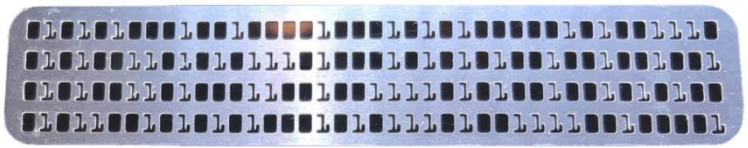
UNIFIED AUDITING

Neil Chandler Oracle ACE Director

Chandler Systems



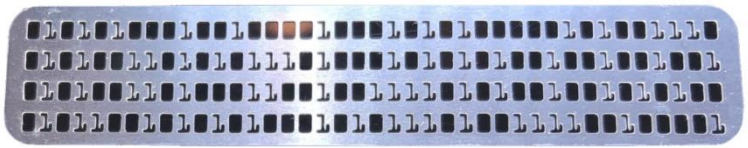
Make IT 2024

Talk relates to 19C and later versions

UNIFIED AUDITING

Neil Chandler Oracle ACE Director

Chandler Systems



MASH PROGRAM

https://mashprogram.wordpress.com

SYM42

https://sym42.org

Talk relates to **19C** and later versions

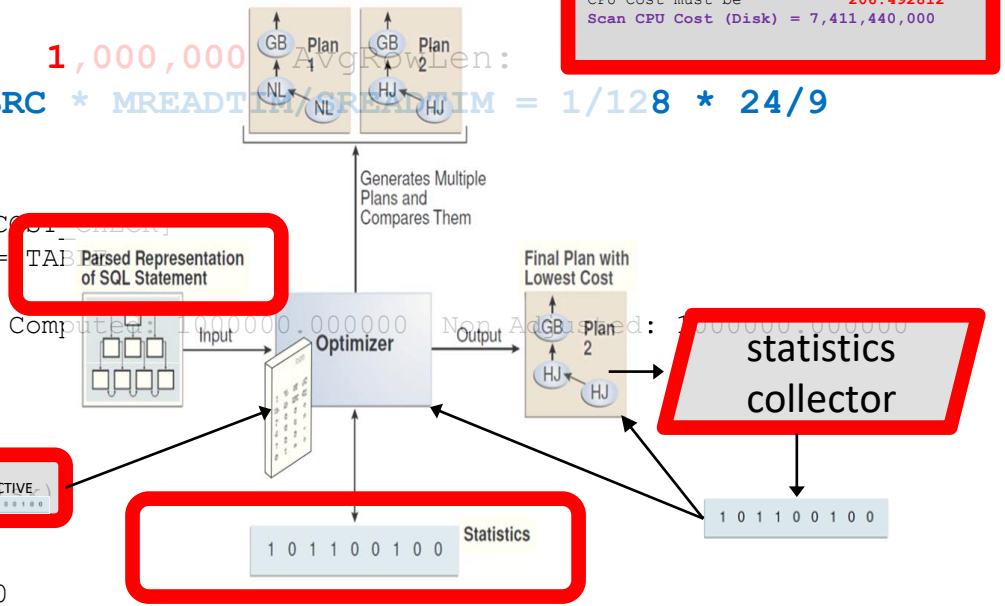
THE COST BASED OPTIMIZER

```
SELECT * FROM cost_check;

Table Stats::
  Table: COST_CHECK  Alias: COST_CHECK
  #Rows: 1000000  SSZ: 0  LGR: 0  #Blks:  1,000,000  AvgRowLen:
_multi block Cost per block=.0206 = 1/MBRC * MREADTIM/SREADTIM = 1/128 * 24/9

[10053] SINGLE TABLE ACCESS PATH
  Single Table Cardinality Estimation for COST_CHECK[COST_CHECK]
  SPD: Return code in qosdDSDirSetup: NOCTX, estType = TABLE
  Table: COST_CHECK  Alias: COST_CHECK
    Card: Original: 1000000.000000  Rounded: 1000000  Computed: 1000000.000000  Non Adjusted: 1000000.000000
  Scan IO  Cost (Disk) =  20631.000000
  Scan CPU Cost (Disk) =  7411440000.000001
  Total Scan IO  Cost  =  20631.000000 (scan (Disk))
                       =  20631.000000
  Total Scan CPU  Cost =  7411440000.000001 (scan (Disk))
                       =  7411440000.000001
  Access Path: TableScan
    Cost:  20902.767101  Resp: 20902.767101  Degree: 0
      Cost_io: 20631.000000  Cost_cpu: 7411440000
      Resp_io: 20631.000000  Resp_cpu: 7411440000
  Best:: AccessPath: TableScan
         Cost: 20902.767101  Degree: 1  Resp: 20902.767101  Card: 1000000.000000  Bytes: 0.000000
```

```
(total)          Cost: 271,041.492812
Scan IO  Cost (Disk) = 270,835
CPU Cost must be          206.492812
Scan CPU Cost (Disk) = 7,411,440,000
```



UNIFIED AUDIT

If you are on Oracle 12.1 or higher, you are using Unified Audit unless you have explicitly disabled it. Which is a very good idea on 12.1.

You could be using it so much better. If you aren't auditing what's going on in your database, how would you know if you have a data breach, who did it and how. Potentially even what was stolen.

You might also be suffering unexpected performance issues in relation to the Unified Audit, and not just in the broken 12.1 implementation of it! It just might not have become visible yet.

