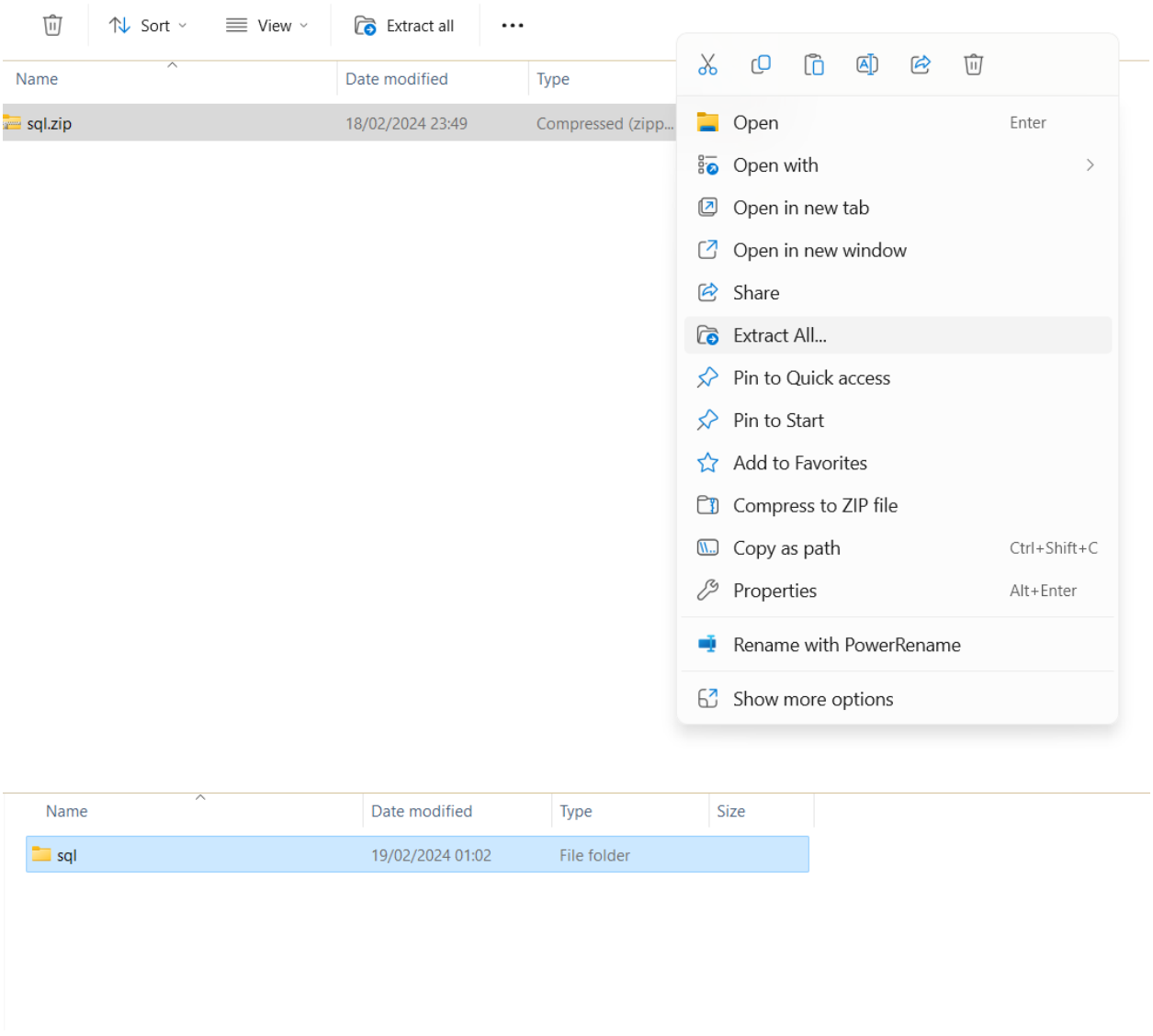
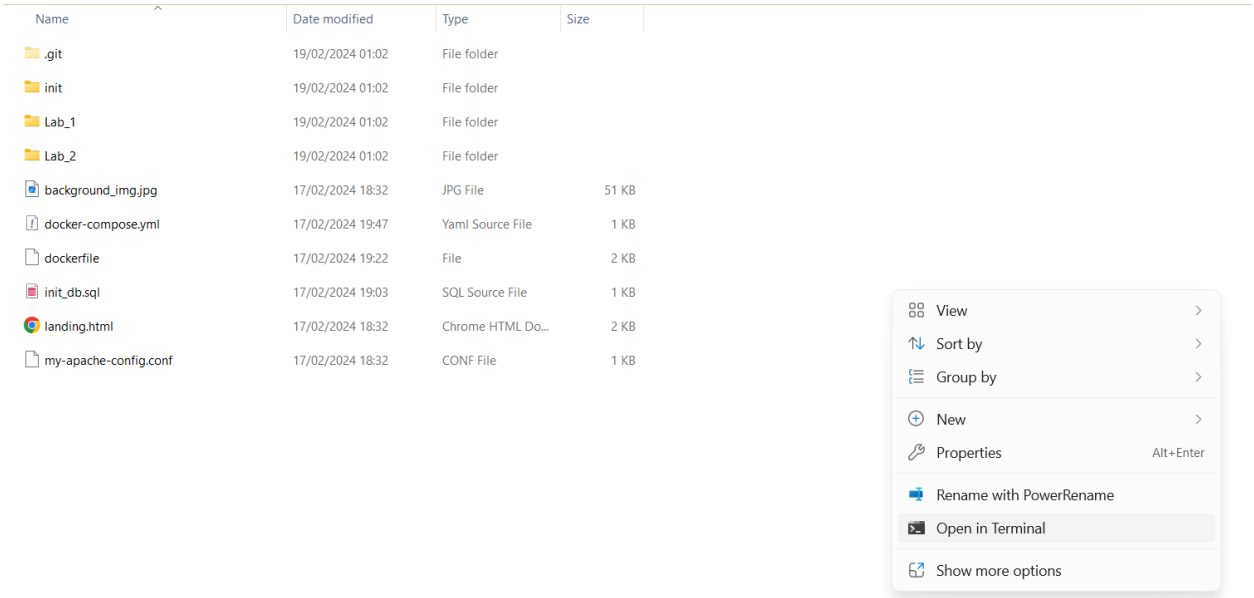
LAB setup

First, you have to Install docker on your system.

Unzip the folder and open the terminal or CMD on that directory.



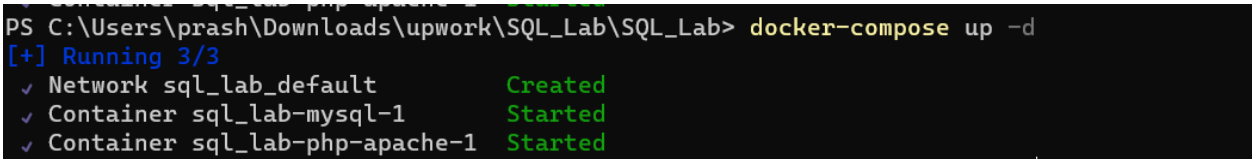
Open Terminal



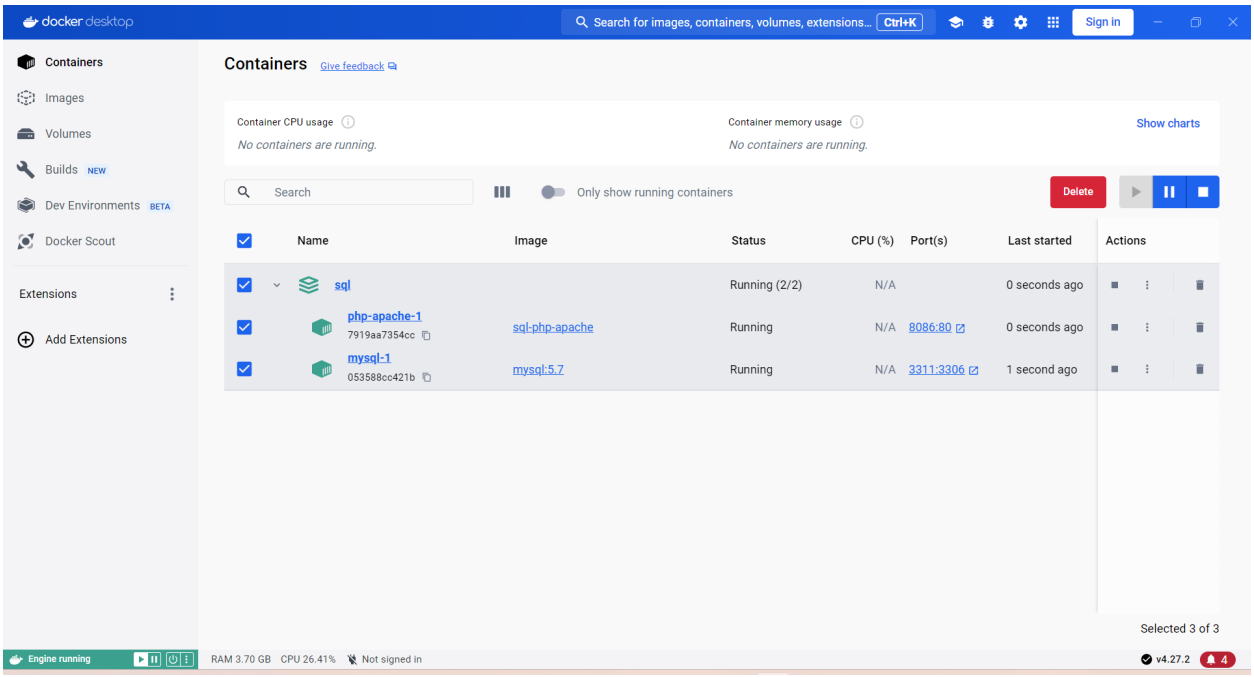
Run this command.

```
docker-compose up -d
```

for Windows



If you are using Windows check docker desktop. and you found that 2 containers are running one Apache with PHP and another with mysql.



You can also check the running containers with docker ps command it work in both linux and Windows.

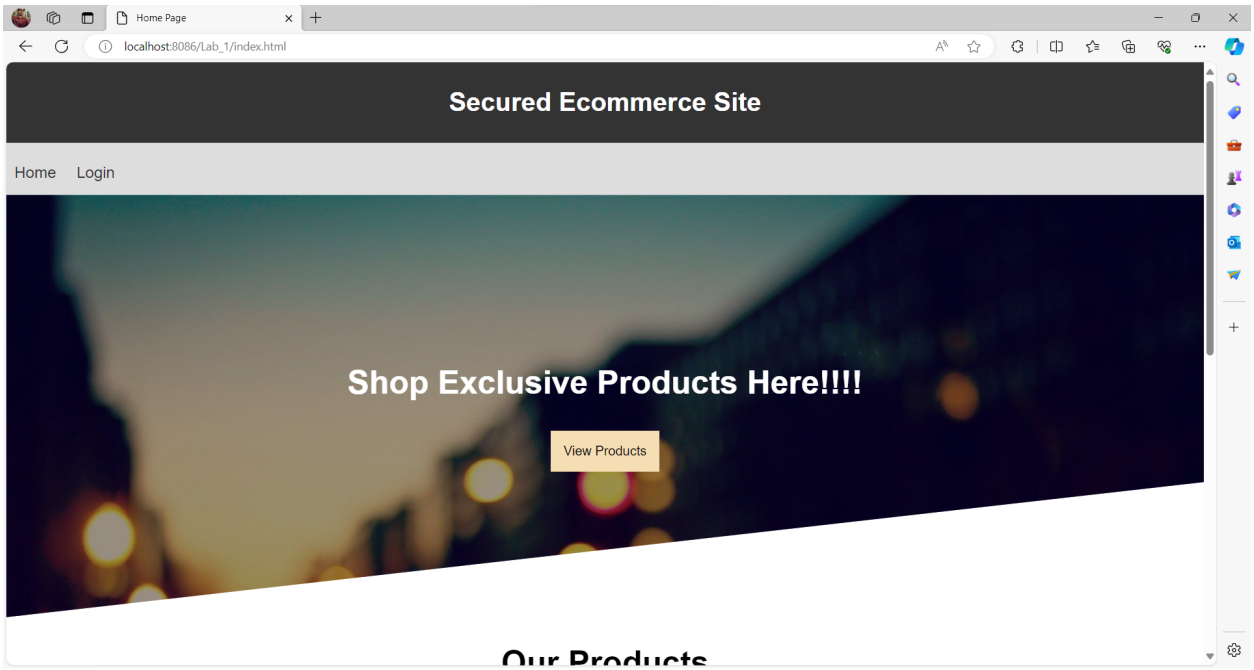
```
# To check running containers
docker ps
```

