Database Security: Identifying Security Risks with the Database Security Assessment Tool

TUTORIAL | NOVEMBER 2017

Table of Contents

TUTORIAL OVERVIEW

Welcome to this Database Security Assessment Tool tutorial!

In this tutorial, you will get hands-on experience in setting up and executing the Database Security Assessment Tool (DBSAT). You will also learn how to interpret the results of DBSAT produced reports.

The goal of this tutorial is the following:

- Understand what DBSAT is and what its components are
- Understand how to execute DBSAT
- Understand the current report formats and their common use cases
- Understand how to interpret the results

Please note that this tutorial was first built for Oracle Open World 2017 DBSAT Hands-on Labs. Your DBSAT report results may vary.

» This tutorial will be using DBSAT 1.0.2. For more information and how to download DBSAT go to: http://www.oracle.com/technetwork/database/security/dbsat.html

» DBSAT supports Oracle Database 10.2.0.5 and later.

» DBSAT Documentation can be found here

» In this tutorial, we have used a 12.2.0.1 multi-tenant database with a pluggable database named **orcl**. Please update passwords and oracle service name/SID in the steps/scripts below as appropriate.

» If for any reason you are not sure if you are connected to the CDB$ROOT container or the PDB, just type *`show con_name`* in SQLPlus.

» While working through this tutorial, you will be copying the required commands from this workbook and paste it by right-clicking on the terminal.

CHALLENGE

Assumption: The database under assessment is badly configured. Really bad as many databases out there. Hackers have already found 10 critical issues that they could use to exploit and get access to your data. Will you be able to find them as well?

Typically, these misconfigurations fall into these domains: Patching, Users, Privileges and Roles, Authorization Control, Data Encryption, Fine-Grained Access Control, Auditing, Database Configuration, Network Configuration, Operating System.

Your first assignment is to find out more than 5 issues in 5 minutes.

Raise your hand, and let us know your findings.

- How hard was it?
- What was the process that you have followed?
- What should you address first?
- How much did you miss?
- How can you repeat the process in a simple way?
- What if you need to share scripts and outputs?
- Will it be interpreted the same way if read by someone else?

It's now time to move to the Database Security Assessment Tool and find out how it could help to identify misconfigurations, users, roles, privileges and the overall security status.

DBSAT OVERVIEW

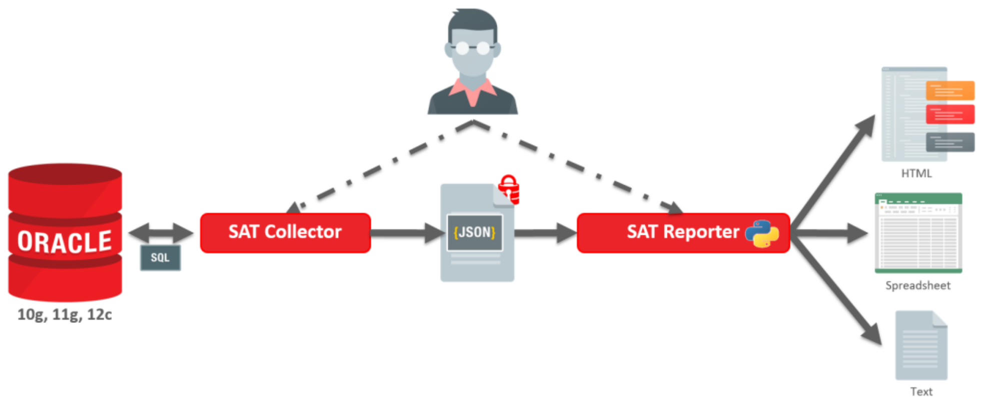
The Oracle Database Security Assessment Tool (DBSAT) analyzes database configurations and security policies to uncover security risks and improve the security posture of Oracle Databases within your organization.

You can use DBSAT to implement and enforce security best practices in your organization. DBSAT reports on the state of user accounts, role and privilege grants, and policies that control the use of various security features in the database.

You can use report findings to:

» Fix immediate short-term risks
» Implement a comprehensive security strategy

Components of DBSAT and Flow



DBSAT consists of two components, the DBSAT Collector and the DBSAT Reporter that correspond to the functions of data collection and data analysis respectively:

» The DBSAT Collector executes SQL queries and runs operating system commands to collect data from the system to be assessed. It does this primarily by querying database dictionary views. The collected data is written to a file that is used by the DBSAT Reporter in the analysis phase.
» The DBSAT Reporter analyzes the collected data and reports its findings and recommendations in multiple formats: HTML, Excel, and Text. The Reporter can run on any machine: PC, laptop, or server. You are not limited to running it on the same server as the Collector.

Benefits of Using DBSAT

You can use DBSAT to:

- Quickly identify security configuration errors in your databases
- Promote security best practices
- Improve the security posture of your Oracle Databases
- Reduce the attack surface and exposure to risk

LAB EXERCISE 01

Creating a Database User to run DBSAT

In this step, you will create a database user with the necessary privileges to be able to collect data with DBSAT.

In order to collect complete data, the DBSAT Collector must be run on a server that contains the database, because it executes some operating system commands to collect process and file system information that cannot be obtained from the database. In addition, the DBSAT Collector must be run as an OS user with read permissions on files and directories under ORACLE_HOME in order to collect and process file system data using OS commands.

The DBSAT Collector collects most of its data by querying database views. It must connect to the database as a user with sufficient privileges to select from these views. You can grant the DBSAT user the individual privileges in the following list, or you can grant this user the DBA role plus the DV_SECANALYST role if needed.

Required privileges and roles:

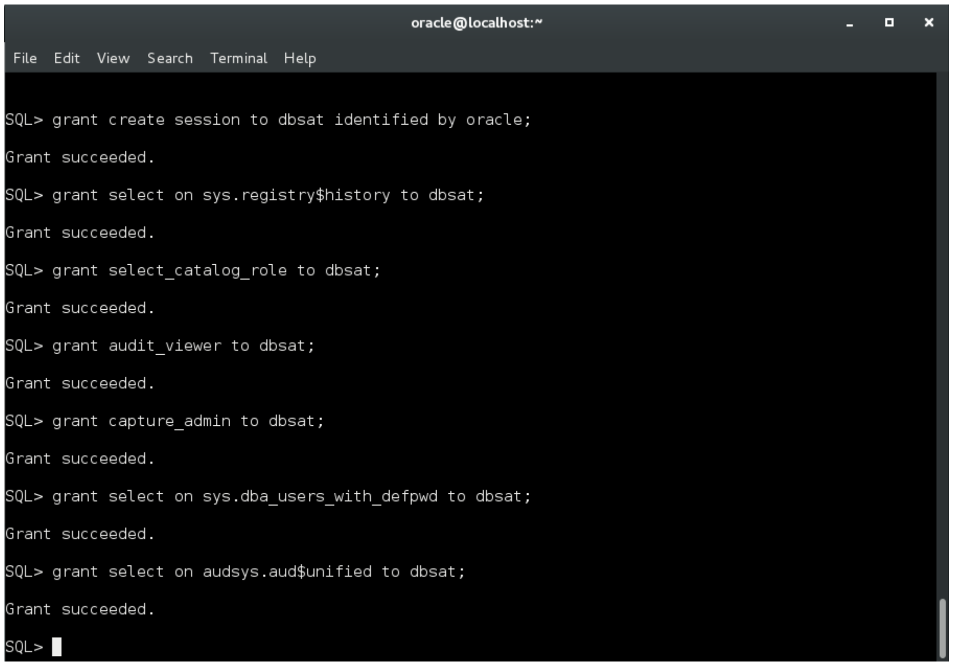
- `CREATE SESSION`
- `SELECT` on `SYS.REGISTRY$HISTORY`
- Role `SELECT_CATALOG_ROLE`
- Role `DV_SECANALYST` (if Database Vault is enabled)
- Role `AUDIT_VIEWER` (12c only)
- Role `CAPTURE_ADMIN` (12c only)
- `SELECT` on `SYS.DBA_USERS_WITH_DEFPWD` (11g and 12c)
- `SELECT` on `AUDSYS.AUD$UNIFIED` (12c only)

In this Lab, we will be running DBSAT with the oracle OS user and will create a database user with the privileges that are strictly needed for its execution.

1. As system, in the `orcl` PDB, execute:

```
grant create session to dbsat identified by oracle;
grant select on sys.registry$history to dbsat;
grant select_catalog_role to dbsat;
grant audit_viewer to dbsat;
grant capture_admin to dbsat;
grant select on sys.dba_users_with_defpwd to dbsat;
grant select on audsys.aud$unified to dbsat;
```

As the output, you should get:



```
SQL> grant create session to dbsat identified by oracle;

Grant succeeded.

SQL> grant select on sys.registry$history to dbsat;

Grant succeeded.

SQL> grant select_catalog_role to dbsat;

Grant succeeded.

SQL> grant audit_viewer to dbsat;

Grant succeeded.

SQL> grant capture_admin to dbsat;

Grant succeeded.

SQL> grant select on sys.dba_users_with_defpwd to dbsat;

Grant succeeded.

SQL> grant select on audsys.aud$unified to dbsat;

Grant succeeded.

SQL> 
```

LAB EXERCISE 02

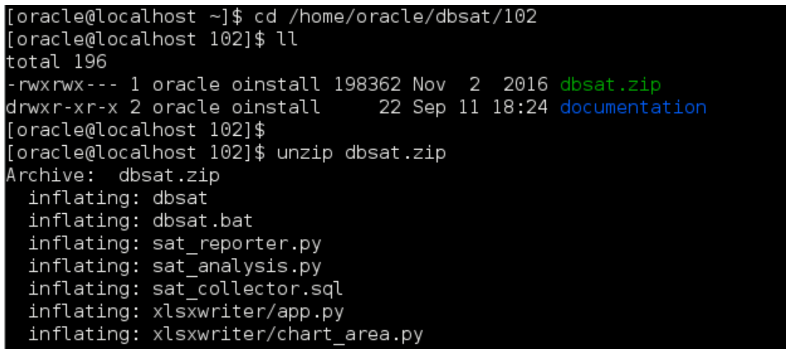
Installing DBSAT

In this exercise, you will learn how to install DBSAT.