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

I'm sorry.

Is it just me, or is audit a really boring subject?

Have I spent too long buried in auditing and security to even suggest it?

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

Traditional audit is… distributed

AUDIT_TRAIL determines location of audit (DB/OS files/XML fileS)

By default does not record the commands which were issued, only that a command was issued!
[use "DB, EXTENDED" or "XML, EXTENDED"]

AUDIT_SYS_OPERATIONS always goes to the O/S

AUDIT_FILE_DEST directory - ON EACH NODE - for OS or XML audit as well as SYSDBA audit

SYS.AUD$ (can move to SYSTEM.AUD$ for Label Security)

SYS.FGAS_LOG$

Gathering them all and proving they were not tampered with… challenging!

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

centralised

better

faster

(12.1 is not. It's broken. MOS: 2063340.1)

1 table contains: FGA, RMAN, Datapump, Oracle Label Security,
Database Vault, Real Application Security, as well as Unified Audit Policies

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

You are already using it*

GET OFF THESE OLD RELEASES!

UPGRADE!

Hacks happen more than you think!

Back to the geekery….

*you ***really*** should not be on 11G, or 10G, 9i, 8i, 8, or 7…

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

You are already using it*

```
SELECT parameter,value FROM v$option WHERE parameter LIKE 'Unified%';
```

12C/18C/19C

```
PARAMETER                                VALUE
Unified Auditing                         FALSE  (which means you are in "mixed mode")
```

21C/23C

```
PARAMETER                                VALUE
Unified Auditing                         TRUE  (traditional audit is deprecated/desupported)
```

Traditional to Unified Audit Syntax Converter - Generate Unified Audit Policies from Current [illegible] Audit Configuration (MOS: **2909718.1**)

- Stop all DB's on that ORACLE_HOME
- `cd $ORACLE_HOME/rdbms/lib`
- `make -f ins_rdbms.mk uniaud_on ioracle`
- restart DB's

*you ***really*** should not be on 11G, or 10G, 9i, 8i, 8, or 7…

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

You are already using it

```
SQL> select * from audit_unified_enabled_policies;
```

POLICY_NAME	ENABLED_OPTION	ENTITY_NAME	ENTITY	SUC	FAI	
ORA_LOGON_FAILURES	BY USER	ALL USERS	USER	NO	YES	
ORA_SECURECONFIG	BY USER	ALL USERS	USER	YES	YES	
ORA$DICTIONARY_SENS_COL_ACCESS	BY USER	ALL USERS	USER	YES	YES	<-23C
ORA_DV_DEFAULT_PROTECTION	BY USER	ALL USERS	USER	YES	YES	<- if DV
ORA_DV_SCHEMA_CHANGES	BY USER	ALL USERS	USER	YES	YES	<- if DV

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

You are already using it

ORA_LOGON_FAILURES

POLICY_NAME	ENABLED_OPTION	ENTITY_NAME	ENTITY_	**SUC**	**FAI**
ORA_LOGON_FAILURES	BY USER	ALL USERS	USER	**NO**	YES (enabled for failed only)

What is audited?

```
select POLICY_NAME,AUDIT_OPTION,AUDIT_OPTION_TYPE,ORACLE_SUPPLIED
from audit_unified_policies where policy_name like 'ORA_LOGON_FAILURES'
```

POLICY_NAME	AUDIT_OPTION	AUDIT_OPTION_TYPE	ORA
ORA_LOGON_FAILURES	LOGON	STANDARD ACTION	YES

Why not all LOGON and LOGOFF actions?

It's only enabled for failed logons…

So what's it for?

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

You are already using it

ORA_LOGON_FAILURES

```
POLICY_NAME                    ENAB
ORA_LOGON_FAILURES             BY U                                    ed for failed only)
```

What is audited?

```
select POLICY_NAME,AUDIT_OPTION,AU
from audit_unified_policies where

POLICY_NAME                      AUI                                    ORA
ORA_LOGON_FAILURES               LOG                                    YES
```

Why not all LOGON and LOGOFF acti

It's only enabled for failed logons…

So what's it for?

