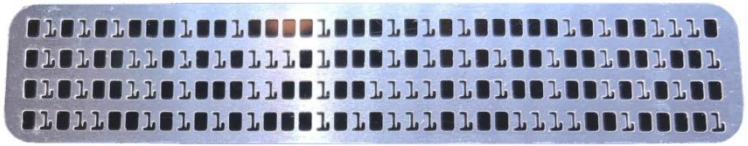
FUNDAMENTAL ORACLE SECURITY

what many of you are not doing

Neil Chandler

ORACLE ACE Director

Chandler Systems



Make IT 2022

JUN 2022

Talk relates to 19C and later versions

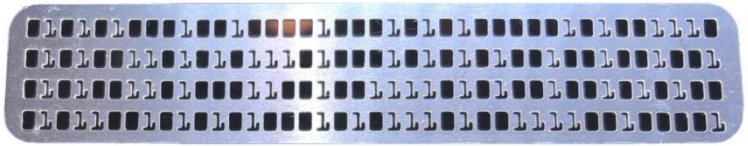
Trouble is, just because things are obvious doesn't mean they're true. *Granny Weatherwax*

FUNDAMENTAL ORACLE SECURITY

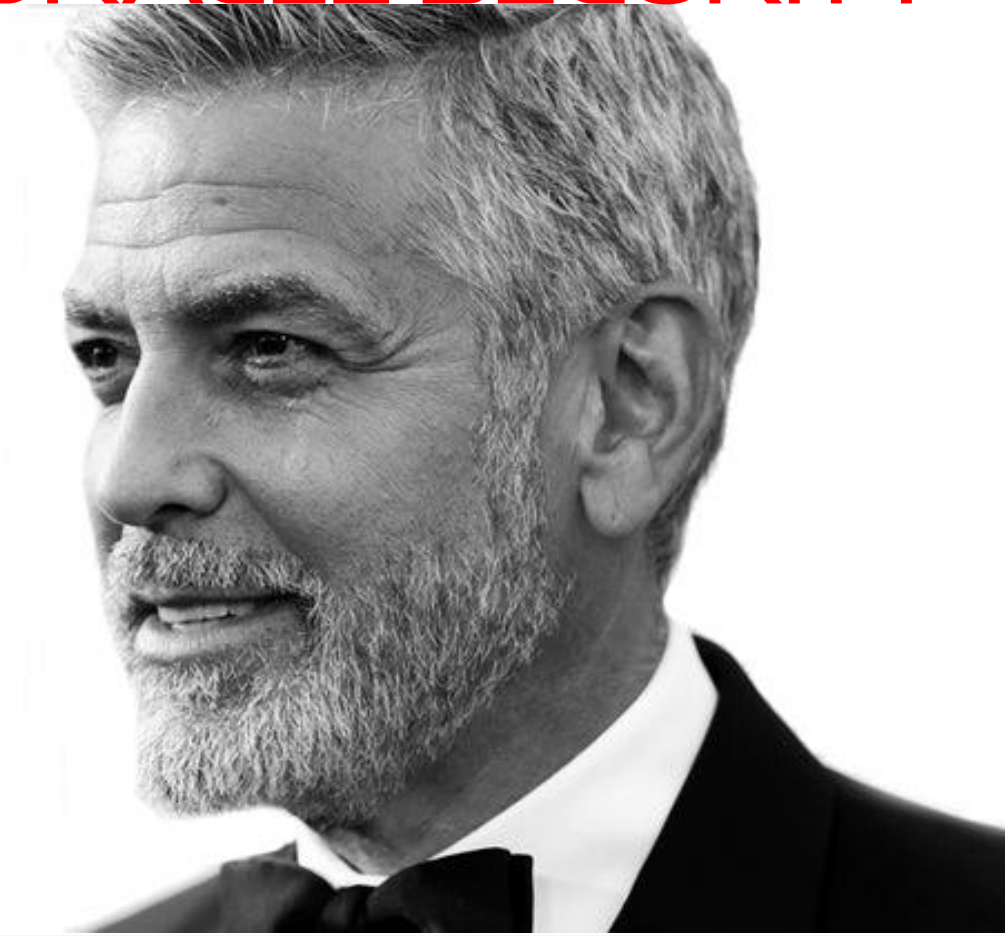
Neil Chandler

Chandler Systems

ORACLE ACE Director



Make IT JUN 2022 2022



Talk relates to 19C and later versions

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

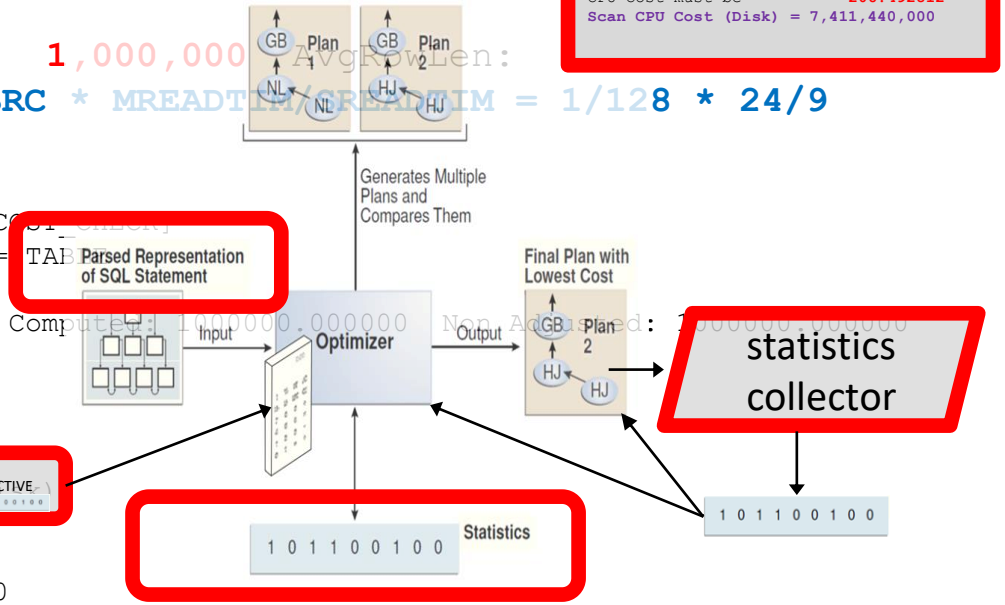
THE COST BASED OPTIMIZER

```
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
```

