**Divya Dileep**

**Report:Installation and Usageof DVWA for SQL InjectionTesting**

CONTENTS

**Installation of DVWA using Docker ...... .......................................................................... 2**

**Performing SQL Injection on DVWA...............................................................................** 4

SQL Injection (Low Security Level) ......................................................................... 4

SQL Injection (Medium Security Level) .................................................................... 5

SQL Injection (High Security Level) .......................................................................... 6

**Conclusion ......................................................................................................................... 8**

1. Installation of DVWA using Docker

To install Damn Vulnerable Web Application (DVWA), I used Docker for a streamlined setup. Below are the steps I followed to complete the installation:

* 1. Cloning the Repository I started by cloning the DVWA repository from pentestlab.github.io using the following command:

**git clone** [**https://github.com/eystsen/pentestlab.git**](https://github.com/eystsen/pentestlab.git)

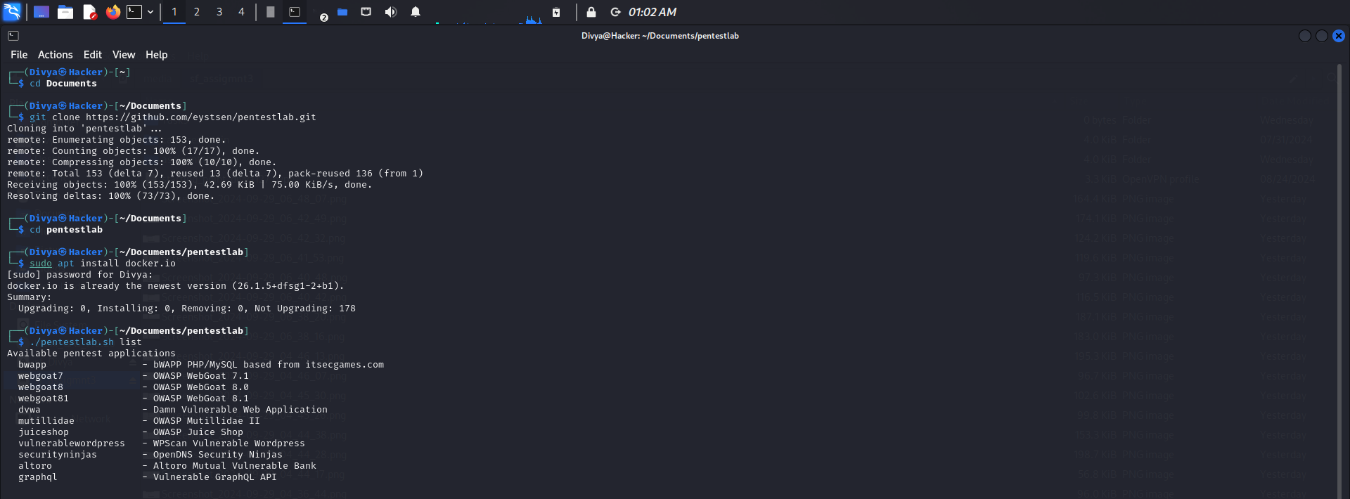
* 1. Starting the Docker Container

After cloning the repository, I navigated to the DVWA folder and ran Docker commands to initiate the web application. The specific steps I followed were:

1. Opened the terminal and navigated to the cloned pentestlab folder.

2. Ran the following command to install Docker container:

**sudo apt install docker.io**



Screenshot 1

* 1. Accessing to the DVWA Web Page

Once the Docker container was running, I run this command for accessing the dvwa web page.