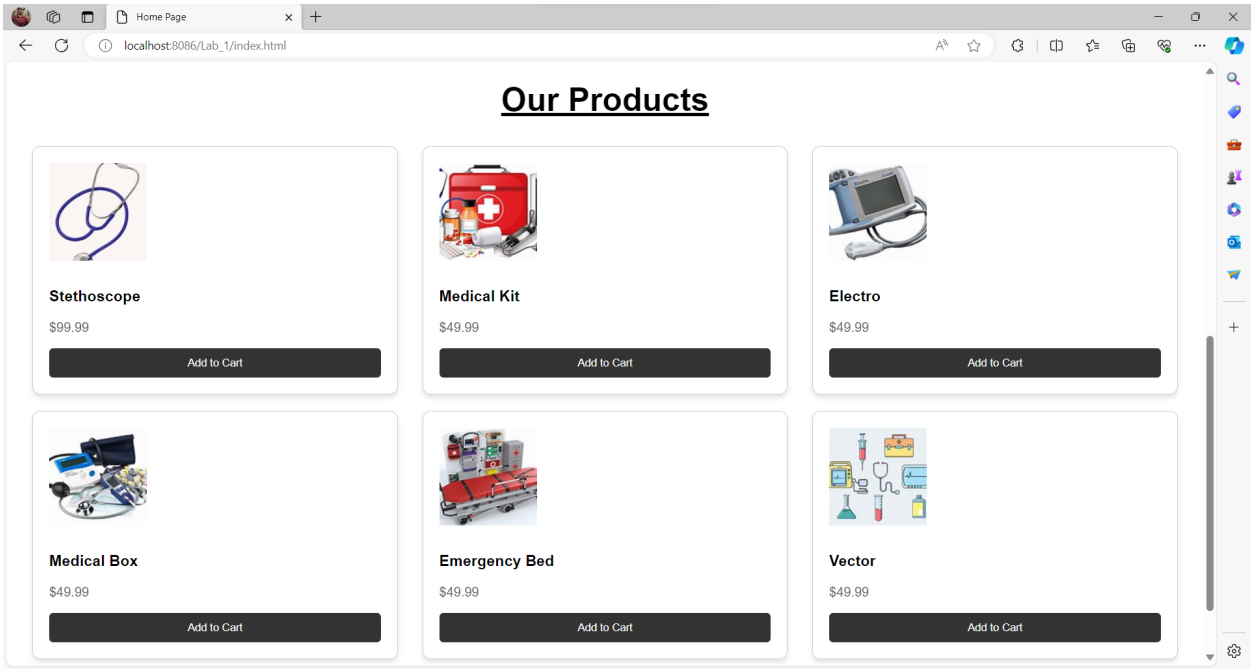
```
C:\Users\prash>docker ps
CONTAINER ID   IMAGE            COMMAND                  CREATED       STATUS         PORTS                                 NAMES
7919aa7354cc   sql-php-apache   "docker-php-entrypoi…"   22 hours ago  Up 6 minutes   0.0.0.0:8086->80/tcp                  sql-php-apache-1
053588cc421b   mysql:5.7        "docker-entrypoint.s…"   22 hours ago  Up 6 minutes   33060/tcp, 0.0.0.0:3311->3306/tcp     sql-mysql-1
```

The port of the web application is mapped to 8086. If any other service is running on port 8086 in the host OS you have to stop it and make it free.

Visit localhost:8086 to open the Lab. If it shows like the below Image you have successfully setup the Lab.

The vulnerable web application lab is based on a medical site.





for Linux

kali Linux

```
┌──(kali㉿kali)-[~/Desktop/SQL_Lab]
└─$ sudo docker-compose up -d
Creating network "sql_lab_default" with the default driver
Creating volume "sql_lab_db_data" with default driver
Pulling mysql (mysql:5.7)...
5.7: Pulling from library/mysql
20e4dcae4c69: Pull complete
1c56c3d4ce74: Pull complete
e9f03a1c24ce: Pull complete
68c3898c2015: Pull complete
6b95a940e7b6: Pull complete
90986bb8de6e: Pull complete
ae71319cb779: Pull complete
ffc89e9dfd88: Pull complete
43d05e938198: Downloading [==================================================>]  55.95MB/56.29MB
064b2d298fba: Download complete
df9a4d85569b: Download complete
```

Wait till the image is downloaded and the installation process.

The web app is running on port 8086.

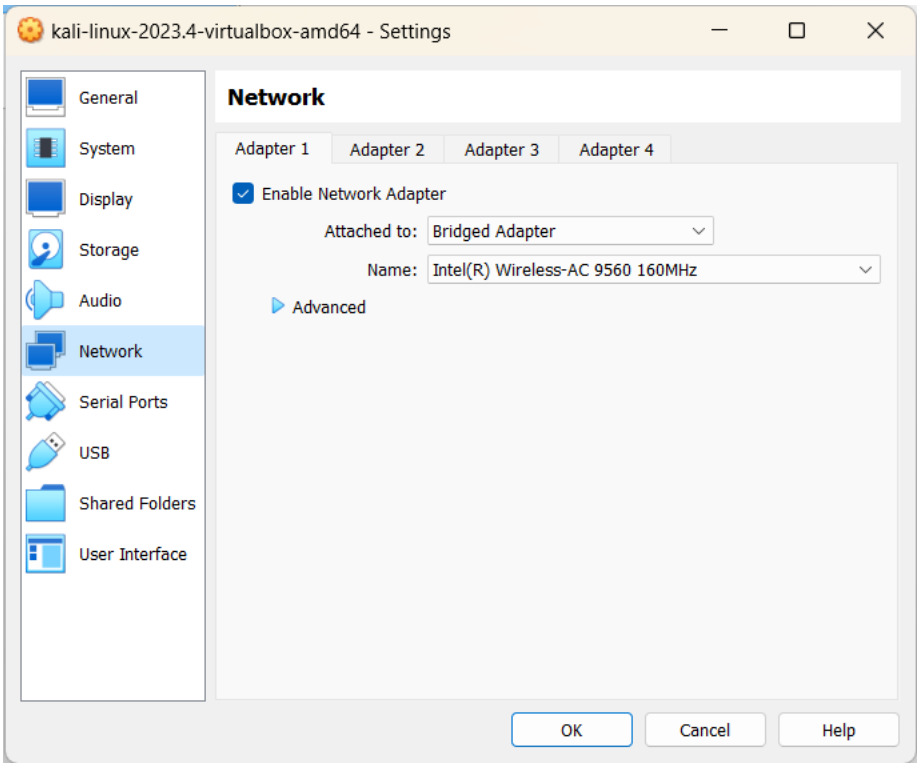
Ubuntu

```
pk@pk-1-2:~/Downloads/sql$ sudo docker-compose up -d
Creating network "sql_default" with the default driver
20e4dcae4c69: Pull complete
1c56c3d4ce74: Pull complete
e9f03a1c24ce: Pull complete
68c3898c2015: Pull complete
6b95a940e7b6: Pull complete
90986bb8de6e: Pull complete
ae71319cb779: Downloading [==========================>                        ]  13.89MB/25.53MB
90986bb8de6e: Download complete
43d05e938198: Download complete
064b2d298fba: Download complete
df9a4d85569b: Download complete
```

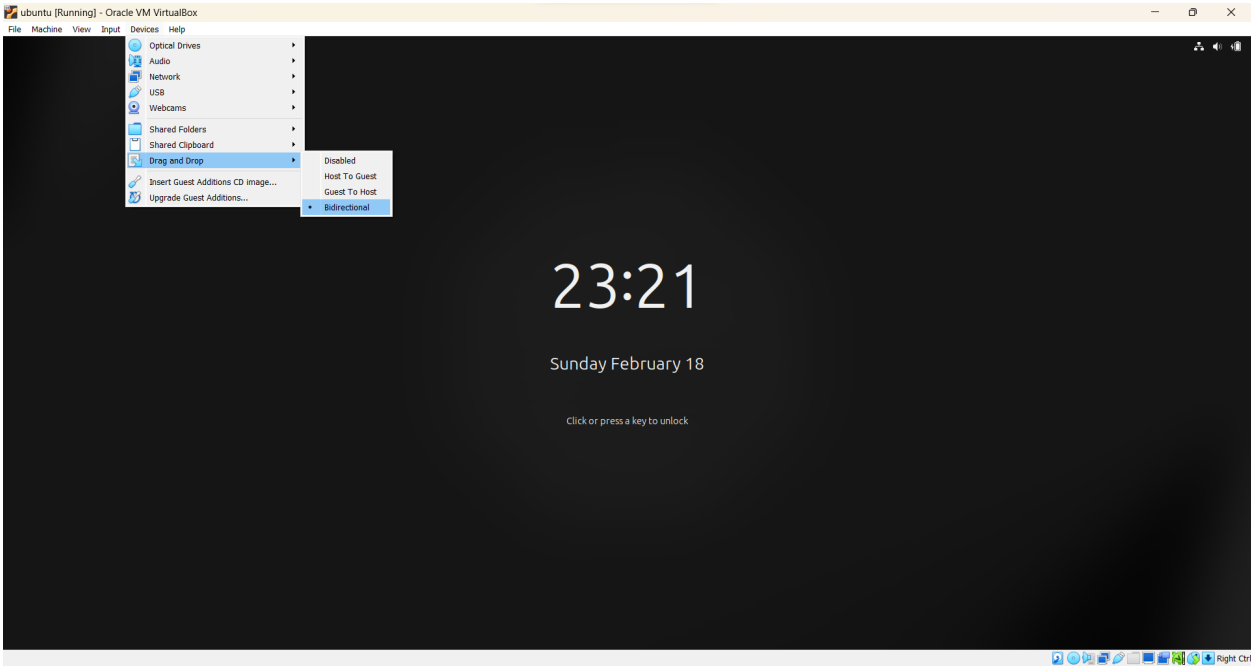
```
pk@pk-1-2:~/Downloads/sql$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS              PORTS                                                  NAMES
be01a0b7c9dd   sql_php-apache   "docker-php-entrypoi…"   About a minute ago   Up About a minute   0.0.0.0:8086->80/tcp, :::8086->80/tcp                  sql_php-apache_1
3c68e111824b   mysql:5.7       "docker-entrypoint.s…"   About a minute ago   Up About a minute   33060/tcp, 0.0.0.0:3311->3306/tcp, :::3311->3306/tcp   sql_mysql_1
```

You can deploy the container in any OS and test the application security using kali Linux or any other Linux-based distro. But use the Kali Linux OS because it has pre-installed most of the tools related to the cyber security domain.

Deploy the docker container with ***docker-compose up -d*** this command and make sure if you are running containers in different OS they have the same network or use a bridge connection if you are using a virtual machine (so the attacker OS i.e Kali Linux can make communication with target web application).



If you are using a container inside the guest OS enable the Drag and drop option bidirectional so you can transfer the zip file easily from host to guest.



The question is in simple format.

1>You have to scan the application list out all the open ports and take a screenshot of the open port and which tool are you using for it.

2>Have you found all the ports? use the switch which scans all the ports.

3>Which switch is used for detail verbose mode?

4>You have to scan all the port ranges, version, OS detection take screenshots, and write down all the switches you used.

5>What is the most commonly used for encoding binary data, decode the secret message from web application source code make screenshot of it, and how you did do this?

6>What do you get after decoding it make a screenshot of it?

7>Test the admin panel security of the web application. How do you exploit it? Which tools are used to automate the payload test to exploit the login page, make a screenshot of it. Write down a detailed solution.

8>Write down some payloads that allow you to exploit it at least 3.

9>Have you gotten access to the admin dashboard take a snapshot of it. What is the final flag present inside the admin dashboard take a snapshot of it.

