Database Security: Identifying Security Risks with the Database Security Assessment Tool
Table of Contents

TUTORIAL OVERVIEW 2

CHALLENGE 3

DBSAT OVERVIEW 4

Components of DBSAT and Flow 4

Benefits of Using DBSAT 5

LAB EXERCISE 01 6

Creating a Database User to run DBSAT 6

LAB EXERCISE 02 8

Installing DBSAT 8

LAB EXERCISE 03 10

Run DBSAT Collector 10

LAB EXERCISE 04 12

Run DBSAT Reporter 12

LAB EXERCISE 05 15

Analyze the generated report - Summary 15

LAB EXERCISE 06 16

Analyze the generated report – Findings 16

LAB EXERCISE 07 17

Analyze the generated report – Details 17

SUMMARY 30
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:

```sql
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 dbset identified by oracle;
Grant succeeded.
SQL> grant select on sys.registry$history to dbset;
Grant succeeded.
SQL> grant select_catalog_role to dbset;
Grant succeeded.
SQL> grant audit_viewer to dbset;
Grant succeeded.
SQL> grant capture_admin to dbset;
Grant succeeded.
SQL> grant select on sys.dba_users_with_defpwd to dbset;
Grant succeeded.
SQL> grant select on audsys.aud$unified to dbset;
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
   ```

   The latest unzip output line should be:

   ```
   inflating: dbsat.zip
   inflating: dbsat.bat
   inflating: dbreporter.py
   inflating: sat_analysis.py
   inflating: sat_collector.sql
   inflating: xlsxwriter/app.py
   inflating: xlsxwriter/chart_area.py
   ```

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

   ![DBSAT Collector Command]

   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:
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

   ```bash
   ./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]$ ./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
[oracle_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.
2. Let’s unzip the file to view the reports

```bash
unzip orcl_hol_report.zip
```

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

