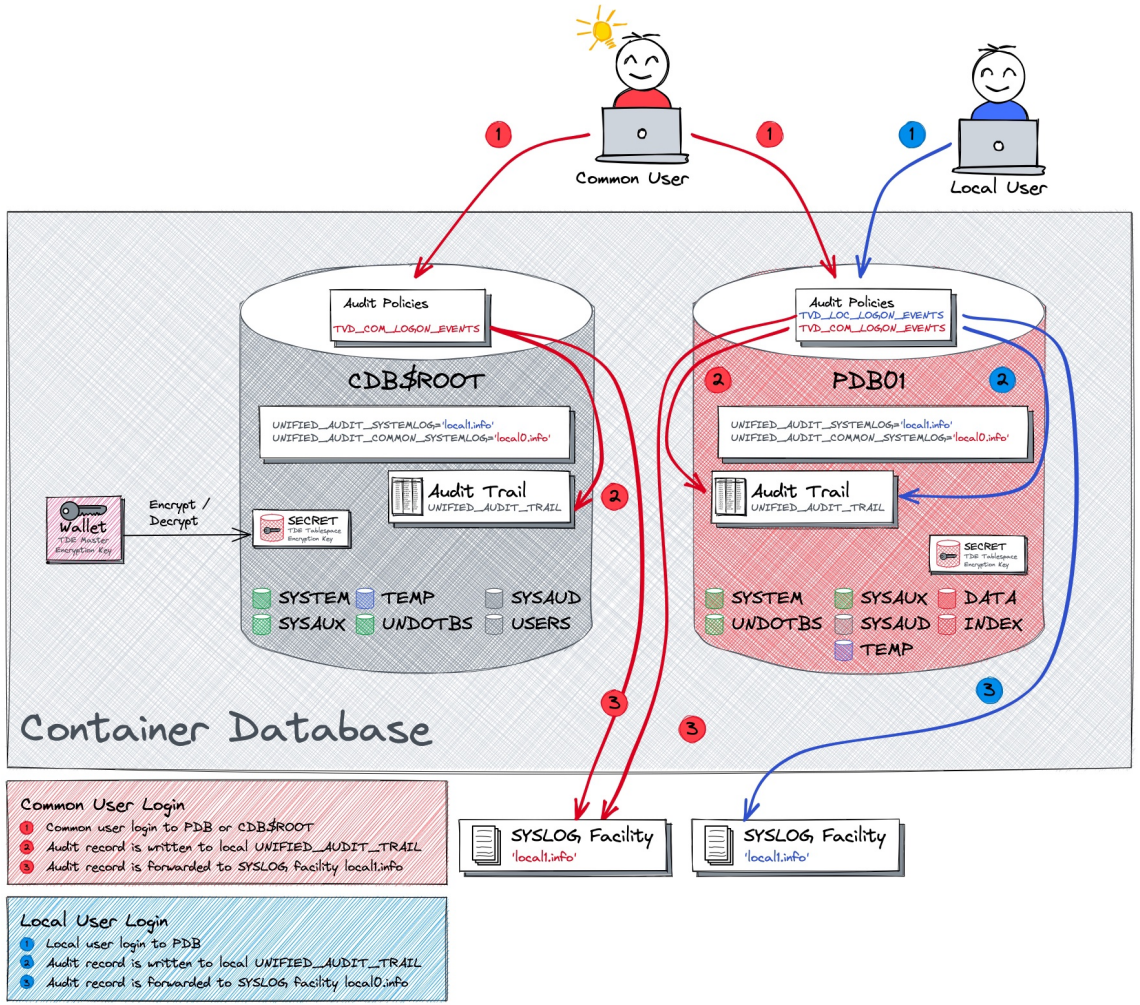
The solutions are usually **limited** to…

- Central Repository
- Reporting
- SOC (Security Operation Center) Integration

Possible Solution Approaches

- **Splunk** Audit data Collection
- **Elasticsearch** or ELK Stack
- **SYSLOG** integration

Usually **no** Audit Policy Management and Database Security Assessment



2

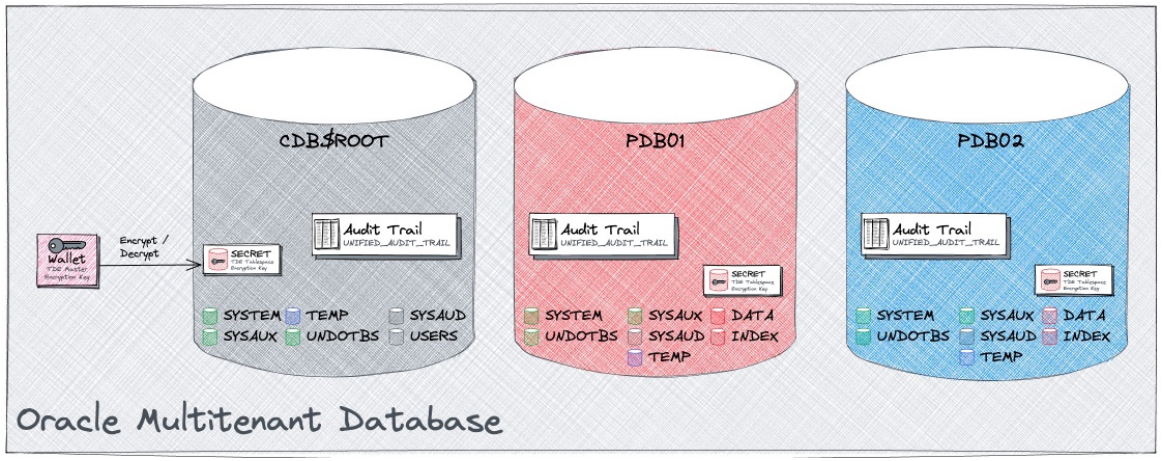
Unified Audit and Multitenant

What is special about auditing in multitenant environments?

Unified Audit in Multitenant Database

Common, local or what else?