The Oracle Database Security Assessment Tool (DBSAT) installation is a simple process. Go to http://www.oracle.com/technetwork/database/security/dbsat.html and download the dbsat zip file. Copy it to your database server destination and simply extract the file dbsat.zip to install the Database Security Assessment Tool:

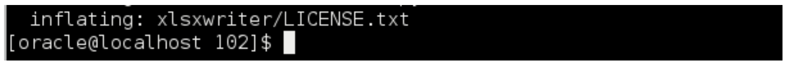
1. In the terminal, type

```
mkdir -p /home/oracle/dbsat/102
#move the zip file from the download location to the created folder
#mv dbsat.zip /home/oracle/dbsat/102/
cd /home/oracle/dbsat/102
unzip dbsat.zip
```



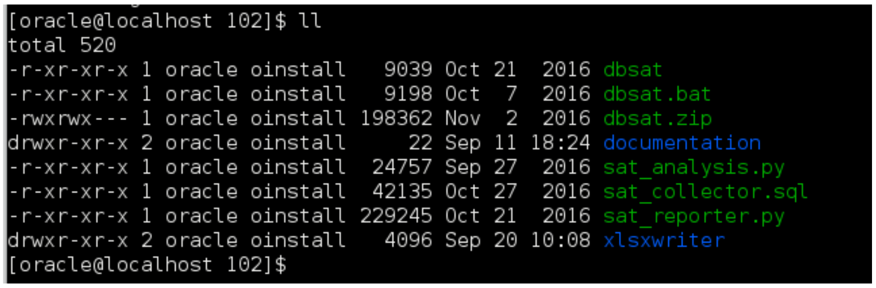
```
[oracle@localhost ~]$ cd /home/oracle/dbsat/102
[oracle@localhost 102]$ ll
total 196
-rwxrwx--- 1 oracle oinstall 198362 Nov  2  2016 dbsat.zip
drwxr-xr-x 2 oracle oinstall     22 Sep 11 18:24 documentation
[oracle@localhost 102]$
[oracle@localhost 102]$ unzip dbsat.zip
Archive:  dbsat.zip
  inflating: dbsat
  inflating: dbsat.bat
  inflating: sat_reporter.py
  inflating: sat_analysis.py
  inflating: sat_collector.sql
  inflating: xlsxwriter/app.py
  inflating: xlsxwriter/chart_area.py
```

The latest unzip output line should be:



```
  inflating: xlsxwriter/LICENSE.txt
[oracle@localhost 102]$
```

2. Validate that the unzipped files match the following list.
   Type `ll`:



```
[oracle@localhost 102]$ ll
total 520
-r-xr-xr-x 1 oracle oinstall   9039 Oct 21  2016 dbsat
-r-xr-xr-x 1 oracle oinstall   9198 Oct  7  2016 dbsat.bat
-rwxrwx--- 1 oracle oinstall 198362 Nov  2  2016 dbsat.zip
drwxr-xr-x 2 oracle oinstall     22 Sep 11 18:24 documentation
-r-xr-xr-x 1 oracle oinstall  24757 Sep 27  2016 sat_analysis.py
-r-xr-xr-x 1 oracle oinstall  42135 Oct 27  2016 sat_collector.sql
-r-xr-xr-x 1 oracle oinstall 229245 Oct 21  2016 sat_reporter.py
drwxr-xr-x 2 oracle oinstall   4096 Sep 20 10:08 xlsxwriter
[oracle@localhost 102]$
```

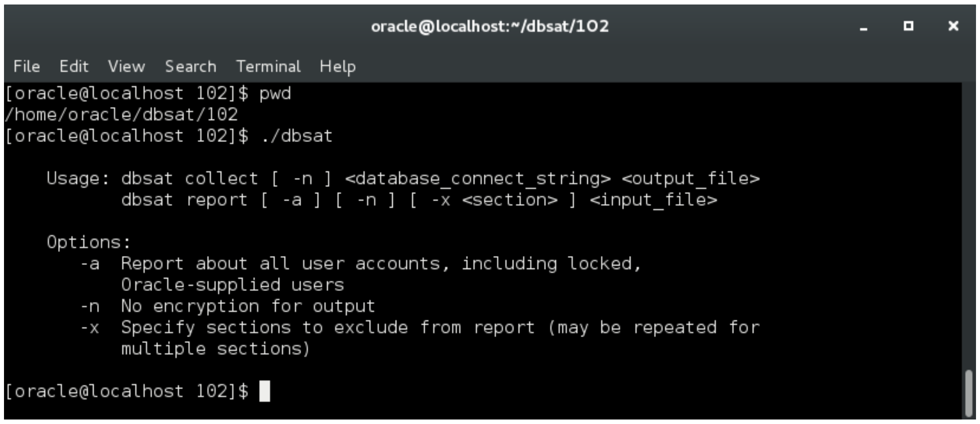
You can now run the DBSAT Collector and DBSAT Reporter from here.

LAB EXERCISE 03

Run DBSAT Collector

In this exercise, you will learn how to execute DBSAT collector. DBSAT collector will connect to the database and collect data needed for analysis. DBSAT will not create any objects in the database. DBSAT only executes queries similar to the ones a Database Administrator would be executing in his daily tasks.

1. To view all the DBSAT execution parameters please type:
   `./dbsat`


oracle@localhost:~/dbsat/102
File Edit View Search Terminal Help

```
[oracle@localhost 102]$ pwd
/home/oracle/dbsat/102
[oracle@localhost 102]$ ./dbsat

    Usage: dbsat collect [ -n ] <database_connect_string> <output_file>
           dbsat report [ -a ] [ -n ] [ -x <section> ] <input_file>

    Options:
       -a  Report about all user accounts, including locked,
           Oracle-supplied users
       -n  No encryption for output
       -x  Specify sections to exclude from report (may be repeated for
           multiple sections)

[oracle@localhost 102]$ 
```

As you can see, dbsat takes different input parameters depending on the component you are running. Unless specified (-n), the output files will be stored in a password protected zip file.

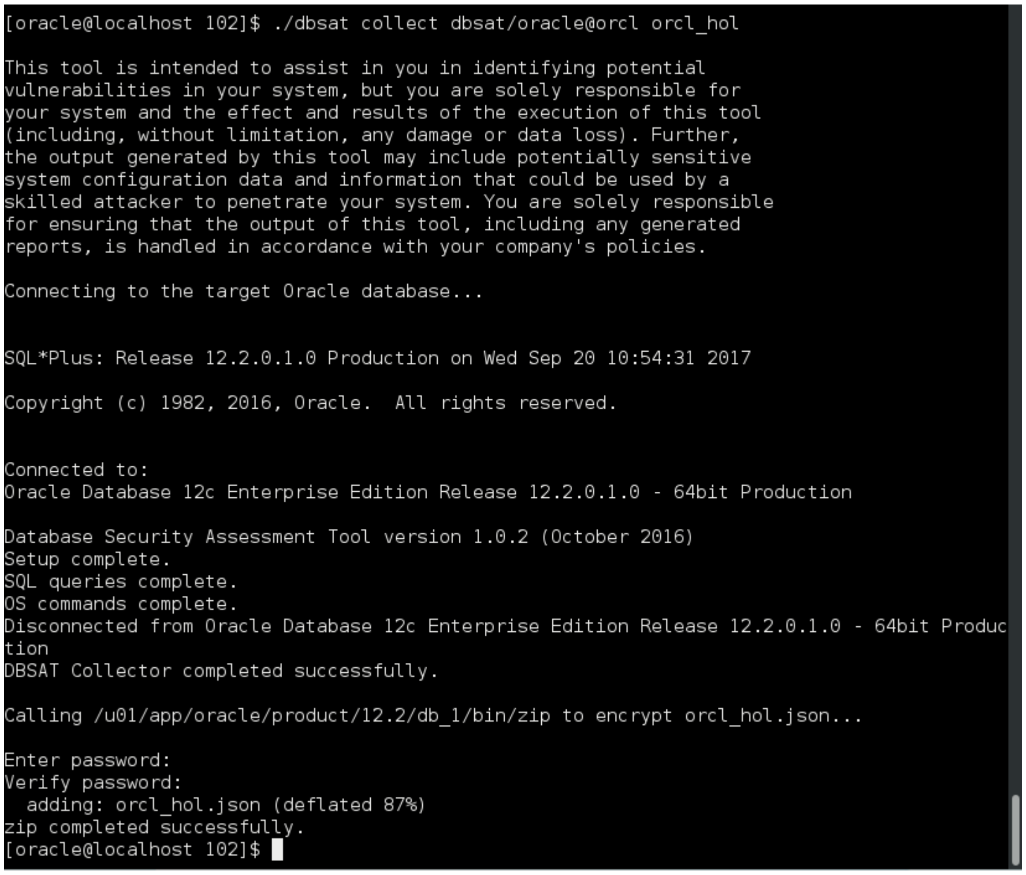
2. Let's run dbsat to collect data from the `orcl` pdb

```
./dbsat collect dbsat/oracle@orcl orcl_hol
```

The time it takes to complete depends on the hardware and the data that needs to be collected. A database that has thousands of users and roles might take hours to run. This lab was created to provide some findings for analysis, and depending on the available hardware in the room, it might take between 2 to 5 minutes.

At the end of the process, you'll be asked to provide a password twice (please use `oracle`).
If you choose a different one, please do not forget it as you'll need it when running `dbsat report`.

This is the expected output:



```
[oracle@localhost 102]$ ./dbsat collect dbsat/oracle@orcl orcl_hol

This tool is intended to assist in you in identifying potential
vulnerabilities in your system, but you are solely responsible for
your system and the effect and results of the execution of this tool
(including, without limitation, any damage or data loss). Further,
the output generated by this tool may include potentially sensitive
system configuration data and information that could be used by a
skilled attacker to penetrate your system. You are solely responsible
for ensuring that the output of this tool, including any generated
reports, is handled in accordance with your company's policies.

Connecting to the target Oracle database...


SQL*Plus: Release 12.2.0.1.0 Production on Wed Sep 20 10:54:31 2017

Copyright (c) 1982, 2016, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

Database Security Assessment Tool version 1.0.2 (October 2016)
Setup complete.
SQL queries complete.
OS commands complete.
Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Produc
tion
DBSAT Collector completed successfully.

Calling /u01/app/oracle/product/12.2/db_1/bin/zip to encrypt orcl_hol.json...

Enter password:
Verify password:
  adding: orcl_hol.json (deflated 87%)
zip completed successfully.
[oracle@localhost 102]$
```

A file named `orcl_hol.zip` is created in the directory (`/home/oracle/dbsat/102`).

You don't need to unzip the file. DBSAT reporter will take either the *json* file (if `-n` was used) or the *zip* file.

The next step is to analyze the collected data using `dbsat reporter`.

LAB EXERCISE 04

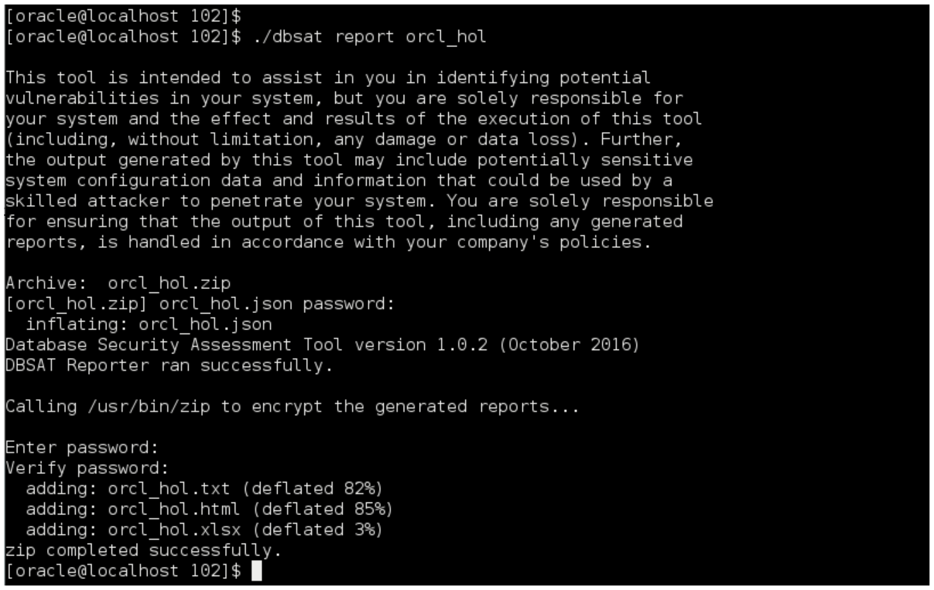
Run DBSAT Reporter

In this exercise, you will learn how to execute the dbsat reporter. DBSAT reporter will take as input the file generated by the collector (*json* or *zip* file) and will produce one zip file containing three reports in different formats: HTML, spreadsheet, and text. If you choose not to encrypt data, the three report files will be generated in the specified directory.