Questions in detailed format

Network Port Analysis:

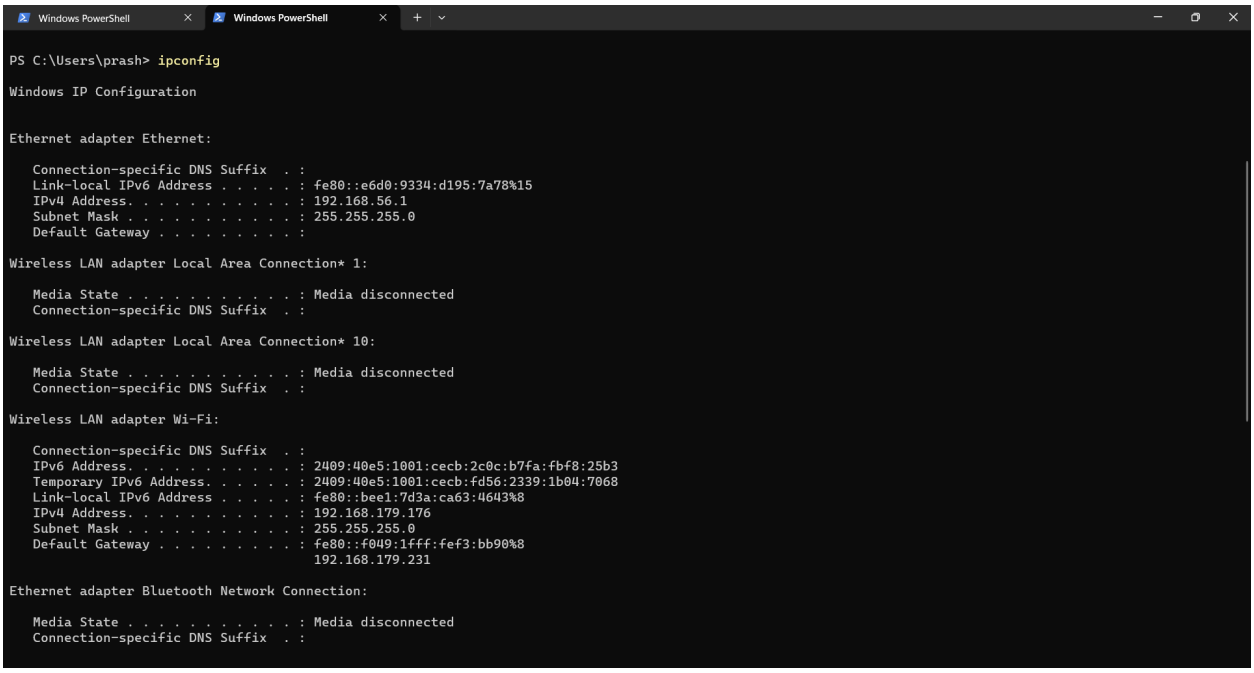
- Task: Conduct a comprehensive scan of the application to identify all open network ports. Capture a screenshot displaying the open ports. Specify the tool utilized for this scan.
- This task involves using network scanning tools to discover open ports on the target application. Document your process by taking a screenshot of the results, which should include the list of open ports, and clearly mention the tool(s) you employed for this scan

Solution:

You have to scan the IP where the docker container is deployed. Here we deploy the container in Windows and the attacker machine is Kali Linux.

You can get the Windows IP by ipconfig command

```
ipconfig
```



Let's scan this by **NMAP** tool.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap  192.168.179.176
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-18 15:18 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.03 seconds
```

Let's use -Pn switch because it's blocking ping.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap  192.168.179.176 -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-18 16:04 EST
Nmap scan report for 192.168.179.176
Host is up (0.013s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
2179/tcp open  vmrdp
8086/tcp open  d-s-n

Nmap done: 1 IP address (1 host up) scanned in 7.21 seconds
```

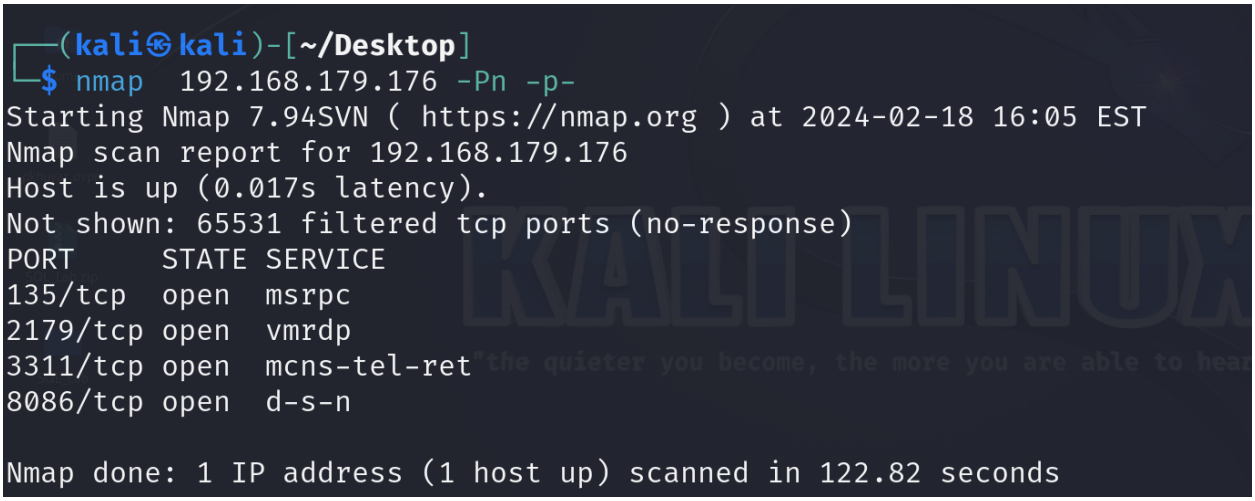
Exhaustive Port Scanning:

- Task: Ensure the identification of all available ports on the application. Employ the specific switch that enables a complete scan across all port ranges.
- This question directs you to perform an exhaustive scan to uncover every open port on the application. Identify and use the command-line switch that allows the scanning tool to check all 65535 ports for any TCP/UDP services.

Solution:

To scan all port range use -p- flag

```
nmap -p- TARGET
```



Verbose Output for Detailed Analysis:

- Task: Identify the switch/command used to enable detailed verbose output during scans.
- When conducting scans, detailed output can be crucial for thorough analysis. Specify the switch or command that provides a verbose mode, offering in-depth details during the scanning process.

Solution:

To enable verbose mode while scanning use -v flag for simple verbose mode and -vv for more detail verbose mode.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap  192.168.179.176 -Pn -p- -vv
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-18 16:11 EST
Initiating Parallel DNS resolution of 1 host. at 16:11
Completed Parallel DNS resolution of 1 host. at 16:11, 0.01s elapsed
Initiating Connect Scan at 16:11
Scanning 192.168.179.176 [65535 ports]
Discovered open port 135/tcp on 192.168.179.176
Discovered open port 2179/tcp on 192.168.179.176
```

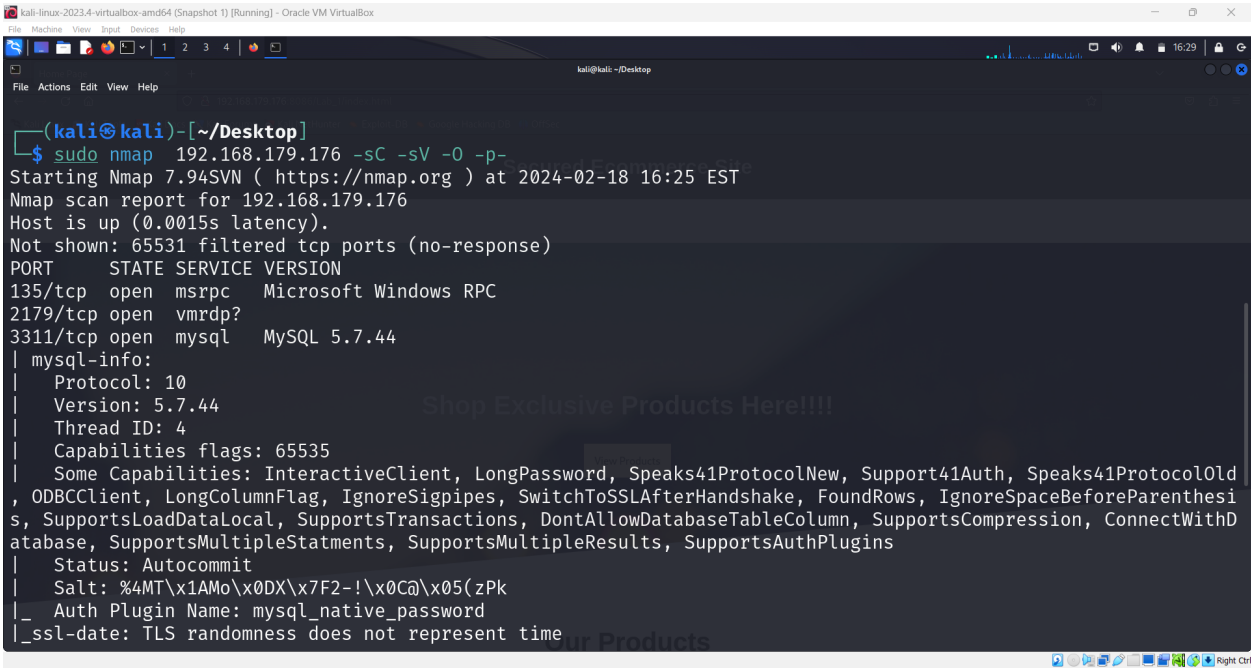
Comprehensive Scanning and Documentation:

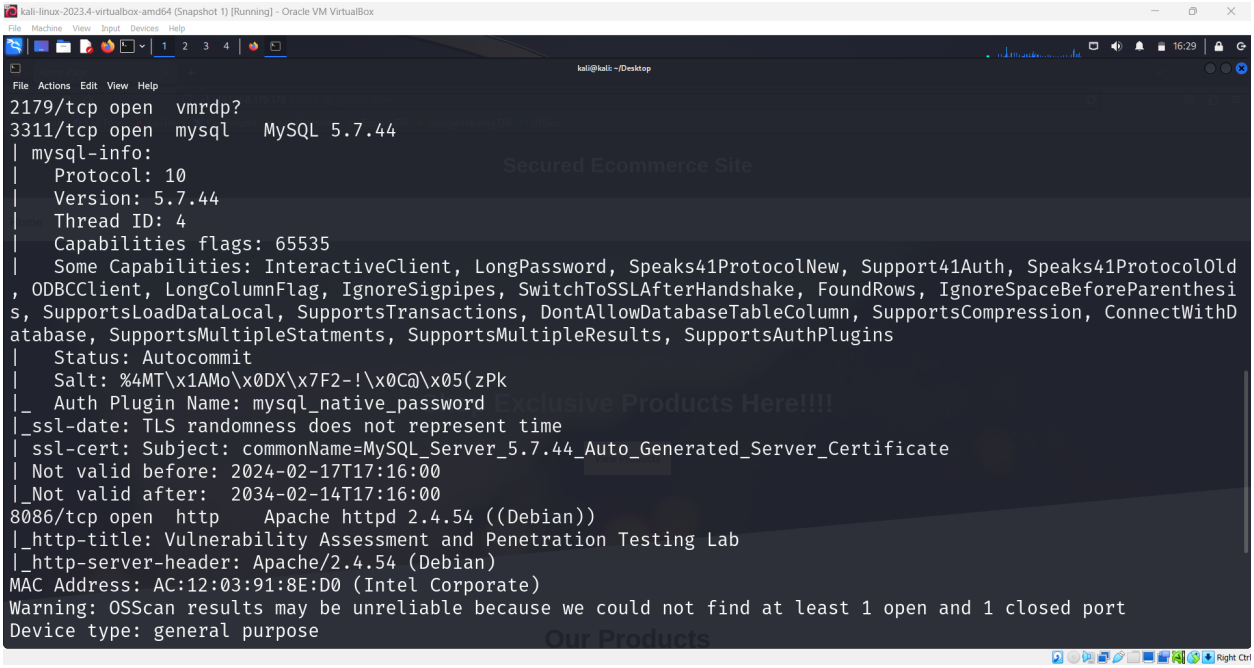
- Task: Execute a scan that covers all port ranges, version detection, and operating system identification. Capture screenshots of your findings and note all switches or commands used during this process.
- This question requires a multifaceted approach to scanning, demanding not just an identification of open ports but also software versions and operating system details. Document your methodology and results comprehensively, including screenshots and a detailed list of all command-line options utilized.

Solution:

The `nmap` command with the `sC` switch enables default script scanning, allowing Nmap to run a selection of scripts against discovered ports for service enumeration and vulnerability detection. With `sV`, Nmap performs version detection, probing target ports to identify the software versions running on