

# Report, Installation and usage of DVWA for SQL injection testing

---

TASK 15

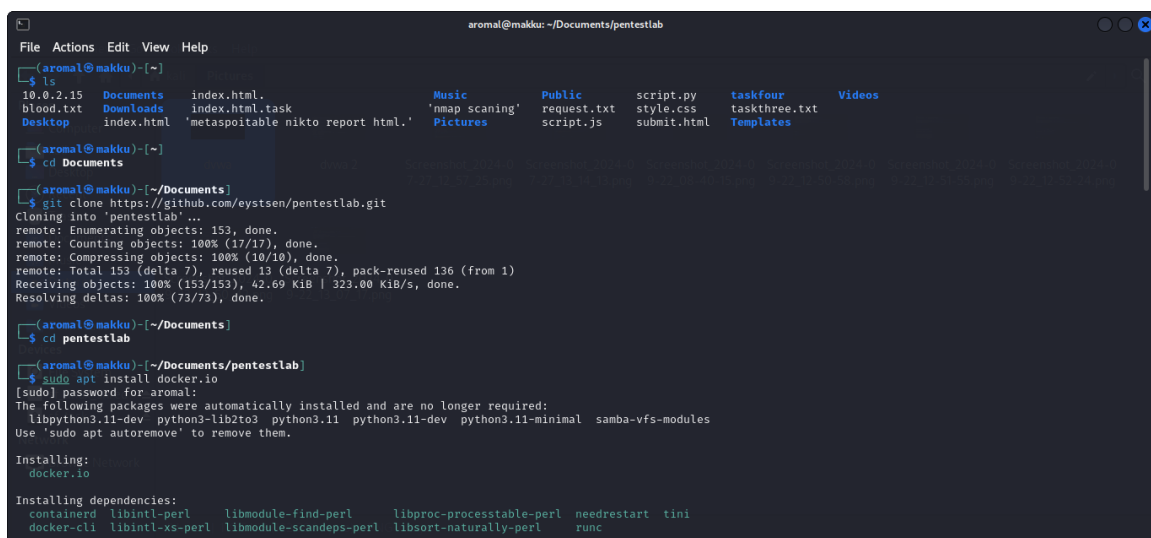
**Aromal Santhosh**

**9/29/2024**

## Installing DVWA

For installing DVWA I used docker.

Then i went to [pentestlab.github](https://github.com/pentestlab). I am sharing my screenshots for better understanding. These are the step by step process I went through, starting from git clone.....



