Oracle Database User Security And Banking Application To The Sector.

Ulker Israfilzade

Abstract

The project's goals are to evaluate ODB's user security, deal with password protection difficulties, and enhance user security in Azerbaijan's banking sector.

The project's objective is to prevent safety problems for database users from evolving as a result of information's rising role in society. One of the databases used to store information is Oracle Database (ODB), and similarly to all ODB sections, user security and privacy should be provided highest priority. In the banking sector, protecting confidential information is of the utmost importance. This includes sensitive data such as customer account information, transaction details, and financial records. ODB administrators in the banking sector must be aware of the potential risks associated with using Oracle, such as security vulnerabilities and data breaches.

Keywords: privilege, safety issues, password mechanism

ODB privacy includes the provision of user security as well as network security. In addition to network security and user security, information security in ODB must be ensured in order to make the database secure. To guarantee the safety of network authentication, ODB encrypts passwords as they are sent.Because databases require certain authentication mechanisms, database administrators do specific database tasks. In order to restrict the DBA's control over all tables and views, ODB makes use of Oracle Database Vault (ODBV) technology. ODBV allows for the restriction of SQL commands that might compromise the application's and the database's security and availability.

Constructing a database

To access a database, a user must start a database application and establish a connection to the database instance using a valid user name defined in the database. For database users' security configurations, Oracle Database offers a variety of options. While creating user accounts, restrictions can be specified. The range of system resources that are made available to each user, for which profiles should be described, can also be restricted as part of a user's security domain. A user's profile is a collection of traits that relate to them and serves as a single point of contact for any number of people who share those traits.

Oracle Database provides a range of configuration options for authentication to users and database managers. For instance, you can authenticate users using the operating system, the network, and the security level of the database. Authentication is the process of confirming the identity of a person, device, or other entity who needs to access data, services, or applications. The validity of that identification is established, creating a base of confidence for subsequent interactions. Accountability is made easier by authentication by providing the link between access, activities, and specific individuals.

If you want to set Oracle Database to authenticate users by encrypting their passwords, it includes a number of built-in password safeguards designed to keep your users' credentials secure. Password reuse time and failed login attempts are two built-in password protection options. Password life time specifies how long to wait before locking a user ID if the password has not changed, and failed login attempts specifies how many failed attempts must occur before the user's username is locked. While connecting to a network, Oracle Database automatically and transparently encrypts passwords and sends them across it using the Advanced Encryption Standard (AES). In order to prevent hackers who try to break into the system by guessing passwords, the verify function 11g password verification function from Oracle Database makes sure that newly created or modified passwords are sufficiently complex. The scope of the entity's appropriate users and actions can then be expanded or decreased following authentication through approval procedures.

The definition of authorization is included in Implementing Privilege and Role Permissions. The ability to run a specific sort of SQL statement, access an object owned by another user, run a PL/SQL package, and do other types of collective is known as a user capability. The various degrees of privileges are set by ODB. By giving new users access to legal powers, database managers, for instance, have strengthened the financial system when compared to other database users.

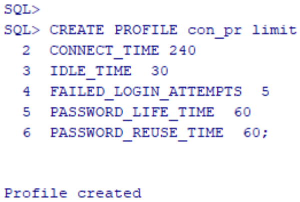
The attributes of the user profile are controlled by its login settings, therefore the post-login privileges of the user must be defined via grants. Privileges don't need to be set up because Oracle has them available. Oracle roles can be created by users (often administrators) to group rights or other tasks. They should give the public access to various tasks or rights as a way to make it more straightforward.

System rights enable the grant to execute common database administrator tasks while restricting them to just authorized users. Object privileges are the rights that are attached to any form of object. User roles are combinations of rights and roles that can be granted to users at the same time as well as taken away from them. Should first activate a role for that user before allowing them to utilize it. Giving away too many excessive powers could put security at risk. For example, you should never grant the SYSDBA or SYSOPER administrator ability to users who don't perform administrative duties.

Practical, essential strategies for user security incidents

In the banking industry, helpful recommendations are provided for the universal application of user security measures and ODBV techniques.

To configure a set of resource limitations and password requirements that restrict a user's access to database and instance resources, a profile must first be built. The seconds that are listed comprise encryption and authentication, and security is guaranteed by carefully guarding these parameters.

