Information Systems Audit
Degree in Computer Engineering in Information Systems - Academic Year 2022/2023

PRACTICE 1: ORACLE DBD AUDIT I

Objectives

- Introduction to auditing in an Oracle database manager.
- Security aspects to evaluate in an Oracle database.
- Obtaining evidence to evaluate the degree of security of the Oracle database.

Content

1. Introduction to auditing in an Oracle database manager

Databases are the storehouse of the most valuable resource that companies have, information. Within the different existing database managers, Oracle occupies a privileged place as a world leader in this market.

From an auditing point of view, Oracle databases provide a comprehensive set of auditing tools to track user activity and identify suspicious trends, allowing us to detect security breaches. Auditing should be focused only on those events that are of interest, as auditing all elements and all actions need not be necessary and negatively affects system performance.

Within an Oracle database we can use different types of auditing:

- **Mandatory auditing**: all Oracle databases audit certain actions regardless of the configuration of the auditing options. The reason is because the database needs to log some database activities, such as connections made by privileged users.

- **Standard database auditing**: it is enabled at system level by means of the dynamic initialization parameter AUDIT_TRAIL. After enabling auditing, the objects and privileges to be audited are selected and the auditing options are configured using the AUDIT command.
  - **Value-based auditing**: extends the standard auditing, capturing not only the audited events that occur, but also the existing values before being inserted, updated or deleted. Value-based auditing is implemented through triggers on database objects (tables) that allow to keep track of accesses, value modifications and record deletions. These audits are developed by the application developers. They mainly consist of triggers. They are therefore left to the free choice of the developer, who can rely on commercial tools that facilitate this task of creating triggers.
  - **Fine-grained auditing (FGA)**: extends the standard auditing of the database, capturing not only the fact that the audited event occurred, but also the SQL statement that was executed.

- **SYSDBA (and SYSOPER) auditee**: separates responsibilities between a DBA (Data Base Administrator) and an auditor or security administrator who monitors the DBA's activities.

In addition to the auditing tools provided by Oracle, the Database auditor must check other aspects that guarantee a correct implementation of security in the Database. Within these aspects to consider, check and evaluate we can include:

- ARCHIVELOG / NOARCHIVELOG operating mode (copy or not copy redo log files)[1])

[1] *The most crucial structure for recovery operations is the redo log file, which consists of two or more preallocated files that store all changes made to the database as they occur.*

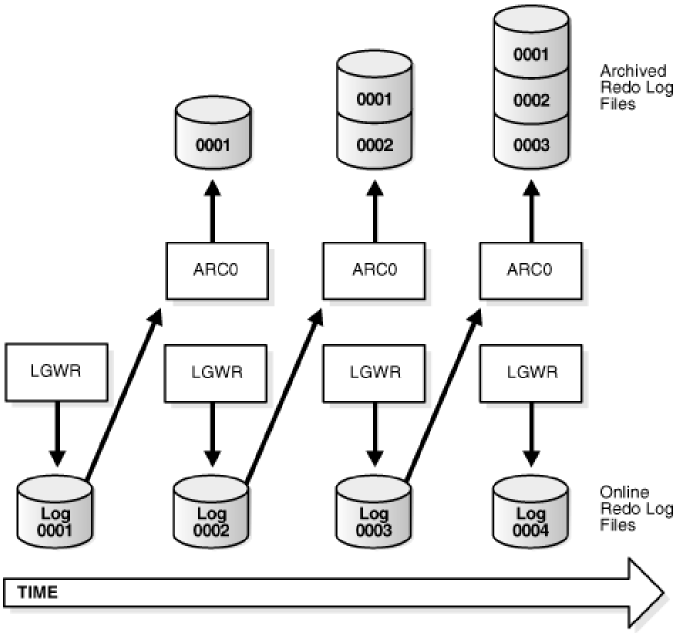
- Application of the Principle of Least Privilege
    - Protecting the data dictionary
    - Revoking unnecessary PUBLIC privileges
    - Limiting users with administrator privileges
    - Restricting remote database authentication
- Default account management
- Implementation of password security

2. ARCHIVELOG / NOARCHIVELOG operating mode

Oracle's ARCHIVELOG mode is a mechanism that protects the database against possible physical disk failures and also against unwanted deletions or modifications of the data. With this mechanism we will have the option to recover the database at a specific moment in time, that is to say, we will be able to recover the state of the database at a specific time and day.