Command: **./pentestlab.sh start dvwa**

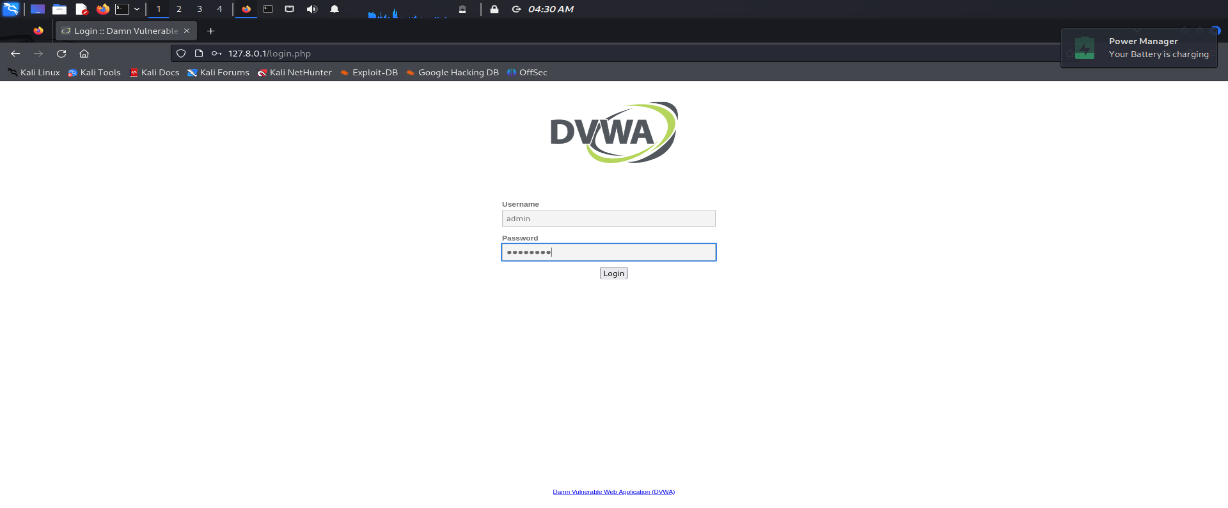


Screenshot 2

* 1. Logging In

At the login page, I used the default credentials: -

* Username: admin
* Password: password



Screenshot 3

* 1. Resetting the Database

After logging in for the first time, I was prompted to reset the database. I clicked the "Reset Database" button (I missed capturing a screenshot of this step). Once the reset was completed, the system redirected me back to the login page.

* 1. Logging In Again

After resetting the database, I logged in again with the default credentials to access the DVWA dashboard

* 1. Completion

At this point, the DVWA setup was complete, and the environment was ready for vulnerability testing.

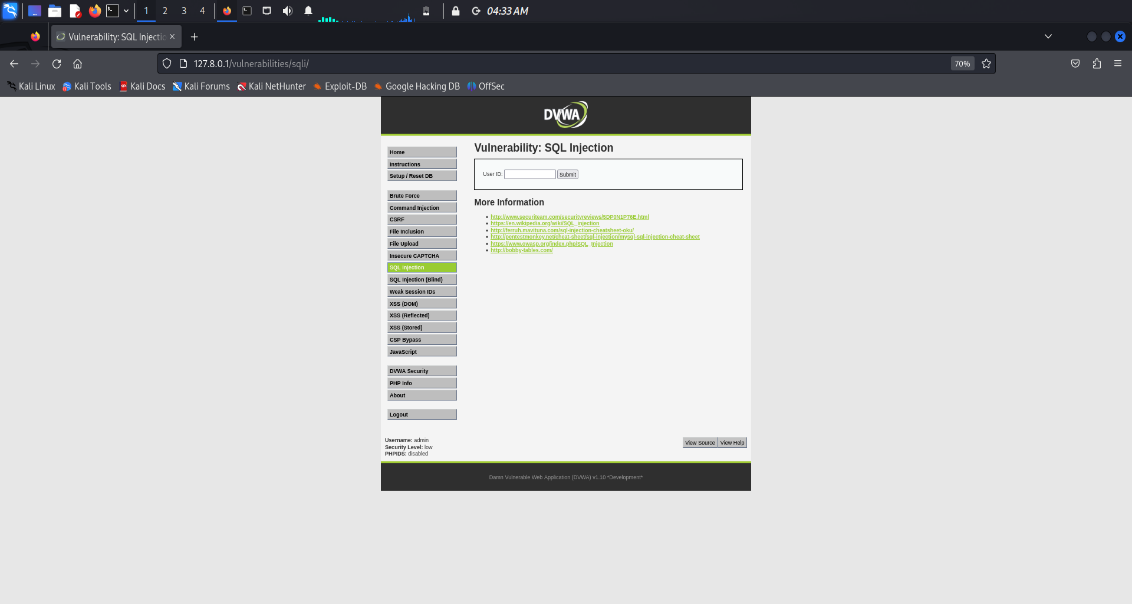
1. Performing SQL Injection on DVWA
   1. SQL Injection (Low Security Level)

I began by testing SQL injection on the Low security level.