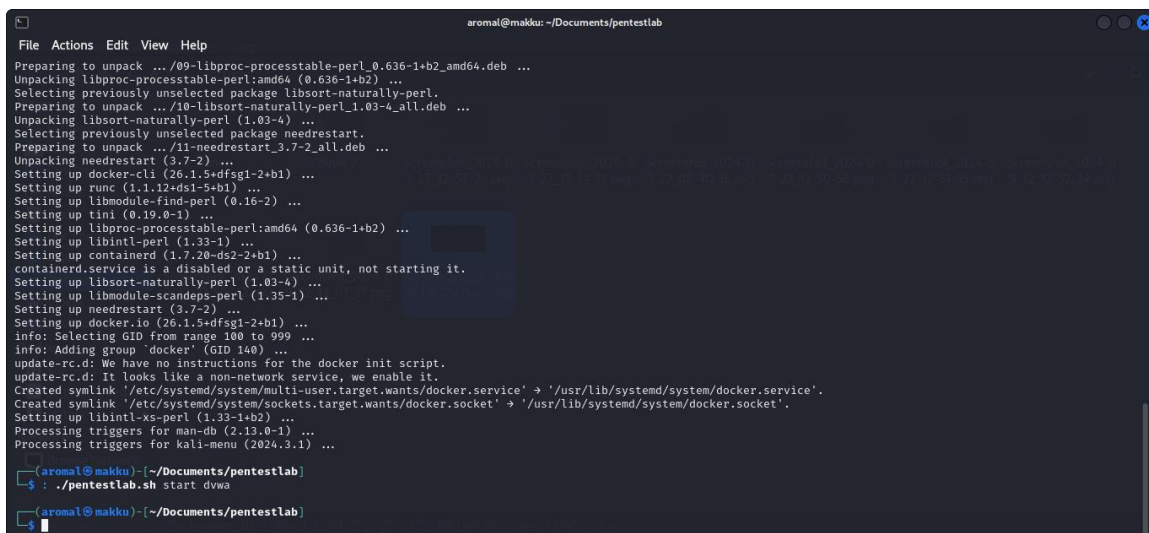
```
aromal@makku: ~/Documents/pentestlab
File Actions Edit View Help
(aromal@makku)~$ ls
10.0.2.15  Documents  index.html  Music  Public  script.py  taskfour  Videos
blood.txt  Downloads  index.html.task  'nmap scanning'  request.txt  style.css  taskthree.txt
Desktop    index.html  'metasploitable nikto report html.'  Pictures  script.js  submit.html  Templates

(aromal@makku)~$ cd Documents
(aromal@makku)~/Documents$ git clone https://github.com/eysysen/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 | 323.00 KiB/s, done.
Resolving deltas: 100% (73/73), done.
(aromal@makku)~/Documents$ cd pentestlab
(aromal@makku)~/Documents/pentestlab$ sudo apt install docker.io
[sudo] password for aromal:
The following packages were automatically installed and are no longer required:
  libpython3.11-dev python3-lib2to3 python3.11 python3.11-dev python3.11-minimal samba-vfs-modules
Use 'sudo apt autoremove' to remove them.

Installing:
  docker.io

Installing dependencies:
  containerd  libintl-perl  libmodule-find-perl  libproc-processtable-perl  needrestart  tini
  docker-cli  libintl-xs-perl  libmodule-scandeps-perl  libsort-naturally-perl  runc
```

