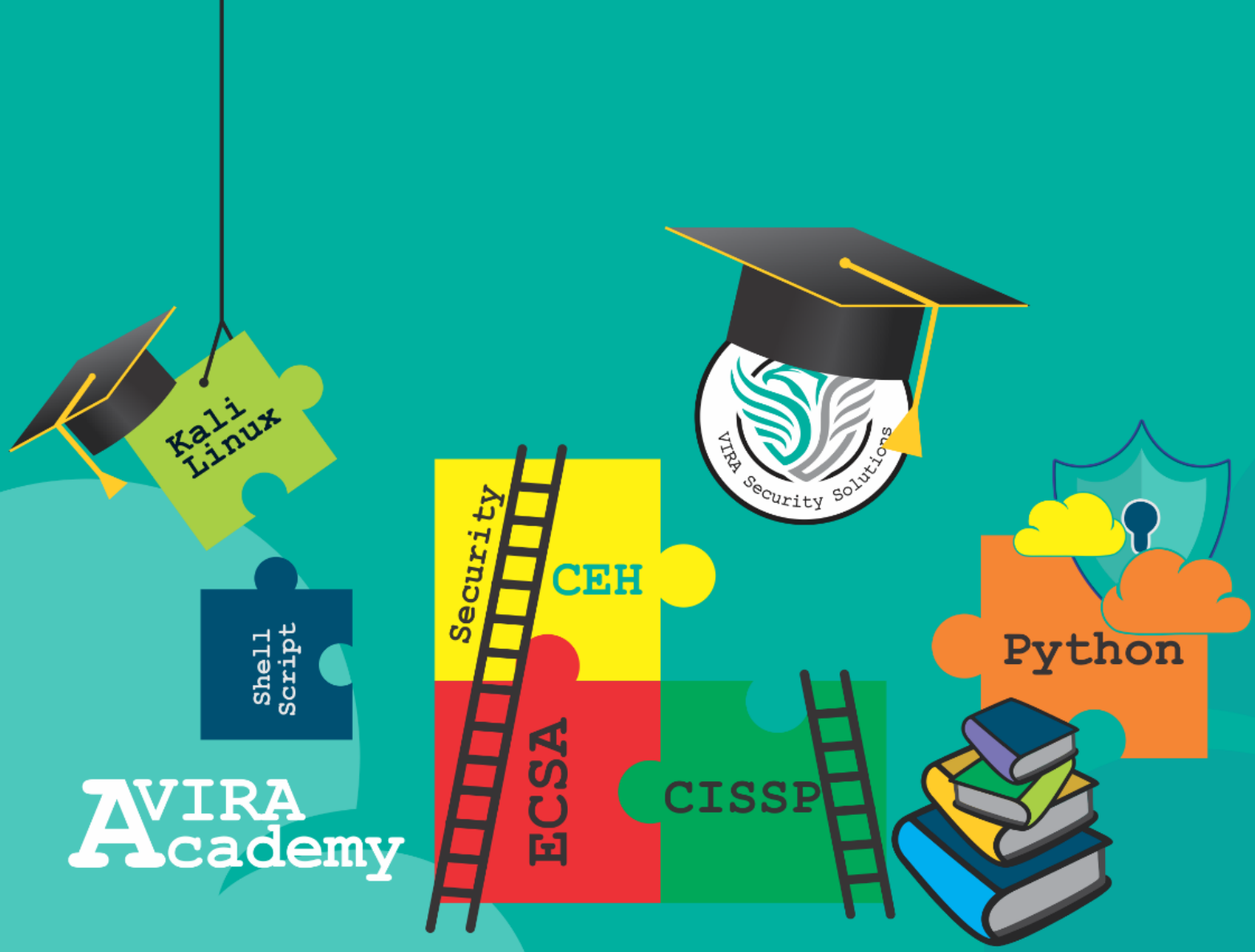

VIRA
Security
Solutions


راهکارهای امن ویرا
شماره ثبت: ۵۴۰۱۳۹
VIRA Security Solutions


Kali Linux
Shell Script
VIRA Academy
Security
CEH
ECSA
CISSP
VIRA Security Solutions
Python

DBA

SQL Server Administration

Part2

Mohammad Reza Gerami
mrgerami@aut.ac.ir
gerami@virasec.ir

Database Users

Database Users

- A database user is the account that has rights to objects inside a database
- The user is mapped to a login, or optionally, can be a user with its own password
- A user mapping must exist for the login to access data
- By default, users have no rights to access any objects inside a database

Database Users

- Lab Demo

sa is a powerful user in your SQL Server system, you must set a strong password for this users and be careful when you use it.