- Each PDB has **its own** audit trail within a dedicated tablespace
- **Central spill** over location for audit during read only or mount state
  - Predefined in $ORACLE_BASE/audit
- Dedicated audit trails per PDB implies…
  - … distribution of relevant audit information, there is not “unified” for the whole CDB
  - … increased administration effort i.e. housekeeping
- Challenges with the terms common and local

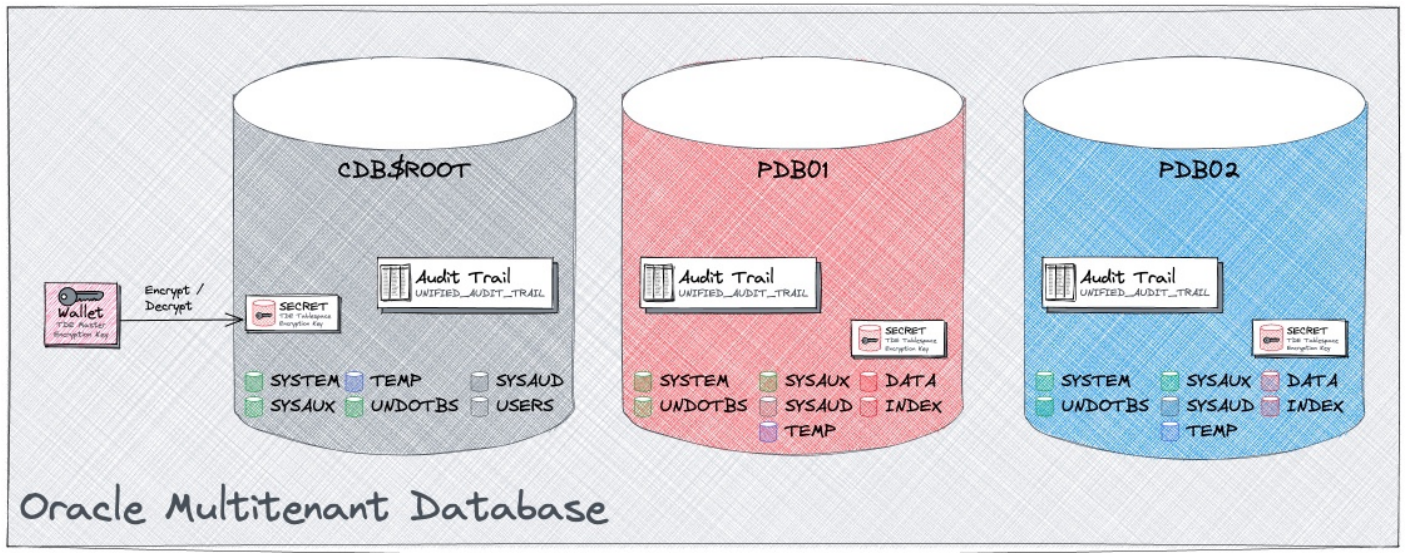


Unified Audit in Multitenant Database

Common, local or what?

Oracle has introduced a couple of common views / package

- CDB Views for Unified Audit display all audit data
- DBMS_AUDIT_MGMT can be used at CDB level or in each PDB
  - Yeah, but a common purge job **will fail** if a PDB is closed when doing housekeeping…
- Never mind still challenging…



Unified Audit at CDB$ROOT

Access all audit unified trails via CDB$RROOT?

- CDB_UNIFIED_AUDIT_TRAIL common VIEW for all audit trails with column CON_DI

```
SQL> SELECT con_id, event_timestamp, dbusername, action_name FROM cdb_unified_audit_trail
WHERE rownum <6;


CON_ID EVENT_TIMESTAMP                DBUSERNAME ACTION_NAME
------ ---------------------------- ---------- ----------------------------
     1 27-MAY-20 07.41.30.212698 PM SYS        ALTER PLUGGABLE DATABASE
     1 27-MAY-20 07.41.45.765554 PM SYS        LOGON
     1 27-MAY-20 07.41.45.978964 PM SYS        LOGON
```

- Be careful old UNIFIED_AUDIT_TRAIL where using EVENT_TIMESTAMP with type TIMESTAMP(6) WITH LOCAL TIME ZONE
  - Could be **really funny** when PDBs use different DB time zones
  - Latest releases due have also EVENT_TIMESTAMP_UTC

Unified Audit at CDB$ROOT

Maintain audit trails in one place using DBMS_AUDIT_MGMT?

- DBMS_AUDIT_MGMT package / procedures allow to specify CONTAINER_CURRENT or ALL

```
BEGIN
    DBMS_AUDIT_MGMT.CREATE_PURGE_JOB (
        AUDIT_TRAIL_TYPE           => DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
        AUDIT_TRAIL_PURGE_INTERVAL => 12,
        AUDIT_TRAIL_PURGE_NAME     => 'Daily_Audit_Purge_Job',
        container                  => dbms_audit_mgmt.container_all,
        USE_LAST_ARCH_TIMESTAMP    => TRUE);
END;
/
```

- DBMS_AUDIT_MGMT allows housekeeping **over all PDB** but…
- … if one of the PDB is **close** the housekeeping **does report** an **error**

Common and local audit policies

Ok, can we create a common policy to rule them all?

But what is a common audit policy? I.e. Policy which is valid for all PDBs?

- **No, my dear**, at least not completely correct

COMMON audit policy

- Policies which are defined on CDB root with CONTAINER=ALL
- valid / visible in all PDBs
- When enabled the will audit actions for **COMMON users** in this particular PDB.
- LOCAL user in PDBs **will not** be audited by COMMON audit policies!

LOCAL audit policy

- Defined locally in the PDB or CDB root
- When enabled a local audit policy is valid for LOCAL and COMMON users in this PDB

Consequences

Your audit concept does not get simplier in multitenant environments….

- There is **no possibility** to **enforce common** audit settings to all PDBs for all users
- Common audit policies will be **visible** in each PDBs
  - Risk to have to much policies active => decrease in performance
- It is highly recommended to have a proper user / role concept before starting with audit
  - Do we have common user?
  - If yes how and why they are used e.g. just common user and rarely local user => no need to have to much local audit policies
- Do we need common policies? If yes how much?
- **Just a hint**: Oracle default audit policies are created as local policies.
  - Audit is threaded individually in each PDB
  - No “umbrella” audit