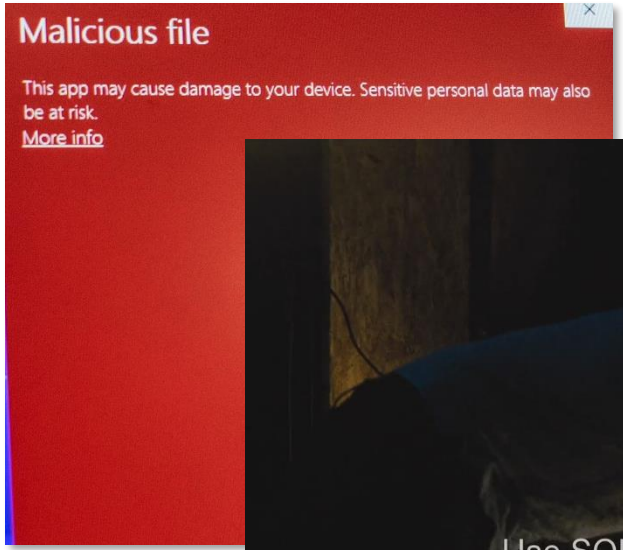
```
(total)            Cost: 271,041.492812
Scan IO  Cost (Disk) = 270,835
CPU Cost must be              206.492812
Scan CPU Cost (Disk) = 7,411,440,000
```



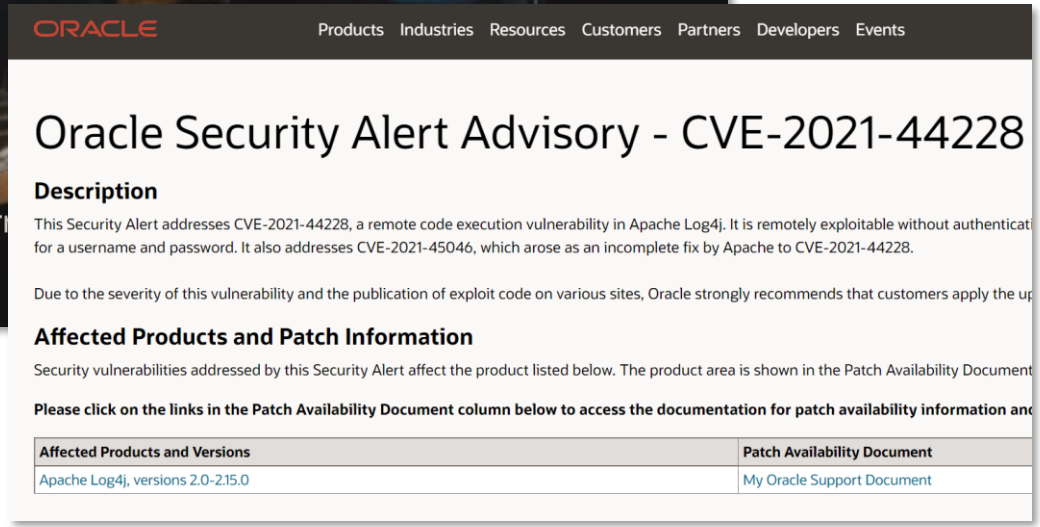
What the f*ck is going on?

Who uses passwords?

SECURITY IS A HOT TOPIC



Viruses and Malware



ORACLE Products Industries Resources Customers Partners Developers Events

Oracle Security Alert Advisory - CVE-2021-44228

Description

This Security Alert addresses CVE-2021-44228, a remote code execution vulnerability in Apache Log4j. It is remotely exploitable without authenticati for a username and password. It also addresses CVE-2021-45046, which arose as an incomplete fix by Apache to CVE-2021-44228.

Due to the severity of this vulnerability and the publication of exploit code on various sites, Oracle strongly recommends that customers apply the up

Affected Products and Patch Information

Security vulnerabilities addressed by this Security Alert affect the product listed below. The product area is shown in the Patch Availability Document

Please click on the links in the Patch Availability Document column below to access the documentation for patch availability information and

Affected Products and Versions	Patch Availability Document
Apache Log4j, versions 2.0-2.15.0	My Oracle Support Document

what many of you are not doing

passwords

Or4cl3

Or4cl3!!!

Admin Accounts With No Passwords at the Heart of Recent MongoDB Ransom Attacks

By Catalin Cimpanu September 11, 2017 06:56 AM 2

The recent wave of ransom attacks on MongoDB databases happened because database owners forgot to set passwords on their administrator accounts, according to Davi Ottenheimer, Senior Director of Product Security at MongoDB, Inc.

Is your SYS password really strong enough?

Is complexity enforced?

