ShadowFox Internship Tasks

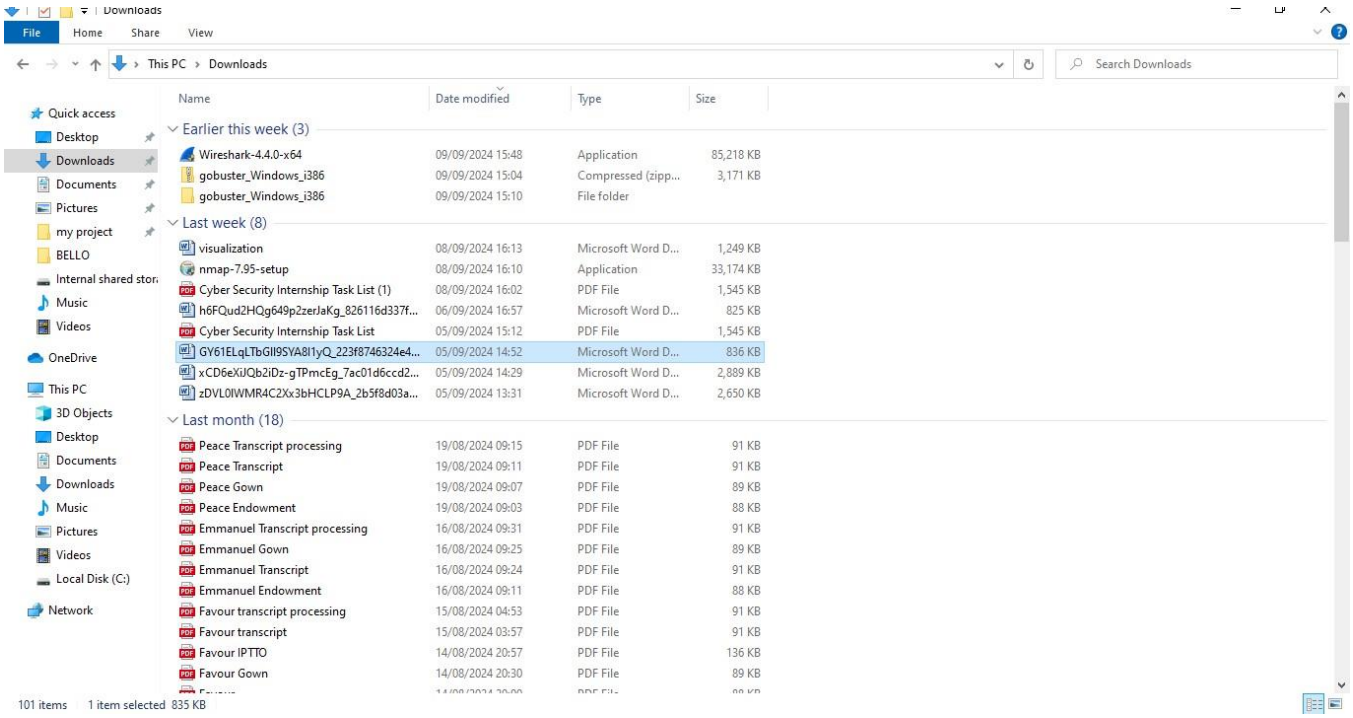
Beginner Level

1. **Find all the ports that are open on the website** http://testphp.vulnweb.com/

In doing this I used the following steps

Step 1

I installed the Nmap app, Nmap is a tool used for network scanning and port discovery. For different operating systems there are several ways to do that. For Windows I went to the site nmap.org and installed it from there.