2.1.1 Initial Injection After accessing the SQL injection page, I quickly identified the input field for injecting SQL code.

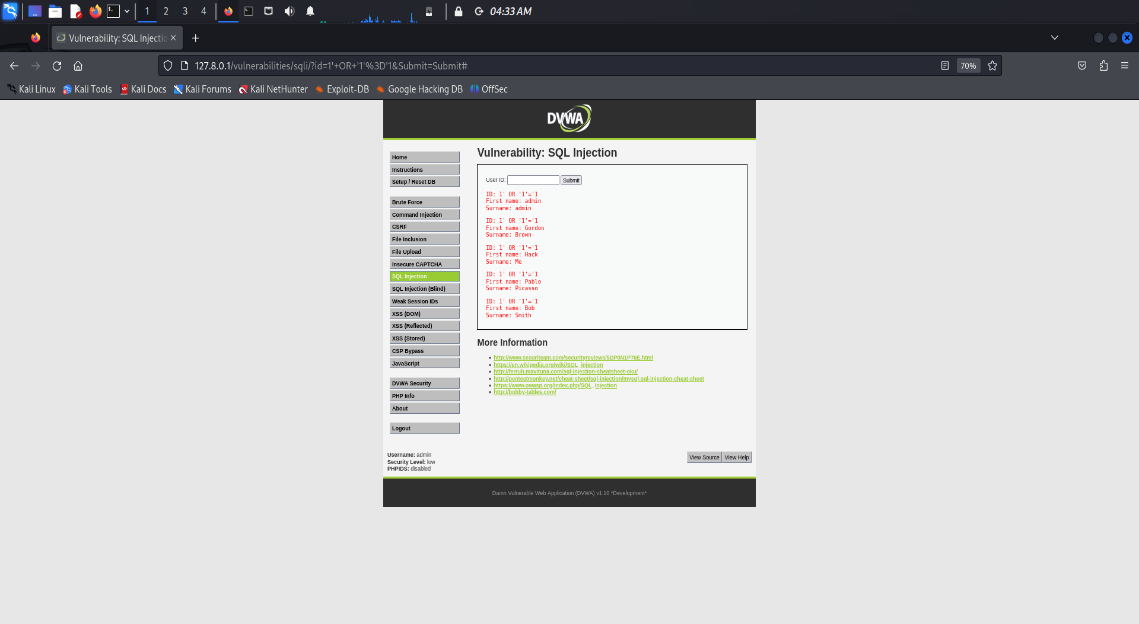
2.1.2 SQL Payload I used the following basic SQL injection string:

1' OR '1'='1



Screenshot 4

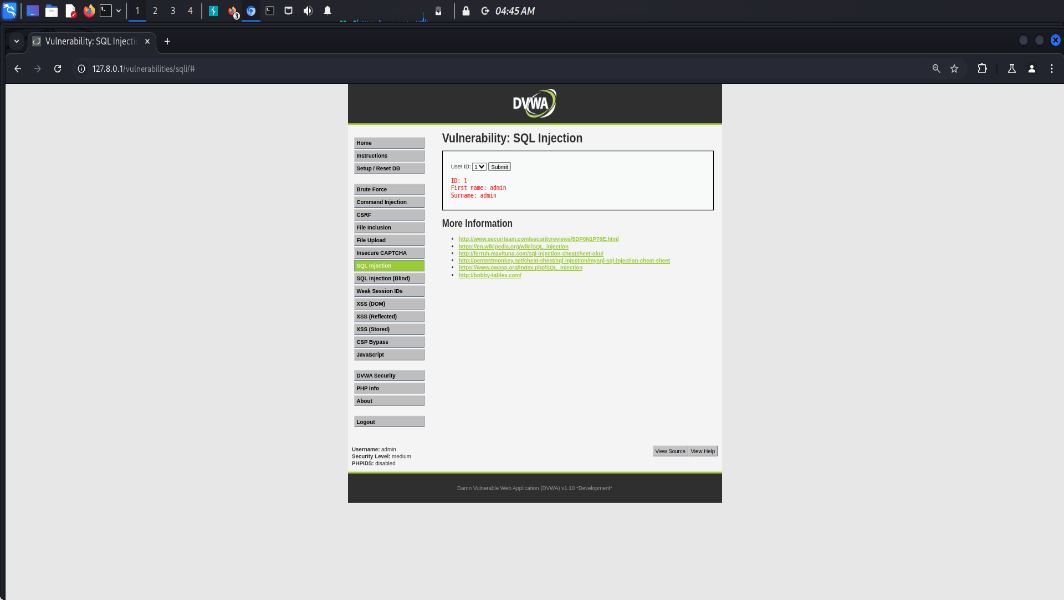
This payload bypassed the need for valid input and displayed the first name and surname of all users.