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

You are already using it

ORA_SECURECONFIG

POLICY_NAME	ENABLED_OPTION	ENTITY_NAME	ENTITY_	SUC	FAI
ORA_SECURECONFIG	BY USER	ALL USERS	USER	YES	YES

What is audited?

```
select AUDIT_OPTION||'('||DECODE(AUDIT_OPTION_TYPE,'OBJECT ACTION',OBJECT_NAME,AUDIT_OPTION_TYPE)||')'
from audit_unified_policies where policy_name like 'ORA_SECURECONFIG' order by AUDIT_OPTION_TYPE,AUDIT_OPTION;
```

```
EXECUTE(DBMS_RLS)
EXECUTE(ADD_AGENT_CERTIFICATE)

ALTER DATABASE DICTIONARY(STANDARD ACTION)
ALTER DATABASE LINK(STANDARD ACTION)
ALTER PLUGGABLE DATABASE(STANDARD ACTION)
ALTER PROFILE(STANDARD ACTION)
ALTER ROLE(STANDARD ACTION)
ALTER USER(STANDARD ACTION)
CREATE DATABASE LINK(STANDARD ACTION)
CREATE DIRECTORY(STANDARD ACTION)
CREATE PLUGGABLE DATABASE(STANDARD ACTION)
CREATE PROFILE(STANDARD ACTION)
CREATE ROLE(STANDARD ACTION)
DROP DATABASE LINK(STANDARD ACTION)
DROP DIRECTORY(STANDARD ACTION)
DROP PLUGGABLE DATABASE(STANDARD ACTION)
DROP PROFILE(STANDARD ACTION)
DROP ROLE(STANDARD ACTION)
SET ROLE(STANDARD ACTION)

ADMINISTER KEY MANAGEMENT(SYSTEM PRIVILEGE)
ALTER ANY PROCEDURE(SYSTEM PRIVILEGE)
ALTER ANY SQL TRANSLATION PROFILE(SYSTEM PRIVILEGE)
ALTER ANY TABLE(SYSTEM PRIVILEGE)
ALTER DATABASE(SYSTEM PRIVILEGE)
ALTER SYSTEM(SYSTEM PRIVILEGE)
AUDIT SYSTEM(SYSTEM PRIVILEGE)
BECOME USER(SYSTEM PRIVILEGE)
CREATE ANY JOB(SYSTEM PRIVILEGE)
CREATE ANY LIBRARY(SYSTEM PRIVILEGE)
CREATE ANY PROCEDURE(SYSTEM PRIVILEGE)
CREATE ANY SQL TRANSLATION PROFILE(SYSTEM PRIVILEGE)
CREATE ANY TABLE(SYSTEM PRIVILEGE)
CREATE EXTERNAL JOB(SYSTEM PRIVILEGE)
CREATE PUBLIC SYNONYM(SYSTEM PRIVILEGE)
CREATE SQL TRANSLATION PROFILE(SYSTEM PRIVILEGE)
CREATE USER(SYSTEM PRIVILEGE)
DROP ANY PROCEDURE(SYSTEM PRIVILEGE)
DROP ANY SQL TRANSLATION PROFILE(SYSTEM PRIVILEGE)
DROP ANY TABLE(SYSTEM PRIVILEGE)
DROP PUBLIC SYNONYM(SYSTEM PRIVILEGE)
DROP USER(SYSTEM PRIVILEGE)
EXEMPT ACCESS POLICY(SYSTEM PRIVILEGE)
EXEMPT REDACTION POLICY(SYSTEM PRIVILEGE)
GRANT ANY OBJECT PRIVILEGE(SYSTEM PRIVILEGE)
GRANT ANY PRIVILEGE(SYSTEM PRIVILEGE)
GRANT ANY ROLE(SYSTEM PRIVILEGE)
LOGMINING(SYSTEM PRIVILEGE)
PURGE DBA_RECYCLEBIN(SYSTEM PRIVILEGE)
TRANSLATE ANY SQL(SYSTEM PRIVILEGE)
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

You are already using it

ORA_SECURECONFIG - *23C* *has added more 11 options for new functionality, and lost EXECUTE(DBMS_RLS) [part of VPD]*

POLICY_NAME	ENABLED_OPTION	ENTITY_NAME	ENTITY_	**SUC**	**FAI**
ORA_SECURECONFIG	BY USER	ALL USERS	USER	YES	YES

EXECUTE(ADD_AGENT_CERTIFICATE)

ALTER DATABASE DICTIONARY(STANDARD ACTION)
ALTER DATABASE LINK(STANDARD ACTION)
ALTER PLUGGABLE DATABASE(STANDARD ACTION)
ALTER PROFILE(STANDARD ACTION)
ALTER ROLE(STANDARD ACTION)
ALTER USER(STANDARD ACTION)
CREATE DATABASE LINK(STANDARD ACTION)
CREATE DIRECTORY(STANDARD ACTION)
CREATE PLUGGABLE DATABASE(STANDARD ACTION)
CREATE PROFILE(STANDARD ACTION)
CREATE ROLE(STANDARD ACTION)
DROP DATABASE LINK(STANDARD ACTION)
DROP DIRECTORY(STANDARD ACTION)
DROP PLUGGABLE DATABASE(STANDARD ACTION)
DROP PROFILE(STANDARD ACTION)
DROP ROLE(STANDARD ACTION)
SET ROLE(STANDARD ACTION)

ADMINISTER FINE GRAINED AUDIT POLICY(SYSTEM PRIVILEGE)
ADMINISTER KEY MANAGEMENT(SYSTEM PRIVILEGE)
ADMINISTER REDACTION POLICY(SYSTEM PRIVILEGE)
ADMINISTER ROW LEVEL SECURITY POLICY(SYSTEM PRIVILEGE)
ADMINISTER SQL FIREWALL(SYSTEM PRIVILEGE)
ALTER ANY DOMAIN(SYSTEM PRIVILEGE)
ALTER ANY MLE(SYSTEM PRIVILEGE)
ALTER ANY PROCEDURE(SYSTEM PRIVILEGE)
ALTER ANY SQL TRANSLATION PROFILE(SYSTEM PRIVILEGE)
ALTER ANY TABLE(SYSTEM PRIVILEGE)
ALTER DATABASE(SYSTEM PRIVILEGE)
ALTER SYSTEM(SYSTEM PRIVILEGE)
AUDIT SYSTEM(SYSTEM PRIVILEGE)
BECOME USER(SYSTEM PRIVILEGE)
CREATE ANY DOMAIN(SYSTEM PRIVILEGE)
CREATE ANY JOB(SYSTEM PRIVILEGE)
CREATE ANY LIBRARY(SYSTEM PRIVILEGE)
CREATE ANY MLE(SYSTEM PRIVILEGE)
CREATE ANY PROCEDURE(SYSTEM PRIVILEGE)
CREATE ANY SQL TRANSLATION PROFILE(SYSTEM PRIVILEGE)
CREATE ANY TABLE(SYSTEM PRIVILEGE)
CREATE EXTERNAL JOB(SYSTEM PRIVILEGE)
CREATE PUBLIC SYNONYM(SYSTEM PRIVILEGE)
CREATE SQL TRANSLATION PROFILE(SYSTEM PRIVILEGE)
CREATE USER(SYSTEM PRIVILEGE)
DROP ANY DOMAIN(SYSTEM PRIVILEGE)
DROP ANY MLE(SYSTEM PRIVILEGE)
DROP ANY PROCEDURE(SYSTEM PRIVILEGE)
DROP ANY SQL TRANSLATION PROFILE(SYSTEM PRIVILEGE)
DROP ANY TABLE(SYSTEM PRIVILEGE)
DROP PUBLIC SYNONYM(SYSTEM PRIVILEGE)
DROP USER(SYSTEM PRIVILEGE)
EXEMPT ACCESS POLICY(SYSTEM PRIVILEGE)
EXEMPT REDACTION POLICY(SYSTEM PRIVILEGE)
GRANT ANY OBJECT PRIVILEGE(SYSTEM PRIVILEGE)
GRANT ANY PRIVILEGE(SYSTEM PRIVILEGE)
GRANT ANY ROLE(SYSTEM PRIVILEGE)
GRANT ANY SCHEMA PRIVILEGE(SYSTEM PRIVILEGE)
LOGMINING(SYSTEM PRIVILEGE)
PURGE DBA_RECYCLEBIN(SYSTEM PRIVILEGE)
TRANSLATE ANY SQL(SYSTEM PRIVILEGE)

MLE is the GraalVM multi-lingual engine (e.g. Javascript in the DB)

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

You are already using it

ORA$DICTIONARY_SENS_COL_ACCESS

