Part 1: Beginner Level

What is SQL Injection?

SQL injection (SQLi) is a code injection technique that exploits vulnerabilities in applications that interact with databases. It occurs when user-supplied data is not properly validated, filtered, or sanitized before being incorporated into SQL statements.

How SQL Injection Works

The core problem occurs when application code constructs SQL queries by concatenating strings that include user input:

sql

-- *Vulnerable code example*

String query = "SELECT * FROM users WHERE username = '" + username + "' AND password = '" + password + "'";

If a user enters admin' -- as the username, the resulting query becomes:

sql

SELECT * FROM users WHERE username = 'admin' *--' AND password = 'anything'*

The -- starts a comment in SQL, causing the password check to be ignored.

Types of SQL Injection (Basic)

1. **In-band SQLi: Results are visible in the application's response**
    - **Error-based: Forces the database to generate error messages that reveal information**
    - **Union-based: Uses UNION SQL operator to combine results of two queries**
2. **Blind SQLi: Results are not visible in responses**
    - **Boolean-based: Asks the database true/false questions and determines answers from responses**
    - **Time-based: Sends queries that cause delays when true, allowing inference**
3. **Out-of-band SQLi: Data is exfiltrated via alternative channels (DNS, HTTP requests)**

Common Entry Points

1. **Form fields: Login forms, registration pages, contact forms**
2. **URL parameters: Query strings in GET requests**
3. **HTTP headers: User-Agent, Cookie values, Referer headers**
4. **File uploads: Metadata extraction or processing**
5. **JSON/XML payloads: API endpoints**

Basic Attack Vectors

1. Authentication Bypass

Username: admin' --

Password: anything

2. Extracting Data with UNION

' UNION SELECT username, password FROM users --

3. Detecting Injection Points

' OR '1'='1

' OR 1=1 --

" OR 1=1 --

or 1=1--

' OR 'x'='x

4. Basic Numeric Injection (no quotes needed)

1 OR 1=1

product_id=1 OR 1=1

Consequences of SQL Injection

1. **Data breach: Unauthorized access to sensitive information**
2. **Authentication bypass: Access to restricted parts of applications**
3. **Data manipulation: Adding, modifying, or deleting records**
4. **Data destruction: Deletion of tables or databases**
5. **Server compromise: In some cases, gaining operating system access**

Basic Prevention Techniques

1. **Parameterized queries (prepared statements):**

java

// Java example

PreparedStatement stmt = conn.prepareStatement("SELECT * FROM users WHERE username = ? AND password = ?");

stmt.setString(1, username);

stmt.setString(2, password);

2. **Input validation: Checking that inputs match expected formats**
3. **Escaping special characters: Using database-specific escaping functions**
4. **Limiting database privileges: Using database accounts with minimal permissions**

5. **Using ORMs: Object-Relational Mapping frameworks often include built-in protections**

Common Tools for Basic Testing

1. **OWASP ZAP: Free security testing tool**
2. **SQLmap: Automated SQL injection detection and exploitation**
3. **Burp Suite Community Edition: Web vulnerability scanner**