Screenshot 5

* 1. SQL Injection (Medium Security Level)

After changing the setting to medium, sqli page was slight different.



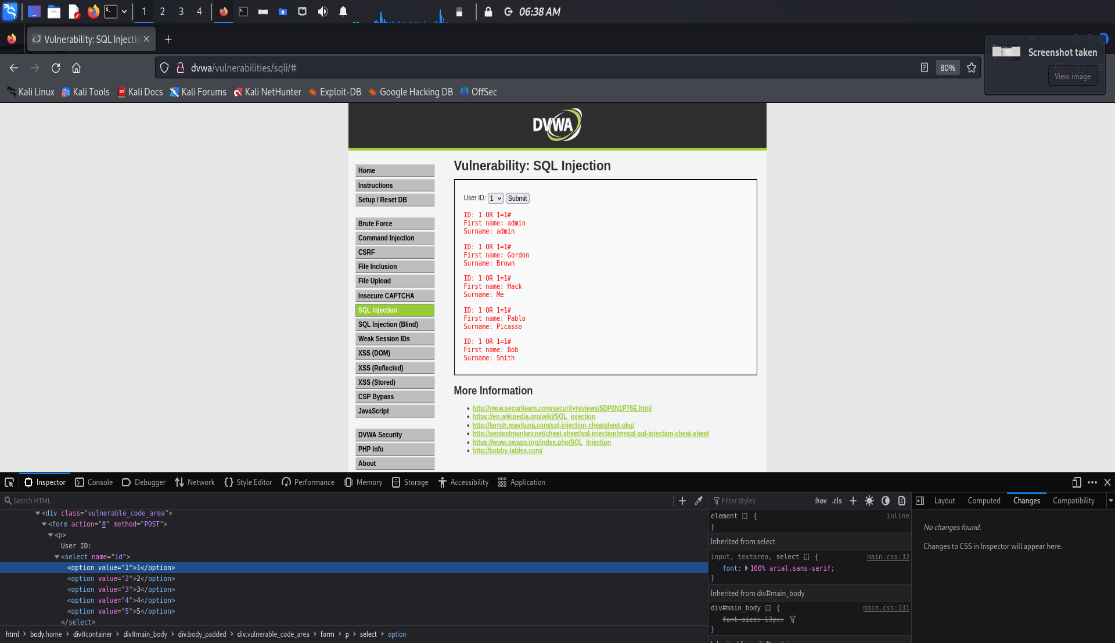
Screenshot 6

There was only buttons of navigating user id and submit, no comment options to inject then I tried some codes from port swigger cheat sheet and medium articles and injected them in URL that was also unsuccessful. Later then went to developer tools hoping that i’ll get an idea.

Then I went to research for sql injection through developer tools contents and used pentestGPT

PentestGPT gave OR 1=1# command to use this on any value of the user ID button and then I clicked submit.

Then on the next page..