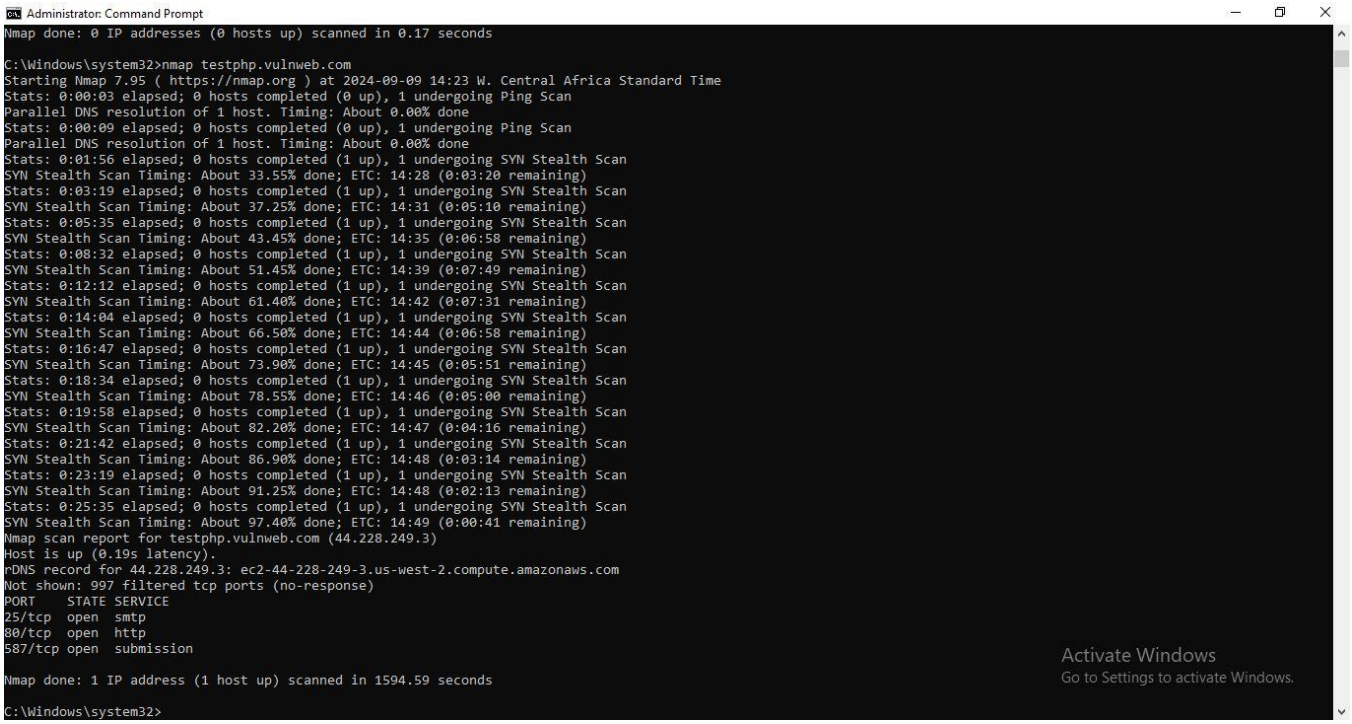


If you take a close look at the picture you will see the downloaded Nmap tool that will be used for the scanning.

Step 2

Perform a port scan

I open command prompt on my laptop, having installed it on my system I ran the following command ***nmap testphp.vulnweb.com***



From the scan we can see that port 25 which provides SMTP service, port 80 for HTTP service and port 587 for submission service is opened.

For the SMTP port which is the abbreviation of Simple Mail Transfer Protocol is the standard communication protocol used to send and relay outgoing emails between mail servers. As we have seen here it operates mainly on port 25 but can also operate on port 465 (With SSL encryption) or port 587 (With TLS encryption). SMTP facilitates sending of emails, receiving of emails and also forwarding of emails across clients and mail servers. Although an opened SMTP port can allow a mail server to send and receive incoming mails and accept incoming connection for other mail servers. If improperly insecured it can lead to spammers sending unsolicited bulk emails which can lead to flooding attacks. Also this can open the server to potential security risk like SMTP relay attacks, where attackers exploit the server to send spam malicious content. Knowing this it tells us that this website is vulnerable to attacks.

One way to mitigate this and still maintain email functionality, I suggest that you restrict access to SMTP ports to specific trusted sources or require authentication to prevent misuse.

For the HTTP port which is the abbreviation of Hypertext Transfer Protocol used for transmitting web pages and other content over the internet. It operates on port 80 for standard unencrypted communication. It has a more secure version HTTPS which operates on port 443 it encrypts data using SSL/TLS.