1. Let's run dbsat to collect data from the orcl pdb

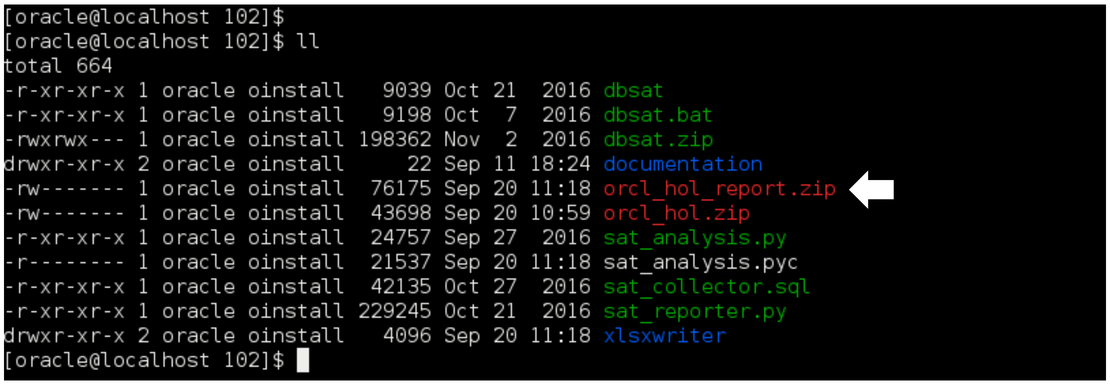
   ```
   ./dbsat report orcl_hol
   ```

   DBSAT will prompt the user for one password – the password used when running the collector so it can unzip the file – followed by another password prompt that will be used to protect the reports zip file, plus the password confirmation.



```
[oracle@localhost 102]$
[oracle@localhost 102]$ ./dbsat report orcl_hol

This tool is intended to assist in you in identifying potential
vulnerabilities in your system, but you are solely responsible for
your system and the effect and results of the execution of this tool
(including, without limitation, any damage or data loss). Further,
the output generated by this tool may include potentially sensitive
system configuration data and information that could be used by a
skilled attacker to penetrate your system. You are solely responsible
for ensuring that the output of this tool, including any generated
reports, is handled in accordance with your company's policies.

Archive:  orcl_hol.zip
[orcl_hol.zip] orcl_hol.json password:
  inflating: orcl_hol.json
Database Security Assessment Tool version 1.0.2 (October 2016)
DBSAT Reporter ran successfully.

Calling /usr/bin/zip to encrypt the generated reports...

Enter password:
Verify password:
  adding: orcl_hol.txt (deflated 82%)
  adding: orcl_hol.html (deflated 85%)
  adding: orcl_hol.xlsx (deflated 3%)
zip completed successfully.
[oracle@localhost 102]$
```

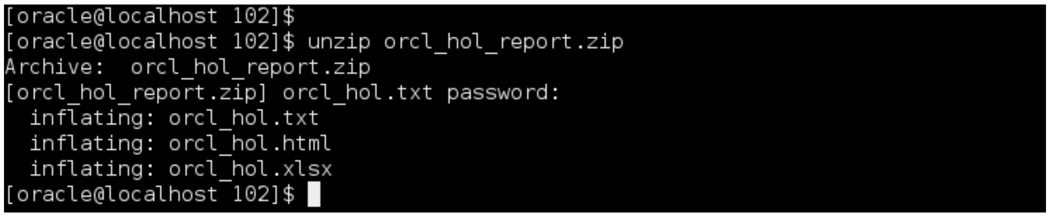
   You will end up with the results of the analysis inside a password protected zip file named `orcl_hol_report.zip`.



```
[oracle@localhost 102]$
[oracle@localhost 102]$ ll
total 664
-r-xr-xr-x 1 oracle oinstall   9039 Oct 21  2016 dbsat
-r-xr-xr-x 1 oracle oinstall   9198 Oct  7  2016 dbsat.bat
-rwxrwx--- 1 oracle oinstall 198362 Nov  2  2016 dbsat.zip
drwxr-xr-x 2 oracle oinstall     22 Sep 11 18:24 documentation
-rw------- 1 oracle oinstall  76175 Sep 20 11:18 orcl_hol_report.zip
-rw------- 1 oracle oinstall  43698 Sep 20 10:59 orcl_hol.zip
-r-xr-xr-x 1 oracle oinstall  24757 Sep 27  2016 sat_analysis.py
-r-------- 1 oracle oinstall  21537 Sep 20 11:18 sat_analysis.pyc
-r-xr-xr-x 1 oracle oinstall  42135 Oct 27  2016 sat_collector.sql
-r-xr-xr-x 1 oracle oinstall 229245 Oct 21  2016 sat_reporter.py
drwxr-xr-x 2 oracle oinstall   4096 Sep 20 11:18 xlsxwriter
[oracle@localhost 102]$
```

2. Let's unzip the file to view the reports

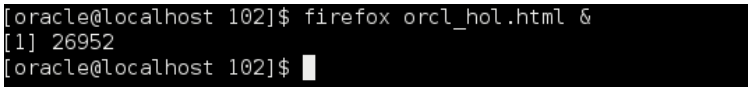
```
unzip orcl_hol_report.zip
```



```
[oracle@localhost 102]$
[oracle@localhost 102]$ unzip orcl_hol_report.zip
Archive:  orcl_hol_report.zip
[orcl_hol_report.zip] orcl_hol.txt password:
  inflating: orcl_hol.txt
  inflating: orcl_hol.html
  inflating: orcl_hol.xlsx
[oracle@localhost 102]$
```

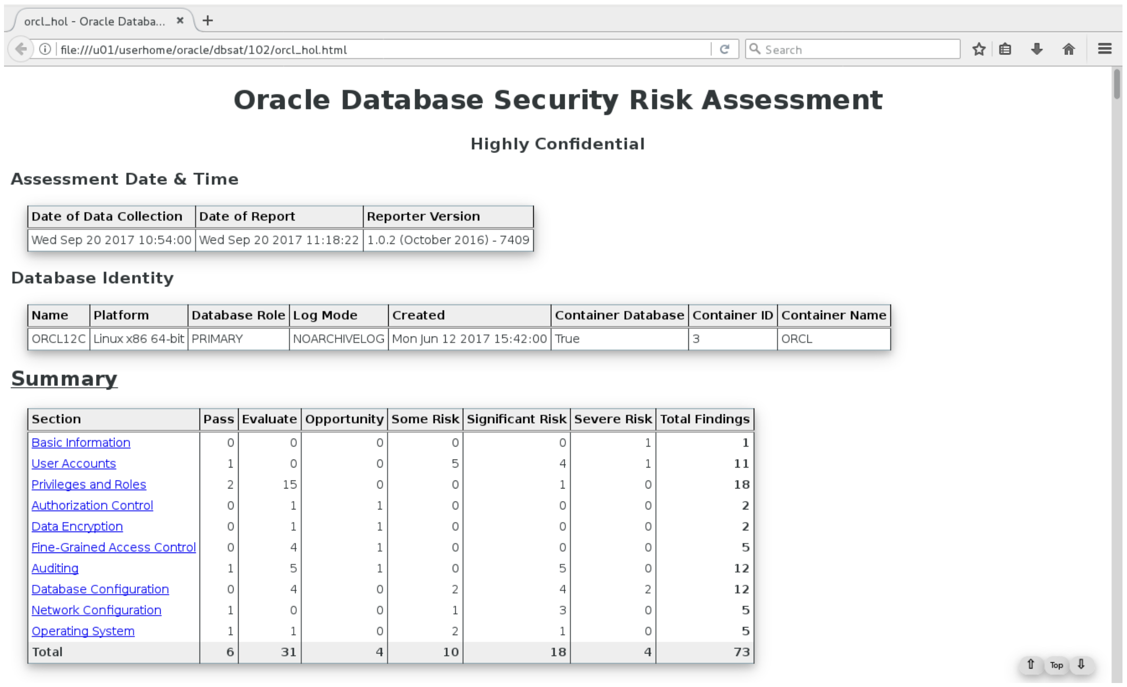
3. Open Firefox to view the html report. For that type:

```
firefox orcl_hol.html &
```



```
[oracle@localhost 102]$ firefox orcl_hol.html &
[1] 26952
[oracle@localhost 102]$
```

Firefox should open and display the html report:



orcl_hol - Oracle Databa...
file:///u01/userhome/oracle/dbsat/102/orcl_hol.html
Search
Top

Oracle Database Security Risk Assessment

Highly Confidential

Assessment Date & Time

Date of Data Collection	Date of Report	Reporter Version
Wed Sep 20 2017 10:54:00	Wed Sep 20 2017 11:18:22	1.0.2 (October 2016) - 7409

Database Identity

Name	Platform	Database Role	Log Mode	Created	Container Database	Container ID	Container Name
ORCL12C	Linux x86 64-bit	PRIMARY	NOARCHIVELOG	Mon Jun 12 2017 15:42:00	True	3	ORCL

Summary

Section	Pass	Evaluate	Opportunity	Some Risk	Significant Risk	Severe Risk	Total Findings
Basic Information	0	0	0	0	0	1	1
User Accounts	1	0	0	5	4	1	11
Privileges and Roles	2	15	0	0	1	0	18
Authorization Control	0	1	1	0	0	0	2
Data Encryption	0	1	1	0	0	0	2
Fine-Grained Access Control	0	4	1	0	0	0	5
Auditing	1	5	1	0	5	0	12
Database Configuration	0	4	0	2	4	2	12
Network Configuration	1	0	0	1	3	0	5
Operating System	1	1	0	2	1	0	5
Total	6	31	4	10	18	4	73

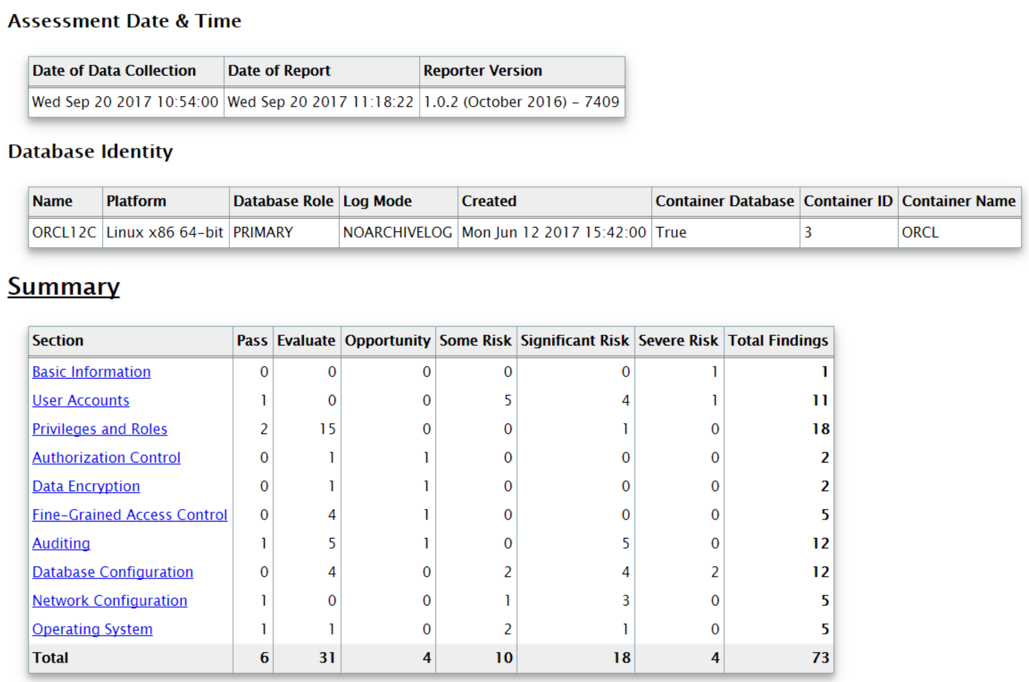
LAB EXERCISE 05

Analyze the generated report - Summary

In this exercise, you will learn how to how to analyze the Risk Assessment Report. We will dive into the summary table, different types of risks, the anatomy of a finding and the actual findings.

Please take a couple of minutes to scroll through the html report. You can click the links in the summary table to go to a specific section or use the navigation arrows at the bottom right.