DBA_PROFILES

```
SELECT
    profile
  , resource_name
  , resource_type
  , limit
FROM
    dba_profiles
WHERE
    resource_type = 'PASSWORD'
ORDER BY
    profile
  , resource_type
  , resource_name;
```

```
PROFILE          RESOURCE_NAME             LIMIT
---------------- ------------------------- ---------
DEFAULT          FAILED_LOGIN_ATTEMPTS     10
DEFAULT          INACTIVE_ACCOUNT_TIME     UNLIMITED
DEFAULT          PASSWORD_GRACE_TIME       7
DEFAULT          PASSWORD_LIFE_TIME        180
DEFAULT          PASSWORD_LOCK_TIME        1
DEFAULT          PASSWORD_REUSE_MAX        UNLIMITED
DEFAULT          PASSWORD_REUSE_TIME       UNLIMITED
DEFAULT          PASSWORD_ROLLOVER_TIME    -1
DEFAULT          PASSWORD_VERIFY_FUNCTION  NULL

ORA_STIG_PROFILE FAILED_LOGIN_ATTEMPTS     3
ORA_STIG_PROFILE INACTIVE_ACCOUNT_TIME     35
ORA_STIG_PROFILE PASSWORD_GRACE_TIME       5
ORA_STIG_PROFILE PASSWORD_LIFE_TIME        60
ORA_STIG_PROFILE PASSWORD_LOCK_TIME        UNLIMITED
ORA_STIG_PROFILE PASSWORD_REUSE_MAX        10
ORA_STIG_PROFILE PASSWORD_REUSE_TIME       365
ORA_STIG_PROFILE PASSWORD_ROLLOVER_TIME    DEFAULT
ORA_STIG_PROFILE PASSWORD_VERIFY_FUNCTION  ORA12C_STIG_VERIFY_FUNCTION
```

https://www.cisecurity.org

```
PROFILE          RESOURCE_NAME             LIMIT       CIS Recommendations
---------------- ------------------------- ---------   ---------------------------------
DEFAULT          FAILED_LOGIN_ATTEMPTS     10          FAIL <= 5
DEFAULT          INACTIVE_ACCOUNT_TIME     UNLIMITED   FAIL <= 120 days (lock if unused)
DEFAULT          PASSWORD_GRACE_TIME       7           FAIL <=   5 days
DEFAULT          PASSWORD_LIFE_TIME        180         FAIL <=  90 days (enforced change)
DEFAULT          PASSWORD_LOCK_TIME        1           PASS >=   1 day  (duration locked)
DEFAULT          PASSWORD_REUSE_MAX        UNLIMITED   FAIL >=  20      (pwd history #)
DEFAULT          PASSWORD_REUSE_TIME       UNLIMITED   FAIL >= 365 days (pwd history len)
DEFAULT          PASSWORD_ROLLOVER_TIME    -1          n/a
DEFAULT          PASSWORD_VERIFY_FUNCTION  NULL        FAIL >= Password Complexity
```

Create your own profile for you accounts – and leave ORACLE_MAINTAINED users to use a modified DEFAULT

```
CREATE PROFILE cis_compliant_profile LIMIT
        FAILED_LOGIN_ATTEMPTS      5
        INACTIVE_ACCOUNT_TIME      120
        PASSWORD_GRACE_TIME        5
        PASSWORD_LIFE_TIME         90
        PASSWORD_LOCK_TIME         1
        PASSWORD_REUSE_MAX         20
        PASSWORD_REUSE_TIME        365
        PASSWORD_ROLLOVER_TIME     0
        PASSWORD_VERIFY_FUNCTION   [what to use?];

ALTER USER myuser PROFILE cis_compliant_profile ;
```

WARNING!
This may cause non-compliant accounts to become LOCKED (later that day)

```
PROFILE          RESOURCE_NAME             LIMIT
---------------- ------------------------- ---------
DEFAULT          PASSWORD_VERIFY_FUNCTION  NULL
```

TIP: Make it the same as your AD validation requirement

Built-In Verify Functions	Len	Upper	Lower	Numeric	Special	Differ By
ORA12C_STIG_VERIFY_FUNCTION	15	1	1	1	1	8
ORA12C_STRONG_VERIFY_FUNCTION	9	2	2	2	2	4
ORA12C_VERIFY_FUNCTION	8	1 *or*	1	1	0	3
VERIFY_FUNCTION_11G	1	0	1	1	0	3

```
ALTER PROFILE default LIMIT PASSWORD_VERIFY_FUNCTION ORA12C_VERIFY_FUNCTION;
```

Probably need to write your own function; base it around code in:

$ORACLE_HOME/rdbms/admin/catpvf.sql

PASSWORD COMPLEXITY FUNCTION

```
CREATE OR REPLACE FUNCTION custom_verify (
    username     VARCHAR2
  , password     VARCHAR2
  , old_password VARCHAR2
) RETURN BOOLEAN IS
    differ  INTEGER;
BEGIN
    IF NOT ora_complexity_check(
                                 password
                               , chars => 15
                               , uppercase => 1
                               , lowercase => 1
                               , digit => 1
                               , special => 1
           ) THEN
        RETURN ( false );
    END IF;

    -- Check if the password differs from the previous password by n characters
    IF old_password IS NOT NULL THEN
        differ := ora_string_distance(old_password, password);
        IF differ < 8 THEN
            raise_application_error(-20000, 'password is too similar to previous password');
        END IF;

    END IF;
RETURN ( true );
END;
/
```

DEFAULTS

DBA_USERS_WITH_DEFPWD

```
SQL > SELECT * FROM dba_users_with_defpwd;

USERNAME                          PRODUCT
----------------------------- ------------------------------
SYS
SYSTEM
CTXSYS

SQL > conn CTXSYS/CTXSYS
ERROR:
ORA-28000: The account is locked.
```

```
SQL > alter user system identified by manager container=all;

User altered.

SQL > conn system/manager
Connected.

SQL > select * from dba_users_with_defpwd;

USERNAME                       PRODUCT
------------------------------ ------------------------------
SYS
CTXSYS
```

Unused and Historic Accounts are a Security Issue

```
SELECT username, oracle_maintained, account_status,
       created,  nvl(last_login,'never') last_login
  FROM dba_users ORDER BY 2, 1;
```