This website is made accessible to me because HTTP is opened but it poses potential threats to the website such as man-in-the-middle attack and eavesdropping since HTTP communication is not encrypted attackers can exploit this open port.

But if I decide to close this port it will prevent accessibility to both legitimate and illegitimate users, so one solution to this is to HTTPS on port 443.

Submission (Message Submission Agent) This service is responsible for submitting emails from client to a mail server for delivery. This is on port 587 as seen in the image above. This service provides secure email submission, Spam prevention.

An open MSA port in this website means that email clients can submit emails successfully. While this is good this poses a threat to the mail server if not secured properly. Attacks like Brute force attacks, spam and phishing, man-in-the-middle attack. Although keeping it closed will automatically eradicate the threats but will prevent the legitimate users who wants to use it. Therefore one of the ways of reducing the threats and leaving the ports opened is by enforcing authentication; this will require clients to authenticate before allowing email submissions through port 587. Another way is by using STARTTLS to encrypt email transmissions and prevent attackers from intercepting sensitive data

2. Brute force the website http://testphp.vulnweb.com/ and find the directories that are present in the website.

For this task we would be using FFUF tool. FFUF (Fuzz Faster U Fool) is a tool that automates the process of fuzzing. Ffuf is designed for security professionals to find vulnerabilities in web applications.

Ffuf does this by sending a large number of requests to a target with various payloads. Ffuf then analyzes the responses and tells us what worked and what didn't.

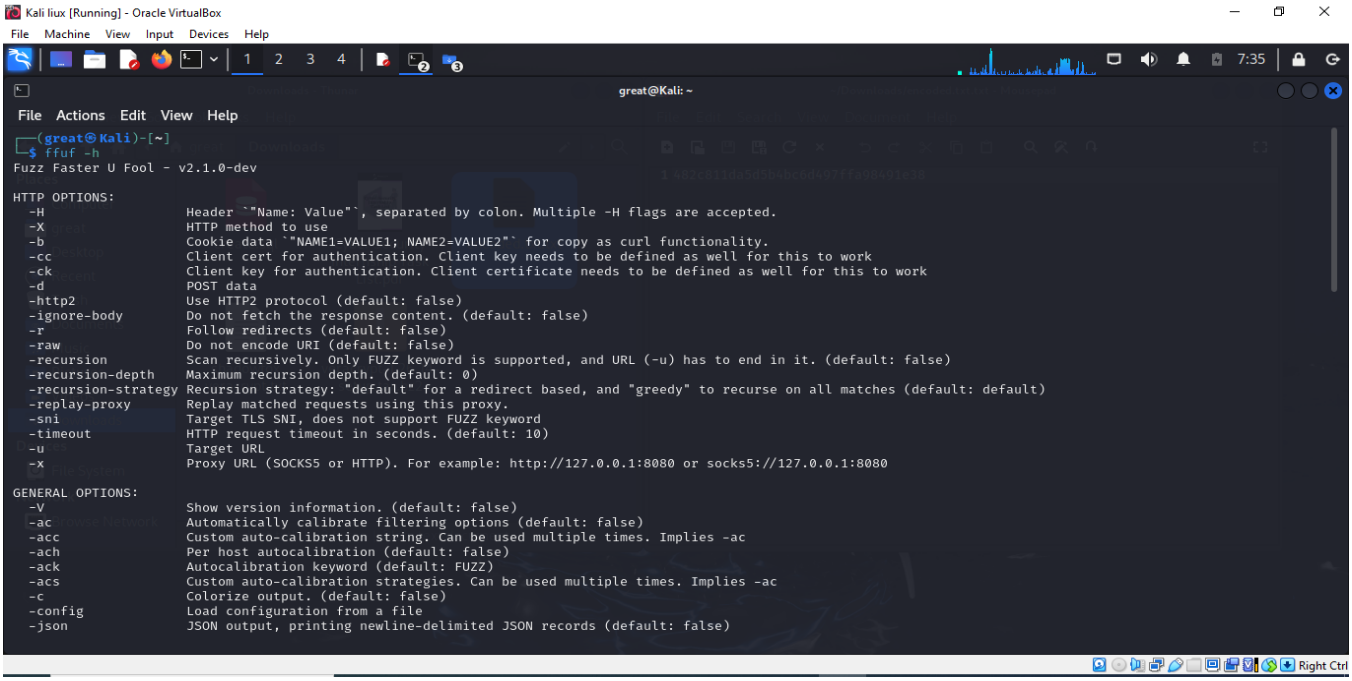
We can use Ffuf to test for a wide variety of vulnerabilities, including input validation issues, access control problems, and other types of security weaknesses.