The report contains *informational tables*, as the one shown below and *findings*. We will get back to the *findings* later. *Informational tables* provide either summary information or additional context to the findings in the same section.



Assessment Date & Time

Date of Data Collection	Date of Report	Reporter Version
Wed Sep 20 2017 10:54:00	Wed Sep 20 2017 11:18:22	1.0.2 (October 2016) - 7409

Database Identity

Name	Platform	Database Role	Log Mode	Created	Container Database	Container ID	Container Name
ORCL12C	Linux x86 64-bit	PRIMARY	NOARCHIVELOG	Mon Jun 12 2017 15:42:00	True	3	ORCL

Summary

Section	Pass	Evaluate	Opportunity	Some Risk	Significant Risk	Severe Risk	Total Findings
Basic Information	0	0	0	0	0	1	1
User Accounts	1	0	0	5	4	1	11
Privileges and Roles	2	15	0	0	1	0	18
Authorization Control	0	1	1	0	0	0	2
Data Encryption	0	1	1	0	0	0	2
Fine-Grained Access Control	0	4	1	0	0	0	5
Auditing	1	5	1	0	5	0	12
Database Configuration	0	4	0	2	4	2	12
Network Configuration	1	0	0	1	3	0	5
Operating System	1	1	0	2	1	0	5
Total	**6**	**31**	**4**	**10**	**18**	**4**	**73**

» At the top of the report, you will find information about the Collector and Reporter run details as the *date of data collection* and the *date of the report* generation along with the reporter version

» Follows the Database Identity information where you will find details about the target database

» Then the Summary table.
The Summary table presents all the findings per section/domain along with their severity level.

LAB EXERCISE 06

Analyze the generated report – Findings

In this exercise, you will learn what is a *Finding*.

The DBSAT reporter resulting analysis is reported in units called Findings.

In each Finding you see:

» **Unique ID for the Rule**
The ID has two parts: the prefix identifies the report section, and the suffix identifies the specific rule.

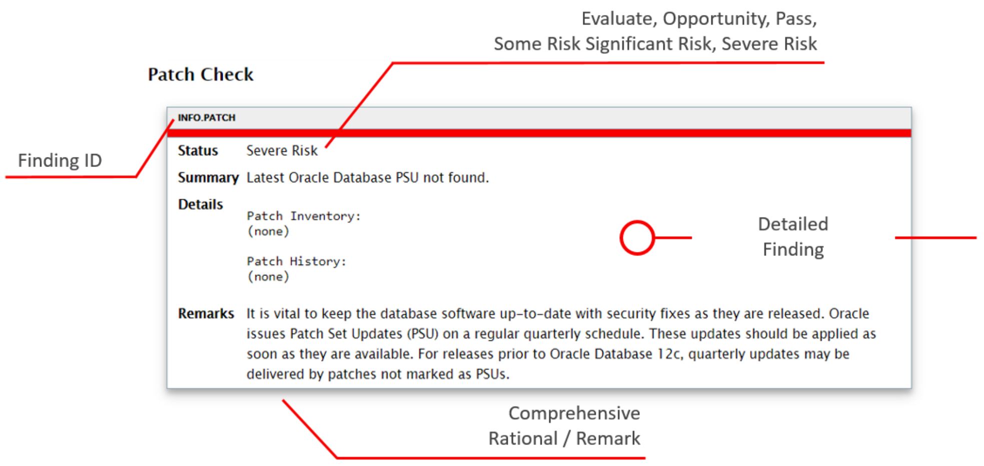
» **Status**
You can use the status values as guidelines for implementing DBSAT recommendations. They can be used to prioritize and schedule changes based on the level of risk, and what it might mean to your organization. Severe risk might require immediate remedial action, whereas other risks might be fixed during a scheduled downtime, or bundled together with other maintenance activities.

- Pass: no error found
- Evaluate: needs manual analysis
- Some Risk: low
- Significant Risk: medium
- Severe Risk: high
- Opportunity: improve security posture by enabling additional security features and technology. Opportunity for Improvement.

» **Summary**
A brief summary of the finding. When the finding is informational, the summary typically reports only the number of data elements that were examined.

» **Details**
Provides detailed information to explain the finding summary, typically results from the assessed database, followed by any recommendations for changes.

» **Remarks**
Explains the reason for the rule and recommended actions for remediation.


Evaluate, Opportunity, Pass,
Some Risk Significant Risk, Severe Risk
Patch Check
Finding ID
Detailed Finding
Comprehensive Rational / Remark

INFO.PATCH

Status	Severe Risk
Summary	Latest Oracle Database PSU not found.
Details	Patch Inventory: (none) Patch History: (none)
Remarks	It is vital to keep the database software up-to-date with security fixes as they are released. Oracle issues Patch Set Updates (PSU) on a regular quarterly schedule. These updates should be applied as soon as they are available. For releases prior to Oracle Database 12c, quarterly updates may be delivered by patches not marked as PSUs.

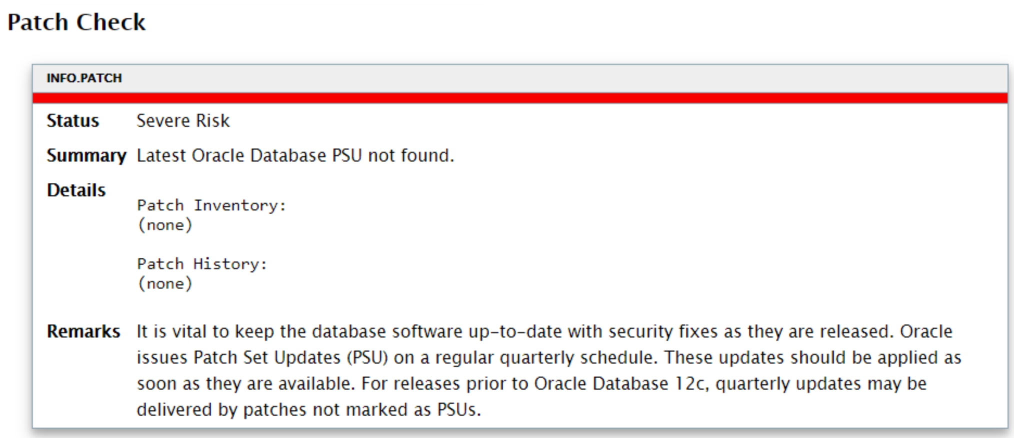
LAB EXERCISE 07

Analyze the generated report – Details

In this exercise, you will be guided by relevant DBSAT findings. This will provide you with knowledge on what DBSAT validates and the value it provides

1. Let's have a look at the "Basic Information" and look after the "Severe Risk" *finding* (red line).

   Click "**Basic Information**" in the Summary table and scroll to **INFO.PATCH**.



Patch Check

INFO.PATCH

Status Severe Risk

Summary Latest Oracle Database PSU not found.

Details

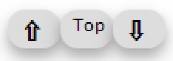
```
Patch Inventory:
(none)

Patch History:
(none)
```

Remarks It is vital to keep the database software up-to-date with security fixes as they are released. Oracle issues Patch Set Updates (PSU) on a regular quarterly schedule. These updates should be applied as soon as they are available. For releases prior to Oracle Database 12c, quarterly updates may be delivered by patches not marked as PSUs.

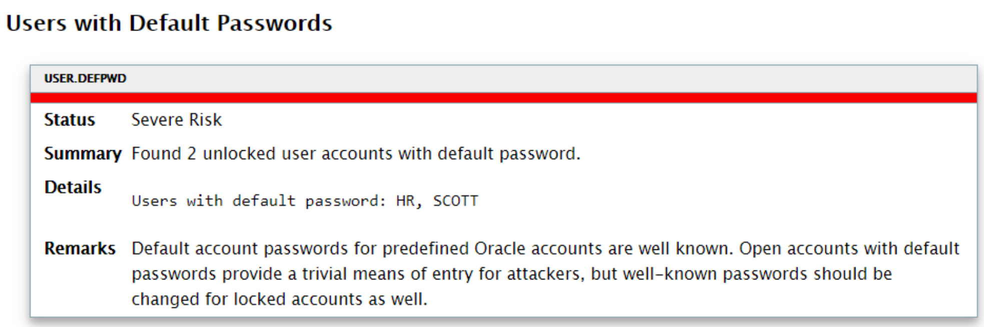
   It seems that the database needs to be patched. This is an important finding as it is one of the most common ways hackers get into databases. They exploit vulnerable, unpatched, databases.

   This is a **Finding.**

2. You can always return back to the Summary table. Click **TOP** on the navigation panel (bottom right).



3. Let's have a look at "User Accounts" and search for the finding marked as "Severe Risk" (red line). Click the "**User Accounts**" link in the Summary table and scroll to **USER.DEFPWD**.



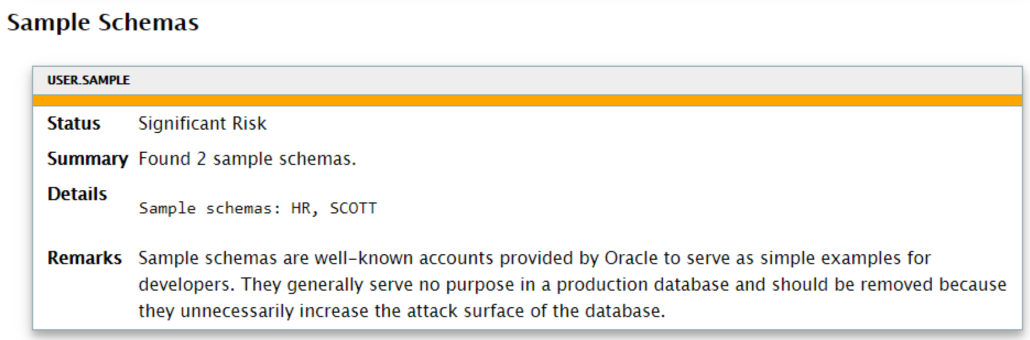
Users with Default Passwords

USER.DEFPWD

Status	Severe Risk
Summary	Found 2 unlocked user accounts with default password.
Details	`Users with default password: HR, SCOTT`
Remarks	Default account passwords for predefined Oracle accounts are well known. Open accounts with default passwords provide a trivial means of entry for attackers, but well-known passwords should be changed for locked accounts as well.

Great! DBSAT pointed out that we have users **HR** and **SCOTT** with default password. What could those be? It's time to either change their passwords or drop these sample schemas as they shouldn't be in my production databases. Keep this in mind.

4. In fact, these users are also highlighted in another finding (scroll up to or search for **USER.SAMPLE**):

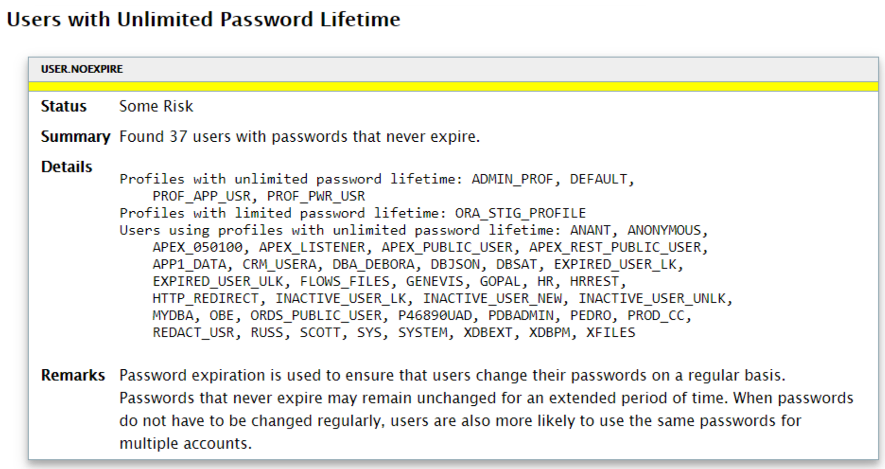


Sample Schemas

USER.SAMPLE

Status	Significant Risk
Summary	Found 2 sample schemas.
Details	`Sample schemas: HR, SCOTT`
Remarks	Sample schemas are well-known accounts provided by Oracle to serve as simple examples for developers. They generally serve no purpose in a production database and should be removed because they unnecessarily increase the attack surface of the database.