After the complete installation of DVWA we will get automatically redirected to the



```
aromal@makku: ~/Documents/pentestlab
File Actions Edit View Help
Preparing to unpack .../09-libproc-processtable-perl_0.636-1+b2_amd64.deb ...
Unpacking libproc-processtable-perl:amd64 (0.636-1+b2) ...
Selecting previously unselected package libsort-naturally-perl.
Preparing to unpack .../10-libsort-naturally-perl_1.03-4_all.deb ...
Unpacking libsort-naturally-perl (1.03-4) ...
Selecting previously unselected package needrestart.
Preparing to unpack .../11-needrestart_3.7-2_all.deb ...
Unpacking needrestart (3.7-2) ...
Setting up docker-cli (26.1.5+dfsg1-2+b1) ...
Setting up runc (1.1.12+ds1-5+b1) ...
Setting up libmodule-find-perl (0.16-2) ...
Setting up tini (0.19.0-1) ...
Setting up libproc-processtable-perl:amd64 (0.636-1+b2) ...
Setting up libintl-perl (1.33-1) ...
Setting up containerd (1.7.20-ds2-2+b1) ...
containerd.service is a disabled or a static unit, not starting it.
Setting up libsort-naturally-perl (1.03-4) ...
Setting up libmodule-scandeps-perl (1.35-1) ...
Setting up needrestart (3.7-2) ...
Setting up docker.io (26.1.5+dfsg1-2+b1) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group 'docker' (GID 140) ...
update-rc.d: We have no instructions for the docker init script.
update-rc.d: It looks like a non-network service, we enable it.
Created symlink '/etc/systemd/system/multi-user.target.wants/docker.service' → '/usr/lib/systemd/system/docker.service'.
Created symlink '/etc/systemd/system/sockets.target.wants/docker.socket' → '/usr/lib/systemd/system/docker.socket'.
Setting up libintl-xs-perl (1.33-1+b2) ...
Processing triggers for man-db (2.13.0-1) ...
Processing triggers for kali-menu (2024.3.1) ...

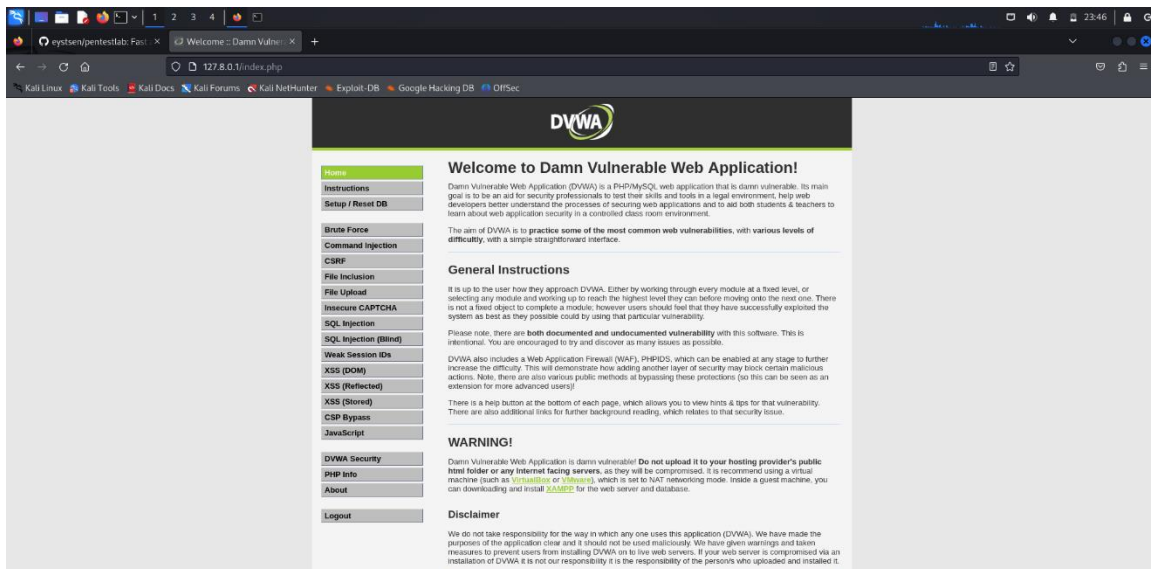
(aromal@makku)~/Documents/pentestlab$ : ./pentestlab.sh start dvwa
(aromal@makku)~/Documents/pentestlab$
```

webpage.





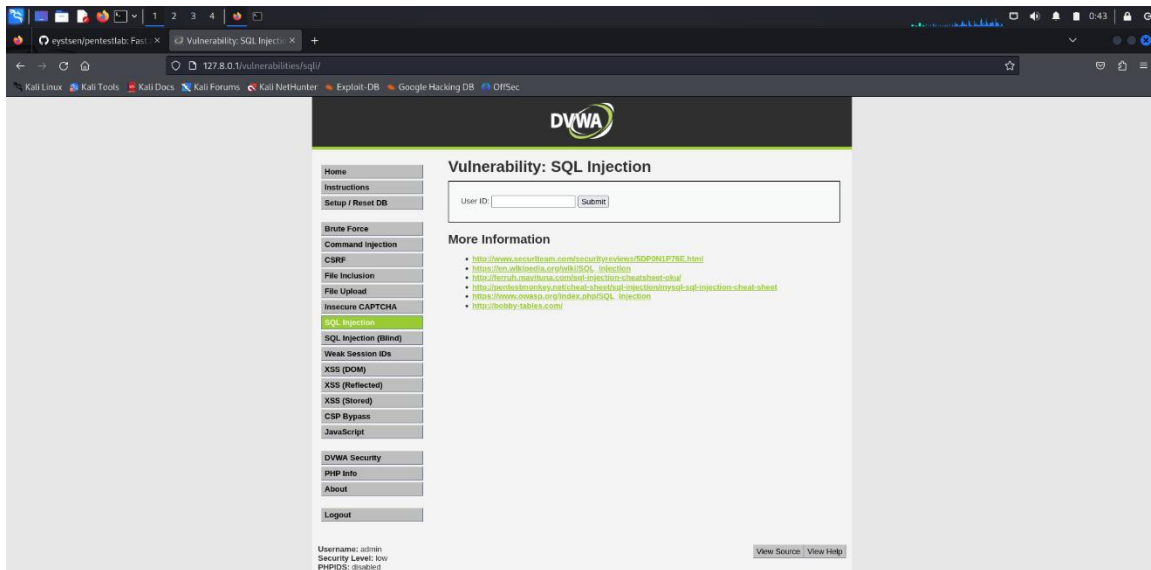
The default username is admin and password is password. Then in the next page you should be asked for reset your database(forgot to take that screenshot). Click that button. Installation complete. Then once again you will be redirected to login page, use the default credentials again. Then you will get to see this page