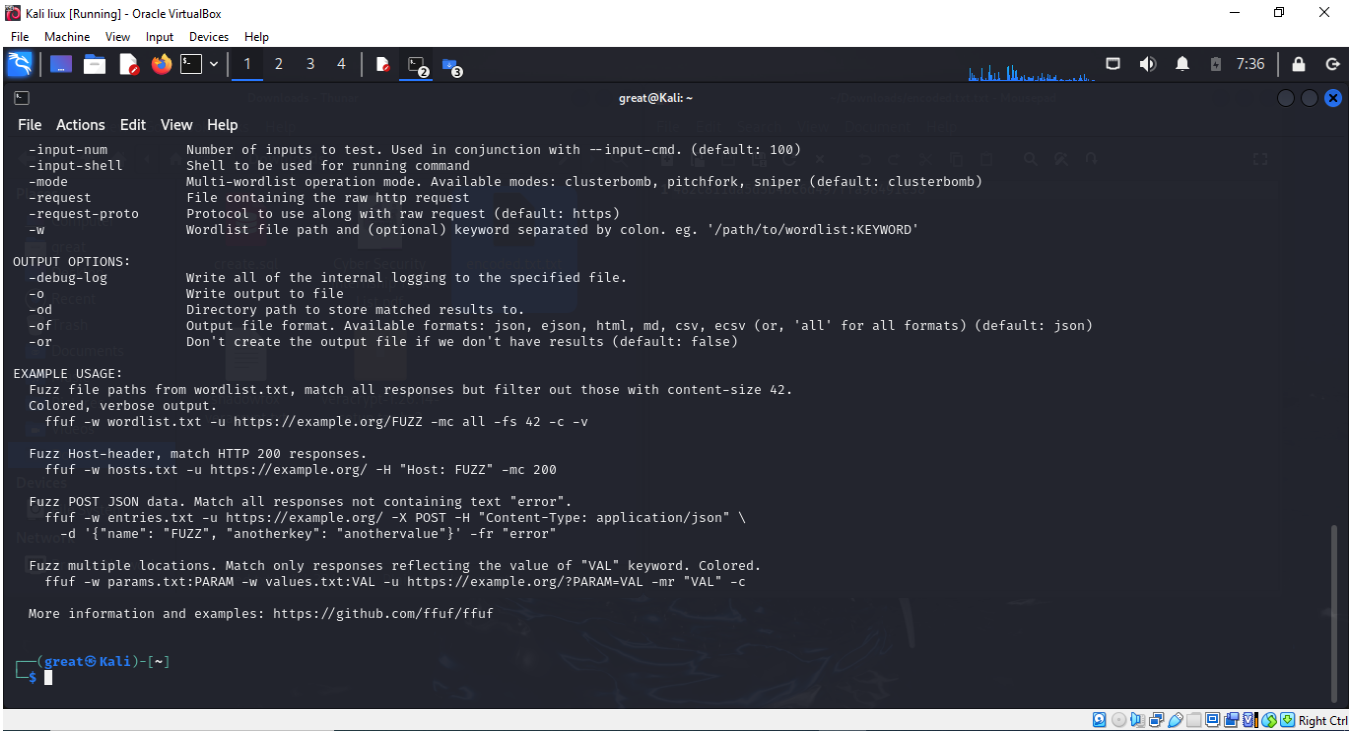
FFUF is also fast and flexible, allowing us to specify the inputs to use for fuzzing and the parameters for the requests sent to the target web application.