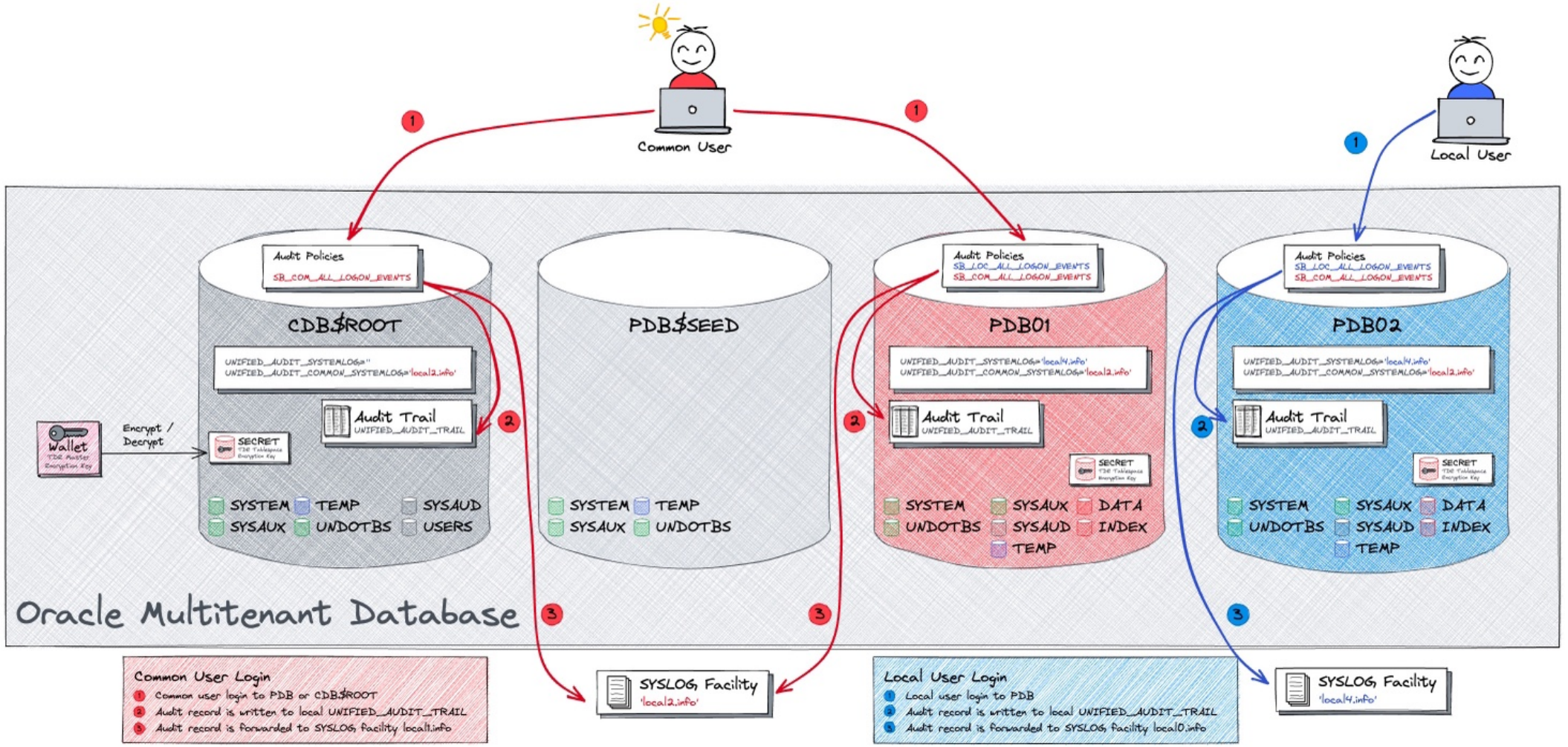
3

Pig Bicture

Where are my audit trails?

Pig Bicture

The Oracle Unified Audit Trails in a multitenant database



4

Basic Configuration

What might a simple use case look like?

Oracle Databases and SYSLOG

SYSLOG improvements over time…

A brief history about Oracle Audit and SYSLOG integration

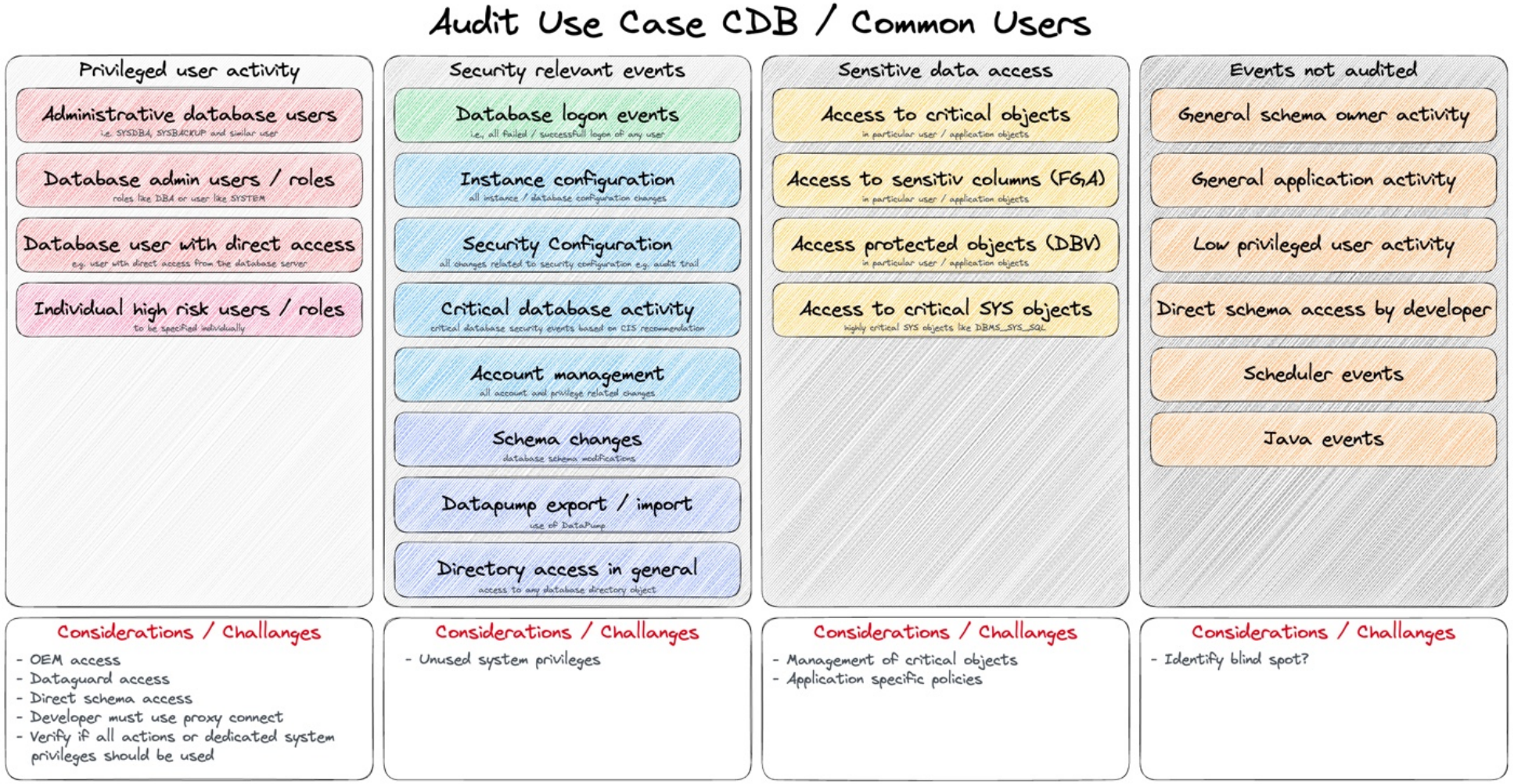
- **AUDIT_SYSLOG_LEVEL** Initialisation parameter in legacy audit (i.e. pre 12c and none unified audit) support a all or nothing approach
- No SYSLOG support in first releases of Oracle 12c i.e., 12.1 and 12.2
- **UNIFIED_AUDIT_SYSTEMLOG** new initialisation parameter introduced in Oracle 18c to configure again syslog facility and level for unified audit
- **UNIFIED_AUDIT_COMMON_SYSTEMLOG** new initialisation parameter introduced in Oracle 19c syslog facility and level for only **common** unified audit records