them. Adding `O` triggers OS detection, where Nmap analyzes various characteristics to determine the target system's operating system. Combining these switches provides a comprehensive scan, aiding in network analysis, security assessments, and system administration

-O flag need root privilege so use sudo.





port 135,2179 is open because docker container host OS is windows.

Decoding Secret Messages:

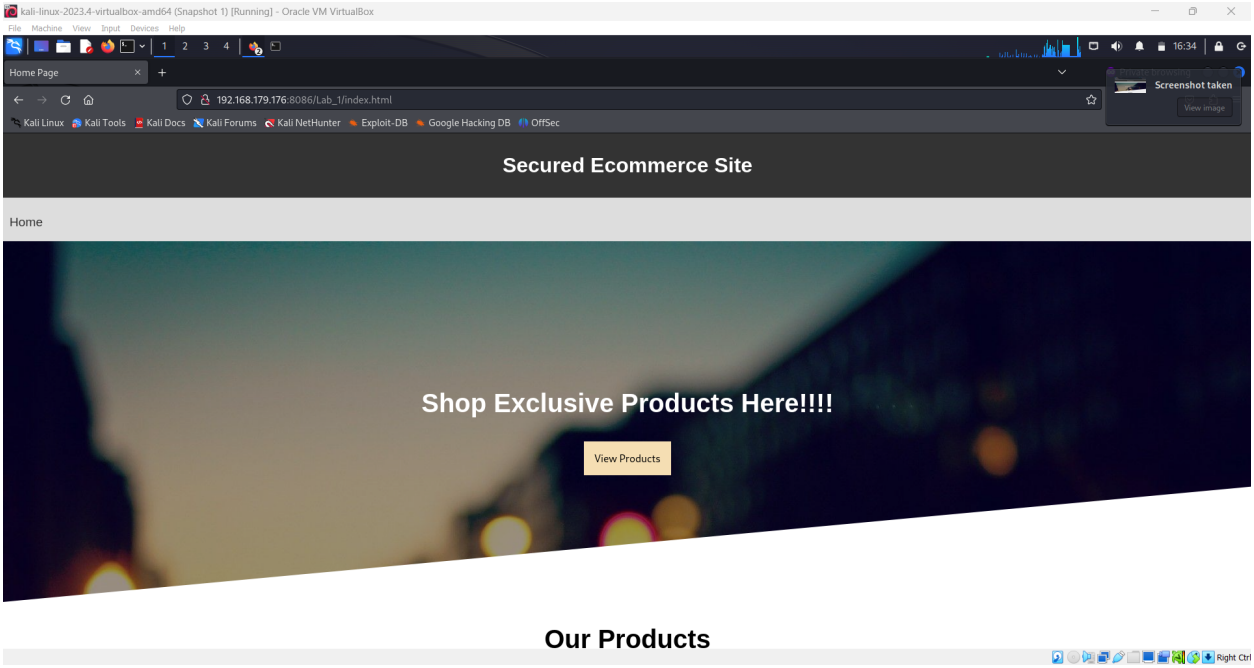
Decoding Secret Messages:

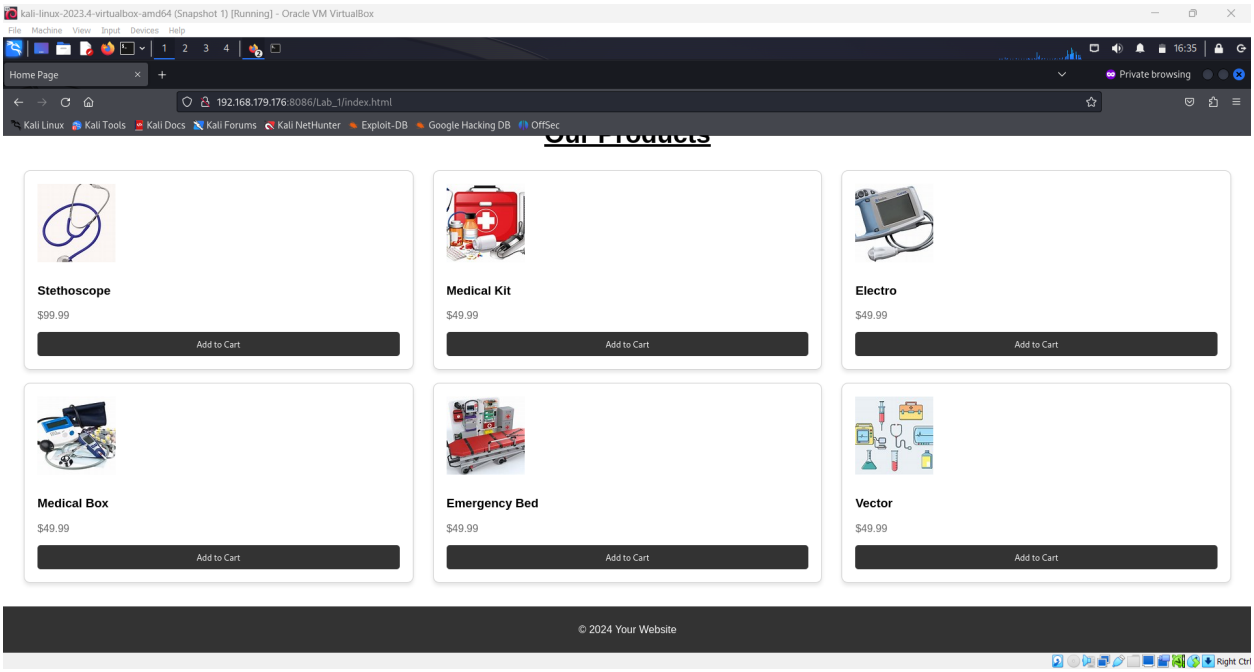
- Task: Identify the most common method used for encoding binary data. Decode the secret message embedded in the web application's source code. Provide screenshots of both the encoded message and the decoding process.
- Focus on the techniques used for encoding binary data within web applications. Once identified, decode the hidden message found within the source code of the web application, capturing the process and outcome through screenshots.

Solution:

We found that port 8086 running the vulnerable web app. Let's investigate this port.

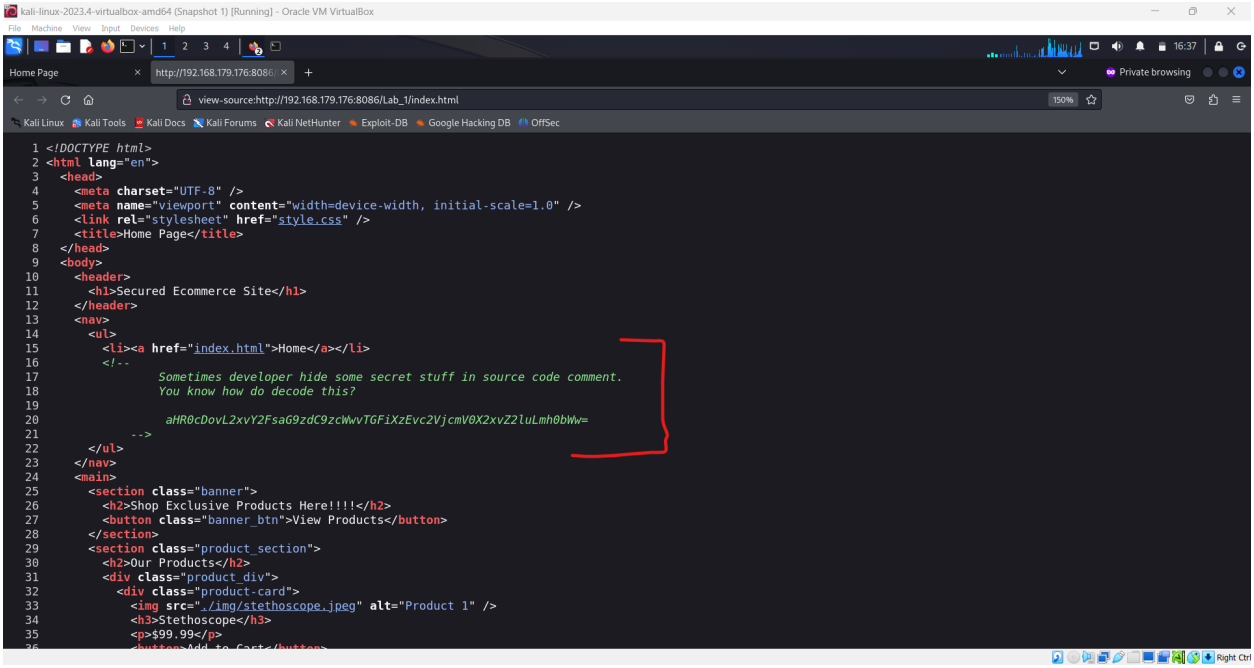
Let's Investigate Lab1.





It's a medical ecommerce site. which is running on port 8086

Check the source code if we get anything interesting.



Outcome of Decoding Efforts:

Outcome of Decoding Efforts:

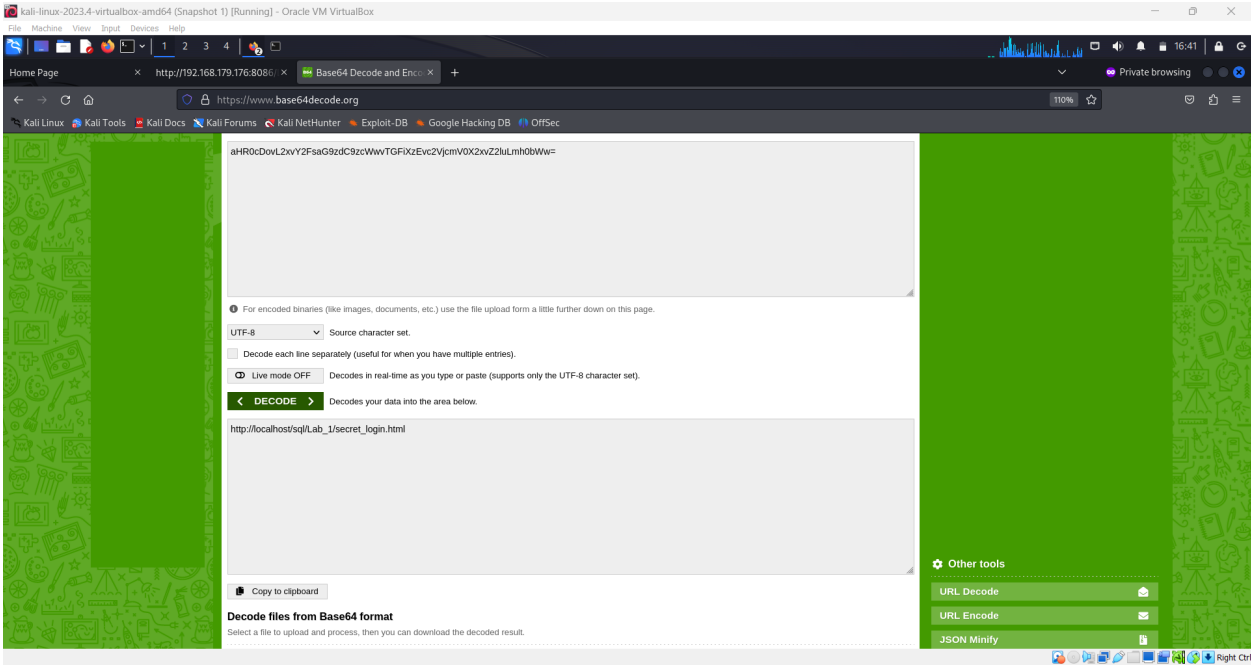
- Task: Present the result obtained after decoding the secret message. Capture a screenshot of the decoded message.