Ffuf is also extensively used in bug-bounty hunting, so if you plan to become a bug-bounty hunter, you will be using Ffuf on a daily basis.

Step 1

Install FFUF on linux or to check if it is installed run the code *ffuf -h* on linux terminal to confirm if it is installed.





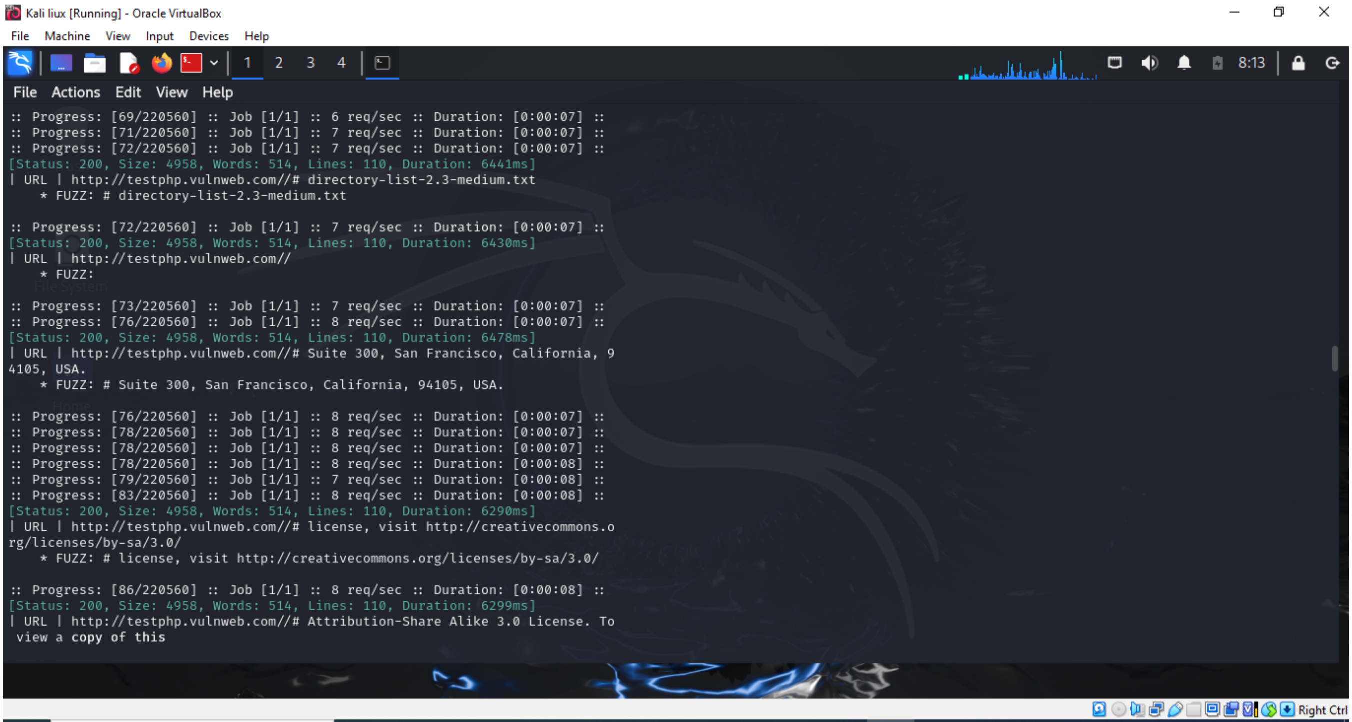
Step 2

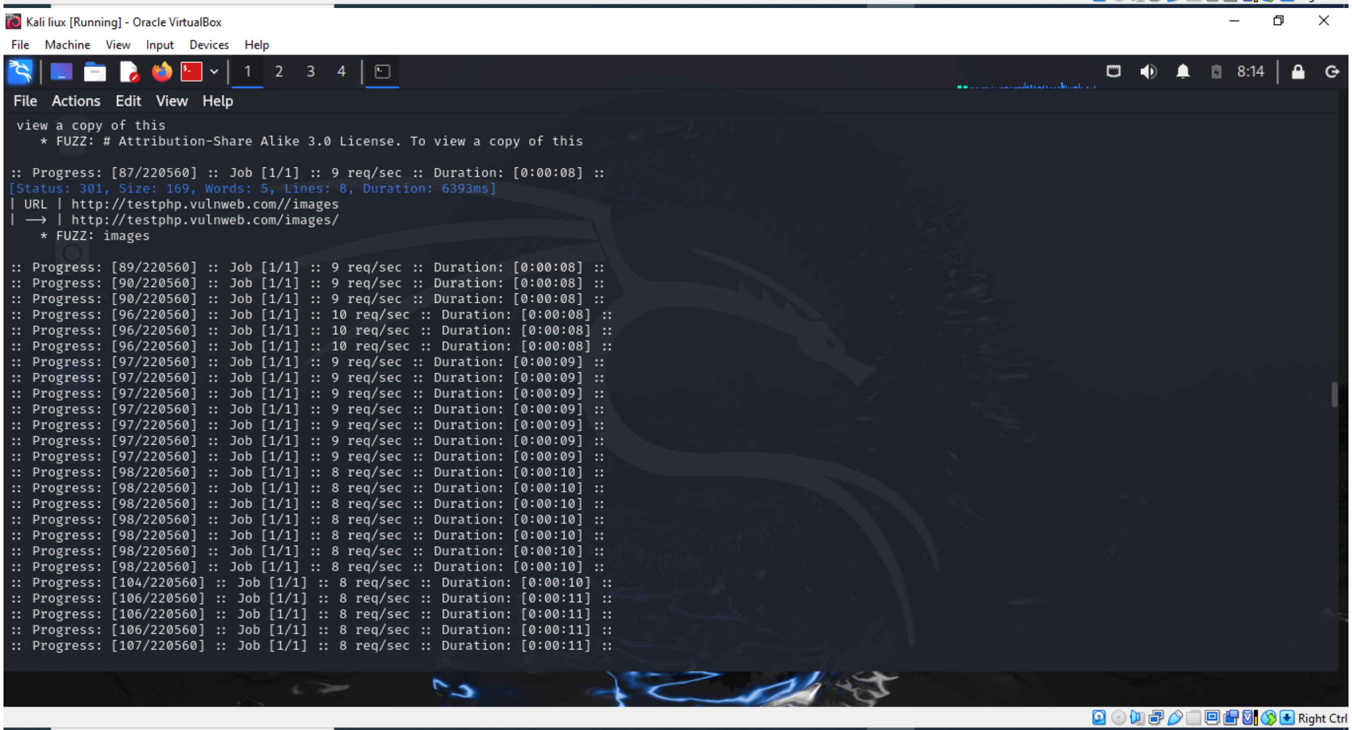
Now that ffuf installation is confirmed we can then run the command to brute force the website for hidden directories.