Major differences:

- **AUDIT_SYSLOG_LEVEL** does define SYSLOG as target for the OS audit trail
  => just one audit trail.
- Unified Audit records will always go to UNIFIED_AUDIT_TRAIL. SYSLOG is an add on

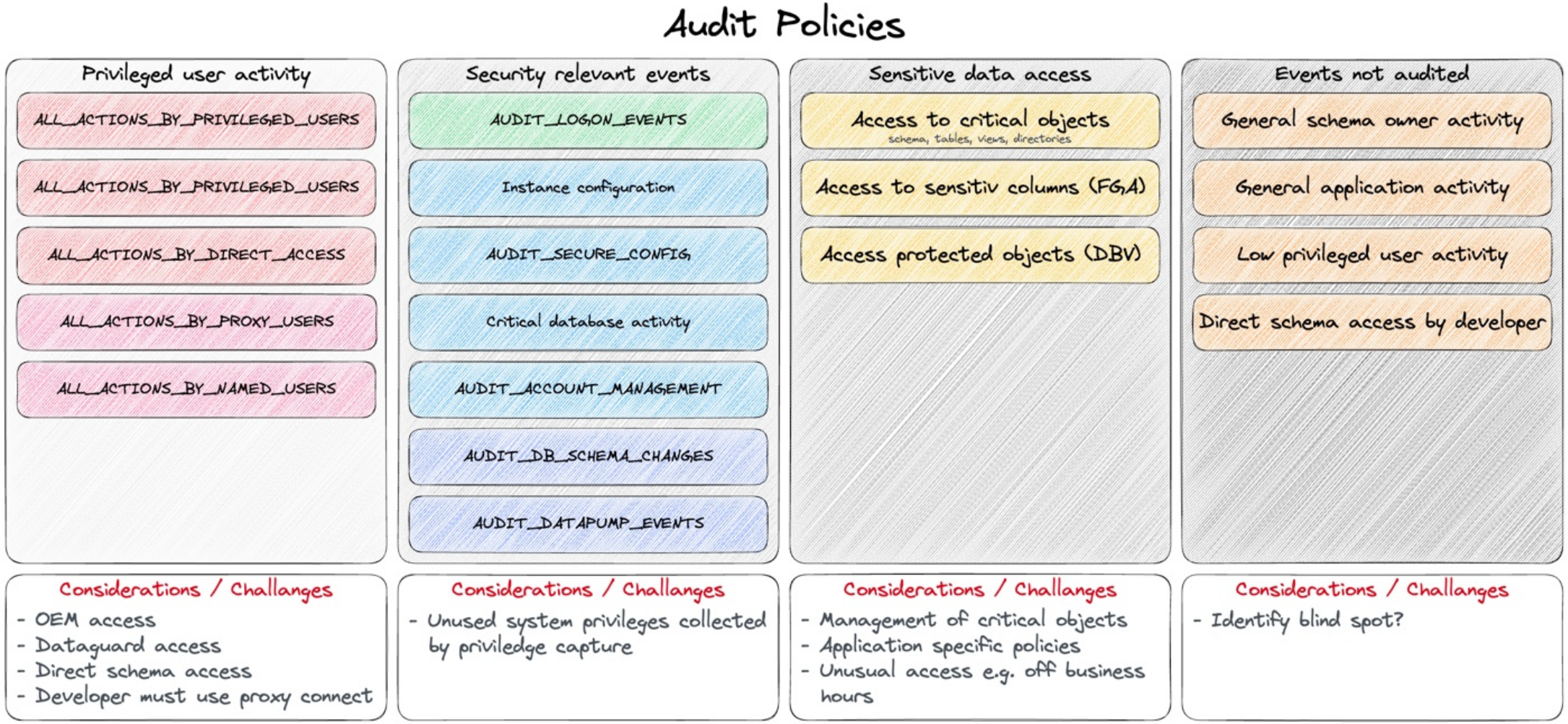
Audit Use Case

Idea to tame the ~~beast~~ audit



Audit Policies based on Use Cases

Idea to tame the ~~beast~~ audit



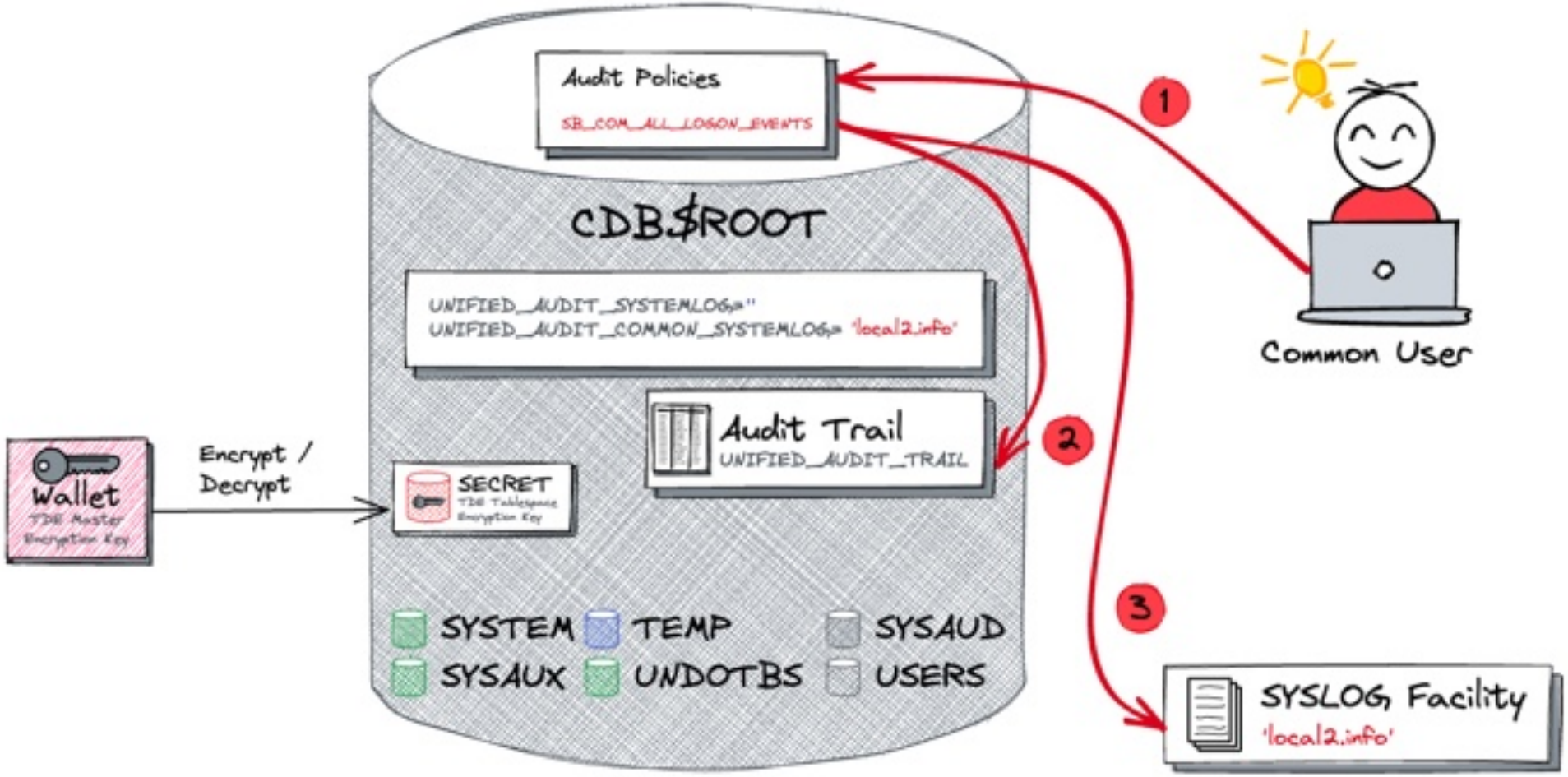
5

Root Container (CDB$ROOT)

What does the audit configuration look like in the root container?

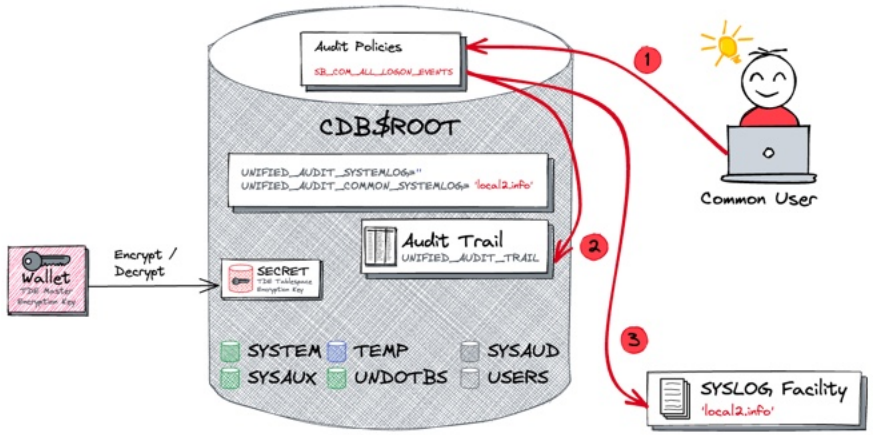
Unified Audit in CDB$ROOT

SYSLOG configuration in the root container



Unified Audit in CDB$ROOT

SYSLOG configuration in the root container explained…



Prerequisites for the use case

- Common audit policy *SB_COM_ALL_LOGON_EVENTS* defined
- SYSLOG facility defined e.g., *local2.info*
- Parameter UNIFIED_AUDIT_COMMON_SYSTEMLOG set to the SYSLOG facility

Audit Event and Records

1. Common user login to CDB$ROOT
2. Audit record is written to local UNIFIED_AUDIT_TRAIL
3. Audit record is forwarded to SYSLOG facility *local1.info*

