8(B) Demonstrate SQL Injection Attacks using Tools

8(B)(i) SQL Injection using Tyrant SQL

Aim

To demonstrate SQL Injection attack using Tyrant SQL tool.

Tools Used

- Tyrant SQL Tool
- Windows Operating System
- Vulnerable Web Application

Theory

Tyrant SQL is an automated SQL Injection tool used to exploit database vulnerabilities by injecting malicious SQL queries into web applications.

Steps / Procedure

1. Download and install Tyrant SQL tool.
2. Launch the Tyrant SQL application.
3. Enter the vulnerable website URL.
4. Specify the injectable parameter.
5. Start the SQL Injection attack.
6. Tyrant SQL analyzes database structure.
7. Extract database names, tables, and columns.
8. Observe retrieved database information.

Result

SQL Injection attack was successfully performed using Tyrant SQL.

8(B)(ii) SQL Injection using Havij

Aim

To perform SQL Injection attack using Havij tool.

Tools Used

• Havij Tool
• Windows Operating System
• Vulnerable Website

Theory

Havij is an automated SQL Injection tool that exploits vulnerabilities in web applications to retrieve database contents without authorization.

Steps / Procedure

1. Download and install Havij tool.
2. Launch the Havij application.
3. Enter the target website URL.
4. Click on **Analyze**.
5. Havij detects SQL Injection vulnerability.
6. Retrieve database details such as:
   - Database name
   - Table names
   - Column values
7. Save extracted data for analysis.

Result

SQL Injection vulnerability was successfully exploited using Havij.

8(B)(iii) SQL Injection using BBQSQL

Aim

To demonstrate Blind SQL Injection using BBQSQL.

Tools Used

• BBQSQL Tool
• Windows Operating System
• Vulnerable Web Application

Theory

BBQSQL is a Blind SQL Injection framework used to exploit applications where error messages are not visible, using inference-based techniques.

Steps / Procedure

1. Download and install BBQSQL.
2. Launch BBQSQL tool.
3. Configure the target URL and injectable parameter.
4. Select Blind SQL Injection method.
5. Start the injection process.
6. BBQSQL infers database responses.
7. Extract database structure information.
8. Analyze retrieved results.

Result

Blind SQL Injection attack was successfully demonstrated using BBQSQL.

Final Conclusion

Practical No. 8 successfully demonstrated protection of web applications using dotDefender and exploitation of SQL Injection vulnerabilities using Tyrant SQL, Havij, and BBQSQL tools.