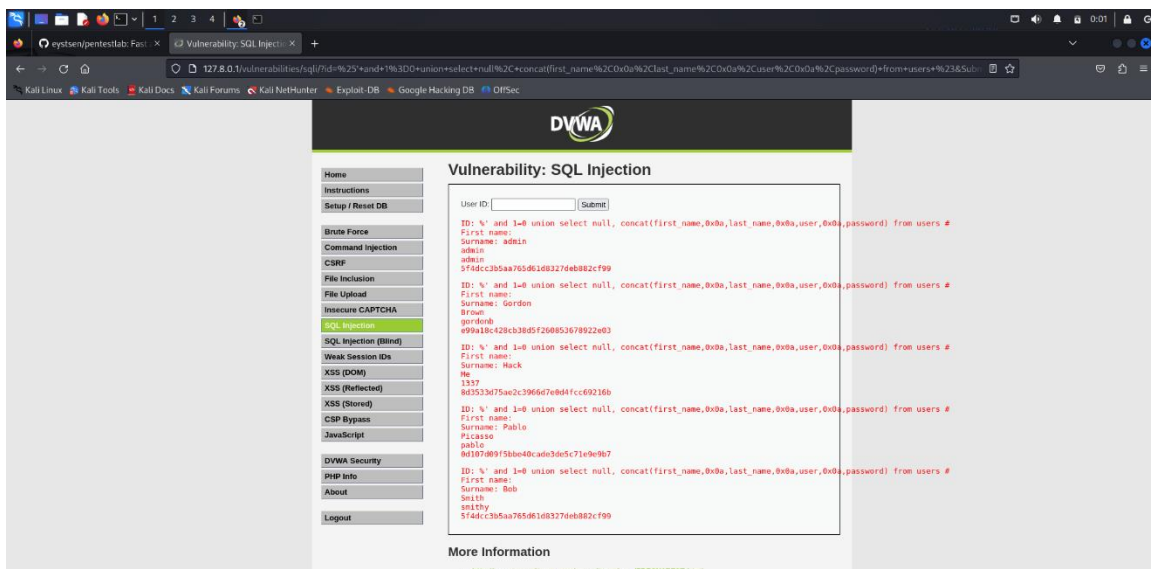
The whole set up is complete.

## SQL injection(low)

First I got to see the page and thought of how to inject the code(no problem for finding the place to inject).