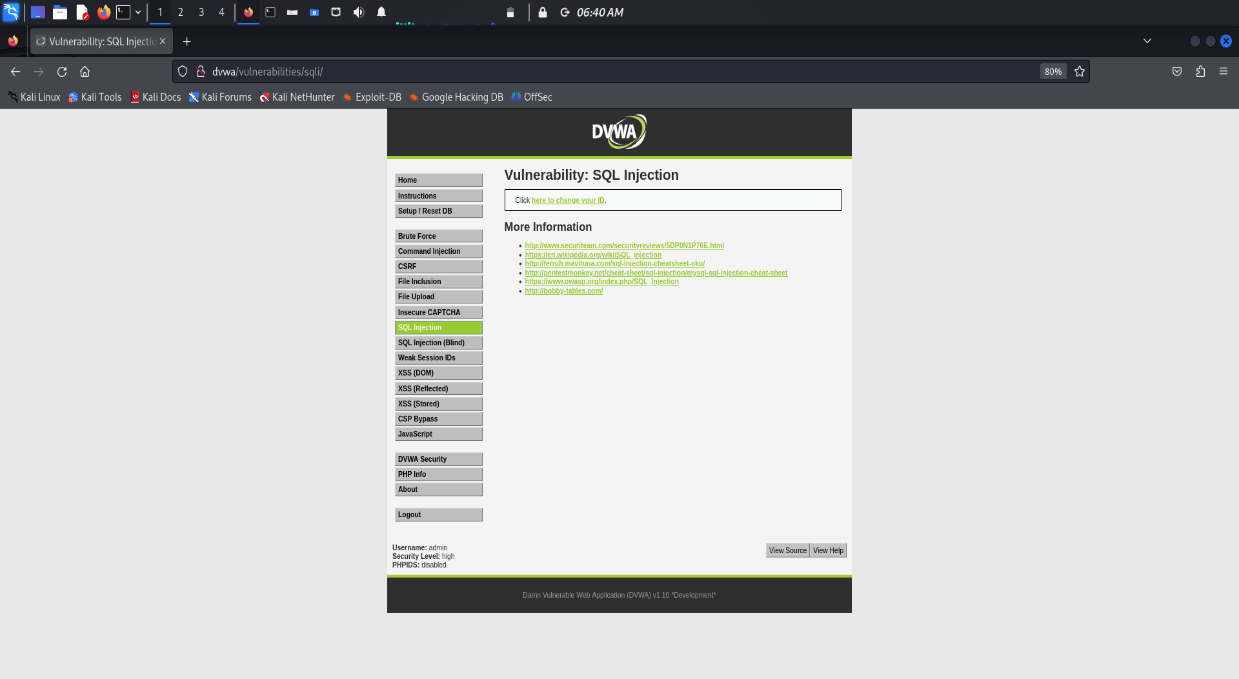


Screenshot 7

I got the first and last name of every user

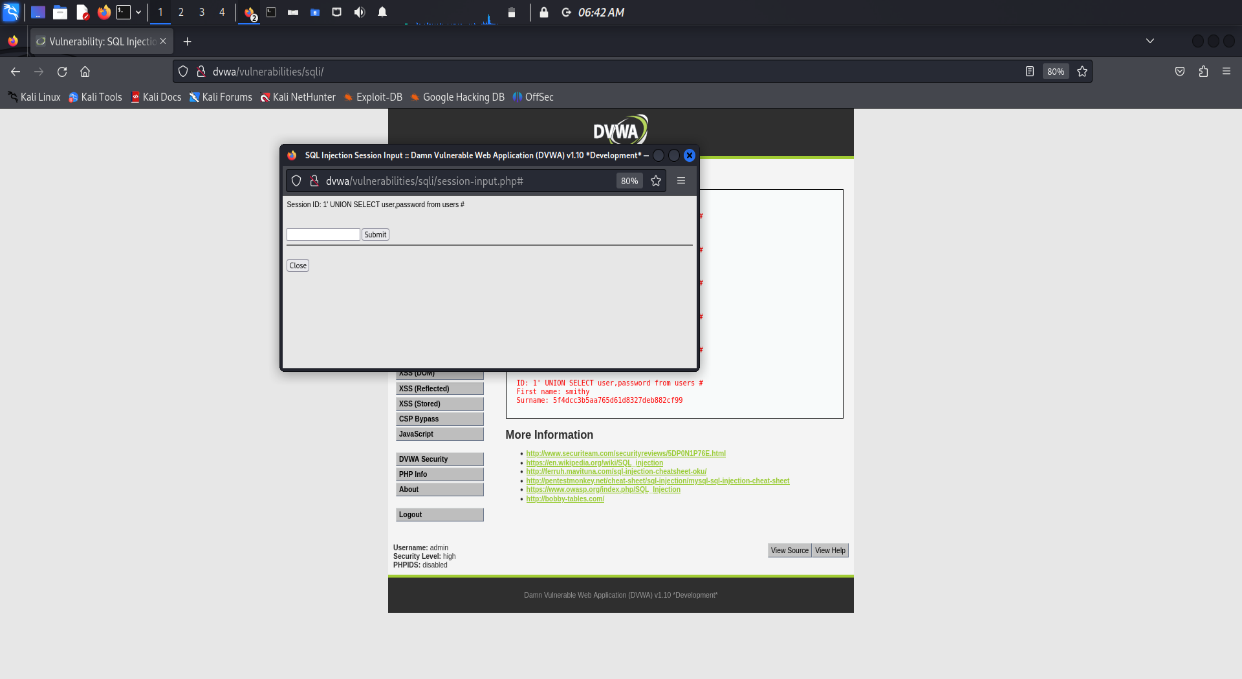
2.3SQL Injection (High Security Level)