USERNAME	O	ACCOUNT_STATUS	CREATED	LAST_LOGIN	
APP_SCHEMA	N	OPEN	2019-11-16	2022-01-01	<- schema owner
APP_USER	N	OPEN	2019-11-16	2022-01-23	<- application user
CHRIS	N	OPEN	2020-11-16	**2021-11-16**	<- should this be open?
NEIL	N	OPEN	2021-11-15	2022-01-23	<- DBA
SCOTT	N	LOCKED	2019-11-15	never	<- should this exist?
SHANE	N	OPEN	2019-11-17	never	<- unused! Delete!
AUDSYS	Y	LOCKED	2019-04-17	never	
CTXSYS	Y	LOCKED	2019-04-17	never	
.					
.					
SYSRAC	Y	LOCKED	2019-04-17	never	
SYSTEM	Y	OPEN	2019-04-17	2021-11-16	
WMSYS	Y	LOCKED	2019-04-17	never	
XDB	Y	LOCKED	2019-04-17	never	
XS$NULL	Y	EXPIRED & LOCKED	2019-04-17	never	

Native Integration in 19C
via Centrally Managed Users (CMU)

Use AD with the previous features, not instead of them

```
sqlplus system/manager <<EOF
SELECT info FROM table;
EOF
```

create a wallet associated with a TNSNAMES.ORA entry:

```
sqlplus /@MYSERVICE <<EOF
SELECT info FROM table;
EOF
```


Wallets are easy to learn about. Use them.

PROXY ACCOUNTS

Don't have ***known*** passwords for high-level or “general” accounts

ALTER USER **app_schema** GRANT CONNECT THROUGH **dba_neil**;

SQL> connect **dba_neil**[**app_schema**]/***dba_neil's_password***
SQL> show user
USER is “APP_SCHEMA”

Wallets are easy to learn about. Use them.

Now you have complex passwords…

https://keepass.info

Mentions are not recommendations – do your own research!

But What Can Users Do?

Permissions Check

```
SELECT * FROM dba_role_privs
WHERE granted_role = 'DBA'
ORDER BY grantee;
```

GRANTEE	GRANTED_ROLE	ADM	DEL	DEF	COM	INH
APP_SCHEMA	DBA	NO	NO	YES	NO	NO
CHRIS	DBA	NO	NO	YES	NO	NO
GRACE	DBA	NO	NO	YES	NO	NO
NEIL	DBA	NO	NO	YES	NO	NO
SHANE	DBA	NO	NO	YES	NO	NO
SYS	DBA	YES	NO	YES	YES	YES
SYSTEM	DBA	NO	NO	YES	YES	YES

Permissions Check

```
SELECT * FROM dba_role_privs
WHERE granted_role = 'IMP_FULL_DATABASE'
ORDER BY grantee
```

GRANTEE	GRANTED_ROLE	ADM	DEL	DEF	COM	INH
DATAPUMP_IMP_FULL_DATABASE	IMP_FULL_DATABASE	NO	NO	YES	YES	YES
DBA	IMP_FULL_DATABASE	NO	NO	YES	YES	YES
SCOTT	**IMP_FULL_DATABASE**	**NO**	**NO**	**YES**	**NO**	**NO**
SYS	IMP_FULL_DATABASE	YES	NO	YES	YES	YES

Permissions Check

```
SELECT * FROM dba_sys_privs
 WHERE privilege LIKE '%ANY%'
 ORDER BY grantee,privilege
```

GRANTEE	PRIVILEGE	ADM	COM	INH
APP_USER	**SELECT ANY TABLE**	**NO**	**NO**	**NO**
AQ_ADMINISTRATOR_ROLE	DEQUEUE ANY QUEUE	YES	YES	YES
.				
CTXSYS	INHERIT ANY PRIVILEGES	NO	YES	YES
DATAPUMP_IMP_FULL_DATABASE	AUDIT ANY	NO	YES	YES
DATAPUMP_IMP_FULL_DATABASE	DELETE ANY TABLE	NO	YES	YES
MDSYS	INHERIT ANY PRIVILEGES	NO	YES	YES
OEM_MONITOR	ANALYZE ANY DICTIONARY	NO	YES	YES
OEM_MONITOR	MANAGE ANY QUEUE	NO	YES	YES
OEM_MONITOR	SELECT ANY DICTIONARY	NO	YES	YES

Permissions Check

```
SELECT owner, table_name, grantee, privilege FROM dba_tab_privs
 WHERE privilege = 'EXECUTE'
   AND grantee   = 'PUBLIC'
   AND type      in ('PROCEDURE','PACKAGE','TYPE','FUNCTION')
 ORDER BY table_name,grantee,privilege

OWNER      TABLE_NAME                     GRANTEE    PRIVILEG TYPE
---------- ------------------------------ ---------- -------- ---------
.
.
SYS        DBMS_LDAP                      PUBLIC     EXECUTE  PACKAGE
SYS        HTTPURITYPE                    PUBLIC     EXECUTE  TYPE
SYS        UTL_HTTP                       PUBLIC     EXECUTE  PACKAGE
SYS        UTL_INADDR                     PUBLIC     EXECUTE  PACKAGE
SYS        UTL_SMTP                       PUBLIC     EXECUTE  PACKAGE
SYS        UTL_TCP                        PUBLIC     EXECUTE  PACKAGE
.
.
```