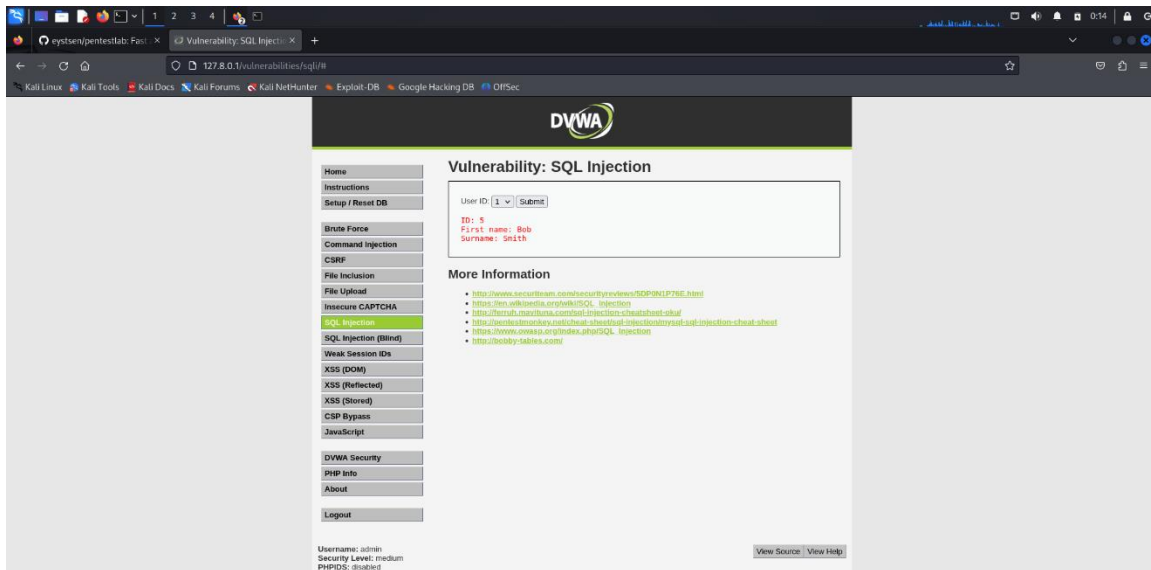
First I used simple one's but that was not helpful to getting inside the database. Went through many websites and at last I used `%' and 1=0 union select null, concat(first_name,oxoa,last_name,oxoa,user,oxoa,password) from users #` which I got from portswigger cheatsheet.



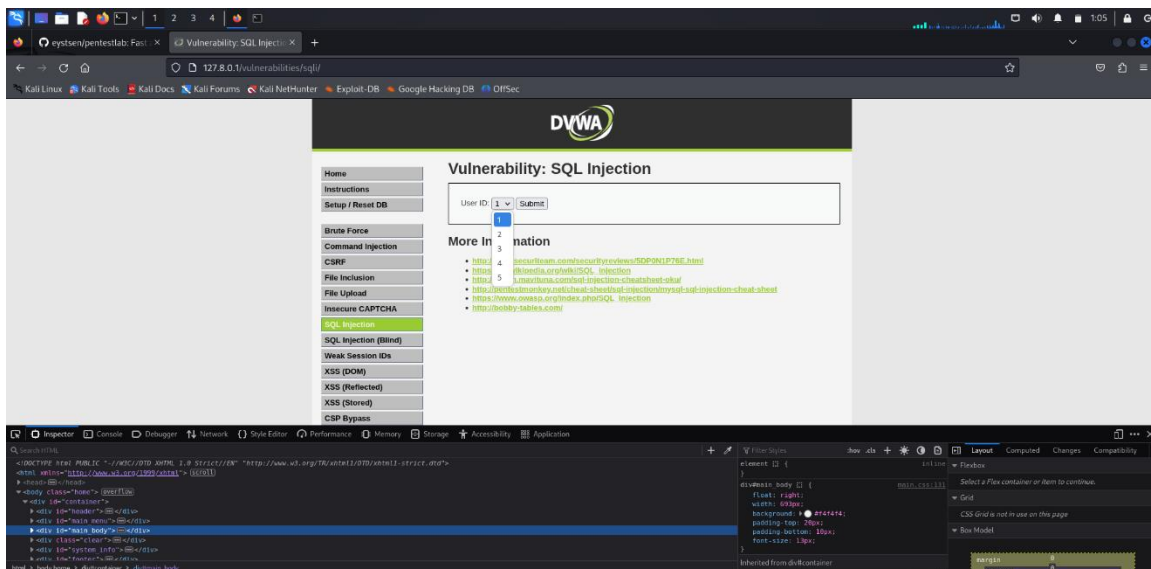
After executing the above mentioned code, this was the result. Got the first and last names and cookies of the users.

