DB2 - DATABASE SECURITY

This chapter describes database security.

Introduction

DB2 database and functions can be managed by two different modes of security controls:

1. Authentication
2. Authorization

Authentication

Authentication is the process of confirming that a user logs in only in accordance with the rights to perform the activities he is authorized to perform. User authentication can be performed at operating system level or database level itself. By using authentication tools for biometrics such as retina and figure prints are in use to keep the database from hackers or malicious users.

The database security can be managed from outside the db2 database system. Here are some type of security authentication process:

- Based on Operating System authentications.
- Lightweight Directory Access Protocol *LDAP*

For DB2, the security service is a part of operating system as a separate product. For Authentication, it requires two different credentials, those are userid or username, and password.

Authorization

You can access the DB2 Database and its functionality within the DB2 database system, which is managed by the DB2 Database manager. Authorization is a process managed by the DB2 Database manager. The manager obtains information about the current authenticated user, that indicates which database operation the user can perform or access.

Here are different ways of permissions available for authorization:

Primary permission: Grants the authorization ID directly.

Secondary permission: Grants to the groups and roles if the user is a member

Public permission: Grants to all users publicly.

Context-sensitive permission: Grants to the trusted context role.

Authorization can be given to users based on the categories below:

- System-level authorization
- System administrator [SYSADM]
- System Control [SYSCTRL]
- System maintenance [SYSMAINT]
- System monitor [SYSMON]

Authorities provide of control over instance-level functionality. Authority provide to group privileges, to control maintenance and authority operations. For instance, database and database objects.

- Database-level authorization
- Security Administrator [SECADM]

- Database Administrator [DBADM]
- Access Control [ACCESSCTRL]
- Data access [DATAACCESS]
- SQL administrator. [SQLADM]
- Workload management administrator [WLMADM]
- Explain [EXPLAIN]

Authorities provide controls within the database. Other authorities for database include with LDAD and CONNECT.

- **Object-Level Authorization**: Object-Level authorization involves verifying privileges when an operation is performed on an object.
- **Content-based Authorization**: User can have read and write access to individual rows and columns on a particular table using Label-based access Control [LBAC].

DB2 tables and configuration files are used to record the permissions associated with authorization names. When a user tries to access the data, the recorded permissions verify the following permissions:

- Authorization name of the user
- Which group belongs to the user
- Which roles are granted directly to the user or indirectly to a group
- Permissions acquired through a trusted context.

While working with the SQL statements, the DB2 authorization model considers the combination of the following permissions:

- Permissions granted to the primary authorization ID associated with the SQL statements.
- Secondary authorization IDs associated with the SQL statements.
- Granted to PUBLIC
- Granted to the trusted context role.

Instance level authorities

Let us discuss some instance related authorities.

System administration authority *SYSADM*

It is highest level administrative authority at the instance-level. Users with SYSADM authority can execute some databases and database manager commands within the instance. Users with SYSADM authority can perform the following operations:

- Upgrade a Database
- Restore a Database
- Update Database manager configuration file.

System control authority *SYSCTRL*

It is the highest level in System control authority. It provides to perform maintenance and utility operations against the database manager instance and its databases. These operations can affect system resources, but they do not allow direct access to data in the database.

Users with SYSCTRL authority can perform the following actions:

- Updating the database, Node, or Distributed Connect Service *DCS* directory
- Forcing users off the system-level

- Creating or Dropping a database-level
- Creating, altering, or dropping a table space
- Using any table space
- Restoring Database

System maintenance authority *SYSMAINT*

It is a second level of system control authority. It provides to perform maintenance and utility operations against the database manager instance and its databases. These operations affect the system resources without allowing direct access to data in the database. This authority is designed for users to maintain databases within a database manager instance that contains sensitive data.

Only Users with SYSMAINT or higher level system authorities can perform the following tasks:

- Taking backup
- Restoring the backup
- Roll forward recovery
- Starting or stopping instance
- Restoring tablespaces
- Executing db2trc command
- Taking system monitor snapshots in case of an Instance level user or a database level user.

