CYBER SECURITY INTERNSHIP – TASK 8

SQL Injection Practical Exploitation (Analysis)

---

1. Introduction

SQL Injection is one of the most critical web application vulnerabilities listed in the OWASP Top 10. It occurs when an application improperly handles user input and directly includes it in SQL queries. This task focuses on understanding how SQL Injection works, how automated tools like SQLMap identify vulnerabilities, and how such attacks can be prevented.

---

2. What is SQL Injection?

SQL Injection is a vulnerability that allows attackers to manipulate backend database queries by injecting malicious SQL statements through user input fields such as login forms or URL parameters.

Example

```
SELECT * FROM users WHERE id = '1';
```

If input is:

```
' OR '1'='1
```

The query becomes always true, allowing unauthorized access.

---

3. Vulnerable Application Used

For this task, testing was performed **only on intentionally vulnerable applications**, such as:

- DVWA (Damn Vulnerable Web Application)
- OWASP Juice Shop (demo)

No real or unauthorized websites were tested.

---

4. Identifying an Injectable Parameter

A URL parameter such as:

```
product.php?id=1
```

was analyzed.
The `id` parameter was identified as a possible injection point due to lack of proper input validation.

---

5. Introduction to SQLMap

SQLMap is an automated penetration testing tool used to detect and exploit SQL Injection vulnerabilities. It can identify vulnerable parameters and extract database information automatically.

---

6. SQLMap Testing (Conceptual)

Basic SQLMap Command

```
sqlmap -u "http://example.com/product.php?id=1"
```

SQLMap analyzes the parameter and confirms whether it is vulnerable to SQL Injection.

---

7. Database Enumeration (Conceptual)

SQLMap can retrieve database information such as database names, tables, and columns.

Example Commands

```
sqlmap -u URL --dbs
sqlmap -u URL -D database_name --tables
sqlmap -u URL -D database_name -T users --dump
```

These commands demonstrate how sensitive data could be extracted if the vulnerability exists.

---

8. Impact of SQL Injection

The impact of SQL Injection can be severe, including:

- Unauthorized access to data
- Leakage of sensitive information
- Exposure of user credentials
- Database modification or deletion
- Complete system compromise

---

9. Prevention and Mitigation Techniques

To prevent SQL Injection attacks:

- Use prepared statements
- Implement parameterized queries
- Validate and sanitize user inputs
- Use least-privilege database accounts
- Deploy Web Application Firewalls (WAF)