5. What else can DBSAT show me on users? I'm curious about SCOTT and HR.
Scroll down or search for **USER.NOEXPIRE**:



Users with Unlimited Password Lifetime

USER.NOEXPIRE

Status Some Risk

Summary Found 37 users with passwords that never expire.

Details

```
Profiles with unlimited password lifetime: ADMIN_PROF, DEFAULT,
    PROF_APP_USR, PROF_PWR_USR
Profiles with limited password lifetime: ORA_STIG_PROFILE
Users using profiles with unlimited password lifetime: ANANT, ANONYMOUS,
    APEX_050100, APEX_LISTENER, APEX_PUBLIC_USER, APEX_REST_PUBLIC_USER,
    APP1_DATA, CRM_USERA, DBA_DEBORA, DBJSON, DBSAT, EXPIRED_USER_LK,
    EXPIRED_USER_ULK, FLOWS_FILES, GENEVIS, GOPAL, HR, HRREST,
    HTTP_REDIRECT, INACTIVE_USER_LK, INACTIVE_USER_NEW, INACTIVE_USER_UNLK,
    MYDBA, OBE, ORDS_PUBLIC_USER, P46890UAD, PDBADMIN, PEDRO, PROD_CC,
    REDACT_USR, RUSS, SCOTT, SYS, SYSTEM, XDBEXT, XDBPM, XFILES
```

Remarks Password expiration is used to ensure that users change their passwords on a regular basis. Passwords that never expire may remain unchanged for an extended period of time. When passwords do not have to be changed regularly, users are also more likely to use the same passwords for multiple accounts.

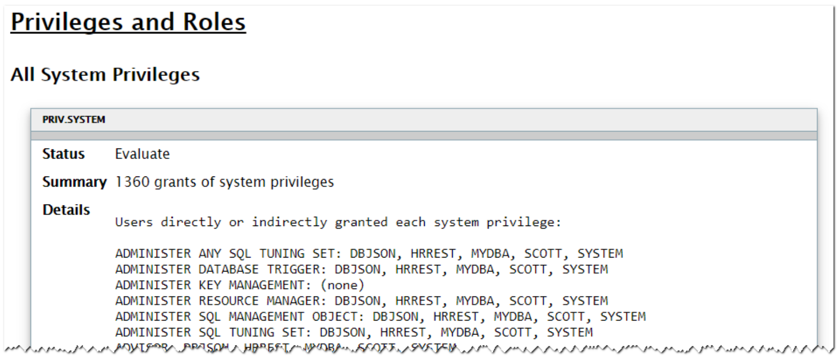
OK, so not only my database isn't patched, I have two sample schemas with default password and these users aren't being requested to change their passwords as they are using profiles with unlimited password lifetime (as shown above).

Findings in this section will provide a view on who are the users in my database, their status, password settings and user profiles.

6. Let's get back to the top again but this time we will review what is going on in the "Privileges and Roles" section. This section shows the largest number of findings (18). Click **TOP** on the navigation panel (bottom right). Followed by clicking the "**Privileges and Roles**" link.

   This section provides information about: System privileges, Roles, Account Management privileges, Privilege Management privileges, Audit Management privileges, Data Access privileges, Access Control exemption privileges, Access to restricted objects, Users with DBA role, Users with Administrative privileges among others.

   You should see this:



Privileges and Roles

All System Privileges

PRIV.SYSTEM

Status Evaluate

Summary 1360 grants of system privileges

Details

```
Users directly or indirectly granted each system privilege:

ADMINISTER ANY SQL TUNING SET: DBJSON, HRREST, MYDBA, SCOTT, SYSTEM
ADMINISTER DATABASE TRIGGER: DBJSON, HRREST, MYDBA, SCOTT, SYSTEM
ADMINISTER KEY MANAGEMENT: (none)
ADMINISTER RESOURCE MANAGER: DBJSON, HRREST, MYDBA, SCOTT, SYSTEM
ADMINISTER SQL MANAGEMENT OBJECT: DBJSON, HRREST, MYDBA, SCOTT, SYSTEM
ADMINISTER SQL TUNING SET: DBJSON, HRREST, MYDBA, SCOTT, SYSTEM
```

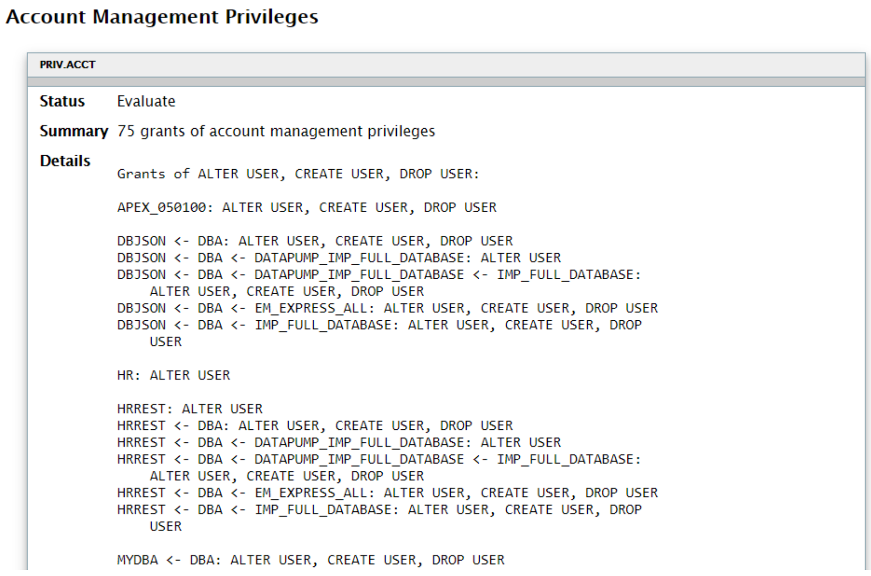
This provides a powerful insight into what can users do that typically is not addressed in common vulnerability management products.

7. Look at the **PRIV.SYSTEM** finding. It seems that **SCOTT** has some powerful privileges. Does he have the DBA role? Good Question! DBSAT provides an answer to that (hint: **PRIV.DBA** finding).

8. DBSAT, where applicable, also shows the grant path (if a privilege was directly granted or indirectly granted) to make it easier to spot wrong grants.

   Scroll down to "Account Management Privileges" (you can also search for **PRIV.ACCT** in the browser).

   You will see this:



Account Management Privileges

PRIV.ACCT

Status Evaluate

Summary 75 grants of account management privileges

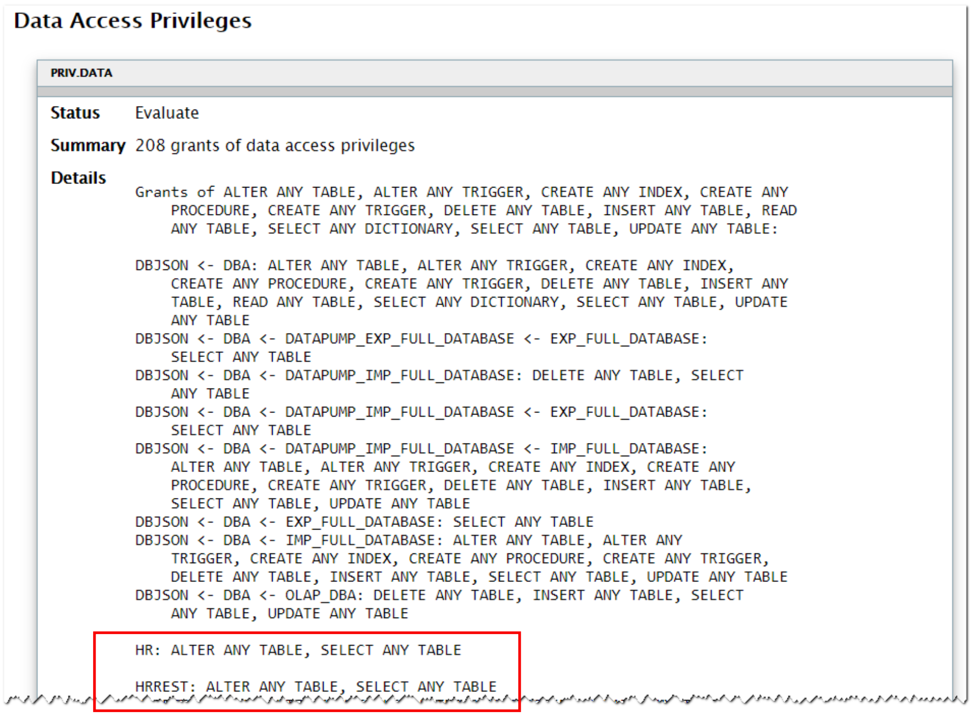
Details

```
Grants of ALTER USER, CREATE USER, DROP USER:

APEX_050100: ALTER USER, CREATE USER, DROP USER

DBJSON <- DBA: ALTER USER, CREATE USER, DROP USER
DBJSON <- DBA <- DATAPUMP_IMP_FULL_DATABASE: ALTER USER
DBJSON <- DBA <- DATAPUMP_IMP_FULL_DATABASE <- IMP_FULL_DATABASE:
    ALTER USER, CREATE USER, DROP USER
DBJSON <- DBA <- EM_EXPRESS_ALL: ALTER USER, CREATE USER, DROP USER
DBJSON <- DBA <- IMP_FULL_DATABASE: ALTER USER, CREATE USER, DROP
    USER

HR: ALTER USER

HRREST: ALTER USER
HRREST <- DBA: ALTER USER, CREATE USER, DROP USER
HRREST <- DBA <- DATAPUMP_IMP_FULL_DATABASE: ALTER USER
HRREST <- DBA <- DATAPUMP_IMP_FULL_DATABASE <- IMP_FULL_DATABASE:
    ALTER USER, CREATE USER, DROP USER
HRREST <- DBA <- EM_EXPRESS_ALL: ALTER USER, CREATE USER, DROP USER
HRREST <- DBA <- IMP_FULL_DATABASE: ALTER USER, CREATE USER, DROP
    USER

MYDBA <- DBA: ALTER USER, CREATE USER, DROP USER
```

This finding will present direct or indirect grants of account management privileges – ALTER USER, CREATE USER, DROP USER – and will show the grant path. Either direct as **HR: ALTER USER** or indirect as **MYDBA <- DBA: ALTER USER, CREATE USER, DROP USER.** MYDBA received those privileges via the DBA role.

9. Let's have a look at another example. This time on "Data Access Privileges" – **PRIV.DATA**.
   Scroll down or search.