A user with SYSMAINT can perform the following tasks:

- Query the state of a tablespace
- Updating log history files
- Reorganizing of tables
- Using RUNSTATS *Collectioncatalogstatistics*

System monitor authority *SYSMON*

With this authority, the user can monitor or take snapshots of database manager instance or its database. SYSMON authority enables the user to run the following tasks:

- GET DATABASE MANAGER MONITOR SWITCHES
- GET MONITOR SWITCHES
- GET SNAPSHOT
- LIST
    - LIST ACTIVE DATABASES
    - LIST APPLICATIONS
    - LIST DATABASE PARTITION GROUPS
    - LIST DCS APPLICATIONS
    - LIST PACKAGES
    - LIST TABLES
    - LIST TABLESPACE CONTAINERS
    - LIST TABLESPACES
    - LIST UTITLITIES
- RESET MONITOR
- UPDATE MONITOR SWITCHES

Database authorities

Each database authority holds the authorization ID to perform some action on the database. These database authorities are different from privileges. Here is the list of some database authorities:

ACCESSCTRL: allows to grant and revoke all object privileges and database authorities.

BINDADD: Allows to create a new package in the database.

CONNECT: Allows to connect to the database.

CREATETAB: Allows to create new tables in the database.

CREATE_EXTERNAL_ROUTINE: Allows to create a procedure to be used by applications and the users of the databases.

DATAACCESS: Allows to access data stored in the database tables.

DBADM: Act as a database administrator. It gives all other database authorities except ACCESSCTRL, DATAACCESS, and SECADM.

EXPLAIN: Allows to explain query plans without requiring them to hold the privileges to access the data in the tables.

IMPLICIT_SCHEMA: Allows a user to create a schema implicitly by creating an object using a CREATE statement.

LOAD: Allows to load data into table.

QUIESCE_CONNECT: Allows to access the database while it is quiesce $temporarilydisabled$.

SECADM: Allows to act as a security administrator for the database.

SQLADM: Allows to monitor and tune SQL statements.

WLMADM: Allows to act as a workload administrator

Privileges

SETSESSIONUSER

Authorization ID privileges involve actions on authorization IDs. There is only one privilege, called the SETSESSIONUSER privilege. It can be granted to user or a group and it allows to session user to switch identities to any of the authorization IDs on which the privileges are granted. This privilege is granted by user SECADM authority.

Schema privileges

This privileges involve actions on schema in the database. The owner of the schema has all the permissions to manipulate the schema objects like tables, views, indexes, packages, data types, functions, triggers, procedures and aliases. A user, a group, a role, or PUBLIC can be granted any user of the following privileges:

- **CREATEIN**: allows to create objects within the schema
- **ALTERIN**: allows to modify objects within the schema.

DROPIN

This allows to delete the objects within the schema.

Tablespace privileges

These privileges involve actions on the tablespaces in the database. User can be granted the USE privilege for the tablespaces. The privileges then allow them to create tables within tablespaces. The privilege owner can grant the USE privilege with the command WITH GRANT OPTION on the tablespace when tablespace is created. And SECADM or ACCESSCTRL authorities have the permissions to USE privileges on the tablespace.

Table and view privileges

The user must have CONNECT authority on the database to be able to use table and view privileges. The privileges for tables and views are as given below:

CONTROL

It provides all the privileges for a table or a view including drop and grant, revoke individual table privileges to the user.

ALTER

It allows user to modify a table.

DELETE

It allows the user to delete rows from the table or view.

INDEX

It allows the user to insert a row into table or view. It can also run import utility.

REFERENCES

It allows the users to create and drop a foreign key.

SELECT

It allows the user to retrieve rows from a table or view.

UPDATE

It allows the user to change entries in a table, view.

Package privileges

User must have CONNECT authority to the database. Package is a database object that contains the information of database manager to access data in the most efficient way for a particular application.

CONTROL

It provides the user with privileges of rebinding, dropping or executing packages. A user with this privileges is granted to BIND and EXECUTE privileges.

BIND

It allows the user to bind or rebind that package.

EXECUTE

Allows to execute a package.

Index privileges

This privilege automatically receives CONTROL privilege on the index.

Sequence privileges

Sequence automatically receives the USAGE and ALTER privileges on the sequence.

Routine privileges

It involves the action of routines such as functions, procedures, and methods within a database.