```
POLICY_NAME                    ENABLED_OPTION  ENTITY_NAME  ENTITY_  SUC FAI
ORA$DICTIONARY_SENS_COL_ACCESS BY USER         ALL USERS    USER     YES YES
```

```
SQL> select * from audit_unified_policies where policy_name = 'ORA$DICTIONARY_SENS_COL_ACCESS' order by policy_name;
POLICY_NAME                    AUDIT_COND CONDITION AUDIT AUDIT_OPTION_TYPE  OBJEC OBJECT_NAME                OBJECT_TYPE                COM INH AUD ORA PRO COLUMN_NAM
ORA$DICTIONARY_SENS_COL_ACCESS NONE       NONE      NONE  NONE               NONE  NONE                       NONE                       NO  NO  NO  YES NO
```

```
SQL> select * from audit_unified_enabled_policies where policy_name = 'ORA$DICTIONARY_SENS_COL_ACCESS' order by policy_name;
POLICY_NAME                    ENABLED_OPTION  ENTITY_NAM ENTITY_ SUC FAI
ORA$DICTIONARY_SENS_COL_ACCESS BY USER         ALL USERS  USER    YES YES
```

from: **audit_unified_policy_comments;**
"Audit policy to audit access to dictionary sensitive columns (01 Jun 2022)"

```
SQL> alter audit policy ORA$DICTIONARY_SENS_COL_ACCESS add actions drop user;
Error report -
ORA-46398: Cannot alter or drop the ORA$DICTIONARY_SENS_COL_ACCESS policy.

SQL> noaudit policy ORA$DICTIONARY_SENS_COL_ACCESS;
Noaudit succeeded.
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

You are already using it

ORA$DICTIONARY_SENS_COL_ACCESS

is auditing access to:

```
HISTGRM$
HIST_HEAD$
WRI$_OPTSTAT_HISTHEAD_HISTORY
WRI$_OPTSTAT_HISTGRM_HISTORY
```

(these tables contain bits of your data)

Used by these views:

```
[ALL/CDB/DBA/USER]_NESTED_TABLE_COLS
[ALL/CDB/DBA/USER]_SUBPART_COL_STATISTICS
[ALL/CDB/DBA/USER]_TAB_COLS
[ALL/CDB/DBA/USER]_TAB_COLS_V$
[ALL/CDB/DBA/USER]_TAB_COLUMNS
EXU8ASC
EXU8ASCU
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

PDB v CDB

- Treat each PDB as a separate database
- Do not try to audit the PDB from the CDB (container=all)
  (does not audit local PDB users!)
- CDB can see audit from all of the PDB's
  [CDB_UNIFIED_AUDIT_TRAIL]

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

So where does it go?

Partitioned table: **AUDSYS.AUD$UNIFIED**

Setup: Move to a dedicated tablespace:

```
DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_LOCATION(
    AUDIT_TRAIL_TYPE     => DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
    AUDIT_TRAIL_LOCATION => 'audit_tablespace');
```

Setup: Set a reasonable partition frequency:

```
DBMS_AUDIT_MGMT.ALTER_PARTITION_INTERVAL(
   INTERVAL_NUMBER        => 1,
   INTERVAL_FREQUENCY     => 'DAY');
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

So where does it go?