```bash
firefox orcl_hol.html &
```

Firefox should open and display the html report:
### Oracle Database Security Risk Assessment

**Highly Confidential**

#### Assessment Date & Time

<table>
<thead>
<tr>
<th>Date of Data Collection</th>
<th>Date of Report</th>
<th>Reporter Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wed Sep 20 2017 10:54:03</td>
<td>Wed Sep 20 2017 11:18:22</td>
<td>1.0.2 (October 2016) - 7409</td>
</tr>
</tbody>
</table>

#### Database Identity

<table>
<thead>
<tr>
<th>Name</th>
<th>Platform</th>
<th>Database Role</th>
<th>Log Mode</th>
<th>Created</th>
<th>Container Database</th>
<th>Container ID</th>
<th>Container Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORCL22</td>
<td>Linux x86-64</td>
<td>PRIMARY</td>
<td>NOARCH</td>
<td>19 Jun 12 17:51</td>
<td>ORCL</td>
<td>3</td>
<td>ORCL</td>
</tr>
</tbody>
</table>

#### Summary

<table>
<thead>
<tr>
<th>Section</th>
<th>Pass</th>
<th>Evaluate</th>
<th>Opportunity</th>
<th>Some Risk</th>
<th>Significant Risk</th>
<th>Severe Risk</th>
<th>Total Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Information</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>User Accounts</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Privileges and Roles</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Authorization Centre</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Data Dictionary</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Fine-Grained Access Control</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>Auditing</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>Database Configuration</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Network Configuration</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Operating System</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>5</td>
<td>33</td>
<td>16</td>
<td>18</td>
<td>6</td>
<td>4</td>
<td>75</td>
</tr>
</tbody>
</table>
LAB EXERCISE 05

Analyze the generated report - Summary

In this exercise, you will learn 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

<table>
<thead>
<tr>
<th>Date of Data Collection</th>
<th>Date of Report</th>
<th>Reporter Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wed Sep 20 2017 10:54:00</td>
<td>Wed Sep 20 2017 11:18:22</td>
<td>1.0.2 (October 2018) – 7409</td>
</tr>
</tbody>
</table>

### Database Identity

<table>
<thead>
<tr>
<th>Name</th>
<th>Platform</th>
<th>Database Role</th>
<th>Log Mode</th>
<th>Created</th>
<th>Container Database</th>
<th>Container ID</th>
<th>Container Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORC12C</td>
<td>Linux x86 64-bit</td>
<td>PRIMARY</td>
<td>NOARCHIVELOG</td>
<td>Mon Jun 12 2017 15:42:00</td>
<td>True</td>
<td>3</td>
<td>ORC1</td>
</tr>
</tbody>
</table>

### Summary

<table>
<thead>
<tr>
<th>Section</th>
<th>Pass</th>
<th>Evaluate</th>
<th>Opportunity</th>
<th>Some Risk</th>
<th>Significant Risk</th>
<th>Severe Risk</th>
<th>Total Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Information</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>User Accounts</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>Privileges and Roles</td>
<td>2</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>Authorization Control</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Data Encryption</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Fine-Grained Access Control</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Auditing</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>Database Configuration</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Network Configuration</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Operating System</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>6</strong></td>
<td><strong>31</strong></td>
<td><strong>4</strong></td>
<td><strong>19</strong></td>
<td><strong>18</strong></td>
<td><strong>4</strong></td>
<td><strong>73</strong></td>
</tr>
</tbody>
</table>

» 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.
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**

   ![Patch Check](image)

   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

**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

**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

**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

**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

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

Sample Schemas

**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

**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

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

---

Database Security: Identifying Security Risks with the Database Security Assessment Tool
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:
This provides a powerful insight into what users can 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

Status  Evaluate
Summary 75 grants of account management privileges
Details
Grants of ALTER USER, CREATE USER, DROP USER:
APEX_0G0100: ALTER USER, CREATE USER, DROP USER
DEFSYS: DBA ALTER USER, CREATE USER, DROP USER
DEFSYS: DBA DATAPUMP_IMP_FULL_DATABASE: ALTER USER
DEFSYS: DBA DATAPUMP_IMP_FULL_DATABASE: IMP_FULL_DATABASE:
ALTER USER, CREATE USER, DROP USER
DEFSYS: DBA EN_EXPRESS_ALL: ALTER USER, CREATE USER, DROP USER
DEFSYS: DBA IMP_FULL_DATABASE: ALTER USER, CREATE USER, DROP USER
HR: ALTER USER
HRST: ALTER USER
HRST: DBA ALTER USER, CREATE USER, DROP USER
HRST: DBA DATAPUMP_IMP_FULL_DATABASE: ALTER USER
HRST: DBA DATAPUMP_IMP_FULL_DATABASE: IMP_FULL_DATABASE:
ALTER USER, CREATE USER, DROP USER
HRST: DBA EN_EXPRESS_ALL: ALTER USER, CREATE USER, DROP USER
HRST: 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:
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** 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.

Privilege Analysis

**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

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

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 Table]

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.
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**

<table>
<thead>
<tr>
<th>Status</th>
<th>Significant Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>Actions of the SYS user are not audited.</td>
</tr>
<tr>
<td>Details</td>
<td>Traditional Audit: AUDIT_SYS_OPERATIONS is set to FALSE. Unified Audit policies enabled for administrators: (none)</td>
</tr>
<tr>
<td>Remarks</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

**Privilege Management Audit**

<table>
<thead>
<tr>
<th>Status</th>
<th>Significant Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>Actions related to privilege management are not sufficiently audited.</td>
</tr>
<tr>
<td>Details</td>
<td>Auditing not enabled: GRANT ANY OBJECT PRIVILEGE, GRANT ANY PRIVILEGE, GRANT ANY ROLE Traditional audit - auditing enabled: (none) Unified audit - auditing enabled: (none)</td>
</tr>
<tr>
<td>Remarks</td>
<td>Creating 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.</td>
</tr>
</tbody>
</table>

**Account Management Audit**

<table>
<thead>
<tr>
<th>Status</th>
<th>Significant Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>Actions related to account management are not sufficiently audited.</td>
</tr>
<tr>
<td>Details</td>
<td>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)</td>
</tr>
<tr>
<td>Remarks</td>
<td>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.</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUDIT_FILE_DEST</td>
<td>/u01/app/oracle/admin/orcl12c/adump</td>
</tr>
<tr>
<td>AUDIT_SYSLOG_LEVEL</td>
<td></td>
</tr>
<tr>
<td>AUDIT_SYS_OPERATIONS</td>
<td>FALSE</td>
</tr>
<tr>
<td>AUDIT_TRAIL</td>
<td>DB</td>
</tr>
<tr>
<td>COMPATIBLE</td>
<td>12.2.0</td>
</tr>
<tr>
<td>DIXP1_140</td>
<td>FALSE</td>
</tr>
<tr>
<td>DISPATCHERS</td>
<td>(PROTOCOL=TCP) (SERVICE=orcl12cXDB)</td>
</tr>
<tr>
<td>ENCRYPT_NEW_TABLESPACES</td>
<td>CLOUD_ONLY</td>
</tr>
<tr>
<td>GLOBAL_NAMES</td>
<td>FALSE</td>
</tr>
<tr>
<td>LDAP_DIRECTORY_ACCESS</td>
<td>NONE</td>
</tr>
<tr>
<td>LDAP_DIRECTORY_SYSAUTH</td>
<td>no</td>
</tr>
<tr>
<td>O7_DICTIONARY_ACCESSIBILITY</td>
<td>TRUE</td>
</tr>
<tr>
<td>OS_AUTHENT_PREFIX</td>
<td>ops$</td>
</tr>
<tr>
<td>O7_ROLES</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

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**

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

![Conf DIR Table](image)

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 the **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**.

![Database Security Assessment Tool](image)

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 than 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:

```sql
sqlplus sys/oracle@orcl as sysdba
drop user scott cascade;
drop 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=CDBSROOT;
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!