

Experiment No. 6

SQL Injection: Use DVWA to practice SQL injection attacks. Demonstrate how an attacker can manipulate input fields to extract, modify, or delete database information.

Setting Up DVWA

1. Install DVWA:

- You can set up DVWA on your local machine using XAMPP or Docker.
- After installation, open DVWA in your web browser (usually accessible at <http://localhost/dvwa>).
- Log in with the default credentials (username: admin, password: password).

2. Set Security Level:

- Go to the DVWA Security tab and set the security level to "Low" for simplicity in this demonstration.

Basic SQL Injection Attack

1. Navigate to the SQL Injection Page:

- In the DVWA menu, click on "SQL Injection".

2. Understanding the Input Field:

- You will see an input field where you are asked to enter a user ID to fetch information from the database.

3. Testing for SQL Injection Vulnerability:

- In the input field, enter a simple SQL injection payload, such as `1' OR '1'='1`. This input attempts to manipulate the SQL query behind the scenes.
- Click "Submit".

4. Analyzing the Result:

- If the application is vulnerable, it should return all user information from the database, because the condition `1' OR '1'='1` is always true.

Extracting Database Information

1. Extracting All Users:

- Try a more sophisticated injection: ' OR 1=1--.
- This payload comments out the rest of the SQL query, causing the database to return all records.

2. Retrieving Specific Information:

- To extract specific information, you can tailor your query. For example: 1' UNION SELECT user, password FROM users--.
- This payload combines the results from the user ID query with a UNION statement that fetches all usernames and passwords from the users table.

Modifying Database Information

1. Altering Data:

- SQL injection can also be used to modify database entries. For example: 1'; UPDATE users SET password='hacked' WHERE user_id=1--.
- This payload attempts to change the password of the user with user_id=1 to 'hacked'.

Deleting Database Information

1. Deleting Data:


- An attacker can delete records with a similar approach. For example: 1'; DELETE FROM users WHERE user_id=1--.
- This payload deletes the user with user_id=1 from the database.

Protecting Against SQL Injection

To prevent SQL injection attacks:

1. Use Prepared Statements: Prepared statements with parameterized queries ensure that SQL code is passed separately from data.

php

 Copy code

```
$stmt = $pdo->prepare('SELECT * FROM users WHERE user_id = :user_id');
$stmt->execute(['user_id' => $user_id]);
```

2. **Validate and Sanitize Input:** Always validate and sanitize user inputs.
3. **Use ORM Libraries:** Object-Relational Mapping (ORM) libraries abstract away the SQL queries.
4. **Least Privilege Principle:** Grant the minimum necessary database privileges to your application.

Conclusion

Practicing SQL injection on DVWA provides hands-on experience on how attackers exploit vulnerabilities in web applications. By understanding these attacks, developers can better secure their applications against such threats. Always ensure you have permission to test and attack systems and never use these skills maliciously.