Built-in Database Roles

- The database roles included in every database
- The privileges of these roles
- Adding users and rights to built-in database roles

Built-in Database Roles

- Just as SQL Server includes prebuilt roles at the server level, each user database has a set of roles included
- These are roles with specific permission preassigned, and updated as objects are created and dropped
- A user can be a member of multiple roles

Built-in Database Roles

Fixed-Database role name	Description
db_owner	Members of the **db_owner** fixed database role can perform all configuration and maintenance activities on the database, and can also drop the database in SQL Server. (In SQL Database and SQL Data Warehouse, some maintenance activities require server-level permissions and cannot be performed by **db_owners**.)
db_securityadmin	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.
db_accessadmin	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.
db_backupoperator	Members of the **db_backupoperator** fixed database role can back up the database.
db_ddladmin	Members of the **db_ddladmin** fixed database role can run any Data Definition Language (DDL) command in a database.
db_datawriter	Members of the **db_datawriter** fixed database role can add, delete, or change data in all user tables.
db_datareader	Members of the **db_datareader** fixed database role can read all data from all user tables.
db_denydatawriter	Members of the **db_denydatawriter** fixed database role cannot add, modify, or delete any data in the user tables within a database.
db_denydatareader	Members of the **db_denydatareader** fixed database role cannot read any data in the user tables within a database.

Built-in Database Roles

- LAB – Demo

Note:

Public Roles exist for all of users and like everyone in Active Directory, with some usual rights

SQL Server has some default roles and you create your custom roles.

Built-in Database Roles

- LAB – Demo

If you want to add a member with command script:

```
ALTER ROLE db_backupoperator ADD MEMBER vira
```

or

```
ALTER ROLE [db_backupoperator] ADD MEMBER vira
```

User-Defined Database Roles

Just as an administrator can create custom server roles, they can also create user-defined database roles inside of each database

These roles are the best way to mange permissions for users

By default, a user-defined database role has no permissions

A database role can own a schema

A database role can have many users, or be a member of other roles

The dbo User Account

The dbo, or database owner, is a user account that has implied permissions to perform all activities in the database. Members of the sysadmin fixed server role are automatically mapped to dbo.

Note

dbo is also the name of a schema, as discussed in Ownership and User-Schema Separation in SQL Server.

The dbo user account is frequently confused with the db_owner fixed database role. The scope of db_owner is a database; the scope of sysadmin is the whole server. Membership in the db_owner role does not confer dbo user privileges.

User-Defined Database Roles

- Lab Demo

Application Roles

An application role is a database principal that enables an application to run with its own, user-like permissions. You can use application roles to enable access to specific data to only those users who connect through a particular application. Unlike database roles, application roles contain no members and are inactive by default. Application roles are enabled by using **sp_setapprole**, which requires a password. Because application roles are a database-level principal, they can access other databases only through permissions granted in those databases to **guest**. Therefore, any database in which **guest** has been disabled will be inaccessible to application roles in other databases.

Application Roles

In SQL Server, application roles cannot access server-level metadata because they are not associated with a server-level principal. To disable this restriction and thereby allow application roles to access server-level metadata, set the global flag 4616. For more information, see Trace Flags (Transact-SQL) and DBCC TRACEON (Transact-SQL).

Application Roles

Connecting with an Application Role

The following steps make up the process by which an application role switches security contexts:
A user executes a client application.
The client application connects to an instance of SQL Server as the user.
The application then executes the **sp_setapprole** stored procedure with a password known only to the application.
If the application role name and password are valid, the application role is enabled.
At this point the connection loses the permissions of the user and assumes the permissions of the application role.
The permissions acquired through the application role remain in effect for the duration of the connection.

Application Roles

Application roles are very similar to user-defined database roles in that they are used to manage permissions

There are no members of an application role. Instead, the role is invoked with sp_setapprole

Once invoked, the user loses their rights in the database and only has the rights of the application role

By default, the role has no rights

Application Roles


```
CREATE APPLICATION ROLE application_role_name
WITH PASSWORD = 'password' [ ,
DEFAULT_SCHEMA = schema_name ]
```

Application Roles


```
CREATE TABLE AppRoleTest (id INT)
GO

INSERT dbo.AppRoleTest
        ( id )
        VALUES ( 1 ) ,  ( 2 )
GO

SELECT * FROM dbo.AppRoleTest
GO
```