Database Security

Introduction

- The protection of the database against unauthorized access, intentional or un- intentional.
- Securing the database includes securing the environment: data center, hardware, software, people and data.
    - Data breaches to other parts of the system affect the database security, e.g. viruses or a hacker taking a copy of the database files.
- Security involves confidentiality, integrity, and availability of system and data.
- DBMS level: 3 levels of access, access to the machine (at the OS level), database access, objects access.
- Authentication vs Authorization
    - Authentication determines whether a user has access to the DBMS server or not.
    - Authorization determines the privileges on the database objects using such commands as GRANT and REVOKE of Select, Insert, Update, Delete, Execute privileges.
- Audit trail may be used as a security mechanism to track who did what when, consume a lot of space in some situations.
- Encryption may be used to secure the content of certain table's data, e.g. social security numbers, credit card numbers.
    - Data is exposed in transition between the application and the server unless the application encrypts the data.
    - You can setup SQL Server to use SSL encryption. There's a performance cost.
    - Encryption can be setup at the column level, table level, and database level.
- Security policies: collection of standards, policies, and practices.
    - Varies by organization
    - Password rotation, system accounts passwords when staff leave company, service accounts,…
    - Access to server room
- Typical regulatory Acts and Standards
    - HIPAA – Health Insurance Portability & Accountability act. The part that relates to a database is the protection of patient information through proper security.
    - SOX – Sarbanes-Oxley act. The goal is to protect the shareholders from unethical practices. The part that relates to a database is the protection of data from unauthorized tampering.
    - PCI – Payment Card Industry data security standard. It is not a government regulation. The part that relates to a database is the protection of the information, and the provision of audit logs to the security team that monitoring the system for potential tampering.

SQL Server Security

- Can configure security using SQL Server Management Studio (SSMS) or Transact-SQL scripts.
    - SSMS is convenient for one off changes.

    - Scripts are better when deploying bulk changes, typically relating to a software installation/upgrade. Scripts are repeatable, makes it easy to reinstall or apply in test prior to production release.
- Authentication
    - Windows Auth, SQL Server, or both (mixed mode). It is specified during server configuration.
    - To Authenticate a user, a Login is created.
    - DDL

      ```
      CREATE LOGIN pstccad\esemaan FROM Windows
      CREATE LOGIN es_test WITH PASSWORD ='eddytest', DEFAULT_DATABASE =
      Eddy_sales, CHECK_POLICY=OFF
      DROP LOGIN eddy
      ```

    - There are policies around password complexity.
- Authorization
    - You authorize a Login by creating a user that is associated with the authenticated login
    - DDL

      ```
      CREATE USER eddy1 FROM LOGIN eddy1
      DROP USER eddy1
      This user has no access yet to any database objects
      ```

- Roles
    - Role-based security allows you to assign permissions to a role instead of to individual users.
    - Fixed server and fixed database roles are pre-configured to have a fixed set of permissions. New server and database roles can be created.
    - Some of the fixed server roles:
        - sysadmin – Can perform any activity on the server.
        - Securityadmin – can manage login IDs and passwords for the server and can gran, deny, and revoke database permissions.
        - Dbcreator – can create, alter, drop, and restore databases
        - Public – minimum permissions
    - Some of the fixed database roles:
        - db_owner – has all permissions to the database
        - db_datareader – can select data from any table
        - db_datawriter – can insert, update, delete from any table
        - public
    - user defined database roles
        - used to group users together. For example, a role can be created for certain group of users that perform a certain job. As employees are hired or as they resign, the DBA adds/removes people from the role.
        - DDL

          ```
          CREATE ROLE role_name
          GRANT privilege ON object TO role_name
          ALTER ROLE role_name ADD MEMBER user_name
          DROP ROLE role_name (have to remove all users first)
          ```

- Schemas

  - A grouping of database objects, typically related objects. The objects are related from a business or technical standpoint, e.g. StudentClasses schema would contain all tables, views, etc. relating to student classes
  - Typically used in large databases
  - An access control mechanism, i.e. group tables, views, and other objects in one schema, then grant users access to the schema.
  - DDL

    ```
    CREATE SCHEMA schema_name
    CREATE TABLE schema_name.table_name …
    ALTER SCHEMA schema_name TRANSFER object_name
    DROP SCHEMA schema_name
    ```

- Database objects
  - A term used to reference tables, views, indexes, stored procedures, etc.
- Granting/Revoking Access to database objects
  - Grant or revoke access of privileges on database objects for users
  - DDL

    ```
    Grant Select on SCHEMA::schema_name TO user_name
    REVOKE SELECT ON SCHEMA::schema_name TO user_name
    ```

  - Most commonly used privileges to objects: select, update, insert, delete, execute

    There are server level and database level permissions, e.g. alter database, alter login, create table, create view, …
- Stored Procedures
  - Prepared T-SQL Database code that can be re-used
  - Any user with EXECUTE privilege may run the stored procedure
  - System stored procedures, e.g. sp_server_info.
  - Can execute by typing stored procedure name. If there more than one statement, you have to precede with EXEC