19.13 has 2,523 permissions granted to public

what many of you are not doing

Revoke from PUBLIC and grant explicitly to accounts which need the functionality

Network Security

DBMS_LDAP
UTL_INADDR
UTL_TCP
UTL_MAIL
UTL_SMTP
UTL_DBWS
UTL_ORAMTS
UTL_HTTP
HTTPURITYPE

Used to leak/spam information outside of the system

Revoke from PUBLIC and grant explicitly to accounts which need the functionality

File Security

DBMS_ADVISOR
DBMS_LOB
UTL_FILE

Used to corrupt/manipulate O/S files and LOB information

Revoke from PUBLIC and grant explicitly to accounts which need the functionality

Encryption

DBMS_CRYPTO
DBMS_OBFUSCATION_TOOLKIT
DBMS_RANDOM

Cryptography-related function

Revoke from PUBLIC and grant explicitly to accounts which need the functionality

Java

DBMS_JAVA
DBMS_JAVA_TEST

Allow execution of O/S commands

Revoke from PUBLIC and grant explicitly to accounts which need the functionality

Scheduler

DBMS_SCHEDULER
DBMS_JOB

Run DB or O/S jobs

Revoke from PUBLIC and grant explicitly to accounts which need the functionality

SQL Injection Helpers

DBMS_SQL
DBMS_XMLGEN
DBMS_XMLQUERY
DBMS_XLMSTORE
DBMS_XLMSAVE
DBMS_REDACT

Privs to help Injection attacks

Other

Not granted to PUBLIC by default, but need to be check as they are extremely powerful

DBMS_BACKUP_RESTORE
DBMS_FILE_TRANSFER
DBMS_SYS_SQL
DBMS_REPCAT_SQL_UTL
INITJVMAUX
DBMS_AQADM_SYS
DBMS_STREAMS_RPC
DBMS_PRVTAQIM
LTADM
DBMS_IJOB
DBMS_PDB_EXEC_SQL

High level access

Sensitive Tables

CDB_LOCAL_ADMINAUTH$
DEFAULT_PWD$
ENC$
HISTGRM$
HIST_HEAD$
LINK$
PDB_SYNC$
SCHEDULER$_CREDENTIAL
USER$
USER_HISTORY$
XS$VERIFIERS

Not granted to PUBLIC by default, but need to be check as they are extremely sensitive

May contain password and other sensitive information

```
SELECT owner, table_name, grantee, privilege, type FROM dba_tab_privs
WHERE grantee='PUBLIC'
  AND table_name IN ('DBMS_LDAP', 'UTL_INADDR', 'UTL_TCP', 'UTL_MAIL', 'UTL_SMTP',
'UTL_DBWS', 'UTL_ORAMTS', 'UTL_HTTP', 'HTTPURITYPE', 'DBMS_ADVISOR', 'DBMS_LOB',
'UTL_FILE', 'DBMS_CRYPTO', 'DBMS_OBFUSCATION_TOOLKIT', 'DBMS_RANDOM', 'DBMS_JAVA',
'DBMS_JAVA_TEST', 'DBMS_SCHEDULER', 'DBMS_JOB', 'DBMS_SQL', 'DBMS_XMLGEN',
'DBMS_XMLQUERY', 'DBMS_XLMSTORE', 'DBMS_XLMSAVE', 'DBMS_REDACT',
'CDB_LOCAL_ADMINAUTH$', 'DEFAULT_PWD$', 'ENC$', 'HISTGRM$', 'HIST_HEAD$', 'LINK$',
'PDB_SYNC$', 'SCHEDULER$_CREDENTIAL', 'USER$', 'USER_HISTORY$', 'XS$VERIFIERS', 'DBMS_BACKUP_RESTORE',
'DBMS_FILE_TRANSFER','DBMS_SYS_SQL','DBMS_REPCAT_SQL_UTL','INITJVMAUX','DBMS_AQADM_SYS','DBMS_STREAMS_RPC',
'DBMS_PRVTAQIM','LTADM',
'DBMS_IJOB','DBMS_PDB_EXEC_SQL')
ORDER BY owner,table_name
```

This does not mean your system is vulnerable, but you may have more open attack vectors than you realise

OWNER	TABLE_NAME	GRANTEE	PRIVILEG	TYPE
SYS	DBMS_ADVISOR	PUBLIC	EXECUTE	PACKAGE
SYS	DBMS_JAVA	PUBLIC	EXECUTE	PACKAGE
SYS	DBMS_JOB	PUBLIC	EXECUTE	PACKAGE
SYS	DBMS_LDAP	PUBLIC	EXECUTE	PACKAGE
SYS	DBMS_LOB	PUBLIC	EXECUTE	PACKAGE
SYS	DBMS_OBFUSCATION_TOOLKIT	PUBLIC	EXECUTE	PACKAGE
SYS	DBMS_RANDOM	PUBLIC	EXECUTE	PACKAGE
SYS	DBMS_SCHEDULER	PUBLIC	EXECUTE	PACKAGE
SYS	DBMS_SQL	PUBLIC	EXECUTE	PACKAGE
SYS	DBMS_XMLGEN	PUBLIC	EXECUTE	PACKAGE
SYS	DBMS_XMLQUERY	PUBLIC	EXECUTE	PACKAGE
SYS	HTTPURITYPE	PUBLIC	EXECUTE	TYPE
SYS	**UTL_FILE**	PUBLIC	EXECUTE	PACKAGE
SYS	UTL_HTTP	PUBLIC	EXECUTE	PACKAGE
SYS	UTL_INADDR	PUBLIC	EXECUTE	PACKAGE
SYS	UTL_SMTP	PUBLIC	EXECUTE	PACKAGE
SYS	**UTL_TCP**	PUBLIC	EXECUTE	PACKAGE

Don't forget to check the **CDB** as well as each **PDB**!