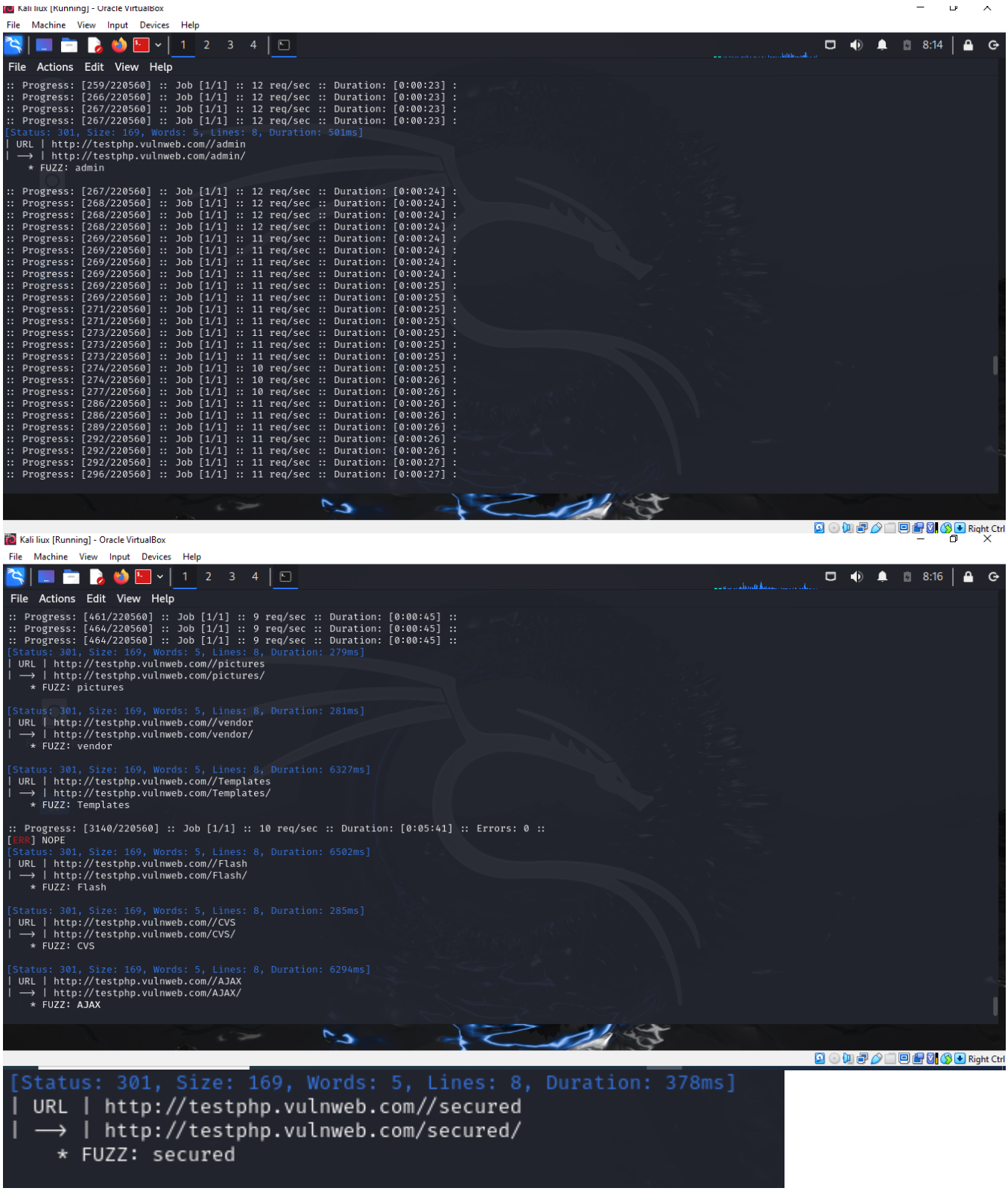
On linux root terminal run the following *command ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://testphp.vulnweb.com//FUZZ -c -v*


Kali liux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
Fuzz multiple locations. Match only responses reflecting the value of "VAL" keyword. Colored.
ffuf -w params.txt:PARAM -w values.txt:VAL -u https://example.org/?PARAM=VAL -mr "VAL" -c
More information and examples: https://github.com/ffuf/ffuf
Encountered error(s): 2 errors occurred.
* -u flag or -request flag is required
* Either -w or --input-cmd flag is required
┌──(root㉿Kali)-[~]
└─# ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://testphp.vulnweb.com//FUZZ -c -v
v2.1.0-dev
:: Method : GET
:: URL : http://testphp.vulnweb.com//FUZZ
:: Wordlist : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout : 10
:: Threads : 40


Kali liux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
:: Progress: [40/220560] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:02] ::
:: Progress: [40/220560] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:03] ::
[Status: 200, Size: 4958, Words: 514, Lines: 110, Duration: 895ms]
| URL | http://testphp.vulnweb.com//#
* FUZZ: #
:: Progress: [40/220560] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:03] ::
:: Progress: [41/220560] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:04] ::
:: Progress: [42/220560] :: Job [1/1] :: 1 req/sec :: Duration: [0:00:04] ::
[Status: 200, Size: 4958, Words: 514, Lines: 110, