SECURITY MECHANISM IN SQL SERVER

SQL SERVER supports four types of security levels

They are

1. Windows level security
2. SQL Server level security
3. Database level security
4. Schema level security

Note: To know security mechanism in SQL Server, it is required to know some terminologies.

- **Principals** are that require access to SQL server or database, to which we grant permissions. Eg: logins, roles etc.

- **Securables** are items we secure or ones on which we grant permissions to principals. Eg: database, schemas, table etc.

- **Authentication** is the process of logging on to SQL Server by which a principal requests access by submitting credentials that the server evaluates. Authentication establishes the identity of the user or process being authenticated. **Authentication means “who you are”**.

- **Authorization** is the process of determining which securable resources a principal can access, and which operations are allowed for those resources. **Authorization means “what you can do”**.
The below figure gives you details about the principals and securables.

Windows-level principals
- Windows Domain Login
- Windows Local Login

SQL Server-level principals
- SQL Server Login
- Server Role

Database-level principals
- Database User
- Database Role
- Application Role
1. Windows Level Security

Windows-level principals

✓ Windows Local Login: Create one user account at windows level and login into user account and enters into sql server. We can create user accounts by using GUI
  In GUI mode: start-->application programs → controlpanel->useraccounts.
✓ Windows Group Login: Create one group account at windows level and add two or more user accounts to that group and login into group user and enters into sql server.
  In GUI mode: start-->application programs → controlpanel->useraccounts.

2. SQL Server Level Security

SQL Server-level principals

The SQL Server sa Login
The SQL Server sa log in is a server-level principal. By default, it is created when an instance is installed.

SQL Server Login
Create SQL server login account with login-name and password and login into SQL server. We can create the sql login accounts in two ways (GUI & CUI).
✓ GUI: explore instance and goto->login->click on mouse right click and select new login with name and password.
✓ CUI: open query window and type the following command:
  
  Create login <login-name> with password='xxxxx'

Server Level Roles
✓ Fixed server roles are provided for convenience and backward compatibility. Assign more specific permissions whenever possible.

✓ SQL Server provides nine fixed server roles. The permissions that are granted to the fixed server roles cannot be changed. Beginning with SQL Server 2012, you can create user-defined server roles and add server-level permissions to the user-defined server roles.

Fixed Server-Level Roles
The following table shows the fixed server-level roles and their capabilities.

<table>
<thead>
<tr>
<th>Fixed Server role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysadmin</td>
<td>Members of the sysadmin fixed server role can perform any activity in the server.</td>
</tr>
<tr>
<td>bulkadmin</td>
<td>Members of the bulkadmin fixed server role can run the BULK INSERT statement.</td>
</tr>
<tr>
<td>dbcreatore</td>
<td>Members of the dbcreatore fixed server role can create, alter, drop, and restore any database.</td>
</tr>
</tbody>
</table>
### diskadmin
The diskadmin fixed server role is used for managing disk files.

### serveradmin
Members of the serveradmin fixed server role can change server-wide configuration options and shut down the server.

### securityadmin
Members of the securityadmin fixed server role manage logins and their properties. They can GRANT, DENY, and REVOKE server-level permissions. They can also GRANT, DENY, and REVOKE database-level permissions if they have access to a database. Additionally, they can reset passwords for SQL Server logins.

### processadmin
Members of the processadmin fixed server role can end processes that are running in an instance of SQL Server. To kill the server processes.

### setupadmin
Members of the setupadmin fixed server role can add and remove linked servers by using Transact-SQL statements. (sysadmin membership is needed when using Management Studio.)

### public
Every SQL Server login belongs to the public server role. When a server principal has not been granted or denied specific permissions on a securable object, the user inherits the permissions granted to public on that object. Only assign public permissions on any object when you want the object to be available to all users. You cannot change membership in public.

---

Note: How to know the default permissions of fixed server-role using the following command.

```sql
SP_SRVROLEPERMISSION <fixed server-role>
```

Eg: SP_SRVROLE PERMISSION bulkadmin

---

**Create Customized Server Roles or User defined Server role**

- The customized server role functionality is also just like a fixed server role.
- We can give server role permissions to logins using customized server roles.
- Open the SSMS -> login into SQL Server account -> Security -> Server Roles- >click on mouse right button -> create new role and give name add to any login account and click ok button.

---

### 3. Database Level Security

- To easily manage the permissions in your databases, SQL Server provides several roles which are security principals that group other principals.
- There are two types of database-level roles in SQL Server: **fixed database roles** that are predefined in the database and **flexible database roles** that you can create.
- Fixed database roles are defined at the database level and exist in each database. Members of the db_owner database role can manage fixed database role membership. There are also some special-purpose fixed database roles in the msdb database.
**Public Database Role**

Every database user belongs to the public database role. When a user has not been granted or denied specific permissions on a securable, the user inherits the permissions granted to public on that securable.

**INFORMATION_SCHEMA and sys**

Every database includes two entities that appear as users in catalog views: INFORMATION_SCHEMA and sys. These entities are required by SQL Server. They are not principals, and they cannot be modified or dropped.

The following are various fixed database level roles:

<table>
<thead>
<tr>
<th>Fixed database role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>db_owner</td>
<td>Members of the db_owner fixed database role can perform all configuration and maintenance activities on the database, and can also drop the database.</td>
</tr>
<tr>
<td>db_securityadmin</td>
<td>Members of the db_securityadmin fixed database role can modify role membership and manage permissions. Adding principals to this role could enable unintended privilege escalation.</td>
</tr>
<tr>
<td>db_accessadmin</td>
<td>Members of the db_accessadmin fixed database role can add or remove access to the database for Windows logins, Windows groups, and SQL Server logins.</td>
</tr>
<tr>
<td>db_backupoperator</td>
<td>Members of the db_backupoperator fixed database role can back up the database.</td>
</tr>
<tr>
<td>db_ddladmin</td>
<td>Members of the db_ddladmin fixed database role can run any Data Definition Language (DDL) command in a database.</td>
</tr>
<tr>
<td>db_datawriter</td>
<td>Members of the db_datawriter fixed database role can add, delete, or change data in all user tables.</td>
</tr>
<tr>
<td>db_datareader</td>
<td>Members of the db_datareader fixed database role can read all data from all user tables.</td>
</tr>
<tr>
<td>db_denydatawriter</td>
<td>Members of the db_denydatawriter fixed database role cannot add, modify, or delete any data in the user tables within a database.</td>
</tr>
<tr>
<td>db_denydatareader</td>
<td>Members of the db_denydatareader fixed database role cannot read any data in the user tables within a database.</td>
</tr>
</tbody>
</table>