**SQL injection Vulnerability: Falls under Injection attacks category in OWASP Top Ten**

SQL injection attacks is performed against relational and non-relational databases. The concept of SQL injection occurs when attacker attempts to pass input that changes the final SQL query sent by web application to the database.

Types of SQL Injection: There are 4 types of SQL injection

1. In-Band SQL Injection:

It occurs when an attacker is using the same communication channel to both launch the attack and gather details from the database.

There are two types of In-Band SQL injection

1. Error based SQL injection:

Error based SQL injection is an In-Band based SQL injection method that relies on error messages is thrown by database server while exploiting or penetration testing the web application or database server.

Prevention:

1. Errors should be disabled in the live site.
2. Errors should be logged to a file with restricted access if necessary.
3. Union based SQL injection:

Union-based SQL injection In-Band based SQL injection method that leverages on the “UNION” SQL operator to combine the results of two or more “SELECT” statements into a single result which is then returned as a part of the HTTP response from the Database server to the attacker.

Prevention:

1. Sanitise the input fields.
2. Use an “allow“ list instead of a “deny” list.
3. Inferential SQL injection (Blind SQL injection):

In an Inferential SQL injection attack no data is transferred via the web application and the attacker would not be able to see the results of an attack. An attacker can reconstruct the database structure by sending payloads observing the web application response and resulting behaviour of the database server.

There are two types of Inferential SQL injection:

1. Boolean Based Blind SQL injection:

It relies on sending a SQL query to the database which force is the application to return a different result depending on whether the query returns true of false.

Depending on result, content in HTTP response will change. This allows the attacker to   
conclude if the payload used returned true or false.

1. Time Based Blind SQL injection:

It relies on sending a SQL query to database which forces is the database to wait for a specified amount of time, in seconds before responding. The response time from the database server would indicate the attacker if payload used in query is executed successfully or not.

1. Out of Band SQL injection:

Out of band SQL injection depends on features being enabled on database server being used by the web app it occurs when an attacker is unable to use the same channel to launch the attack and gather results out of bank SQL injection methods would rely on the database server’s ability to make DNS or HTTP requests to deliver do an attacker.

1. Second Order SQL injection

The attacker supplied input is stored in the database and later it is used without sanitation in new SQL query when a user accesses some other functionality of the same application.

Discovery method: Uses ‘quotes’ or “double quotes” in input field to verify the SQL injection vulnerability on the specific input field.

Note that input field checkbox is drop boxes any field containing text candy vulnerable to SQL injection attack to identify search wonder abilities use burp suit to capture the request.

Impact of SQL Injection Attack:

1. Successful attack leads in gaining administrative access over sensitive data stored in database server.

Example for sensitive data: - Usernames, Passwords, Phone numbers, Credit/Debit card numbers, Govt/Private provided Identity numbers, Health Records, Financial Wallet information.

1. Successful attack causes data beach against website.
2. Successful attack allows attackers to read and write files directly on the backend server which instead leaving back door on backend servers.

Prevention and Mitigation for SQL Injection Vulnerabilities:

1. Input Sanitization: Removing, Replacing, Encoding, Escaping characters which are unwanted for the query’s logic.
2. Input Validation: User input should be validated based on the data used in query to ensure that it matches the expected input from user.
3. Parameterised queries: Parameterised queries contain placeholders for the input data which is then escaped and passed on by the drivers instead of directly passing the data into SQL query. Placeholders are used and then filled by PHP functions.

Example:

mysqli\_stmt\_bind\_param() function binds the place holders with the value/input which will safely escape any quotes and insert the values into query.

1. User Privileges: Ensuring that the user querying the database has only minimum permissions required to perform the authorised intended functions smoothly.
2. Web Application Firewall: WAF’s are used to detect malicious input and reject any HTTP request containing suspicious contents.

Example:

HTTP Requests made to the database server containing “INFORMATION\_SCHEMA” word would be rejected as it is used while exploiting SQL injection, which is also an example for a “deny” list.

Types of payloads that can be used

* Subverting query logic

Example: <sql query> OR ‘1’=’1’<sql query>

* Using comments: once the payload is inserted into a position the rest of the query will be commented using comments (-- ) (note the space after double slashes) & (#) so that the compiler ignores the rest of the web applications logic and execute the payload.
* Using union clause: It is used to combine the results of multiple select statement. Union statement can only operate on select statement with equal number of columns to detect the number of columns use order by clause.

Tool to detect SQL related vulnerabilities and exploit them

1. SQLMap is a free and open-source penetration testing tool written in python that automates detecting and exploiting SQL Injection Vulnerabilities.

Practice the SQL Injection Labs

PortSwigger, HackTheBox, any other open-source labs available