This is a database table; what if I can't write to it?

There is a "spillover" area on each node at:
`$ORACLE_BASE/audit/instance-id/[PDB-GUID]`
which contains **".bin"** files

Upload the "bin" files to the AUD$UNIFIED table:
`DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES`

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

Performance Problem

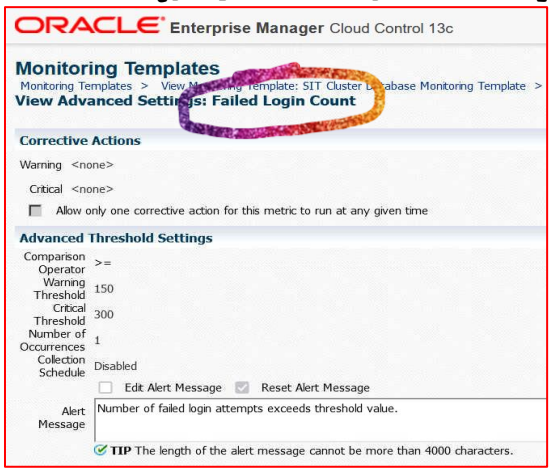
A client was getting a regular alert:

`EM Event: Critical: database-server.domain - Disk Device` **`sda`** `is 98.01% busy.`
(this is on an Exadata, so that's /u01 - an internal ssd)

We used ***pidstat*** to ID "sda“ disk heavy processes & linked to the database [`gv$process, gv$session`]

`sql_id=`**`3wybn83gkuc4k`**

```
.
.
UNION
SELECT COUNT(dbusername) AS failed_count
      ,MIN(event_timestamp) AS first_occur_time
      ,MAX(event_timestamp) AS last_occur_time
  FROM unified_audit_trail
 WHERE action_name = 'LOGON'
   AND return_code <> 0
   AND event_timestamp >= current_timestamp - to_dsinterval('0 0:30:00')
)
```



Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

Performance Problem

"UNIFIED_AUDIT_TRAIL" joins "AUD$UNIFIED" to the contents of the ".bin" directory

Every time we
SELECT … FROM UNIFIED_AUDIT_TRAIL
we read **every** unindexable ".bin" file in there.

Overspill is usually fairly empty directory (audits of startup/shutdown/etc)

NOT on a R/O reporting DB using Active Data Guard

We wrote some special "tidy" scripts to retain the older audit file "elsewhere" once the audit data had been extracted [*better option: copy to Primary and load regularly!*]

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

Performance Advice

You are probably going to read/extract the Audit regularly.
For consistency, probably using "UTC" timezone.

AUD$UNIFIED is partitioned on column EVENT_TIMESTAMP
Try to query using this column for partition elimination.

Avoid predicates on "EVENT_TIMESTAMP_UTC" column - cast event_timestamp

```
SELECT CAST((event_timestamp at TIME zone 'UTC') AS TIMESTAMP)) as ET_UTC
      ,dbusername
      ,action_name
      ,etc…
  FROM UNIFIED_AUDIT_TRAIL
 WHERE ET_UTC > :1
 ORDER BY ET_UTC ASC;
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

LINUX SYSLOG

```
Add to /etc/rsyslog.conf:
local0.* /var/log/oracle.audit.log

systemctl restart rsyslog

in CDB:
> alter system set unified_audit_common_systemlog='LOCAL0.WARNING' scope=spfile;
In (some or all) PDB's:
> alter system set unified_audit_systemlog='LOCAL0.WARNING' scope=spfile;
restart DB

01:06:35 SYS @ UTF8 > show parameter unified
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
unified_audit_common_systemlog       string      LOCAL0.WARNING
unified_audit_sga_queue_size         integer     1048576
unified_audit_systemlog              string      LOCAL0.WARNING
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

WRITING TO SYSLOG

```
SQL> delete from audsys.aud$unified --just an auditable action
ERROR at line 1:
ORA-46385: DML and DDL operations are not allowed on table "AUDSYS"."AUD$UNIFIED".
```

/var/log/oracle.audit.log

```
May  7 01:09:56 ora19-rh79-1 journal: Oracle Unified Audit[25763]: LENGTH: '217' TYPE:"4" DBID:"2032249806"
SESID:"1068262675" CLIENTID:"" ENTRYID:"2" STMTID:"5" DBUSER:"SYS" CURUSER:"SYS" ACTION:"7" RETCODE:"46385"
SCHEMA:"AUDSYS" OBJNAME:"AUD$UNIFIED" PDB_GUID:"D055D5F6F1F5404EE055D6DA69BE4D61"
```