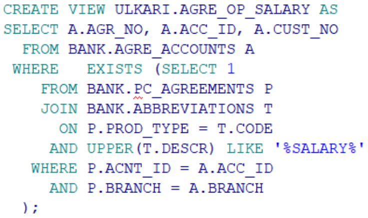


```
SQL>
SQL> CREATE PROFILE con_pr limit
  2  CONNECT_TIME 240
  3  IDLE_TIME  30
  4  FAILED_LOGIN_ATTEMPTS  5
  5  PASSWORD_LIFE_TIME  60
  6  PASSWORD_REUSE_TIME  60;


Profile created
```

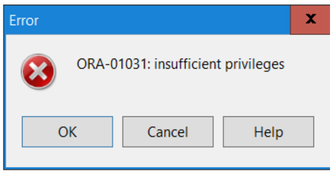
For security reasons, make sure to use a distinct profile for every database and refrain from using the default profile. Put this profile into the user management parameters after that:

```
SQL> alter user ulkari profile con_pr;
```

Then, using grants, the user's object privileges must be provided. There will be a problem if the user doesn't have authority to create the thing:



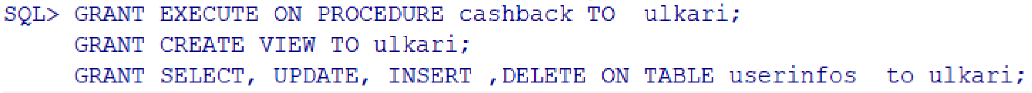
```
CREATE VIEW ULKARI.AGRE_OP_SALARY AS
SELECT A.AGR_NO, A.ACC_ID, A.CUST_NO
  FROM BANK.AGRE_ACCOUNTS A
 WHERE   EXISTS (SELECT 1
    FROM BANK.PC_AGREEMENTS P
    JOIN BANK.ABBREVIATIONS T
      ON P.PROD_TYPE = T.CODE
     AND UPPER(T.DESCR) LIKE '%SALARY%'
   WHERE P.ACNT_ID = A.ACC_ID
     AND P.BRANCH = A.BRANCH
  );
```



Error

ORA-01031: insufficient privileges

Because of this, "execute on" permission is needed when compiling system objects in a production environment that depend on security events. The "create" permission can be used to produce something new, and the "select, insert, update, delete" grants can be applied to user objects to modify their data:



```
SQL> GRANT EXECUTE ON PROCEDURE cashback TO  ulkari;
     GRANT CREATE VIEW TO ulkari;
     GRANT SELECT, UPDATE, INSERT ,DELETE ON TABLE userinfos  to ulkari;
```

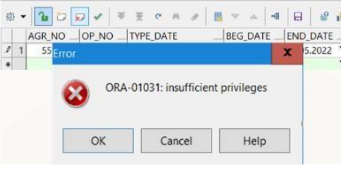
After that, the user will supply the view object in upper permission:

```
SQL> VIEW CREATED;
```

Grants must be revoked if any of the following conditions exist:

```
SQL> REVOKE DELETE ON userinfos FROM ulkari;
     REVOKE ALL ON userinfos FROM ulkari;
```

The user encounters an error after rescinding a command:



ORA-01031: insufficient privileges

As a result, roles must be provided for database administrators in order to save time and effort:

```
 SQL>  CREATE ROLE dev_role IDENTIFIED BY dev12c;

 Role created
```

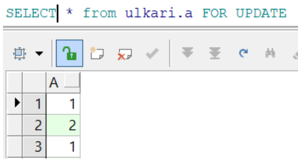
And award grants to “dev_role”;

```
SQL> GRANT select, insert, update, delete ON ulkari.a TO dev_role;

Grant succeeded
```

The user has full access to the position:

```
SQL> GRANT dev_role TO ulkari;

Grant succeeded
```

If an user tries to remove a record, the following will happen:



```
SELECT * from ulkari.a FOR UPDATE
```

	A
1	1
2	2
3	1

Conclusion

This session provides an organized introduction to Oracle user security and security vault techniques. In conclusion, creative research on roles and privileges in database modules, approaches to more effectively handle Oracle user security in the banking industry. In the financial sector, ODB user security is a more secure and authentic approach to database management. At the end of the workshop article, we looked at an example of creating a profile for using a password mechanism to ensure security for each user, which is an important step in resolving security issues.

References

[1] Effective Oracle Database 10g Security by Design 1st Edition by David Knox (Author), mcgraw-Hill (Author)
[2] Oracle Database SQL Exam Guide 2018 by mcgraw-Hill Education
[3] Oracle Database 12c Security by Scott Gaetjen, David Knox, William Maroulis Publisher Oracle Press
[4] Oracle Database Vault July 19, 2019 by Donald K. Burleson
[5] Oracle® Database Vault Administrator's Guide November 2022 by Patricia Huey
[6] Oracle PL/SQL programming by Steven Feuerstein with Bill Pribyl
[7] Oracle Privacy Security Auditing Book by Donald K. Burleson
[8] Oracle Security by Released October 2018 Publisher O’Reilly Media, Inc.
[9] Oracle Security Step-by-Step (Version 2.0) by Pete Finnigan
[10] Practical Oracle Security: Your Unauthorized Guide to Relational Database Security 1st Edition by Josh Shaul (Author), Aaron Ingram (Author)