OBSERVABILITY

WHAT ELSE IS OUT THERE?

what many of you are not doing

packet capturing

i love the whooshing sound of data as it flys past

network encryption

i love the whooshing sound of data as it flys past

Transport Layer Security (TLS)
[using certificates]

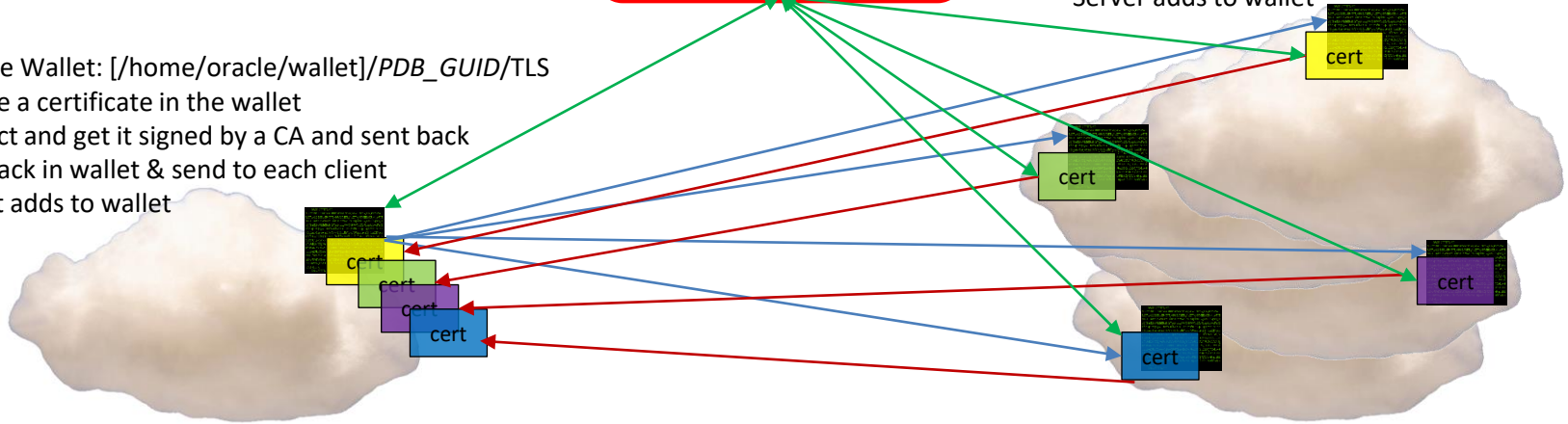
or

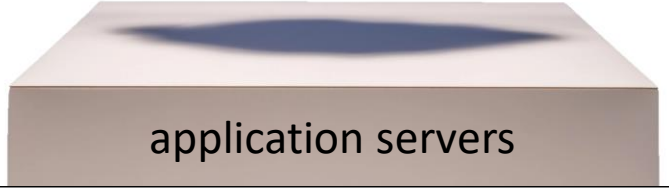
Oracle Native Network Encryption
and Integrity

Certificate Process

Create Wallet on each client server. Add DB cert.
create a certificate in the wallet
Extract and get it signed by a CA
Put back in wallet & send to server
Server adds to wallet

Create Wallet: [/home/oracle/wallet]/*PDB_GUID*/TLS
create a certificate in the wallet
Extract and get it signed by a CA and sent back
Put back in wallet & send to each client
Client adds to wallet





Oracle Native Network Encryption and Integrity [formerly: Oracle Advanced Networking Option]

change the sqlnet.ora file and add:

```
SQLNET.ENCRYPTION_SERVER       = REQUESTED
SQLNET.CRYPTO_CHECKSUM_SERVER = REQUESTED
```

```
ACCEPTED   - encrypt if requested [DEFAULT]
REJECTED   - refuse to encrypt (reject requests, don't connect)
REQUESTED  - encrypt if you can, don't if you can't, but CONNECT
REQUIRED   - encrypt otherwise the connection is refused
```

change the sqlnet.ora file and add:

```
SQLNET.ENCRYPTION_SERVER       = REQUESTED
SQLNET.CRYPTO_CHECKSUM_SERVER = REQUESTED
```

```
SQL> SELECT sys_context('USERENV', 'NETWORK_PROTOCOL') as protocol
     FROM dual;

PROTOCOL
----------------
tcp
```

change the sqlnet.ora file and add:

```
SQLNET.ENCRYPTION_SERVER       = REQUESTED
SQLNET.CRYPTO_CHECKSUM_SERVER = REQUESTED
```

```
SQL> SELECT network_service_banner FROM v$session_connect_info
      WHERE sid IN (SELECT DISTINCT sid FROM v$mystat) ORDER BY 1;

NETWORK_SERVICE_BANNER
----------------------------------------------------------------------
AES256 Encryption service adapter for Linux: Version 19.0.0.0.0 - Production
Crypto-checksumming service for Linux: Version 19.0.0.0.0 - Production
Encryption service for Linux: Version 19.0.0.0.0 - Production
SHA1 Crypto-checksumming service adapter for Linux: Version 19.0.0.0.0 - Production
TCP/IP NT Protocol Adapter for Linux: Version 19.0.0.0.0 - Production
```

change the sqlnet.ora file and add:

```
SQLNET.ENCRYPTION_SERVER        = REQUESTED
SQLNET.ENCRYPTION_TYPES_SERVER = (AES256)
SQLNET.CRYPTO_CHECKSUM_SERVER = REQUESTED
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (SHA384)
```

```
SQL> SELECT network_service_banner FROM v$session_connect_info
      WHERE sid IN (SELECT DISTINCT sid FROM v$mystat) ORDER BY 1;

NETWORK_SERVICE_BANNER
----------------------------------------------------------------------
AES256 Encryption service adapter for Linux: Version 19.0.0.0.0 - Production
Crypto-checksumming service for Linux: Version 19.0.0.0.0 - Production
Encryption service for Linux: Version 19.0.0.0.0 - Production
SHA384 Crypto-checksumming service adapter for Linux: Version 19.0.0.0.0 - Producti
TCP/IP NT Protocol Adapter for Linux: Version 19.0.0.0.0 - Production
```

Implementation Flow

```
SQLNET.ENCRYPTION_SERVER       = REQUESTED
SQLNET.ENCRYPTION_TYPES_SERVER = (AES256)
SQLNET.CRYPTO_CHECKSUM_SERVER = REQUESTED
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (SHA384)
```