Oracle's treatment of transactions is as follows. When users make modifications to the database, Oracle saves all the transactions in files called "online redo log files". This mechanism allows the information to be reconstructed from the last backup. The writing in these files is cyclical, so that you start writing in the first online redo log file, when it is full it goes to the second one, and so on until the last online redo log file is reached, when a background process called LGWR (Log Writer) is started, which overwrites the contents of the first online redo log file and so on.

When Oracle runs in ARCHIVELOG mode there is a background process called ARC0 that makes a copy of each online redo log file once the LGWR process finishes writing to it, saving that copy to the offline redo log files on disk. This process can slow down the database minimally and, logically, requires more disk space to hold those files.



The auditor must check if the ARCHIVELOG or NONARCHIVELOG mode of operation is active in the database. If it is not active, the evidence of attack or changes will be overwritten by a new redo.

It can be determined by performing a SQL statement to the

database: SQL> SELECT name, log_mode FROM

v$database;

3. Principle of least privilege

This principle is based on the premise that a user should only have the minimum privileges necessary to carry out the tasks assigned to him. This reduces the possibility of users accidentally or intentionally modifying or viewing data for which they do not have the respective privileges.

Actions to be taken to comply with this principle include:

- **Protect the data dictionary: prevents** users with ANY TABLE systems privilege from accessing data dictionary tables. In addition, it prevents the SYS user (root) from logging in with a role other than SYSDBA (DB administrator role). The appropriate value is indicated by parameter O7_DICTIONARY_ACCESSIBILITY:

  ```
  SQL> SELECT name, value FROM v$parameter WHERE UPPER(name)='O7_DICTIONARY_ACCESSIBILITY';
  ```

- **Revoke unnecessary PUBLIC privileges:** remove all unnecessary privileges and roles from the PUBLIC group database. The database grants execution permissions on certain packages to the PUBLIC role, allowing any user to execute those packages. They can be queried with the following SELECT:

  ```
  SQL> SELECT table_name FROM dba_tab_privs WHERE owner='SYS' AND privilege = 'EXECUTE' AND
  grantee='PUBLIC' and table_name like 'UTL%';
  ```

- **Limit users with administrator privileges:** users must have the minimum privileges necessary to carry out their tasks. Therefore, they must be controlled:
  - Permissions on System and Object privileges
  - Privileged SYS connections, such as SYSDBA and SYSOPER (DB operator role)
  - Other DBA type privileges, such as DROP ANY TABLE

    ```
    --Users with DBA role
    SQL> SELECT grantee FROM dba_role_privs WHERE granted_role = 'DBA

    --Users with SYSDBA and SYSOPER SQL
    privileges> SELECT * FROM V$PWFILE_USERS;
    ```

- **Restrict remote database authentication:** when external authentication of database users is enabled, it is delegated to the remote system. This means that the instance implicitly trusts that users have been properly authenticated on the client PC and does not request a new authentication credential. Users can connect to the database without providing a password. The operating system user name must be the same as the database user name in order to be externally authenticated. The value is specified by the REMOTE_OS_AUTHENT parameter:

  ```
  SQL> SELECT name, value FROM v$parameter WHERE UPPER(name) = 'REMOTE_OS_AUTHENT';
  ```

4. Default account management

You should check that all accounts that are created at the time of installing the database are locked except:

– SYS
– SYSTEM
– SYSMAN
– DBSNMP

All other accounts are created automatically on installation with default passwords. Leaving them unlocked may allow unauthorized people to log into the database. To obtain a list of these accounts and their status (assuming the database installation date is 01/31/2006):

```
SQL> SELECT * FROM dba_users WHERE TRUNC(created)= '01/31/2006';
```