## SQL injection(medium)

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

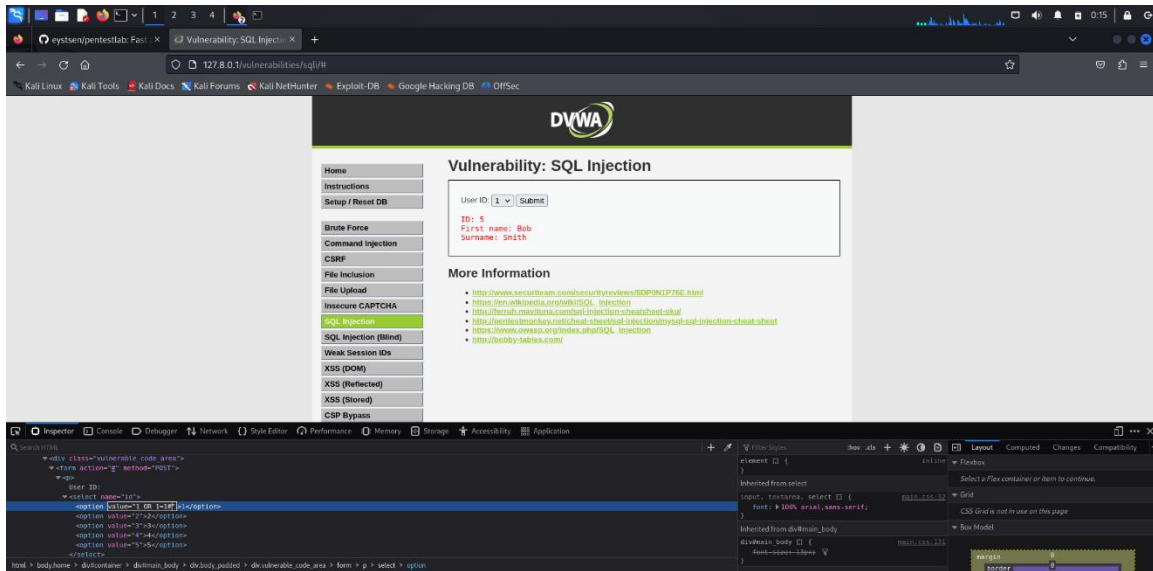


There was only buttons of navigating userid and submit, no comment options to inject then I tried some codes from port swigger cheatsheet and medium articles and injected them in URL that was also unsuccessful. Later then went to developer tools hoping that i'll get an idea.