In SQLi high mode the interface different. There was a hyperlink texting click here to change your id. After clicking that link another window showed up.



Screenshot 8

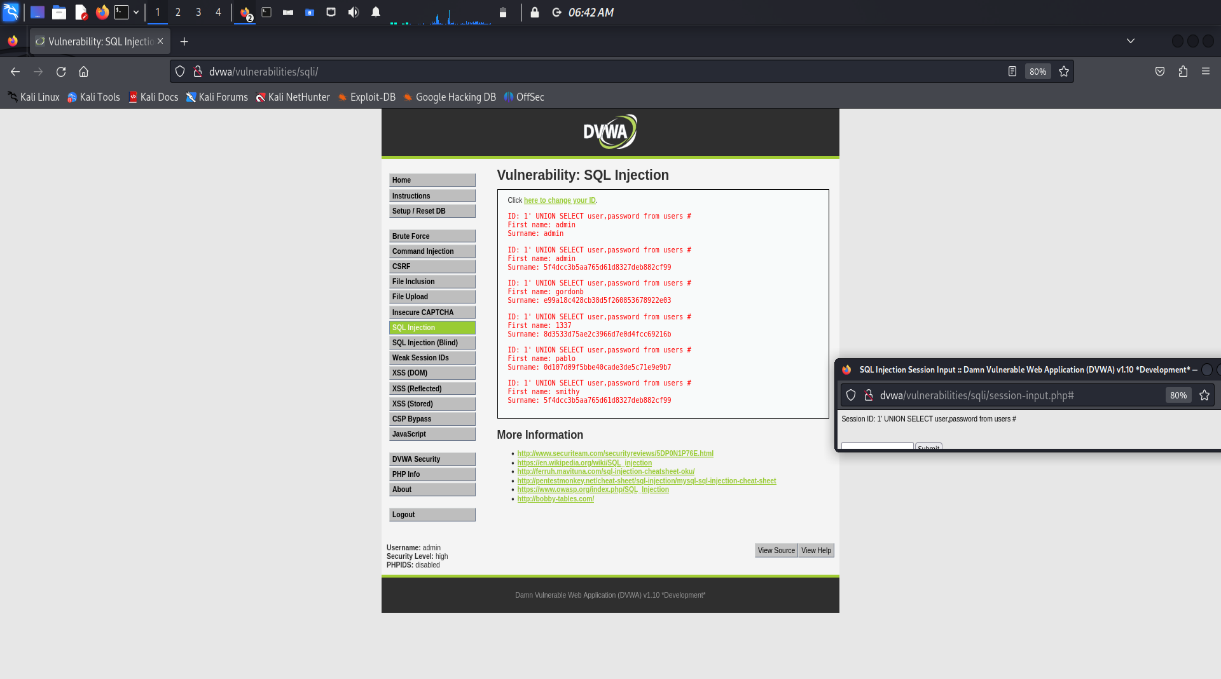
In that window there was a box to text session id. I enter 1 and gave the first and surnames of user number 1.



Screenshot 9

Then I again went for researching for what to do in this snenario.

**1' UNION SELECT user,password from users #** i got this command from medium article of Kamal S and said to use this command after the execution of session id



Screenshot 10

After entering that command I got the first names and session cookies of the users.

**Conclusion**

I successfully installed DVWA using Docker and tested SQL injection vulnerabilities at various security levels. By using simple and advanced SQL injection payloads, I was able to extract sensitive information from the database across all security settings, demonstrating the risks associated with poorly secured web applications. This exercise highlights the importance of securing applications against SQL injection attacks, especially by implementing proper input validation and parameterized queries.