5. Implementation of password security

In Oracle, password management is implemented through user profiles. These profiles allow you to specify certain security features such as:

- **Account locking:** Enables automatic account locking when the user fails a specified number of attempts to log in to the system.
- **Password aging and expiration:** Enables the user password to have an activation time or duration, after which the password expires and must be changed.
- **Password history:** Checks the new passwords and verifies that they are not reused for a period of time or until a specific number of new passwords is reached.
- **Password complexity verification:** Checks the complexity of the password and verifies that it meets certain characteristics. The check allows passwords to be sufficiently complex to provide greater protection to the system.

Parameters to be reviewed:

- FAILED_LOGIN_ATTEMPTS: Number of failed connection attempts before the account is locked.
- PASSWORD_LOCK_TIME: Number of days the account is locked after the number of failed attempts has been exceeded.
- PASSWORD_LIFE_TIME: Password validity time in days; after that, the password expires.
- PASSWORD_GRACE_TIME: Grace period in days to change the password after the first login attempt and after the password has expired.
- PASSWORD_REUSE_TIME: Specifies that the user cannot reuse a password until the specified number of days.
- PASSWORD_REUSE_MAX: Specifies the number of passwords modified before the current password can be reused.
- PASSWORD_VERIFY_FUNCTION: A PL/SQL function that ensures the complexity of the password is checked before it is assigned.

To obtain information about the profiles and their characteristics:

```
--User profiles
SQL> SELECT USERNAME, PROFILE FROM dba_users ORDER BY 1;

--Characteristics of each profile
SQL> SELECT * FROM dba_profiles ORDER BY 1;
```

DEFAULT values mean that Oracle uses the values defined in the DEFAULT profile. By default, those values are defined as UNLIMITED in that profile.

Bibliography

ORACLE 11GR2 Administration Workshop I Oracle University.
ORACLE DATABASE ONLINE DOCUMENTATION: https://docs.oracle.com/cd/E11882_01/index.htm
ORACLE DATABASE REFERENCE: https://docs.oracle.com/en/database/

Problems

P1. Create an Excel document where to compile all the information from the different sections. This document will gather all the evidence obtained and will serve as a basis for formulating our opinion and making recommendations in a supposed audit report. Among the information to be included is:

- Auditor's name
- Audited company data
- Scope of the audit
- Date of audit
- Control of revisions
- Database name
- List of the evidences obtained The

credentials to access the DB are:

Ubuntu:
OSBOXES: Olavide2019
root: oracle
oracle:oracleO

ORACLE SQL DEVELOPER
host: ubuntu IP port:
1521
SID: XE

user sys as sysdba
(pass Oracle_4U)

user: auditor
pass: auditor

login: hr
pass: hr

If we access directly from Ubuntu we open a black window:
C:> sqlplus
login: sys as sysdba
pass: Oracle_4U

login:auditor
pass: auditor

login: hr
pass: hr

Estimated time: 25 minutes

P2. Enter the SQL Developer application using the AUDITOR user. Once inside the application consult the name and operation mode of the database. Store the information in Excel.

Estimated time: 5 minutes

P3. From SQL Developer check the protection of the data dictionary.

Estimated time: 5 minutes

P4. From SQL Developer, extract a list of the packages that can be executed by any user of the system. Store the information in Excel.

Estimated time: 10 minutes

P5. From SQL Developer, extract a list of users with DBA role and SYSDBA and SYSOPER privileges. Store the information in Excel.

Estimated time: 5 minutes

P6. From SQL Developer, check if remote database authentication is enabled. Store the information in Excel.

Estimated time: 5 minutes

P7. From SQL Developer, check if users that were created by default in the database installation are unlocked. Store the information in Excel.

Estimated time: 10 minutes

P8. From SQL Developer, list the different profiles that are created in the database and their characteristics in terms of the security of the passwords of the users assigned to those profiles. Store the information in Excel.

Estimated time: 15 minutes