/var/log/messages

```
May  7 01:09:56 ora19-rh79-1 journal: Oracle Unified Audit[25763]: LENGTH: '217' TYPE:"4" DBID:"2032249806"
SESID:"1068262675" CLIENTID:"" ENTRYID:"2" STMTID:"5" DBUSER:"SYS" CURUSER:"SYS" ACTION:"7" RETCODE:"46385"
SCHEMA:"AUDSYS" OBJNAME:"AUD$UNIFIED" PDB_GUID:"D055D5F6F1F5404EE055D6DA69BE4D61"
```

```
SQL> select event_timestamp,dbusername,action_name,object_name,return_code from
unified_audit_trail where return_code = 46385 order by event_timestamp

EVENT_TIMESTAMP                 DBUSER ACTION_NAME             OBJECT_NAME                    RETURN_CODE
------------------------------- ------ ----------------------- ------------------------------ -----------
07-MAY-23 01.09.56.372459 SYS    DELETE                  AUD$UNIFIED                         46385
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

WRITING TO SYSLOG

- /var/log/messages gets big. Really big, really quickly. /var fills up…

- Unix admins are not amused
- Splunk admins are even less amused

- There is a max line length (usually 1k or 2K)
- Not all fields are recorded

- Has the advantage of locking the audit away from the DBA's - as long as they don't have root access... 👀

- Only write into the SYSLOG if you have a compelling reason
  Argue against it!

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT POLICIES

CIS Audit Recommendations

because they are quite sensible

Switch on the build-in policies

```
audit policy ORA_CIS_RECOMMENDATIONS;
audit policy ORA_ACCOUNT_MGMT;
audit policy ORA_DATABASE_PARAMETER;
audit policy ORA_SECURECONFIG;
```
(it's on anyway)

Ignore unless you using DV / RAS

```
ORA_DV_AUDPOL
ORA_DV_AUDPOL2
ORA_RAS_POLICY_MGMT
ORA_RAS_SESSION_MGMT
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT POLICIES

CIS Audit Recommendations

You need to Audit all DB connections for compliance too

```
noaudit policy ORA_LOGON_FAILURES;

create audit policy NEIL_LOGON_LOGOFF actions logon,logoff;
audit policy neil_logon_logoff;
```

In 23c, Oracle have noticed this omission:

```
audit policy ORA_LOGON_LOGOFF;
```

WARNING! If you do not have connection pooling, or it is not correctly configured with static connection levels (they rarely are), this can create a **lot** of audit records

(however, this can provide evidence of badly configured connection pools)

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT POLICIES

...and audit what else?

- You can audit by Privileges, Action, Objects and Roles
- By Role will audit all privileges within the role
- A Policy can be conditional, giving a really narrow focus
- Policies can be **enabled** conditionally (all users or specified user).
- Can add context attributes to enhance audit records too

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT POLICIES

…and audit what else?

What are the high privilege users up to?

Unless a user has been explicitly granted access to a table, why are they accessing it?

```
CREATE AUDIT POLICY neil_any_dml
PRIVILEGES
select any table,
  read any table,
update any table,
delete any table,
insert any table;

AUDIT POLICY neil_any_dml;
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT POLICIES

Lets see "neil_**any**_dml" policy in action

```
SELECT e.empno,d.deptno  FROM neil.emp e INNER JOIN neil.dept d ON (e.deptno = d.deptno);
```

Lets check what happened in the audit

```
SELECT dbusername ,unified_audit_policies ,object_name ,system_privilege_used ,statement_id,sql_text FROM unified_audit_trail;
```

DBUSERNAME	UNIFIED AUDIT POLICIES	OBJECT	SYSTEM PRIVILEGE USE	STATEMENT ID	SQL TEXT
DODGY_DBA	NEIL_ANY_DML	EMP	SELECT ANY TABLE	54	SELECT e.empno,d.deptno FROM neil.emp e INNER JOIN neil.dept d……
DODGY_DBA		DEPT	SELECT ANY TABLE	54	SELECT e.empno,d.deptno FROM neil.emp e INNER JOIN neil.dept d……

is has audited BOTH tables separately.

What if the SQL is malformed?

DBUSERNAME	UNIFIED AUDIT POLICIES	OBJECT	SYSTEM PRIVILEGE USE	STATEMENT ID	SQL TEXT
DODGY_DBA	NEIL_ANY_DML			55	SELECT e.empno,**d.deptnoX** FROM neil.emp e INNER JOIN neil.dept d ON

The audit takes place at PARSE time
The lack of object_name and system_privilege_used in the audit shows objects were not accessed
BUT the user *does* have access to the objects in the SQL

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

HOUSEKEEPING

Create a DBMS_SCHEDULER job to tidy up according to your corporate standards

```
BEGIN
dbms_scheduler.create_job(q'[AUDIT_HOUSEKEEPING]',
job_type=>'PLSQL_BLOCK', job_action=>
q'[
BEGIN
     dbms_audit_mgmt.set_last_archive_timestamp(audit_trail_type    => dbms_audit_mgmt.audit_trail_unified
                                               ,last_archive_time   => trunc(systimestamp - INTERVAL '3' MONTH));

     dbms_audit_mgmt.clean_audit_trail(audit_trail_type        => dbms_audit_mgmt.audit_trail_unified
                                      ,use_last_arch_timestamp => true);
END;
]'
,number_of_arguments=>0
,start_date=>trunc(systimestamp + interval '1' day)
,repeat_interval=> q'[FREQ = DAILY; INTERVAL = 1]'
,end_date=>NULL
,job_class=>q'[SCHED$_LOG_ON_ERRORS_CLASS]'
,enabled=>FALSE
,auto_drop=>FALSE
,comments=> q'[Daily clean-up Unified Audit older than 3 months]'
);
COMMIT;
dbms_scheduler.enable(q'[AUDIT_HOUSEKEEPING]');
END;
```

- delete from aud$unified
  where event_timestamp < :1
  and (dbid =:2 or dbid=0)
- removes empty partitions
- removes overspill files

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

How vulnerable is UNIFIED_AUDIT_TRAIL

Lets add some data to an application table…