- Set to **REQUESTED**
- Observe connection encryption status
- Resolve client issues

```
SQLNET.ENCRYPTION_CLIENT              = REQUESTED
SQLNET.ENCRYPTION_TYPES_CLIENT        = (AES256)
SQLNET.CRYPTO_CHECKSUM_CLIENT         = REQUESTED
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT = (SHA384)
```

Implementation Flow

```
SQLNET.ENCRYPTION_SERVER       = REQUIRED
SQLNET.ENCRYPTION_TYPES_SERVER = (AES256)
SQLNET.CRYPTO_CHECKSUM_SERVER = REQUIRED
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (SHA384)
```

- Set to REQUESTED
- Observe connection encryption status
- Resolve client issues
- Set to **REQUIRED**

Problem

1. It's not *actually* TLSv1.2
2. Non-repudiation of servers

BUT

1. You don't have to manage certificates
2. You probably don't need to make any client changes

Performance

1% to 15% CPU overhead for encryption and decryption

Almost identical for TLS and Native Network Encryption

Encrypting Data-at-Rest

What's the point?

Use your SAN

(or the O/S with dm-crypt/LUKS/etc)

[no good for file hacking]

Transparent Data Encryption (TDE)

- DB Files are encrypted by Oracle
- Encrypt columns, tablespaces or the entire DB
- cannot hack files from the O/S
- Oracle Cloud (or ExaCC), it's free and mandatory
- On-Prem, or anyone else's cloud, it's expensive
- Only realistic option for Exadata

Simple TDE Implementation

create a keystore (in CDB)

```
SQL> ADMINISTER KEY MANAGEMENT CREATE KEYSTORE
/home/oracle/keystore/' IDENTIFIED BY mypwd;

SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
mypwd CONTAINER=ALL;

SQL> ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY mypwd WITH
BACKUP CONTAINER=ALL;

SQL> SELECT * FROM v$encryption_wallet;

WRL_TYPE  WRL_PARAMETER                STATUS  WALLET_TYPE          WALLET_OR KEYSTORE FULLY_BAC     CON_ID
--------- ------------------------- ------- -------------------- --------- -------- --------- ----------
FILE      /home/oracle/keystore/      OPEN    PASSWORD             SINGLE    NONE     NO                 1
FILE                                  OPEN    PASSWORD             SINGLE    UNITED   NO                 2
FILE                                  OPEN    PASSWORD             SINGLE    UNITED   NO                 3
FILE                                  OPEN    PASSWORD             SINGLE    UNITED   NO                 5
```

```
sqlnet.ora:
ENCRYPTION_WALLET_LOCATION =
  (SOURCE =(METHOD = FILE)(METHOD_DATA =
    (DIRECTORY = /home/oracle/keystore/)))
```

Simple TDE Implementation

```
conn neil/oracle@UTF8PDB1
Connected.

SQL> create table t_enc (c1 number,c2 varchar2(10) encrypt);
Table created.

SQL> insert into t_enc values (1,'encrypt');
1 row created.

SQL> commit;
Commit complete.

SQL> select * from t_enc;

        C1 C2
---------- ----------
         1 encrypt
```

```
shutdown/startup

SQL> conn neil/oracle@UTF8PDB1

SQL> select c1 from t_enc;
        C1
----------
         1

SQL> select c1,c2 from t_enc;
ERROR at line 1:
ORA-28365: wallet is not open

SQL> connect / as sysdba

SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN
     IDENTIFIED BY mypwd container=all;
keystore altered.

SQL> conn neil/oracle@UTF8PDB1
Connected.

SQL> select * from t_enc;

        C1 C2
---------- ----------
         1 encrypt
```

Simple TDE Implementation

Create Encrypted Tablespace

```
CREATE TABLESPACE enc_ts
datafile '/u01/oradata/UTF8/UTF8PDB1/enc_ts01.dbf' SIZE 128K
AUTOEXTEND ON
NEXT 64K
ENCRYPTION USING 'AES256'
DEFAULT STORAGE (ENCRYPT);
```

```
Tablespace Created
```

Simple TDE Implementation

Always Create Encrypted Tablespaces

```
SQL > show parameter encrypt
NAME                          TYPE        VALUE
------------------------------ ----------- ------------------------
encrypt_new_tablespaces       string      CLOUD_ONLY


SQL> ALTER SYSTEM SET encrypt_new_tablespaces='ALWAYS' scope=both
```

Simple TDE Implementation

Convert Tablespace

```
SQL> !ls /u01/oradata/UTF8/UTF8PDB1/users*
/u01/oradata/UTF8/UTF8PDB1/users01.dbf

SQL> ALTER TABLESPACE users ENCRYPTION ONLINE USING 'AES256'
     ENCRYPT
     FILE_NAME_CONVERT=
    ('/u01/oradata/UTF8/UTF8PDB1/users01.dbf',
     '/u01/oradata/UTF8/UTF8PDB1/users01enc.dbf');
Tablespace altered.

SQL> !ls /u01/oradata/UTF8/UTF8PDB1/users*
/u01/oradata/UTF8/UTF8PDB1/users01enc.dbf
```

Transparent Data Encryption (TDE)

Performance

- Exadata can offload some decryption to storage cells
- Encryption is always on your database (compute) nodes
- Overhead usually in the 5%-40% range [some workloads can be much worse]

Audit

Traditional Audit

Places files in AUDIT_FILE_DEST on each node
Data in SYS.AUD$ (for standard audit)
Data in SYS.FGA_LOG$ (for fine-grained auditing)
Does not record the command by default, only the action
(set AUDIT_TRAIL to “DB, EXTENDED” or “XML, EXTENDED”)