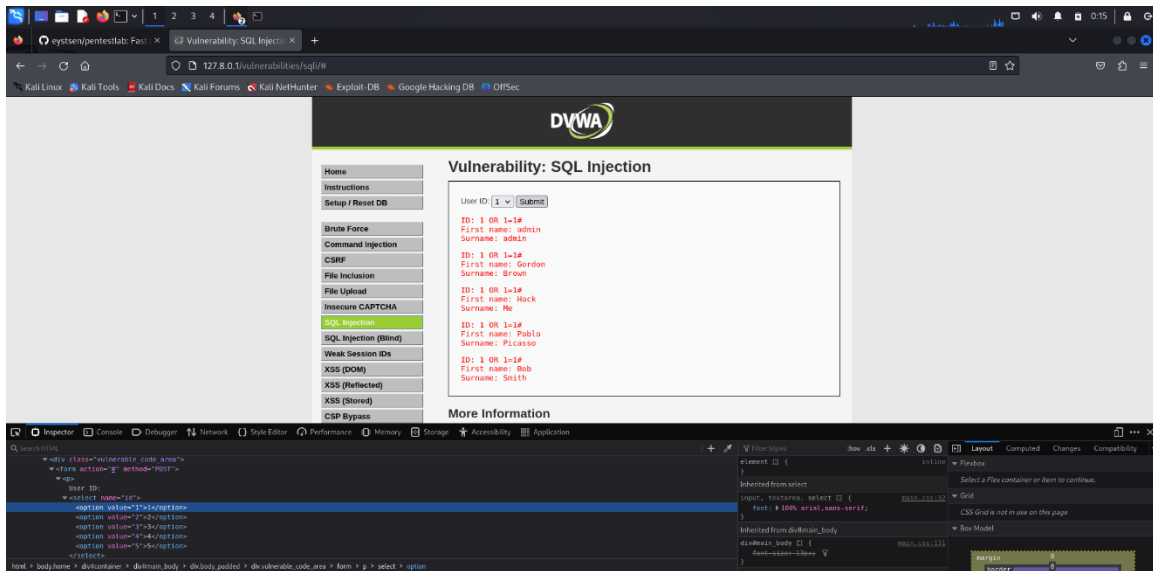


Then I went to research for sql injection through developer tools contents and used pentestGPT.

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



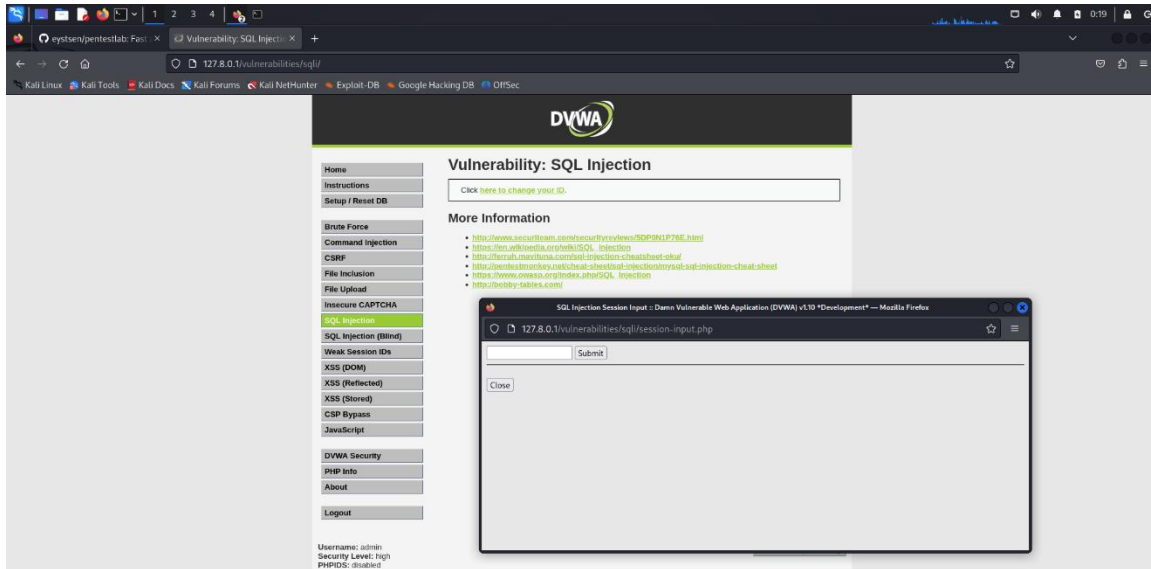
Then on the next page.....