```
SQL> connect dodgy_dba/oracle

SQL> insert into neil.emp values
    (9999,'Chandler','DBA',7839,to_date('19/09/1988','DD/MM/YYYY'),99999,null,10);

SQL> commit;

SQL> SELECT event_timestamp,dbusername,action_name,object_name,system_privilege_used
       FROM unified_audit_trail;

SESSIONID  EVENT_TIMESTAMP      DBUSERNAME ACTION_NAME OBJECT_NAME  SYSTEM_PRIVILEGE
3367125493 2023-05-08 00:59:07 DODGY_DBA  INSERT      EMP          INSERT ANY TABLE
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

How vulnerable is UNIFIED_AUDIT_TRAIL

SESSIONID	EVENT_TIMESTAMP	DBUSERNAME	ACTION_NAME	OBJECT_NAME	SYSTEM_PRIVILEGE
3367125493	2023-05-08 00:59:07	DODGY_DBA	INSERT	EMP	INSERT ANY TABLE

```
$ sqlplus / as sysdba

SQL> alter session set container=utf8pdb1;
Session altered.

SQL> delete from audsys.aud$unified where sessionid=3367125493;
delete from audsys.aud$unified where sessionid=3367125493
                   *
ERROR at line 1:
ORA-46385: DML and DDL operations are not allowed on table "AUDSYS"."AUD$UNIFIED".
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

rico2 (or bbed)

Get some info about the data location:

DBA_OBJECTS

```
DATA_OBJECT_ID  OBJECT_ID OBJECT_NAME                    OBJECT_TYPE
         75729      74952 AUD$UNIFIED                    TABLE PARTITION
                    18567 AUD$UNIFIED                    TABLE
```

DBA_TAB_PARTITIONS

```
TABLESPACE_NAME                PARTITION_NAME
SYSAUX                         SYS_P776
```

DBA_DATA_FILES

```
FILE_NAME                                                    FILE_ID
/u01/oradata/UTF8/UTF8PDB1/sysaux01.dbf                           10
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

rico2

DATA_OBJECT_ID
75729

```
$ cat listfile.rico2
10 /u01/oradata/UTF8/UTF8PDB1/sysaux01.dbf

$ python rico2.py listfile.rico2

rico2 > find -f 10 -o 75729

Found in block: 37728 block type: FIRST LEVEL BITMAP BLOCK
Found in block: 37729 block type: SECOND LEVEL BITMAP BLOCK
Found in block: 37730 block type: PAGETABLE SEGMENT HEADER
Found in block: 37731 block type: DATA
Found in block: 37732 block type: DATA
Found in block: 37733 block type: DATA
Found in block: 37734 block type: DATA
Found in block: 37735 block type: DATA

rico2 > set dba 10,37731
        DBA             0x2809363 (41980771 10,37731)

rico2 > select col2=n:3367125493
Found at *kdbr[2]
Found at *kdbr[3]
Found at *kdbr[4]
```

audsys.aud$unified

Name	Null?	Type
INST_ID		NUMBER
AUDIT_TYPE		NUMBER
SESSIONID		**NUMBER**
PROXY_SESSIONID		NUMBER
OS_USER		VARCHAR2(128)
HOST_NAME		VARCHAR2(128)
TERMINAL		VARCHAR2(30)
INSTANCE_ID		NUMBER
DBID		NUMBER
AUTHENTICATION_TYPE		VARCHAR2(1024)
USERID		VARCHAR2(128)
PROXY_USERID		VARCHAR2(128)
EXTERNAL_USERID		VARCHAR2(1024)
GLOBAL_USERID		VARCHAR2(32)
CLIENT_PROGRAM_NAME		VARCHAR2(48)
DBLINK_INFO		VARCHAR2(4000)
XS_USER_NAME		VARCHAR2(128)
XS_SESSIONID		RAW(33)
ENTRY_ID	NOT NULL	NUMBER
STATEMENT_ID	NOT NULL	NUMBER
EVENT_TIMESTAMP	NOT NULL	TIMESTAMP(6)
ACTION	NOT NULL	NUMBER

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

rico2

```
rico2 > p *kdbr[3]

rowdata[603]                            @5688   0x2c
-------------
flag@5688:      0x2c
lock@5689:      0x0
cols@5690:      48


col    0[0]     @5691:  *NULL*
col    1[2]     @5692:  c105
col    2[6]     @5695:  c522440d375e
col    3[1]     @5702:  80
col    4[6]     @5704:  6f7261636c65
col    5[12]    @5711:  6f726131392d726837392d31
col    6[5]     @5724:  7074732f30
col    7[2]     @5730:  c102
col    8[6]     @5733:  c51521196307
col    9[115]   @5740:  28545950453d28444154414241534529293b28434c49454e54204144445245 [snip]
col   10[9]     @5856:  444f4447595f444241
col   11[0]     @5866:  *NULL*
...
col   33[0]     @6204:  *NULL*
col   34[0]     @6205:  *NULL*
col   35[0]     @6206:  *NULL*
col   36[16]    @6207:  494e5345525420414e59205441424c45
col   37[1]     @6224:  80
...
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

rico2

rico2 > x /rnnnnccccnnccccccccxnnntnncxnccexxccccccnnccccccccccccnccccccc

