SQL Injection Practical Exploitation Report

Cyber Security Internship – Task 8

Tool Used: SQLMap

Target Application: DVWA (Localhost Test Environment)

Operating System: Windows

Author: Mohammed Nihal

Abstract

This report presents a detailed SQL Injection exploitation exercise conducted as part of a cyber security internship. The objective of this task is to understand how SQL Injection vulnerabilities are identified, exploited, and analyzed using automated tools such as SQLMap. The assessment was performed on DVWA, a deliberately vulnerable web application, hosted locally to ensure ethical and legal testing.

1. Introduction

SQL Injection (SQLi) is a web application vulnerability that allows attackers to manipulate database queries by injecting malicious SQL code through user input. It is consistently ranked as one of the most dangerous vulnerabilities in the OWASP Top 10 due to its potential impact on data confidentiality, integrity, and availability. This task focuses on practical exploitation to better understand the real-world risks associated with SQL Injection.

2. Testing Environment

The testing environment was configured on a Windows system using XAMPP to host DVWA locally. Apache and MySQL services were enabled to support the application. DVWA security level was set to 'Low' to intentionally disable security controls for learning purposes. SQLMap was executed via command line to automate SQL Injection exploitation.

3. Testing Methodology

A combination of manual testing and automated exploitation was used during this task. Initially, manual SQL Injection payloads were injected to confirm vulnerability presence. Once confirmed, SQLMap was used to automate detection, enumeration, and data extraction processes. Valid session cookies were supplied to ensure authenticated access to the vulnerable endpoint.

4. Identification of Injectable Parameters

The SQL Injection vulnerability was identified in the 'id' GET parameter of the DVWA SQL Injection module. Manual payloads such as ' OR '1'='1 -- resulted in abnormal responses, confirming that user input was directly incorporated into SQL queries without proper sanitization.

5. SQLMap Exploitation Process

SQLMap was executed against the vulnerable URL with appropriate session cookies. The tool automatically detected the injectable parameter and identified the backend database management system as MySQL. SQLMap then proceeded with enumeration of databases and tables without manual intervention.

6. Database Enumeration

Using SQLMap, all available databases on the server were enumerated. The 'dvwa' database was identified as the primary target. Further enumeration revealed tables such as 'users' containing sensitive authentication data.

7. Data Extraction and Analysis

SQLMap successfully extracted data from the 'users' table, including usernames and password hashes. This demonstrated how attackers can retrieve sensitive information without proper authorization. Extracted password hashes could potentially be cracked offline, leading to account compromise.

8. Impact Analysis

The successful exploitation of SQL Injection vulnerabilities can lead to severe security consequences. These include authentication bypass, unauthorized data access, credential theft, data manipulation, and in some cases complete compromise of the database server. Such vulnerabilities pose a critical risk to organizations and users.

9. Mitigation and Prevention Strategies

To mitigate SQL Injection vulnerabilities, developers should implement prepared statements and parameterized queries to separate SQL code from user input. Additional protections include server-side input validation, least privilege database access, disabling verbose error messages, regular security testing, and deployment of Web Application Firewalls (WAF).

10. Conclusion

This task provided in-depth practical exposure to SQL Injection exploitation using SQLMap. By performing enumeration and data extraction in a controlled lab environment, a strong understanding of SQL Injection risks and mitigation techniques was developed. These skills are essential for web application security testing and secure software development.