- After successfully decoding the secret message, document the final output. Provide a clear screenshot of the message that was uncovered through your decoding efforts.

Solution:

We have to decode this data which we got in source code of the web app. It's seem base64 encoding.

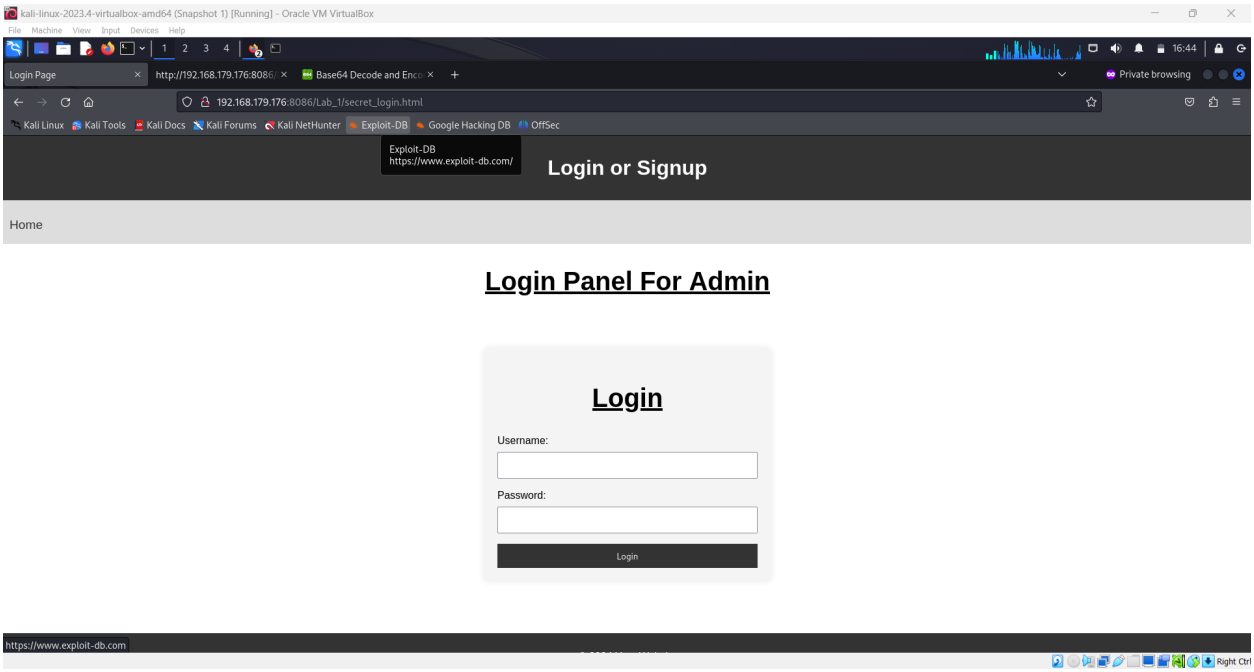
```
aHR0cDovL2xvY2FsaG9zdC9zcWwvTGFiXzEvc2VjcmV0X2xvZ2luLmh0bWw=
```

Use this site to decode it. https://www.base64decode.org/



Let's go to that endpoint.

secret_login.html



We got an admin login page.

Web Application Admin Panel Security Assessment:

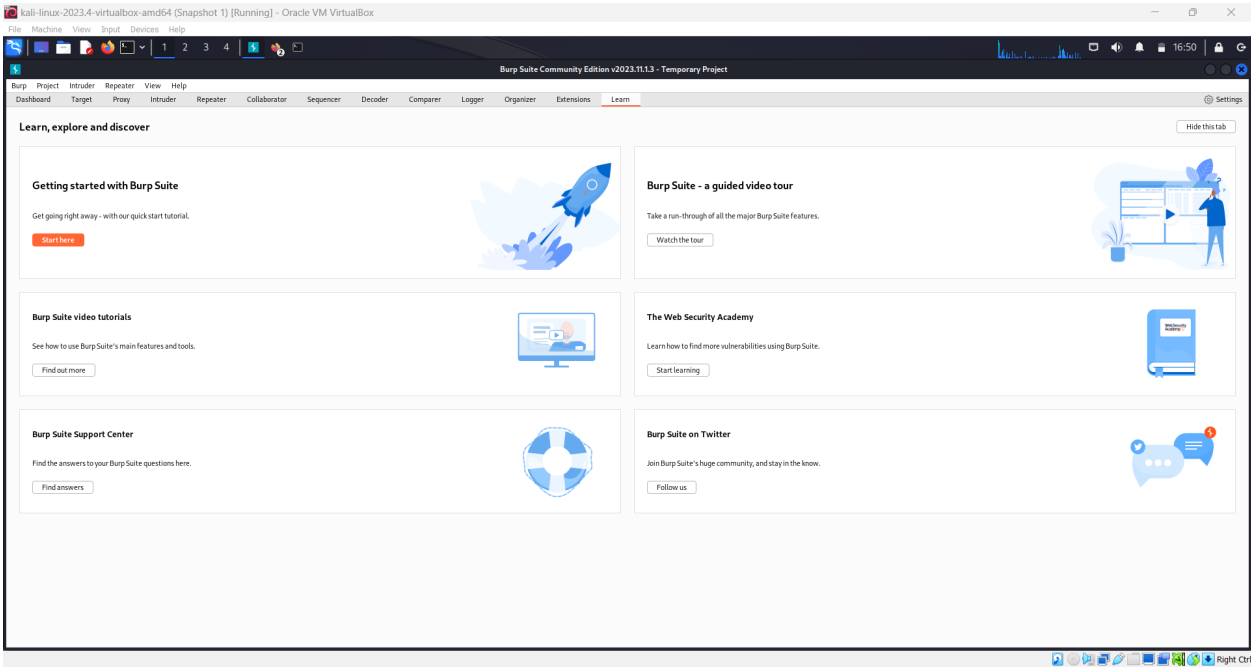
- Task: Evaluate the security of the web application's admin panel. Describe your approach to exploiting potential vulnerabilities. Identify the tools used for automating payload tests to exploit the login page. Document your process with screenshots and a detailed explanation.
- This question challenges you to test and potentially exploit the security of the admin panel within a web application. Outline your strategy for identifying vulnerabilities, the tools employed to automate testing, and the specific payloads used. Capture this process through detailed documentation and screenshots.

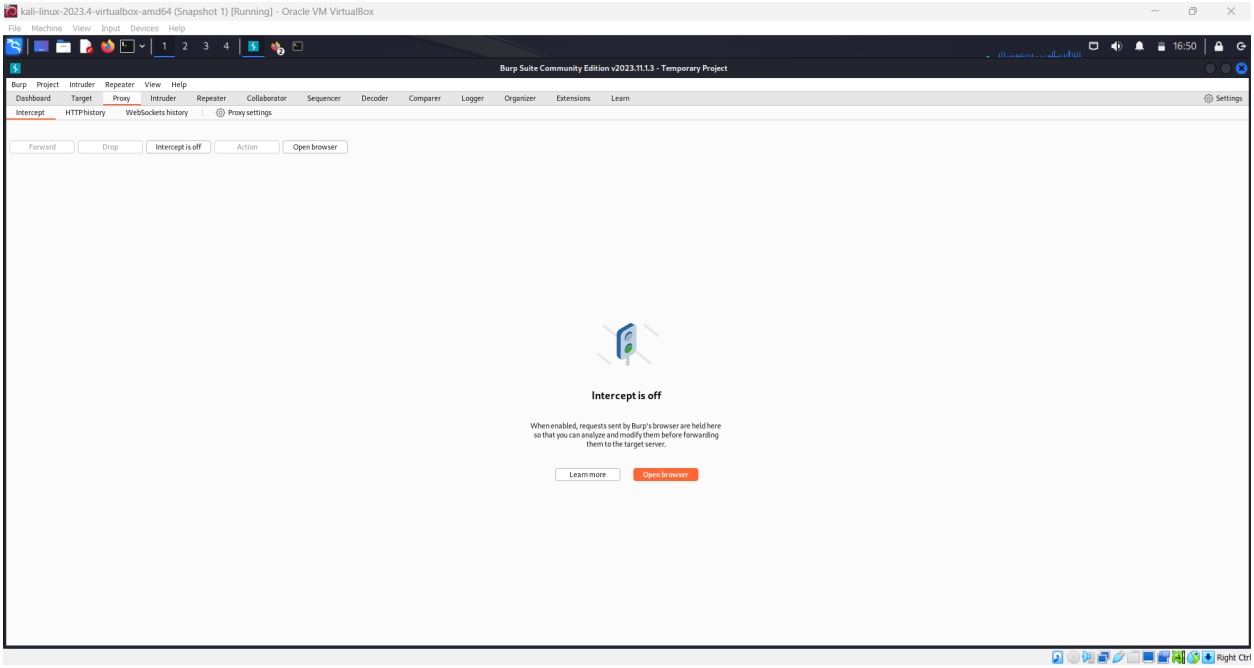
Solution:

Let's try to bypass the admin panel with SQL injection payload. To do this I am using burp proxy to try list of admin bypass payload.

Open the burp suite in Kali Linux.

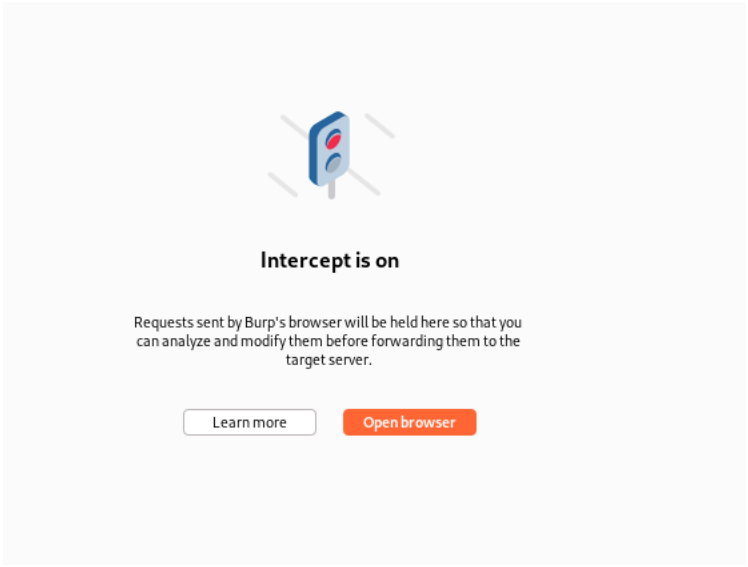
Go to proxy tab



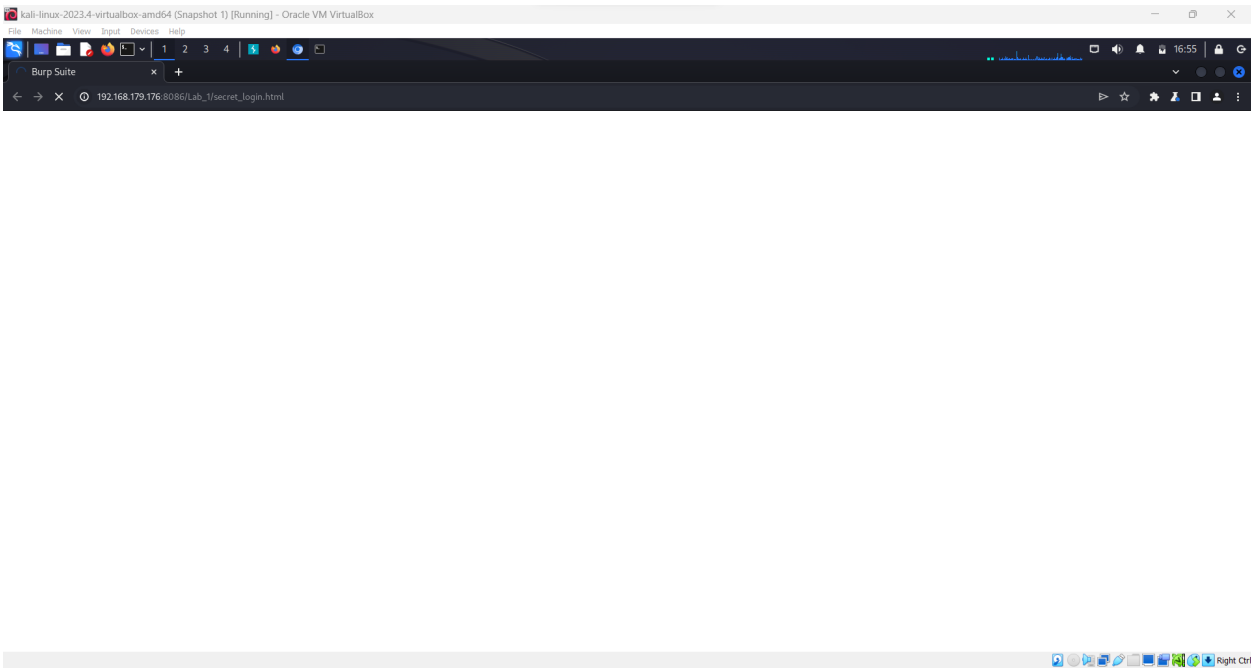