```
rowdata[603]                            @5688   0x2c
-------------
flag@5688:      0x2c
lock@5689:      0x0
cols@5690:      48


col    0[0]     @5691:  *NULL*
col    1[2]     @5692:  c105                                    4
col    2[6]     @5695:  c522440d375e                            3367125493
col    3[1]     @5702:  80                                      0
col    4[6]     @5704:  6f7261636c65                            oracle
col    5[12]    @5711:  6f726131392d726837392d31                ora19-rh79-1
col    6[5]     @5724:  7074732f30                              pts/0
col    7[2]     @5730:  c102                                    1
col    8[6]     @5733:  c51521196307                            2032249806
col    9[115]   @5740:  28545950453d284441544142415345293b2... (TYPE=(DATABASE));(CLIENT ...
col   10[9]     @5856:  444f4447595f444241                      DODGY_DBA
col   11[0]     @5866:  *NULL*
.
col   33[0]     @6204:  *NULL*
col   34[0]     @6205:  *NULL*
col   35[0]     @6206:  *NULL*
col   36[16]    @6207:  494e5345525420414e59205441424c45        INSERT ANY TABLE
col   37[1]     @6224:  80                                      0
.
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

rico2

```
col   10[9]     @5856:  444f4447595f444241                    DODGY_DBA
```

rico2 > set offset 5857

rico2 > d

```
 File: /u01/oradata/UTF8/UTF8PDB1/sysaux01.dbf(10)
 Block: 37731 Offsets: 5857 to 6369            Dba: 0x2809363
------------------------------------------------------------------
444f4447 595f4442 41ffffff 2073716c | DODGY_DBA... sql
706c7573 406f7261 31392d72 6837392d | plus@ora19-rh79-
```

rico2 > modify -h 445550415F44555041

```
You want to modify block: 37731 at offset: 5857
New value: 445550415f44555041
Are you sure? (Y/N)  y
Block data changed. To save changes set edit mode and type: save
```

rico2 > d

```
 File: /u01/oradata/UTF8/UTF8PDB1/sysaux01.dbf(10)
 Block: 37731 Offsets: 5857 to 6369            Dba: 0x2809363
------------------------------------------------------------------
44555041 5f445550 41ffffff 2073716c | DUPA_DUPA... sql
706c7573 406f7261 31392d72 6837392d | plus@ora19-rh79-
```

rico2 > sum apply

```
checksum int = 49605
checksum hex = 0xc1c5
Block data changed. To save changes set edit mode and type: save
```

rico2 > set mode edit

rico2 > save

```
Current block data successfully saved to disk. To revert changes, type:
dupa
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

rico2

```
sqlplus / as sysdba

SQL > alter system flush buffer_cache;

System altered.

SQL > select sessionid,event_timestamp,userid,action from AUDSYS.AUD$UNIFIED where sessionid=3367125493;

 SESSIONID EVENT_TIMESTAMP     USERID         ACTION
---------- ------------------- ---------- ----------
3367125493 2023-05-08 01:33:02 DODGY_DBA         100
3367125493 2023-05-08 01:33:04 DUPA_DUPA           2
3367125493 2023-05-08 01:34:13 DODGY_DBA         101
```

but what if we want to hack a *lot* of data?

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

How vulnerable is UNIFIED_AUDIT_TRAIL

```
SESSIONID  EVENT_TIMESTAMP      DBUSERNAME ACTION_NAME OBJECT_NAME  SYSTEM_PRIVILEGE
3367125493 2023-05-08 00:59:07 DODGY_DBA  INSERT      EMP          INSERT ANY TABLE


$ sqlplus / as sysdba

SQL> alter session set container=utf8pdb1;
Session altered.

SQL> delete from audsys.aud$unified where sessionid=3367125493;
delete from audsys.aud$unified where sessionid=3367125493
                   *
ERROR at line 1:
ORA-46385: DML and DDL operations are not allowed on table "AUDSYS"."AUD$UNIFIED".

SQL> [do some naughty things]
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

How vulnerable is UNIFIED_AUDIT_TRAIL

```
SQL> [do some naughty things]
$ sqlplus / as sysdba
SQL> alter session set container=utf8pdb1;
Session altered.

SQL> select sessionid,event_timestamp,userid,action from AUDSYS.AUD$UNIFIED where sessionid=3367125493;
 SESSIONID EVENT_TIMESTAMP                    USERID                    ACTION
---------- ---------------------------------- -------------------- ----------
3367125493 08-MAY-23 01.33.02.440149 AM   DODGY_DBA                        100
3367125493 08-MAY-23 01.33.04.342422 AM   DUPA_DUPA                          2
3367125493 08-MAY-23 01.34.13.787265 AM   DODGY_DBA                        101

SQL> delete from audsys.aud$unified where sessionid=3367125493;
3 rows deleted.

SQL> commit;
Commit complete.

SQL> select sessionid,event_timestamp,userid,action from AUDSYS.AUD$UNIFIED where sessionid=3367125493;
no rows selected
```


Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

Audit is not just about compliance and security

Use it day-to-day

- troubleshooting connection problems
  (did you get to the DB)
- explaining connection pools and storms
- proving Development *did* change something
  (especially those CI/CD Developers!)

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

UNIFIED AUDIT

WHY

are we auditing?

Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

THANK YOU. . .

BLOG: http://chandlerdba.com
Twitter: **@chandlerDBA**
E: neil@chandler.uk.com