## SQL Injection (High Level)

### Vulnerability Overview

At the High level, DVWA introduces prepared statements with MySQLi, which helps protect the SQL structure. However, the session variable $\_SESSION['id'] is still directly used in the query.

**1. SQL Injection Vulnerability**



Figure 12:SQL Injection Vulnerability

The $id value from the session is inserted directly into the query string without any validation or sanitization. This means if an attacker manages to control or manipulate the $\_SESSION['id'], they could inject malicious SQL code.

**2. Unsafe Error Message**



Figure 13:Unsafe Error Message

While this doesn’t expose detailed server info, it still stops execution and can hint to an attacker that they triggered something — which helps in crafting further attacks.

### Exploitation

**1. Authentication Bypass**

Payload: 1' or '1' = '1

A screenshot of a computer

AI-generated content may be incorrect.

Figure 14:Authentication Bypass

This always-true condition retrieves all rows, bypassing any intended filtering.

**2. Extracting Database User and Name**

Payload: 1' union select database(),user() #

A screenshot of a computer

AI-generated content may be incorrect.

Figure 15:Extracting Database User and Name

This payload forces the database to return the current database name and the user running the query. The result showed that the database being used is dvwa and the user is dvwa\_user@localhost, confirming SQLi vulnerability.

**3. Enumerating Tables**

Payload: 1' union select 1,table\_name from information\_schema.tables #

A screenshot of a computer

AI-generated content may be incorrect.

Figure 16:Enumerating Tables

All the tables in the database has been listed out.

**4. List Tables in DVWA Only**

Payload: 1' union select 1,table\_name from information\_schema.tables where table\_schema = 'dvwa'#

A screenshot of a computer

AI-generated content may be incorrect.

Figure 17:List Tables in DVWA Only

Filters only DVWA-specific tables.

**5. Finding Columns Inside the Users Table**

Payload: 1' union select 1,table\_name from information\_schema.tables where table\_name = 'users'#

A screenshot of a computer

AI-generated content may be incorrect.

Figure 18:Finding Columns Inside the Users Table

**6. Dumping User Credentials**

Payload: ' UNION SELECT user, password FROM users#

A screenshot of a computer

AI-generated content may be incorrect.

Figure 19:Dumping User Credentials

This exposed user accounts along with their hashed passwords.

Then we can use crack station to crack the hashed passwords.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 20:crack the hashed passwords

Now I will use **SQLmap** to exploit data on DVWA.

**1. Check vulnerability**

Command:

A screen shot of a computer

AI-generated content may be incorrect.

Figure 21:Command of checking vulnerability

Output:

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 22: Output of checking vuln

This output shows that SQLMap successfully found a SQL injection vulnerability in the id parameter of the DVWA SQLi page. It confirmed multiple injection types including Boolean-based, Error-based, Time-based, and UNION-based. The database was identified as MySQL (MariaDB), and the server is running on Apache with a Debian Linux system. This proves that the page is vulnerable and that SQLMap can be used to extract data from the backend database.

**2.** **Display all the tables in database**

Command:

A screen shot of a computer

AI-generated content may be incorrect.

Figure 23: Command of Display all the tables in database

Output:

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 24: Output of display all the tables in database

The output showing all tables of dvwa database.

**3. Display all the columns of the table user in the database**

Command:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 25: Command of displaying all the columns of the table user in the database

Output:

A screenshot of a computer

AI-generated content may be incorrect.

**4. Dump user credentials**

Command:

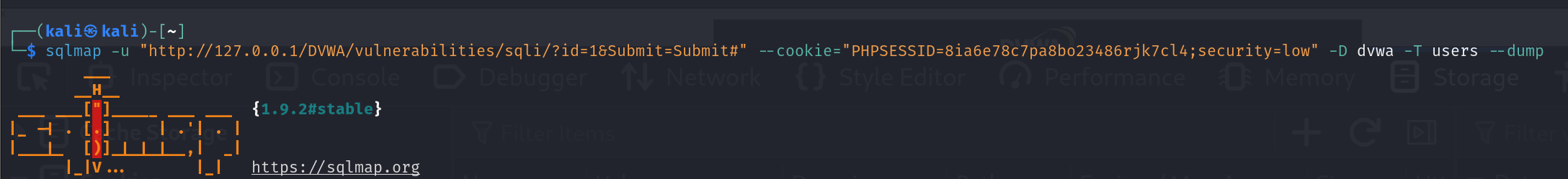


Figure 26: Command of dumping user credentials

Output:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 27: Output of dumping user credentials

When I access and dump all data in the users table, it shows all the dvwa users’ passwords. I can use these passwords to login/logout their account on DWVA.