Full audit record in UNIFIED_AUDIT_TRAIL but limited in SYSLOG

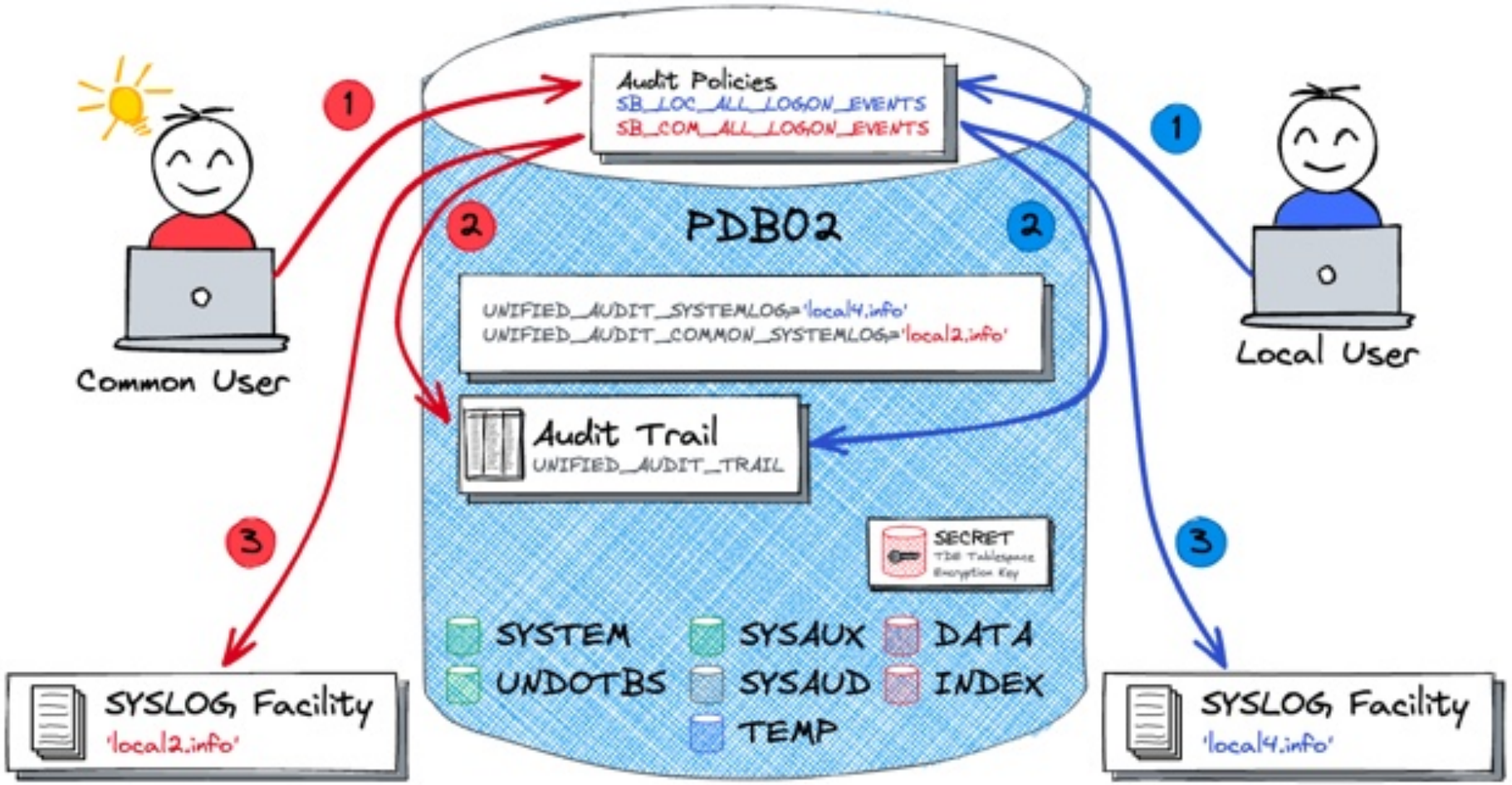
6

Pluggable Database (PDB)

What is the audit configuration in pluggable databases?

Unified Audit in PDB

SYSLOG configuration in the pluggable database



Unified Audit in PDB

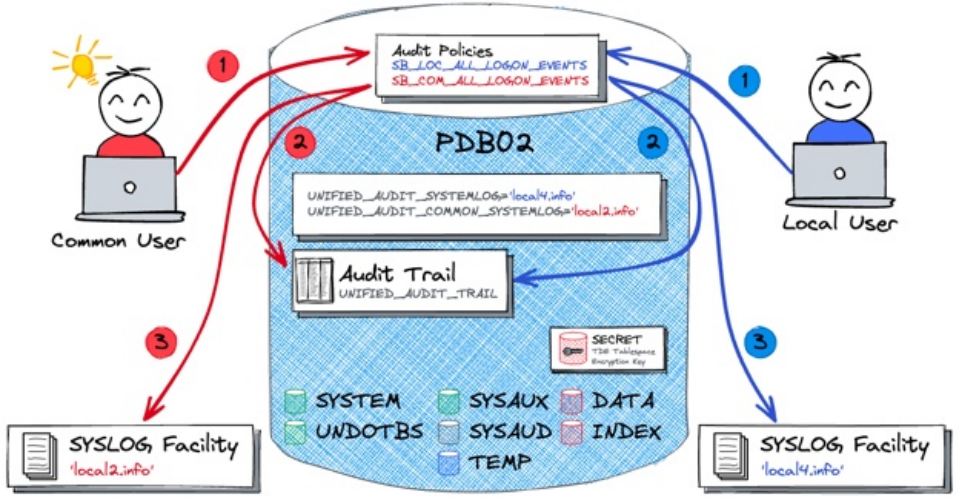
What happens when by a common user access?

Prerequisites for the use case

- Common audit policy SB_COM_ALL_LOGON_EVENTS defined
- SYSLOG facility defined e.g., *local2.info*
- Parameter UNIFIED_AUDIT_COMMON_SYSTEMLOG set to the SYSLOG facility

Audit Event and Records

1. Common user login to PDB
2. Audit record is written to local UNIFIED_AUDIT_TRAIL
3. Audit record is forwarded to SYSLOG facility *local2.info*



Full audit record in UNIFIED_AUDIT_TRAIL but limited in SYSLOG

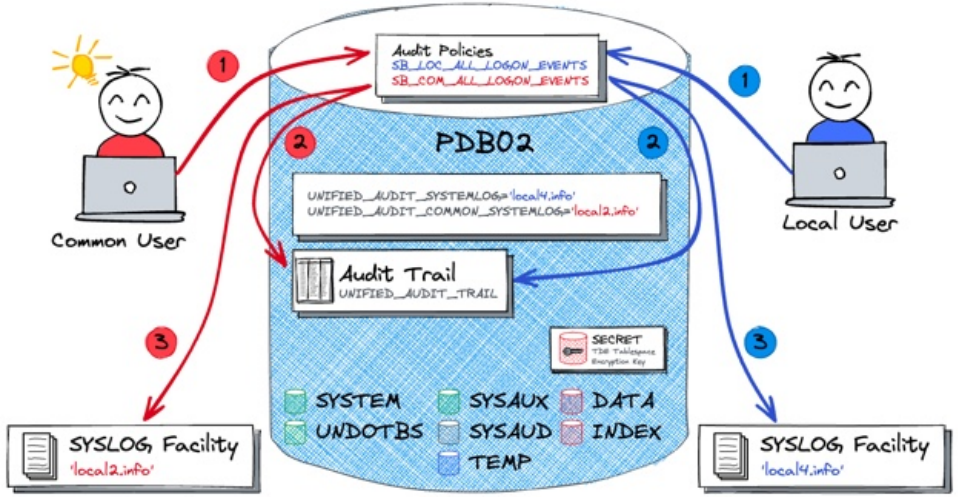
Unified Audit in PDB

What happens when by a common user access?