   You will see this:



Data Access Privileges

PRIV.DATA

Status Evaluate

Summary 208 grants of data access privileges

Details

```
Grants of ALTER ANY TABLE, ALTER ANY TRIGGER, CREATE ANY INDEX, CREATE ANY
    PROCEDURE, CREATE ANY TRIGGER, DELETE ANY TABLE, INSERT ANY TABLE, READ
    ANY TABLE, SELECT ANY DICTIONARY, SELECT ANY TABLE, UPDATE ANY TABLE:

DBJSON <- DBA: ALTER ANY TABLE, ALTER ANY TRIGGER, CREATE ANY INDEX,
    CREATE ANY PROCEDURE, CREATE ANY TRIGGER, DELETE ANY TABLE, INSERT ANY
    TABLE, READ ANY TABLE, SELECT ANY DICTIONARY, SELECT ANY TABLE, UPDATE
    ANY TABLE
DBJSON <- DBA <- DATAPUMP_EXP_FULL_DATABASE <- EXP_FULL_DATABASE:
    SELECT ANY TABLE
DBJSON <- DBA <- DATAPUMP_IMP_FULL_DATABASE: DELETE ANY TABLE, SELECT
    ANY TABLE
DBJSON <- DBA <- DATAPUMP_IMP_FULL_DATABASE <- EXP_FULL_DATABASE:
    SELECT ANY TABLE
DBJSON <- DBA <- DATAPUMP_IMP_FULL_DATABASE <- IMP_FULL_DATABASE:
    ALTER ANY TABLE, ALTER ANY TRIGGER, CREATE ANY INDEX, CREATE ANY
    PROCEDURE, CREATE ANY TRIGGER, DELETE ANY TABLE, INSERT ANY TABLE,
    SELECT ANY TABLE, UPDATE ANY TABLE
DBJSON <- DBA <- EXP_FULL_DATABASE: SELECT ANY TABLE
DBJSON <- DBA <- IMP_FULL_DATABASE: ALTER ANY TABLE, ALTER ANY
    TRIGGER, CREATE ANY INDEX, CREATE ANY PROCEDURE, CREATE ANY TRIGGER,
    DELETE ANY TABLE, INSERT ANY TABLE, SELECT ANY TABLE, UPDATE ANY TABLE
DBJSON <- DBA <- OLAP_DBA: DELETE ANY TABLE, INSERT ANY TABLE, SELECT
    ANY TABLE, UPDATE ANY TABLE

HR: ALTER ANY TABLE, SELECT ANY TABLE

HRREST: ALTER ANY TABLE, SELECT ANY TABLE
```

HR and **HRREST** have been granted the powerful **SELECT ANY TABLE** and **ALTER ANY TABLE** privileges.

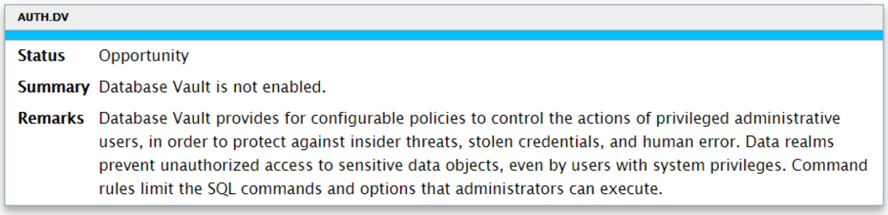
Do these users really need it?
That’s something that DBSAT can’t define as it lacks organizational and processes awareness. That is why it is marked for review (Status = Evaluate).

10. We have spent some time now looking into users, privileges, and roles. What about “Authorization Control”?
Click Top and in the summary table go to “**Authorization Control**”.

You will see this:

Authorization Control

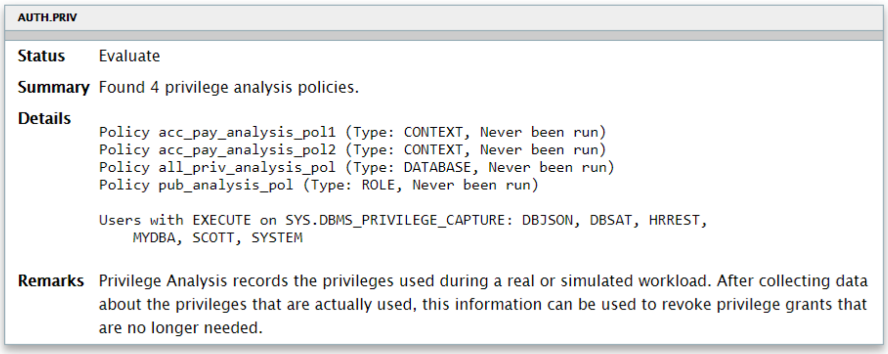
Database Vault



AUTH.DV

Status	Opportunity
Summary	Database Vault is not enabled.
Remarks	Database Vault provides for configurable policies to control the actions of privileged administrative users, in order to protect against insider threats, stolen credentials, and human error. Data realms prevent unauthorized access to sensitive data objects, even by users with system privileges. Command rules limit the SQL commands and options that administrators can execute.

Privilege Analysis



AUTH.PRIV

Status Evaluate

Summary Found 4 privilege analysis policies.

Details

```
Policy acc_pay_analysis_pol1 (Type: CONTEXT, Never been run)
Policy acc_pay_analysis_pol2 (Type: CONTEXT, Never been run)
Policy all_priv_analysis_pol (Type: DATABASE, Never been run)
Policy pub_analysis_pol (Type: ROLE, Never been run)

Users with EXECUTE on SYS.DBMS_PRIVILEGE_CAPTURE: DBJSON, DBSAT, HRREST,
    MYDBA, SCOTT, SYSTEM
```

Remarks Privilege Analysis records the privileges used during a real or simulated workload. After collecting data about the privileges that are actually used, this information can be used to revoke privilege grants that are no longer needed.

The **AUTH.DV** Finding is marked Blue (Opportunity) as it presents an opportunity for improvement. Database Vault enables to define Realms around sensitive data to prevent unauthorized access, even from privileged users. Database Vault also enables to control command execution according to a certain factor(s). As an example, you can disable DROP TABLE in your production database or ALTER SYSTEM if not coming from a certain IP ADDRESS or day/time of day.

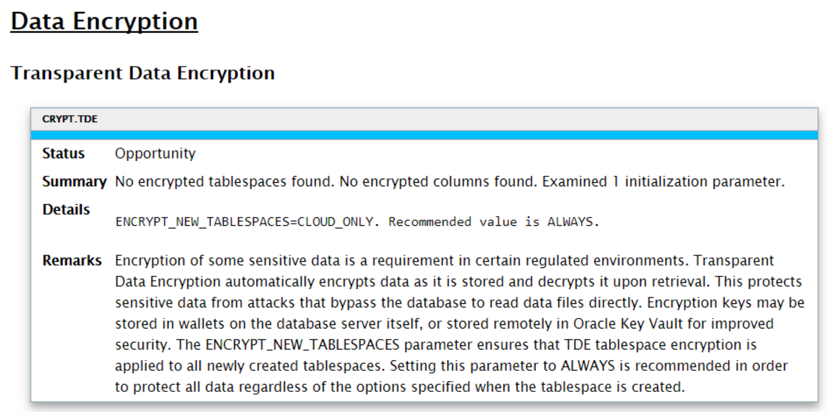
AUTH.PRIV Finding is showing that, apparently, someone already created Database Vault Privilege Analysis policies (maybe concerned with the current user, roles, privs status) to assess the current needs. However, they weren't run.

This feature (Privilege Analysis) extends the capabilities of Oracle Database Vault to include least privilege analysis for existing applications and a continuous analysis of privileges used during new application development. Privilege Analysis allows to:

» Report on actual privileges and roles used in the database
» Identify unused privileges and roles by users and applications
» Reduce risk by helping enforce least privilege for users and applications

11. Scroll down to "**Data Encryption**"

You should see this:



Data Encryption

Transparent Data Encryption

CRYPT.TDE

Status Opportunity

Summary No encrypted tablespaces found. No encrypted columns found. Examined 1 initialization parameter.

Details

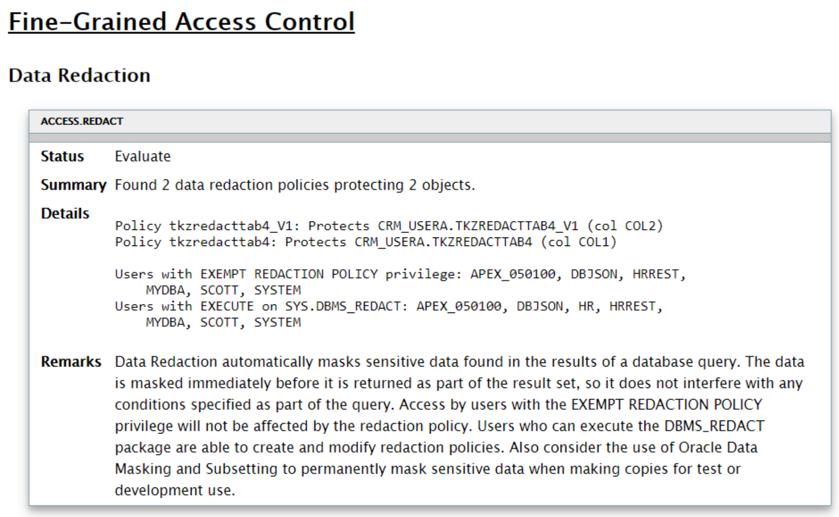
```
ENCRYPT_NEW_TABLESPACES=CLOUD_ONLY. Recommended value is ALWAYS.
```

Remarks Encryption of some sensitive data is a requirement in certain regulated environments. Transparent Data Encryption automatically encrypts data as it is stored and decrypts it upon retrieval. This protects sensitive data from attacks that bypass the database to read data files directly. Encryption keys may be stored in wallets on the database server itself, or stored remotely in Oracle Key Vault for improved security. The ENCRYPT_NEW_TABLESPACES parameter ensures that TDE tablespace encryption is applied to all newly created tablespaces. Setting this parameter to ALWAYS is recommended in order to protect all data regardless of the options specified when the tablespace is created.

Data is not being encrypted. No encrypted tablespaces found, nor encrypted columns. Is this database storing sensitive data? Is the data it holds subject to any regulation? Make sure that you understand the data that it is stored in your databases and if they are subject to any regulation.

12. Scroll down to the “**Fine-Grained Access Control**”.

    You will see this:



Fine-Grained Access Control

Data Redaction

ACCESS.REDACT

Status Evaluate

Summary Found 2 data redaction policies protecting 2 objects.

Details

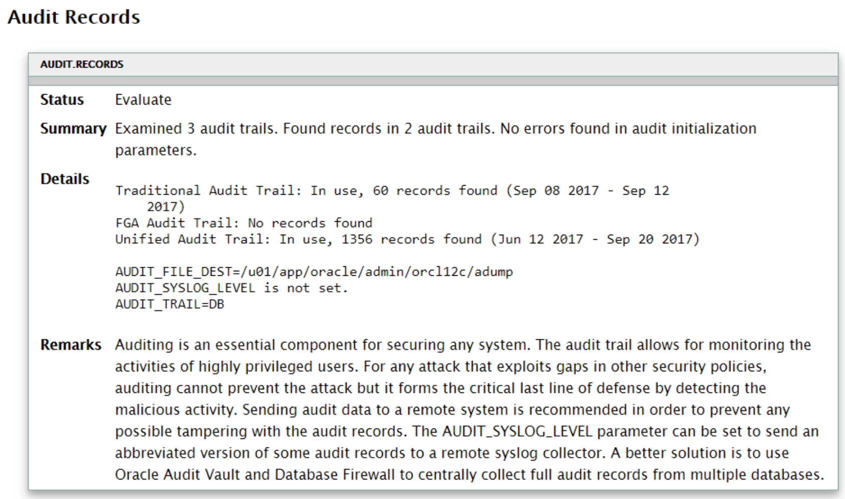
```
Policy tkzredacttab4_V1: Protects CRM_USERA.TKZREDACTTAB4_V1 (col COL2)
Policy tkzredacttab4: Protects CRM_USERA.TKZREDACTTAB4 (col COL1)

Users with EXEMPT REDACTION POLICY privilege: APEX_050100, DBJSON, HRREST,
    MYDBA, SCOTT, SYSTEM
Users with EXECUTE on SYS.DBMS_REDACT: APEX_050100, DBJSON, HR, HRREST,
    MYDBA, SCOTT, SYSTEM
```

Remarks Data Redaction automatically masks sensitive data found in the results of a database query. The data is masked immediately before it is returned as part of the result set, so it does not interfere with any conditions specified as part of the query. Access by users with the EXEMPT REDACTION POLICY privilege will not be affected by the redaction policy. Users who can execute the DBMS_REDACT package are able to create and modify redaction policies. Also consider the use of Oracle Data Masking and Subsetting to permanently mask sensitive data when making copies for test or development use.