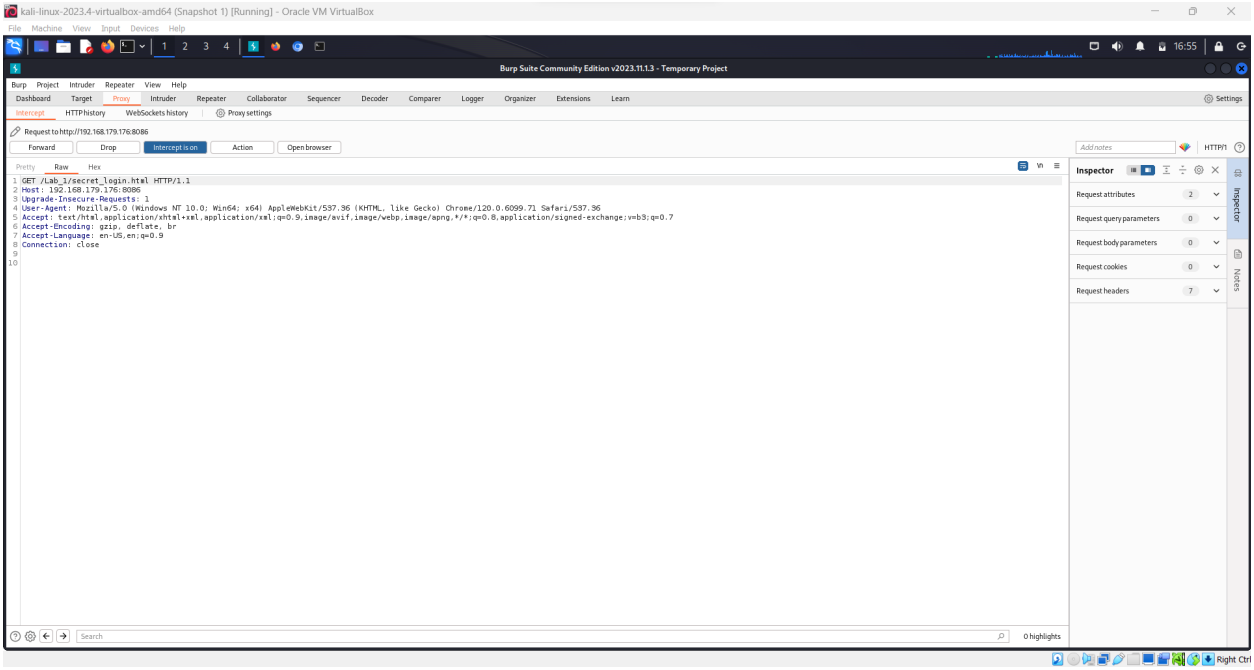
Turn on the intercept.

Open burp embedded browser.



Go to the admin URL of the vulnerable web application.

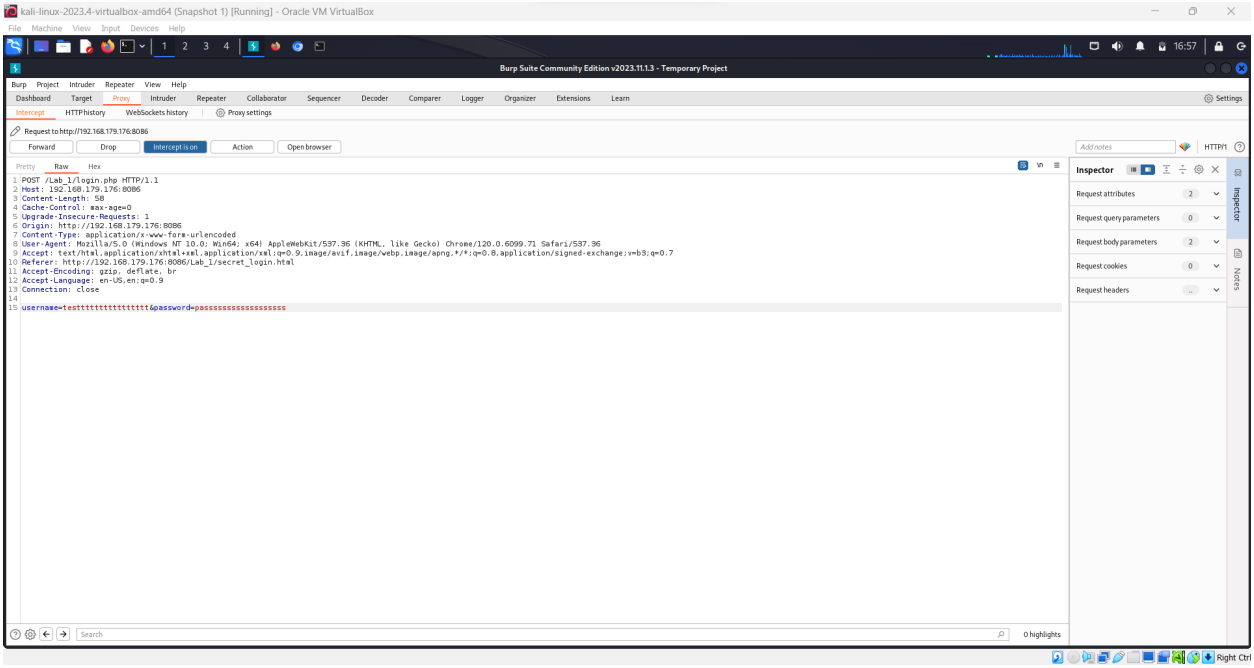




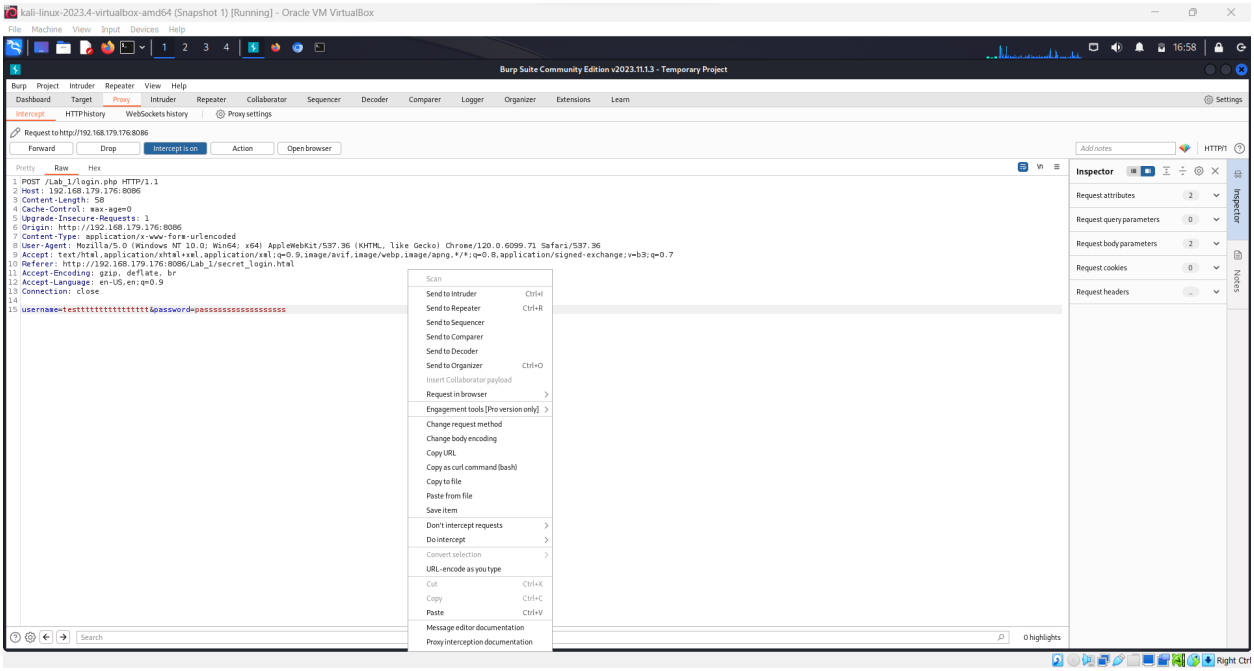
Forward the request from the burp proxy.

enter some test data to the user and password field and submit.

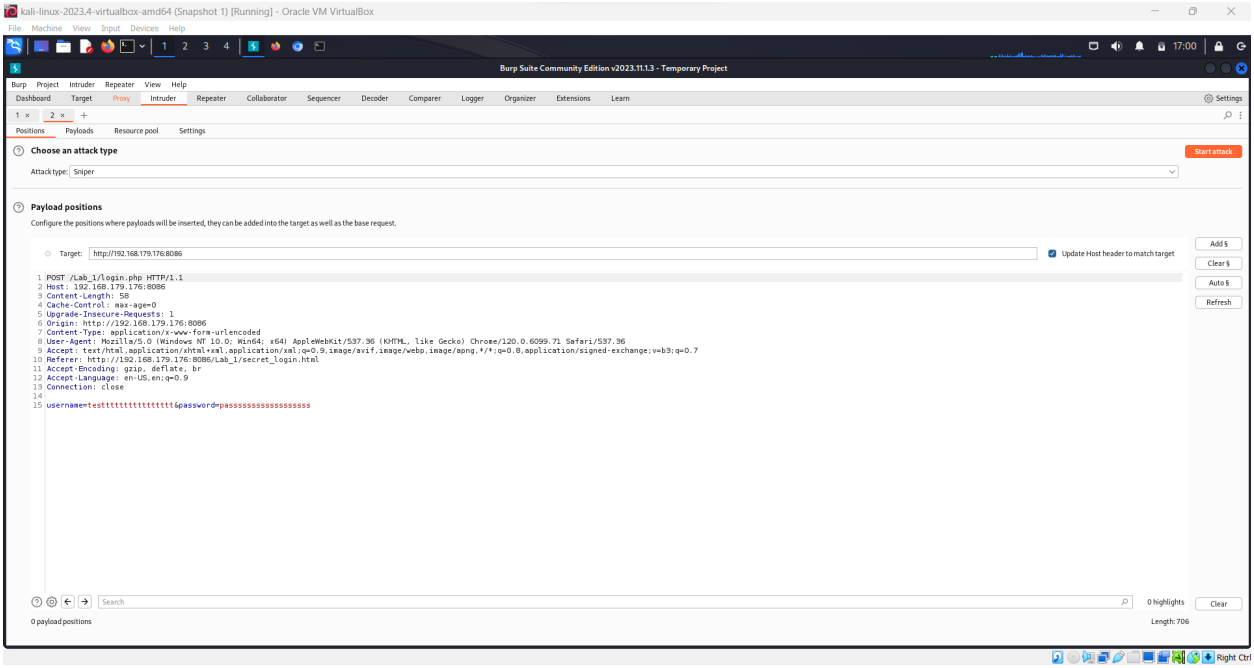
It open the burp proxy and the http request get captured.



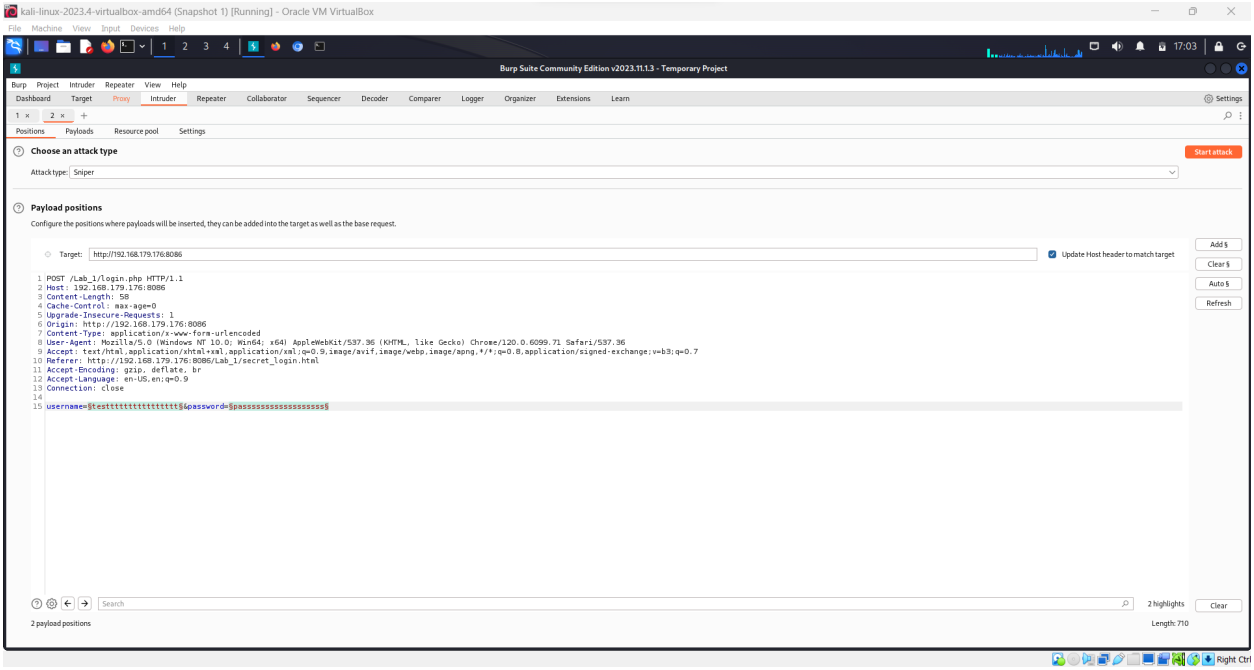
Send this request to intruder by right click on it and chose intruder option.



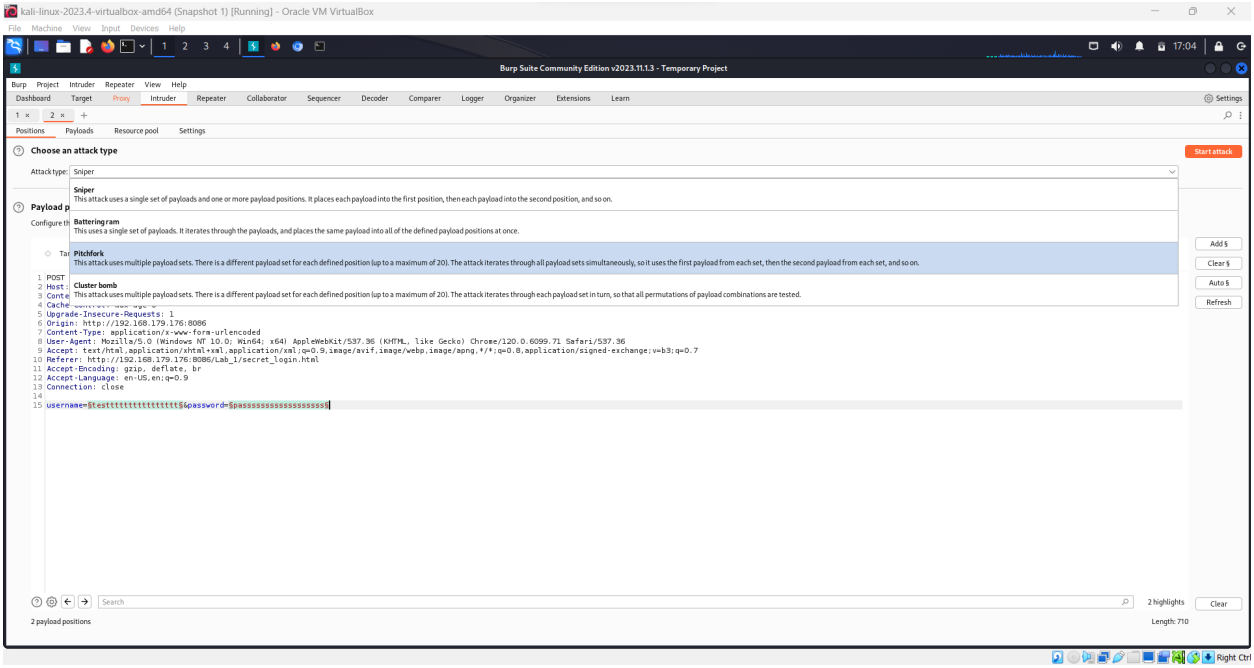
Go to the Intruder tab of burp suit.



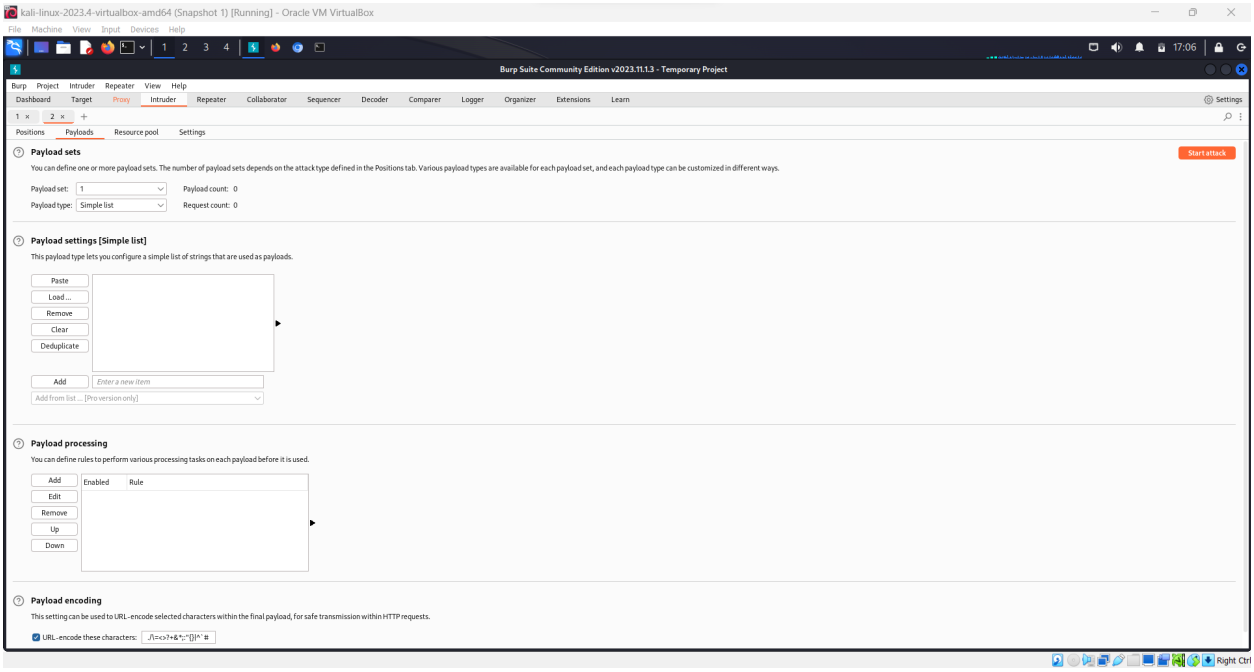
Add the username and password parameter by selecting its value and clicking on add button.



Go to attack type option and chose Pitchfork option.

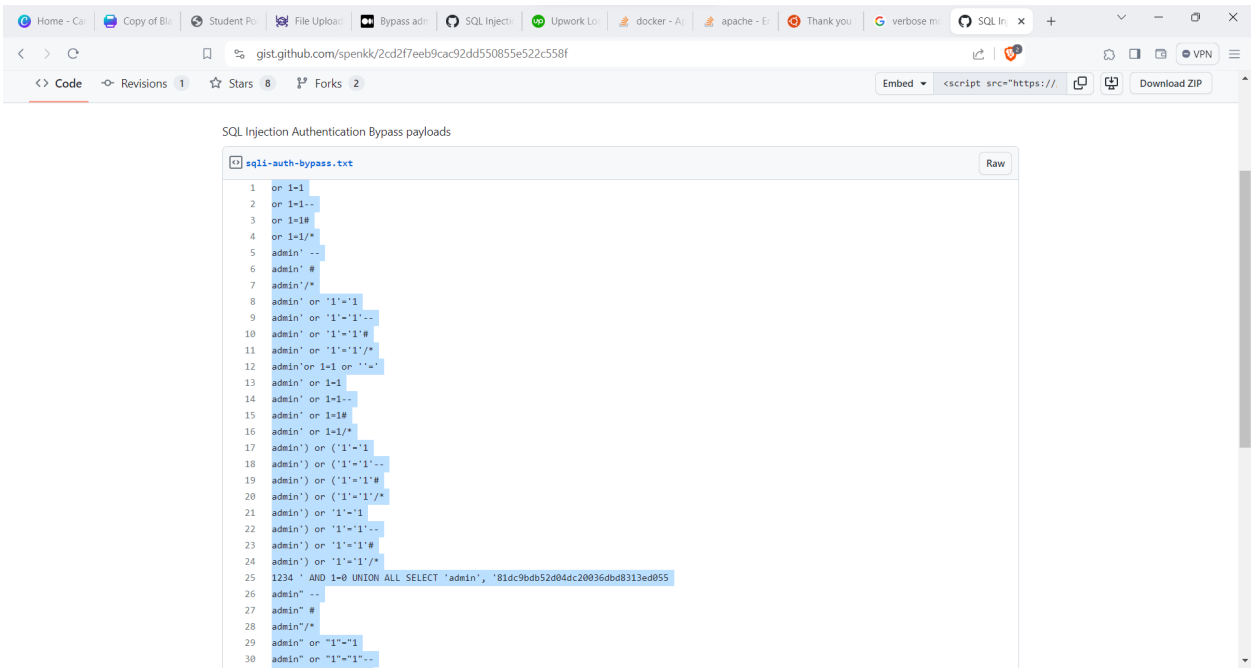


Now go to payload tab.

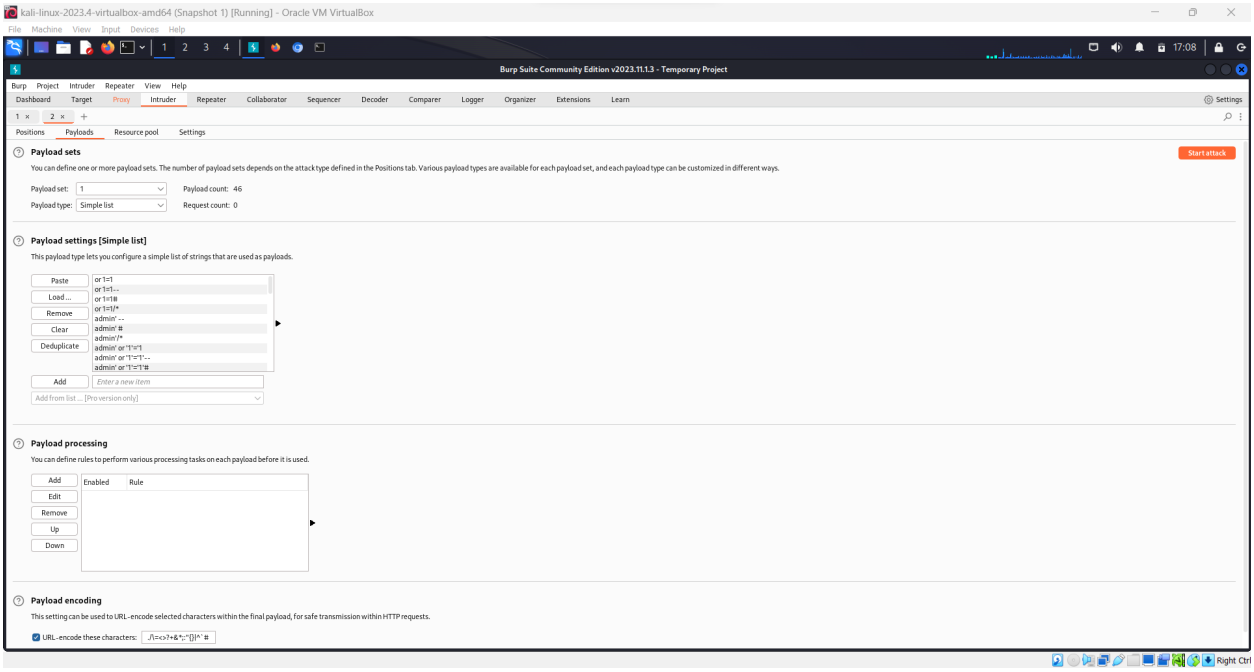


Go to this page and copy all the SQL injection login bypass payload.

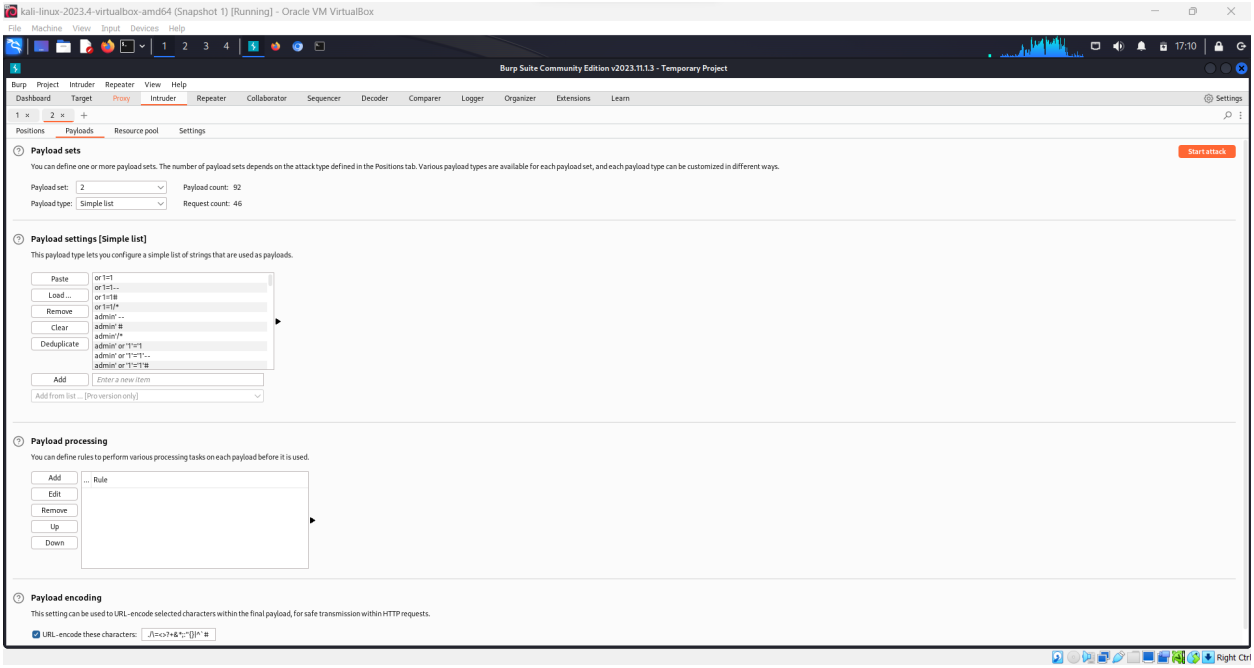
https://gist.github.com/spenkk/2cd2f7eeb9cac92dd550855e522c558f



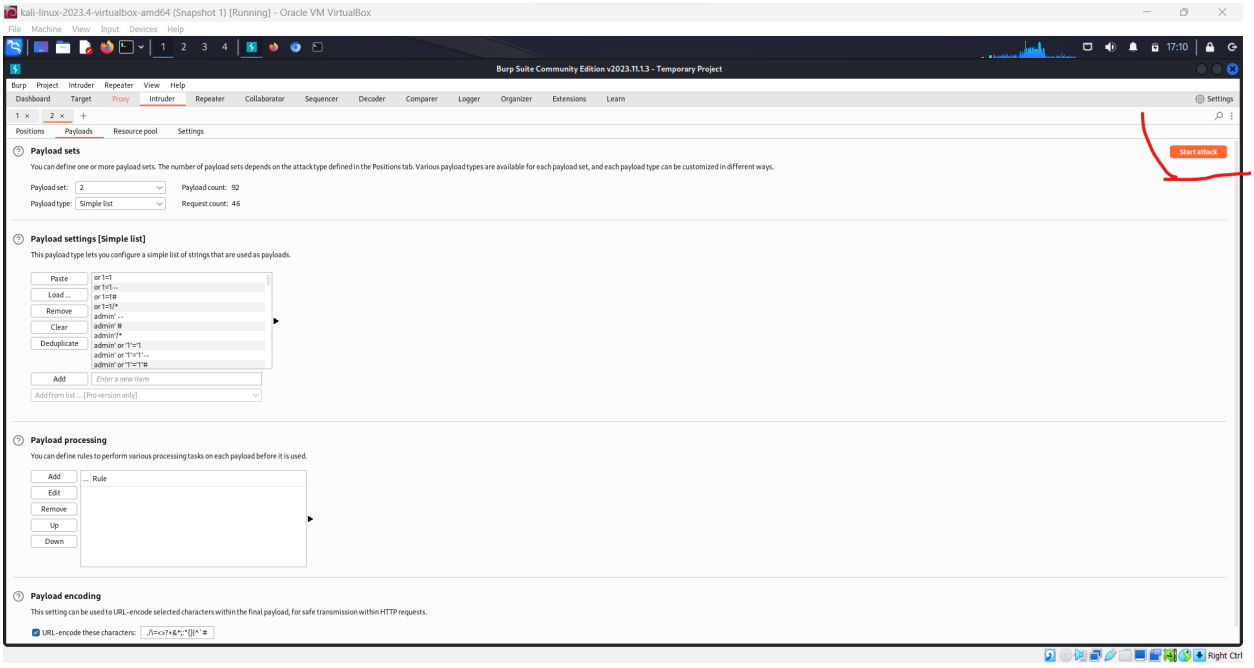
Now go to payload option in burp and click on paste and past all the payloads.



Now go to Payload set 2 and do the same.

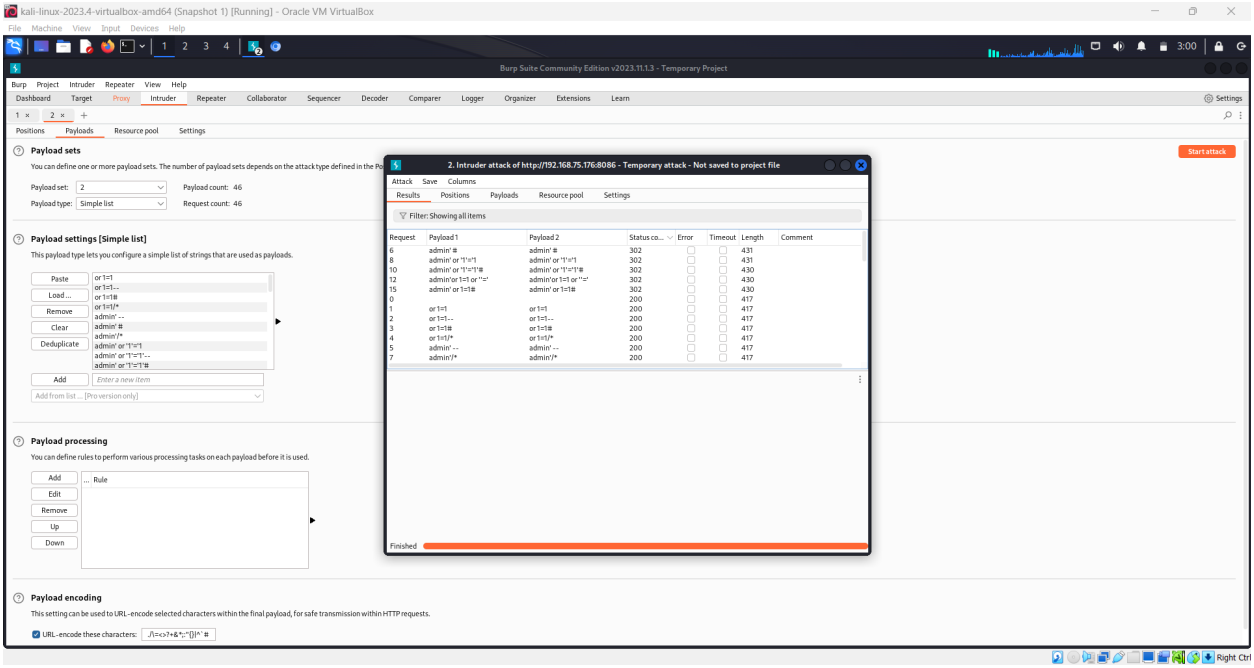


Now click on start attack button.



Now it's try all the payload one by one.

Filter out 302 status code response payloads which show successful exploited and bypass the login page.