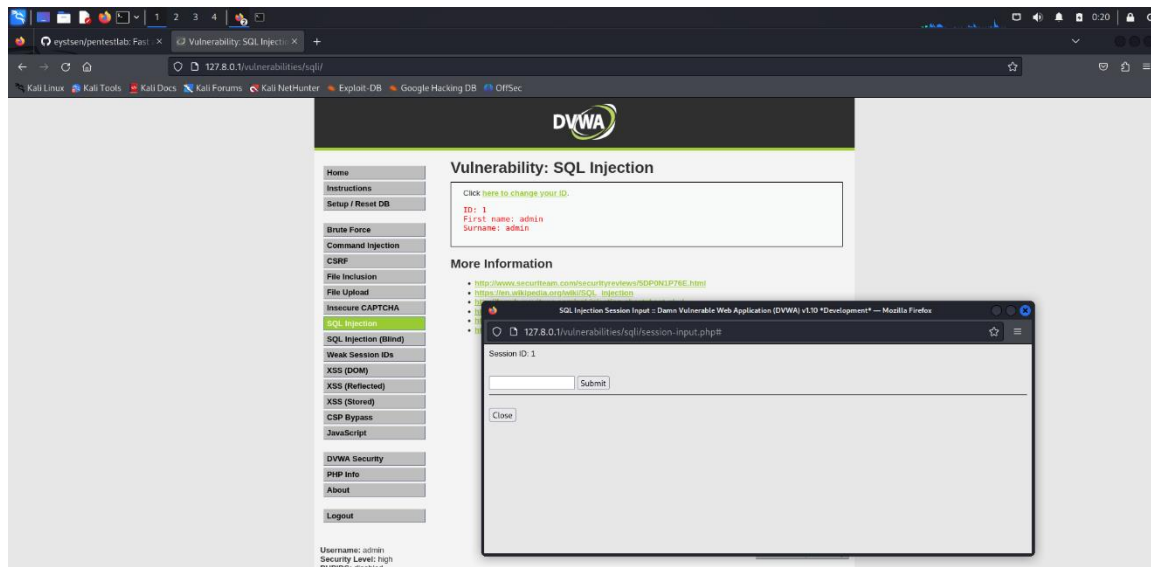
I got the first and last name of every user.

## SQL injection(High)

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.



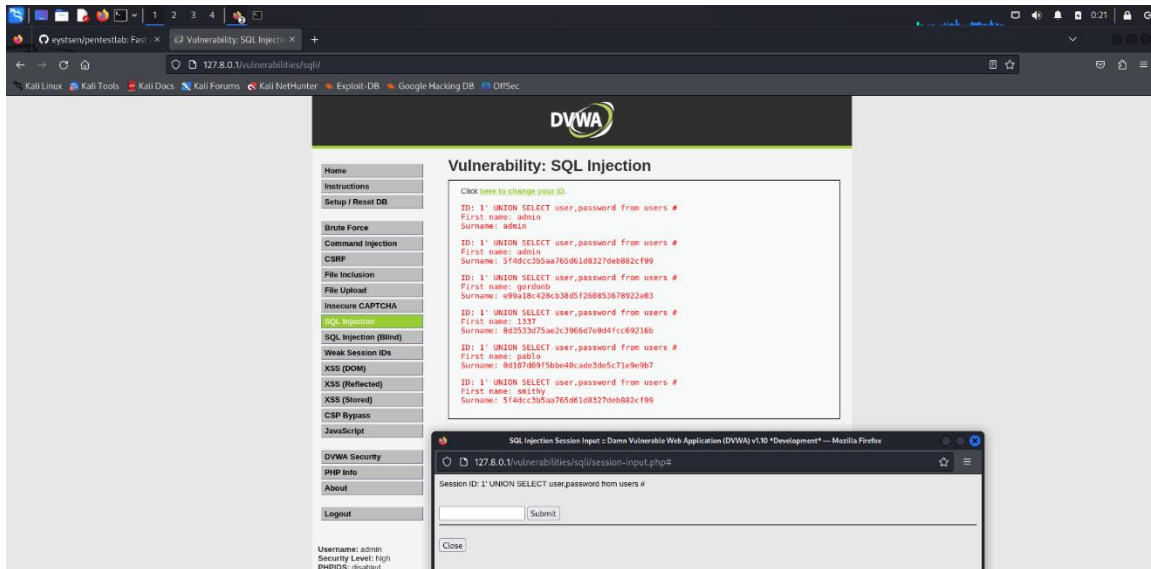
In that window there was a box to text session id. I enter 1 and gave the first and surnames of user number 1.





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

**I' 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.



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 different security levels. Using simple and advanced SQL injection payloads, along with Burp Suite for request interception, I was able to extract sensitive information from the database across all security settings, demonstrating the effectiveness of these attacks