This section displays information on Data Redaction, VPD, RAS Policies, Label Security and Transparent Sensitive Data Protection (TSDP) policies. This Database has some policies configured so you can have a look at how a finding would look like in these cases.

In the finding above, you’ll notice that there are **2 Data Redaction policies** in place on table CRM_USERA.TKZREDACTTAB4 and below you will immediately see that there are 6 users that are exempted from those policies. DBSAT also displays the list of users with execute privilege on SYS.DBMS_REDACT, and hence, are able to manage data redaction policies.

13. Have a look and when finished move to the next section – “**Auditing**”.



Audit Records

AUDIT.RECORDS

Status	Evaluate
Summary	Examined 3 audit trails. Found records in 2 audit trails. No errors found in audit initialization parameters.

Details

```
Traditional Audit Trail: In use, 60 records found (Sep 08 2017 - Sep 12
    2017)
FGA Audit Trail: No records found
Unified Audit Trail: In use, 1356 records found (Jun 12 2017 - Sep 20 2017)

AUDIT_FILE_DEST=/u01/app/oracle/admin/orcl12c/adump
AUDIT_SYSLOG_LEVEL is not set.
AUDIT_TRAIL=DB
```

Remarks Auditing is an essential component for securing any system. The audit trail allows for monitoring the activities of highly privileged users. For any attack that exploits gaps in other security policies, auditing cannot prevent the attack but it forms the critical last line of defense by detecting the malicious activity. Sending audit data to a remote system is recommended in order to prevent any possible tampering with the audit records. The AUDIT_SYSLOG_LEVEL parameter can be set to send an abbreviated version of some audit records to a remote syslog collector. A better solution is to use Oracle Audit Vault and Database Firewall to centrally collect full audit records from multiple databases.

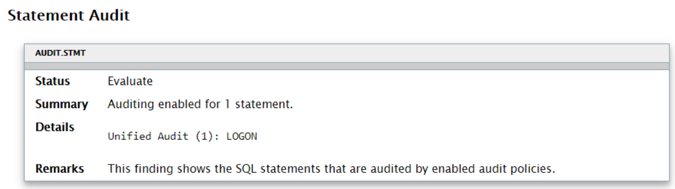
We have traditional audit trail records and Unified Audit trail records as well. To know more about the actual auditing policies in place we need to have a look at the next findings. Just enabling auditing does not generate any audit records. Audit policies need to be in place.

14. Looking at the findings below, this looks pretty bad.

    Statement Audit – Just LOGON actions are being audited with a unified audit policy.

    Object Audit – Audit policies for DBV and OLS are in place but nothing else.

    Privilege Audit – No privileges are being audited.



Statement Audit

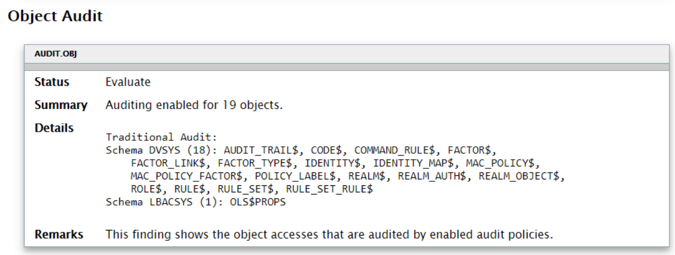
AUDIT.STMT

Status	Evaluate
Summary	Auditing enabled for 1 statement.

Details

```
Unified Audit (1): LOGON
```

Remarks This finding shows the SQL statements that are audited by enabled audit policies.



Object Audit

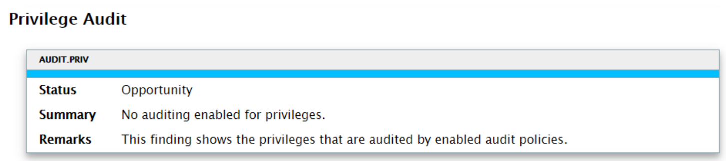
AUDIT.OBJ

Status	Evaluate
Summary	Auditing enabled for 19 objects.

Details

```
Traditional Audit:
Schema DVSYS (18): AUDIT_TRAIL$, CODE$, COMMAND_RULE$, FACTOR$,
    FACTOR_LINK$, FACTOR_TYPE$, IDENTITY$, IDENTITY_MAP$, MAC_POLICY$,
    MAC_POLICY_FACTOR$, POLICY_LABEL$, REALM$, REALM_AUTH$, REALM_OBJECT$,
    ROLE$, RULE$, RULE_SET$, RULE_SET_RULE$
Schema LBACSYS (1): OLS$PROPS
```

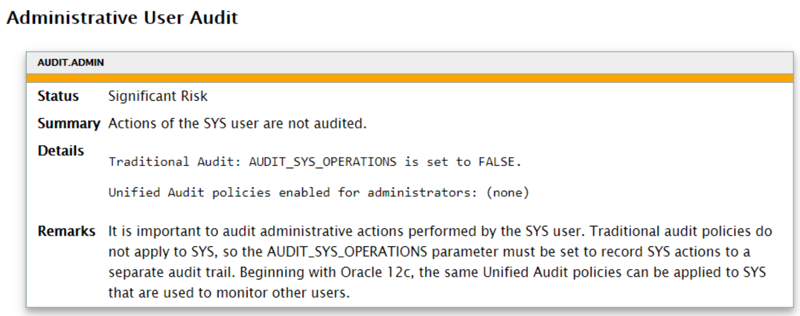
Remarks This finding shows the object accesses that are audited by enabled audit policies.


Privilege Audit

AUDIT.PRIV

Status	Opportunity
Summary	No auditing enabled for privileges.
Remarks	This finding shows the privileges that are audited by enabled audit policies.

15. Moving to the next finding – **AUDIT.ADMIN** – and we find out that auditing for administrative actions by **SYS** is not being performed.


Administrative User Audit

AUDIT.ADMIN

Status Significant Risk

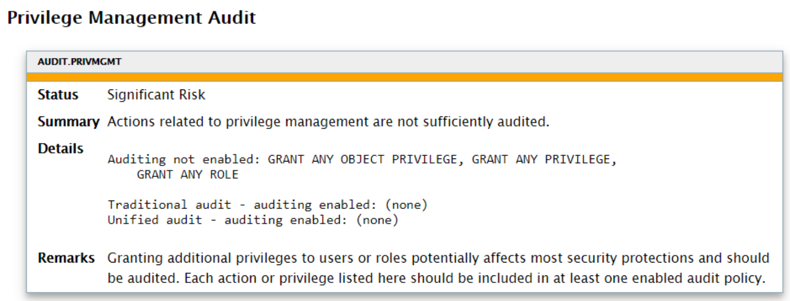
Summary Actions of the SYS user are not audited.

Details

```
Traditional Audit: AUDIT_SYS_OPERATIONS is set to FALSE.

Unified Audit policies enabled for administrators: (none)
```

Remarks It is important to audit administrative actions performed by the SYS user. Traditional audit policies do not apply to SYS, so the AUDIT_SYS_OPERATIONS parameter must be set to record SYS actions to a separate audit trail. Beginning with Oracle 12c, the same Unified Audit policies can be applied to SYS that are used to monitor other users.

16. Neither **CREATE USER**, **DROP USER**, nor, **GRANT ANY ROLE/PRIVILEGE**. See below:


Privilege Management Audit

AUDIT.PRIVMGMT

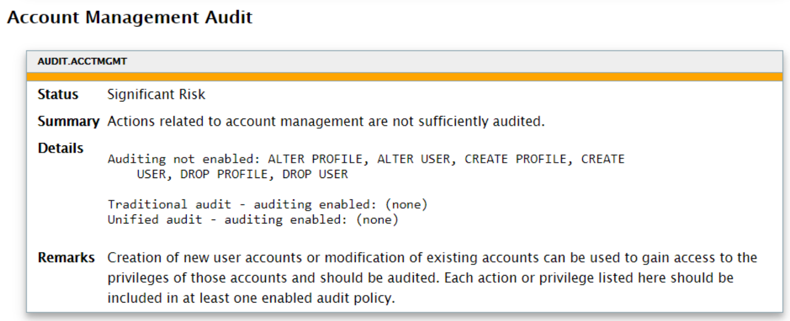
Status Significant Risk

Summary Actions related to privilege management are not sufficiently audited.

Details

```
Auditing not enabled: GRANT ANY OBJECT PRIVILEGE, GRANT ANY PRIVILEGE,
    GRANT ANY ROLE

Traditional audit - auditing enabled: (none)
Unified audit - auditing enabled: (none)
```

Remarks Granting additional privileges to users or roles potentially affects most security protections and should be audited. Each action or privilege listed here should be included in at least one enabled audit policy.


Account Management Audit

AUDIT.ACCTMGMT

Status Significant Risk

Summary Actions related to account management are not sufficiently audited.

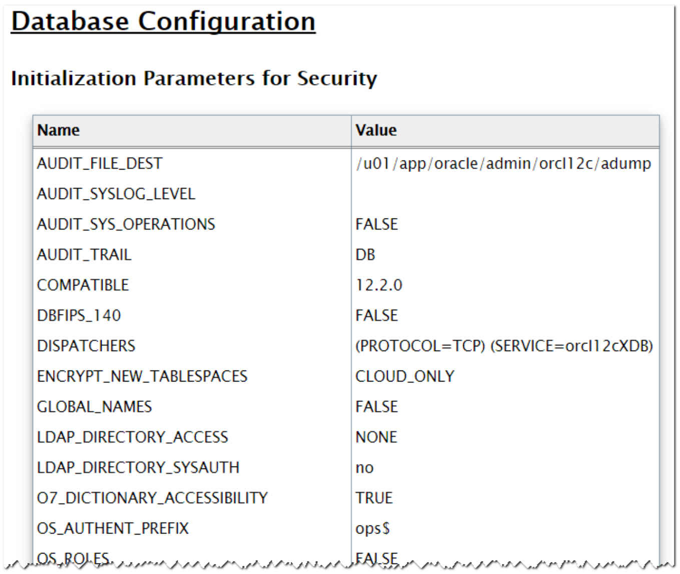
Details

```
Auditing not enabled: ALTER PROFILE, ALTER USER, CREATE PROFILE, CREATE
    USER, DROP PROFILE, DROP USER

Traditional audit - auditing enabled: (none)
Unified audit - auditing enabled: (none)
```

Remarks Creation of new user accounts or modification of existing accounts can be used to gain access to the privileges of those accounts and should be audited. Each action or privilege listed here should be included in at least one enabled audit policy.

Let’s get to the next section – “**Database Configuration**”- and have a look at some of the “Severe” and “Significant” findings.

This section starts with an informational table that provides a summary of relevant security-related initialization parameters.

