Introduction to Structured Query Language (SQL).
Database Security

UMKC
School of Medicine

Database Security

- Database Backup & Recovery
  - Database can suffer physical damage – water damage, ransomware virus, break-in.
  - Database backup allows you to recover any deleted or accidentally changed file to the version required.
  - Backup all pertinent databases using a reliable, cloud backup or tape backup.
  - Design backup and recovery solutions
  - The backup and recovery procedures should be documented.
  - Backup and recovery procedures should be periodically tested.
  - Backup retention intervals should be sufficient to meet the business resumption requirements.

Database Security

- Database Encryption & Key Management
    - Restricted data is encrypted during transmission over the network using encryption measures strong enough to minimize the risk of the data's exposure if intercepted or misrouted from database to client workstation.
    - If database-level encryption for restricted data is implemented, procedures for secure key management should be documented. (Check National Institute of Standards and Technology (NIST) for current recommendations.) *Note: It is recommended that all application layers (network, application, client workstation) are already encrypted before encrypting the database. Database encryption is not a substitute for any of the above requirements. Database encryption of restricted data is not mandatory to meet this standards document.*
    - For data subject to disclosure that is encrypted at storage, the means to decrypt must be available to more than one person.
    - Backup tapes store backups of the database in an encrypted format, and the tapes should not store the plain text encryption keys necessary to decrypt the backups.
    - Key management procedures for decrypting backups should be documented, and available to more than one person.

Database Security

- User Database Roles, Permissions, Passwords
  - Use secure authentication to the database.
  - Only authorized users should have access to the database.
  - Users should be granted the minimal permissions necessary for their job function in the database.
  - Permissions are managed through roles or groups, and not by direct grants to user IDs where possible.
  - Strong passwords in the database should be enforced when technically possible
  - Database passwords are encrypted when stored in the database or transmitted over the network.
  - Applications should require individual database login/password and roles/grants when possible.

Database Security

- User Database Roles, Permissions, Passwords
  - The login ID and password must be secure.
  - Applications should manage user permissions and have audit processes in place.
  - User database objects with restricted data should not have public grants when possible.
  - Document any public grants if needed in databases with restricted data.
  - Non-DBA accounts should not allow the granting of roles or permissions in any environment (Quality Assurance, Production, Development).
  - Database accounts should be locked after at most six failed logins.
  - Procedure to address inactive users should be documented and approved.
  - A report of elevated database permissions should be audited.
  - A report of all access rights for users is should be audited. Twice a year is the recommended interval.

Security - SQL Injection attacks

- SQL injection (SQLI) is a common attack vector that uses malicious SQL code for backend database manipulation to access private, sensitive information.
- For example, A user-provided input http://www.estore.com/items/items.asp?itemid=999 can generates the following SQL query

```
SELECT ItemName, ItemDescription
FROM Item
WHERE ItemNumber = 999
```

Security - SQL Injection attacks

The above-mentioned input, which pulls information for a specific product, can be altered to read http://www.estore.com/items/items.asp?itemid=999 or 1=1.

```
SELECT ItemName, ItemDescription
FROM Items
WHERE ItemNumber = 999 OR 1=1
```

And since the statement 1 = 1 is always true, the query returns all of the product names and descriptions in the database, even those that user may not be eligible to access.

Best practices to prevent SQL Injection Attacks

- Using Prepared Statements (with Parameterized Queries) instead of dynamic SQL.
- Validating user input
- Limiting privileges
- Hiding info from the error message
- Updating your system - apply patches and updates your system to the most up-to-date version
- Keeping database credentials separate and encrypted
- Disabling shell and any other functionalities you don’t need

Database Security

- **Questions**