- Prerequisites for the use case
- Common audit policy SB_LOC_ALL_LOGON_EVENTS defined
- SYSLOG facility defined e.g. *local4.info*
- Parameter UNIFIED_AUDIT_SYSTEMLOG set to the SYSLOG facility

- Audit Event and Records

1. Local user login to PDB
2. Audit record is written to local UNIFIED_AUDIT_TRAIL
3. Audit record is forwarded to SYSLOG facility *local4.info*



Full audit record in UNIFIED_AUDIT_TRAIL but limited in SYSLOG

7

Setup Example

Let’s setup a basic example…

SYSLOG Configuration

Preparation on OS and the SYSLOG

- Setup SYSLOG facilities as user root

```
sudo vi /etc/rsyslog.conf
*.info;mail.none;authpriv.none;cron.none;local2.none;local4.none
/var/log/messages
# Unified Audit Rules
local2.info                /var/log/oracle_common_audit_records.log
local4.info                /var/log/oracle_audit_records.log
```

- Restart the RSYSLOG service as user root

```
sudo systemctl restart rsyslog.service
```

Database Configuration

Setup initialisation parameter on CDB$ROOT and PDB level

- Connect as *SYS* to *CDB$ROOT* and change *UNIFIED_AUDIT_COMMON_SYSTEMLOG*

```
CONNECT / AS SYSDBA
ALTER SYSTEM SET unified_audit_common_systemlog='local2.info'
SCOPE=SPFILE;
```

- Connect as *SYS* to *PDB1* and change *UNIFIED_AUDIT_SYSTEMLOG*

```
ALTER SESSION SET CONTAINER=PDB1;
ALTER SYSTEM SET unified_audit_systemlog='local4.info' SCOPE=SPFILE;
```

- Restart the whole container database

```
CONNECT / AS SYSDBA
STARTUP FORCE;
```

Database Configuration

Review the instance parameter

- List the current settings of audit related *init.ora* parameter in CDB$ROOT

```
SQL> SHOW PARAMETER unified_audit
NAME                 TYPE   VALUE
-------------------------------- ------- ------------------------
unified_audit_common_systemlog     string  LOCAL2.INFO
unified_audit_systemlog            string
```

- List the current settings of audit related *init.ora* parameter in PDB

```
SQL> ALTER SESSION SET CONTAINER=pdb1;
SQL> SHOW PARAMETER unified_audit

NAME                                 TYPE         VALUE
------------------------------------ ------------ ---------------------
unified_audit_common_systemlog       string       LOCAL2.INFO
unified_audit_systemlog              string       LOCAL4.INFO
```

Define Unified Audit Policy

Test if there are some audit records in SYSLOG

- Create an audit policy for all logon events of common users in CDB$ROOT and any PDB

```
CONNECT / AS SYSDBA
CREATE AUDIT POLICY sb_com_all_logon_events ACTIONS LOGON CONTAINER=ALL;
AUDIT POLICY sb_com_all_logon_events;
```

- Create a local audit policy for all logon events of local users in a particular PDB

```
ALTER SESSION SET CONTAINER=pdb1;
CREATE AUDIT POLICY sb_loc_all_logon_events ACTIONS LOGON;
AUDIT POLICY sb_loc_all_logon_events;
```

Review Unified Audit Policy

Review the current audit setup in CDB$ROOT

- Check which audit policies are enabled in the root container

```
CONN / AS SYSDBA
SET LINESIZE WINDOW
COL policy_name FOR A20
COL entity_name FOR A20

SELECT * FROM audit_unified_enabled_policies;

POLICY_NAME          ENABLED_OPTION  ENTITY_NAME ENTITY_ SUC FAI
-------------------- --------------- ------------ ------- --- ---
ORA_SECURECONFIG     BY USER         ALL USERS   USER    YES YES
ORA_LOGON_FAILURES   BY USER         ALL USERS   USER    NO  YES
SB_COM_ALL_LOGON_EVENTS BY USER      ALL USERS   USER    YES YES
```

Review Unified Audit Policy

Review the current audit setup in PDB

- Check which audit policies are enabled in the pluggable database

```
CONN / AS SYSDBA
ALTER SESSION SET container=pdb1;
SET LINESIZE WINDOW
COL policy_name FOR A20
COL entity_name FOR A20

SELECT * FROM audit_unified_enabled_policies;

POLICY_NAME          ENABLED_OPTION  ENTITY_NAME ENTITY_ SUC FAI
-------------------- --------------- ------------ ------- --- ---
ORA_SECURECONFIG     BY USER         ALL USERS    USER    YES YES
ORA_LOGON_FAILURES   BY USER         ALL USERS    USER    NO  YES
SB_COM_ALL_LOGON_EVENTS BY USER      ALL USERS    USER    YES YES
SB_LOC_ALL_LOGON_EVENTS BY USER      ALL USERS    USER    YES YES
```

Test Audit Events

Test a couple of audit events and see if we have data in SYSLOG

- Finally create a few audit events e.g.
  - Login to CDB$ROOT as *SYSDBA*
  - Login to PDB as *SYSTEM*
  - Login to PDB as *SCOTT*

Test Audit Events

Test a couple of audit events and see if we have data in SYSLOG

- Connect as SYSDBA to the root conntainer

```
SQL> CONN / AS SYSDBA
Connected
```

- Check the syslog file */var/log/oracle_common_audit_records.log*

```
...
May 23 14:32:07 db21 journal[332142]: Oracle Unified Audit[332142]:
LENGTH: '198' TYPE:"4" DBID:"2330528275" SESID:"2629419256" CLIENTID:""
ENTRYID:"1" STMTID:"1" DBUSER:"SYS" CURUSER:"SYS" ACTION:"100" RETCODE:"0"
SCHEMA:"" OBJNAME:"" PDB_GUID:"C9D29836D5F7297CE0531501000A3469"
...
```