These are some payloads which exploit the login page.

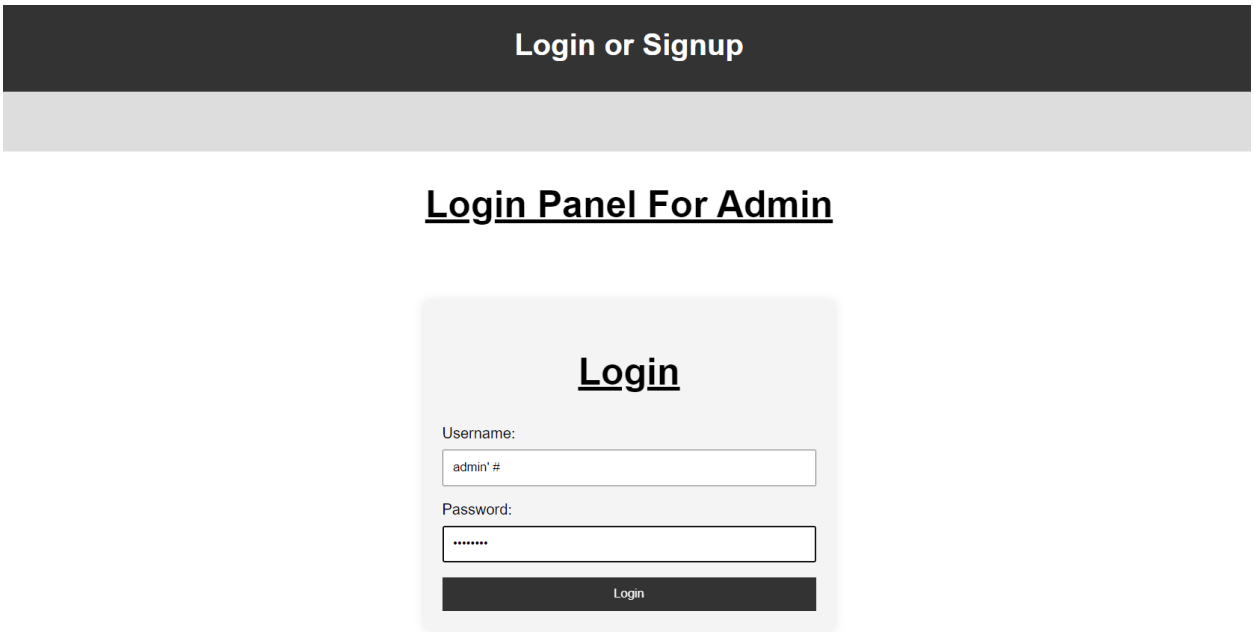
Effective Payloads for Exploitation:

- Task: List at least three payloads that could be used to exploit vulnerabilities in the web application. Provide a detailed explanation of each.
- Develop a list of effective payloads that have the potential to exploit known or discovered vulnerabilities within the web application. For each payload, offer a detailed description of its purpose and how it can be applied in an attack scenario.

These are some payloads of that successfully exploit the login page.

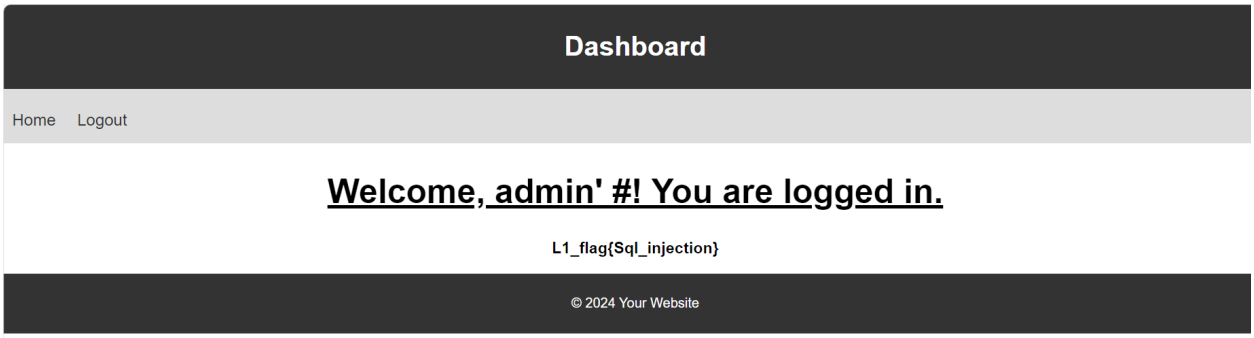
Let try any of them

6	admin' #	admin' #	302	false	false	431	
8	admin' or '1'='1	admin' or '1'='1	302	false	false	431	
10	admin' or '1'='1'#	admin' or '1'='1'#	302	false	false	430	
12	admin'or 1=1 or ''='	admin'or 1=1 or ''='	302	false	false	430	
15	admin' or 1=1#	admin' or 1=1#	302	false	false	430	



Admin Dashboard Access and Final Flag Identification:

- Task: Confirm whether access to the admin dashboard was successfully obtained. Capture a snapshot of the dashboard. Additionally, identify the final flag located within the admin dashboard and provide a screenshot of it.
- This final question requires you to document access to the admin dashboard of the web application, if such access was achieved. Capture evidence of this access and locate the final flag present within the dashboard area. Provide clear screenshots to confirm your findings.



L1_flag{Sql_injection}