Audit

Use Unified Audit

- Everything is in a single immutable location [AUD$UNIFIED]
- Can also write to the Linux SYSLOG – kept away from DBAs
- It's faster – less DB impact

Unified Audit

Setup

Re-link the Oracle binaries to switch to exclusive mode

[DB/listener/etc must be down for this]

```
cd $ORACLE_HOME/rdbms/lib
make -f ins_rdbms.mk uniaud_on ioracle
```

Validate in each database that unified auditing mode is set:

```
SELECT VALUE FROM V$OPTION WHERE PARAMETER = 'Unified Auditing';

VALUE
----------------------------------------------------------------
TRUE
```

Unified Audit

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';
```

Set a reasonable partition frequency:

```
DBMS_AUDIT_MGMT.ALTER_PARTITION_INTERVAL(
   INTERVAL_NUMBER        => 7,
   INTERVAL_FREQUENCY     => 'DAY');
```

Unified Audit

Switch off all built-in policies

```
NOAUDIT POLICY ora_logon_failures ;
NOAUDIT POLICY ora_secureconfig;
NOAUDIT POLICY ora_account_mgmt;
NOAUDIT POLICY ora_cis_recommendations;
NOAUDIT POLICY ora_database_parameter;
```

Unified Audit

Enable some built-in policies

~~AUDIT POLICY ora_logon_failures ;~~ <- NOT THIS ONE!
AUDIT POLICY ora_secureconfig;
AUDIT POLICY ora_account_mgmt;
AUDIT POLICY **ora_cis_recommendations**;
AUDIT POLICY ora_database_parameter;

These will enable all CIS recommendations, but that policy alone does not monitor admin activity!

Unified Audit

Add your policies

~~audit policy ORA_LOGON_FAILURES ;~~ <- not this one!

```
CREATE AUDIT POLICY all_logons
PRIVILEGES CREATE SESSION CONTAINER=CURRENT;

AUDIT POLICY all_logons;
```

Captures every logon, not just unsuccessful ones

Unified Audit

Add your policies

```
CREATE AUDIT POLICY all_selects
PRIVILEGES SELECT ANY TABLE, READ ANY TABLE
CONTAINER=CURRENT;

AUDIT POLICY all_selects;
```

Captures every SELECT or READ using the ANY privilege
Who is not using a specifically granted privilege to read application data?

Unified Audit

This is the only audit control you have in the Autonomous Database

Add Fine Grained Audit Policies (if needed)

```
DBMS_FGA.ADD_POLICY (
   object_schema       =>  'your_schema',
   object_name         =>  'person',
   policy_name         =>  'person_info',
   audit_condition     =>  null,
   audit_column        =>  'salary,age',
   enable              =>   TRUE,
   statement_types     =>  'SELECT, INSERT, UPDATE, DELETE',
   audit_column_opts   =>   DBMS_FGA.ANY_COLUMNS);
```

Who is accessing or changing the SALARY or AGE column?

Unified Audit

Housekeeping - create a scheduler job

```
BEGIN
dbms_scheduler.create_job('"MY_AUDIT_HOUSEKEEPING"',
job_type=>'PLSQL_BLOCK', job_action=>
'DECLARE
 v_instance_number number := 1;
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)
                , rac_instance_number => v_instance_number);
 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=> 'FREQ = DAILY; INTERVAL = 7'
,end_date=>NULL
,job_class=>'"SCHED$_LOG_ON_ERRORS_CLASS"'
,enabled=>FALSE
,auto_drop=>FALSE
,comments=> 'Cleanup Unified Audit older than 3 months'
);
COMMIT;
dbms_scheduler.enable('MY_AUDIT_HOUSEKEEPING');
END;
/
```

Unified Audit

Extract the data

Company specific:

- create an "audit-read" user and allow security to extract the data to [*Splunk/LogRhythm/your corp security package*] directly from the DB for analysis
- Extract the data (as JSON/XML/CSV file) from AUD$UNIFIED to a secure NFS drive
- etc

Patch Management

- Patches are released every 3 months on a known date
- 83% of exploits are against systems where the vulnerability patch has been released over 6 months previously
- “Management” frequently don't see the point, until it's too late
- Audit and Compliance is your friend

Critical Patch Updates
19 July 2022
18 October 2022
17 January 2023
18 April 2023

DBSAT

Oracle Database Security Assessment Tool (DBSAT) (Doc ID 2138254.1)
Oracle semi-supported Database security tool on MOS

Data Safe

Now available for on-premises databases
(DBSAT with a pretty GUI)

https://www.oracle.com/security/database-security/data-safe/

MISSING!

There's ***lots*** missing from what I just talked about

initialisation parameters

IP whitelisting

listener parameters

PDB lockdown profiles

database vault

database firewall

Virtual Private Database

Real Application Security

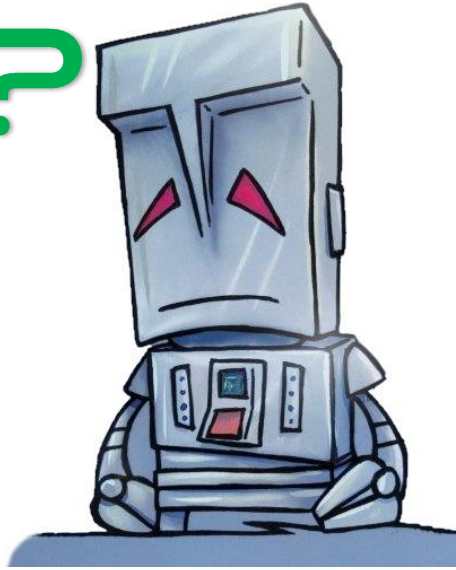
etc

PLUS

your role privileges

your data!

ANY QUESTIONS?



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

I have a million ideas. They all point to certain death.

THANK YOU...

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

...and may your god go with you