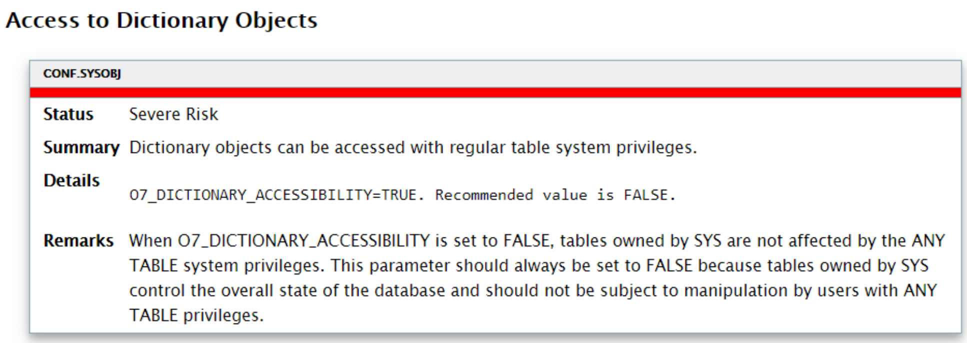


Database Configuration

Initialization Parameters for Security

Name	Value
AUDIT_FILE_DEST	/u01/app/oracle/admin/orcl12c/adump
AUDIT_SYSLOG_LEVEL	
AUDIT_SYS_OPERATIONS	FALSE
AUDIT_TRAIL	DB
COMPATIBLE	12.2.0
DBFIPS_140	FALSE
DISPATCHERS	(PROTOCOL=TCP) (SERVICE=orcl12cXDB)
ENCRYPT_NEW_TABLESPACES	CLOUD_ONLY
GLOBAL_NAMES	FALSE
LDAP_DIRECTORY_ACCESS	NONE
LDAP_DIRECTORY_SYSAUTH	no
O7_DICTIONARY_ACCESSIBILITY	TRUE
OS_AUTHENT_PREFIX	ops$
OS_ROLES	FALSE

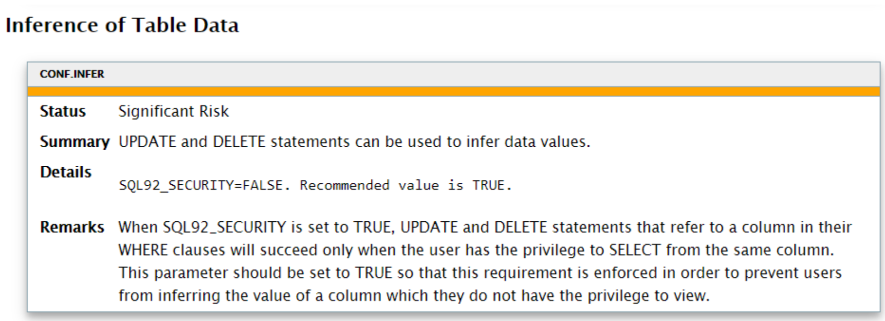
17. The next finding is marked with the status "**Severe Risk**" – **O7_DICTIONARY_ACCESSIBILITY=TRUE** – If set to TRUE this parameter will allow the **ANY TABLE** system privileges to apply to **SYS** owned tables. This parameter should always be **FALSE** as, as an example, a user with **SELECT ANY TABLE** privilege could read **SYS** owned tables.



Access to Dictionary Objects

CONF.SYSOBJ	
Status	Severe Risk
Summary	Dictionary objects can be accessed with regular table system privileges.
Details	`O7_DICTIONARY_ACCESSIBILITY=TRUE. Recommended value is FALSE.`
Remarks	When O7_DICTIONARY_ACCESSIBILITY is set to FALSE, tables owned by SYS are not affected by the ANY TABLE system privileges. This parameter should always be set to FALSE because tables owned by SYS control the overall state of the database and should not be subject to manipulation by users with ANY TABLE privileges.

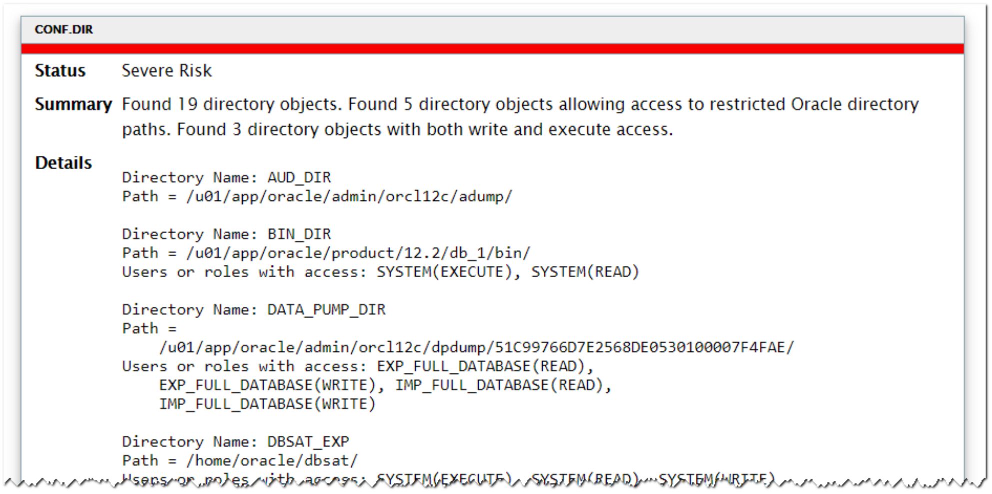
18. The next finding is also a common one – **SQL92_SECURITY=FALSE**. Please have a look at the Remarks to find out why it should be **TRUE**.


Inference of Table Data

CONF.INFER

Status	Significant Risk
Summary	UPDATE and DELETE statements can be used to infer data values.
Details	`SQL92_SECURITY=FALSE. Recommended value is TRUE.`
Remarks	When SQL92_SECURITY is set to TRUE, UPDATE and DELETE statements that refer to a column in their WHERE clauses will succeed only when the user has the privilege to SELECT from the same column. This parameter should be set to TRUE so that this requirement is enforced in order to prevent users from inferring the value of a column which they do not have the privilege to view.

19. Scroll down or search for **CONF.DIR**. It is also marked as posing a "Severe Risk".


CONF.DIR

Status Severe Risk

Summary Found 19 directory objects. Found 5 directory objects allowing access to restricted Oracle directory paths. Found 3 directory objects with both write and execute access.

Details

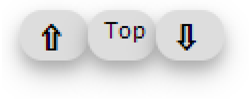
```
Directory Name: AUD_DIR
Path = /u01/app/oracle/admin/orcl12c/adump/

Directory Name: BIN_DIR
Path = /u01/app/oracle/product/12.2/db_1/bin/
Users or roles with access: SYSTEM(EXECUTE), SYSTEM(READ)

Directory Name: DATA_PUMP_DIR
Path =
    /u01/app/oracle/admin/orcl12c/dpdump/51C99766D7E2568DE0530100007F4FAE/
Users or roles with access: EXP_FULL_DATABASE(READ),
    EXP_FULL_DATABASE(WRITE), IMP_FULL_DATABASE(READ),
    IMP_FULL_DATABASE(WRITE)

Directory Name: DBSAT_EXP
Path = /home/oracle/dbsat/
Users or roles with access: SYSTEM(EXECUTE), SYSTEM(READ), SYSTEM(WRITE)
```

A special look needs to be taken into these **DIRECTORY** Objects as they allow access to the server's file system from PL/SQL code within the database.

Access to files that are used by the database kernel itself should not be permitted.

Make sure all those **DIRECTORY** objects are needed and for the ones that are, point them to other directories rather than inside `$ORACLE_HOME`, `$ORACLE_BASE`.

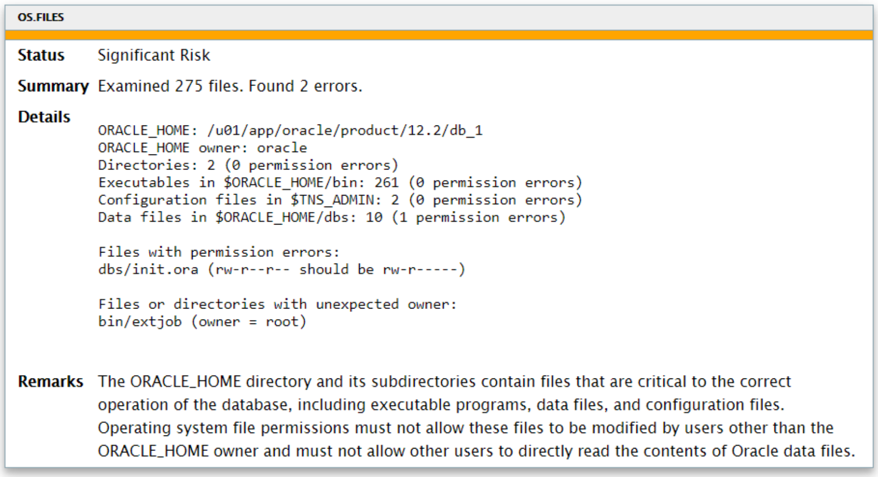
20. Click **TOP** on the navigation panel (bottom right).



Let's have a look at "**Operating System**" section and search for the finding marked as "Significant Risk" (orange line).

Click the "**Operating System**" link in the Summary table.

21. Scroll down to **OS.FILES.**



OS.FILES

Status Significant Risk

Summary Examined 275 files. Found 2 errors.

Details

```
ORACLE_HOME: /u01/app/oracle/product/12.2/db_1
ORACLE_HOME owner: oracle
Directories: 2 (0 permission errors)
Executables in $ORACLE_HOME/bin: 261 (0 permission errors)
Configuration files in $TNS_ADMIN: 2 (0 permission errors)
Data files in $ORACLE_HOME/dbs: 10 (1 permission errors)

Files with permission errors:
dbs/init.ora (rw-r--r-- should be rw-r-----)

Files or directories with unexpected owner:
bin/extjob (owner = root)
```

Remarks The ORACLE_HOME directory and its subdirectories contain files that are critical to the correct operation of the database, including executable programs, data files, and configuration files. Operating system file permissions must not allow these files to be modified by users other than the ORACLE_HOME owner and must not allow other users to directly read the contents of Oracle data files.

In this finding, DBSAT will identify operating system file permissions that are wrongly set up.
In this case, it flagged init.ora has had the wrong permission settings and bin/extjob as being owned by root.

Make sure **OS file permissions** are rightly setup to avoid having database binaries and files modified by users other that the ORACLE_HOME owner.

22. In case you still have time, you can now "**fix it or break it**".

Play around with the database and run the collector and reporter again and have a look at the results.

Some snippets that you can use:

```
sqlplus sys/oracle@orcl as sysdba
drop user scott cascade;
alter user hr identified by oracle;
revoke app_read from public;
alter user expired_user_ulk identified by oracle;
drop trigger logon_info_trig;
alter session set container=CDB$ROOT;
alter system set audit_sys_operations=true scope=spfile;
alter system set o7_dictionary_accessibility=FALSE scope=spfile;
alter system set sql92_security=true scope=spfile;
alter system set utl_file_dir='' scope=spfile;
```

```
shutdown immediate
startup
host
cd $ORACLE_HOME/dbs
chmod 640 init.ora
cd $ORACLE_HOME/bin
sudo chown oracle:oinstall extjob
```

SUMMARY

In this tutorial, we were able to play with the **Database Security Assessment Tool** and understand how it works and the immediate value it provides.

The **Top 10 findings from running Database Security Assessments** (Interview + DBSAT) at customers has proven to be:

» No Database Security Policies / Strategy in place
» No patching/patch management policy in place
» No encryption of sensitive/regulated data
» No monitoring/auditing in place
» Over-privileged accounts; No personalized accounts; NO SoD
» Weak/inexistent password policies; Weak password management
» Data sent in clear to third parties
» No OS hardening
» No sensitive data anonymization in production to DEV/TEST/Training/etc.
» Still some sample schemas in production environments out there

TRY DBSAT TODAY AT:

HTTP://WWW.ORACLE.COM/TECHNETWORK/DATABASE/SECURITY/DBSAT.HTML

TO REFER TO DBSAT ON SOCIAL MEDIA PLEASE USE:

#DBSAT

#SECURITY @ORACLESECURITY @ORACLEDATABASE

CONGRATULATIONS!

YOU HAVE SUCCESSFULLY COMPLETED THIS DBSAT TUTORIAL!

ORACLE®

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Integrated Cloud Applications & Platform Services

Database Security: Identifying Security Risks with the Database Security Assessment Tool – October November 2017
Author: Pedro Lopes