Test Audit Events

Test a couple of audit events and see if we have data in SYSLOG

- Connect as SYSDBA to the root conntainer

```
SQL> CONN system/manager@db21:1521/pdb1.trivadislabs.com
Connected
```

- Check the syslog file */var/log/oracle_common_audit_records.log*

```
...
May 23 14:32:42 db21 journal[332264]: Oracle Unified Audit[332264]:
LENGTH: '204' TYPE:"4" DBID:"2257451541" SESID:"3460983953" CLIENTID:""
ENTRYID:"1" STMTID:"1" DBUSER:"SYSTEM" CURUSER:"SYSTEM" ACTION:"100"
RETCODE:"0" SCHEMA:"" OBJNAME:""
PDB_GUID:"C9D32D6F1DD56EC3E0531501000A2496"
...
```

Test Audit Events

Test a couple of audit events and see if we have data in SYSLOG

- Connect as SCOTT to the PDB1

```
SQL> CONN SCOTT/tiger@db21:1521/pdb1.trivadislabs.com
Connected
```

- Check the syslog file */var/log/oracle_audit_records.log*

```
...
May 23 14:39:41 db21 journal[333781]: Oracle Unified Audit[333781]:
LENGTH: '201' TYPE:"4" DBID:"2257451541" SESID:"498406760" CLIENTID:""
ENTRYID:"1" STMTID:"1" DBUSER:"SCOTT" CURUSER:"SCOTT" ACTION:"100"
RETCODE:"0" SCHEMA:"" OBJNAME:""
PDB_GUID:"C9D32D6F1DD56EC3E0531501000A2496"
...
```

8

Conclusion

Should you start configure Unified Audit for SYSLOG?

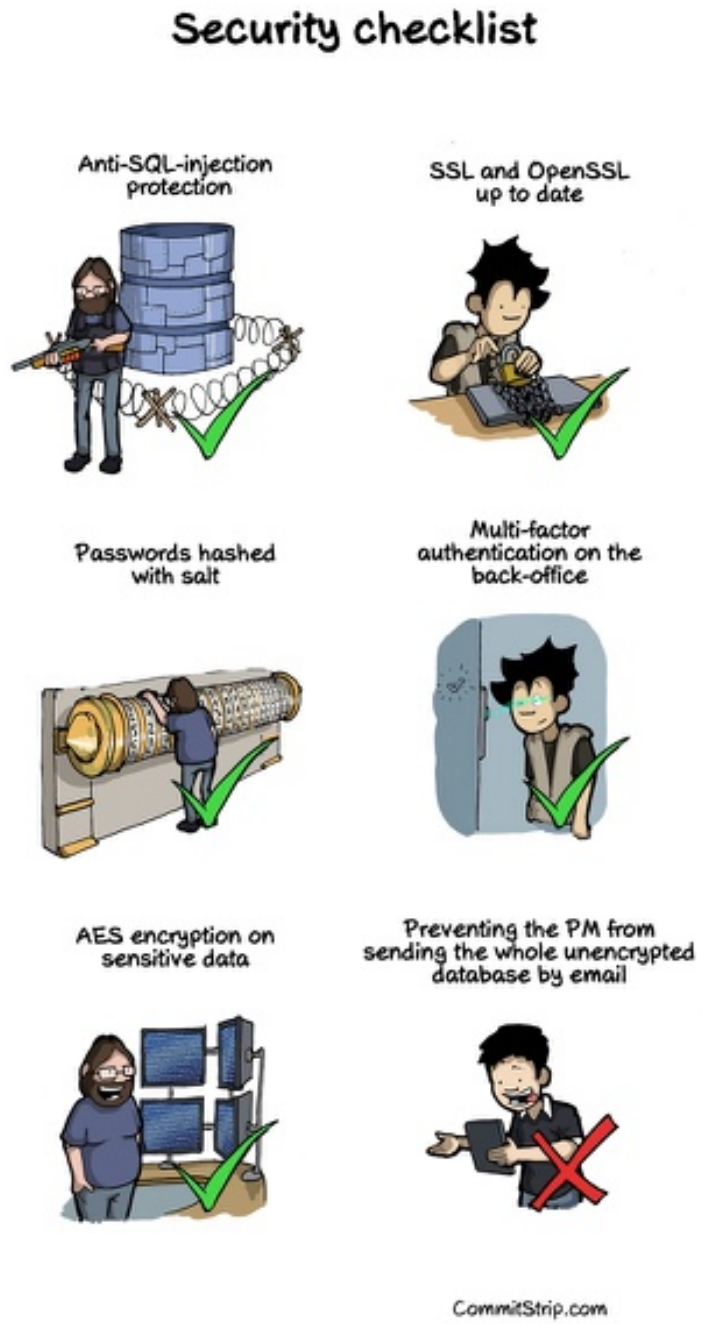
Conclusion

Can SYSLOG used instead of table based audit trails?

- Oracle SYSLOG integration has disappeared and came back with a clearer defined purpose
- Easy possibility to distinct local and common user audit events in PDB e.g. private cloud environment
- Common Audit Events in PDB are **protected** from Customer Housekeeping activity
- SYSLOG allows to forward information to KAFKA, Splunk, SOC etc.

A few major **challenges** remain:

- Redundant Audit Information e.g. SYSLOG and UNIFIED_AUDIT_TRAIL
- Full Audit information including SQL Text, Enterprise User etc only in UNIFIED_AUDIT_TRAIL
- SYSLOG **is limited** compared to UNIFIED_AUDIT_TRAIL



The biggest challenge is still a decent audit concept. What, where, how long, ...

References

Not enough yet? Below a few links to explore the topic in more depth.

- OraDBA How to write Unified Audit Trail Records to SYSLOG
- Oracle® Database SQL Language Reference 21c AUDIT (Unified Auditing)
- Oracle® Database Database Reference 21c UNIFIED_AUDIT_SYSTEMLOG
- Oracle® Database Database Reference 21c UNIFIED_AUDIT_COMMON_SYSTEMLOG
- Oracle Support Document 2623138.1 How to write Unified Audit Trail Records to SYSLOG in 18c
- Oracle Support Document 1582627.1 How To Purge The UNIFIED AUDIT TRAIL
- Oracle Database Unified Audit Best Practice Guidelines

Thank You