Database Security

CO527 : Advanced Database Systems
Lab 05

Kanchana Jayasinghe
kc43224jayasinghe@eng.pdn.ac.lk

Outline

- What is Database Security
- Types of Security
- Some Database Security Controls
- Why Database Security important ?
- Privileges and Roles
- SQL GRANT command
- SQL REVOKE command
- SQL Injection Attacks

What is Database Security ?

Database security is the technique that protects and secures the database against intentional or accidental threats.

Database security encompasses a range of security controls designed to protect the Database Management System (DBMS).

We consider database security about the following situations:

- Theft and fraudulent.
- Loss of confidentiality or secrecy.
- Loss of data privacy.
- Loss of data integrity.
- Loss of availability of data.

Types of Security

- The types of database security measures your business/system should use include protecting the underlying infrastructure that houses the database such as the network and servers), securely configuring the DBMS, and the access to the data itself.
- Database security encompasses multiple controls, including system hardening, access, DBMS configuration, and security monitoring. These different security controls help to manage the circumventing of security protocols.

Some Database Security Controls

- System hardening and monitoring
- DBMS configuration
- Authentication
- Access
- Database auditing
- Backups
- Encryption
- Application security

Why is Database Security important ?

Database security can guard against a compromise of your database, which can lead to financial loss, reputation damage, consumer confidence disintegration, brand erosion, and non-compliance of government and industry regulation.

Database security safeguards can help protect your system from:

- Deployment failure
- Excessive privileges
- Privilege abuse
- Platform vulnerabilities
- Unmanaged sensitive data
- Backup data exposure
- Weak authentication
- Database injection attacks

Privileges and Roles

Privileges

- Privileges defines the access rights provided to a user on a database object. There are two types of privileges.
  - ➢ System privileges
  - ➢ Object privileges

1. System privileges

This allows the user to CREATE, ALTER, or DROP database objects. Few CREATE system privileges are listed below:

System Privileges	Description
CREATE object	Allows users to create the specified object in their own schema.
CREATE ANY object	Allows users to create the specified object in any schema.

** The above rules also apply for ALTER and DROP system privileges.

2. Object privileges

This allows the user to EXECUTE, SELECT, INSERT, UPDATE, or DELETE data from database objects to which the privileges apply. Few object privileges are listed below:

Object Privileges	Description
INSERT	Allows users to insert rows into a table within the schema.
SELECT	Allows users to select data from a database object within the schema.
UPDATE	Allows user to update data in a table within the schema.
EXECUTE	Allows user to execute a stored procedure or a function within the schema.
DELETE	Allows a user to delete rows from tables within the schema
REFERENCES	Allows a user to set up references to primary keys within the schema
TRIGGER	Allows a user to create triggers on tables within the schema

Roles

- Roles are a collection of privileges or access rights.
- When there are many users in a database it becomes difficult to grant or revoke privileges to users. Therefore, if you define roles, you can grant or revoke privileges to users, thereby automatically granting or revoking privileges.
- You can either create Roles or use the system roles pre-defined by oracle.

Some of the privileges granted to the system roles are as given below:

System Role	Privileges Granted to the Role
CONNECT	CREATE TABLE, CREATE VIEW, CREATE SYNONYM, CREATE SEQUENCE, CREATE SESSION etc.
RESOURCE	CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER etc. The primary usage of the RESOURCE role is to restrict access to database objects.

GRANT & REVOKE

SQL GRANT command

- Used to provide access or privileges on the database objects to the users.

  Syntax :

  GRANT privilege_name
  ON object_name
  TO {user_name | PUBLIC | role_name}
  [WITH GRANT OPTION];

Example :

```
GRANT SELECT
ON employee
TO user1;
```

- This command grants a SELECT permission on employee table to user1.
- You should use the WITH GRANT option carefully because for example if you GRANT SELECT privilege on employee table to user1 using the WITH GRANT option, then user1 can GRANT SELECT privilege on employee table to another user, such as user2 etc.
- Later, if you REVOKE the SELECT privilege on employee from user1, still user2 will have SELECT privilege on employee table.

SQL REVOKE command

- This command removes user access rights or privileges to the database objects.

  Syntax :

  REVOKE privilege_name
  ON object_name
  FROM {user_name | PUBLIC | role_name} ;

Example :

```
REVOKE SELECT
ON employee
FROM user1;
```

- This command will REVOKE a SELECT privilege on employee table from user1.
- When you REVOKE SELECT privilege on a table from a user, the user will not be able to SELECT data from that table anymore.
- However, if the user has received SELECT privileges on that table from more than one users, he/she can SELECT from that table until everyone who granted the permission revoked it.
- You cannot REVOKE privileges if they were not initially granted by you.

SQL Injection Attacks

References :

- https://portswigger.net/web-security/sql-injection
- https://www.imperva.com/learn/application-security/sql-injection-sqli/

Summary

- What is Database Security
- Types of Security
- Some Database Security Controls
- Why Database Security important ?
- Privileges and Roles
- SQL GRANT command
- SQL REVOKE command
- SQL Injection Attacks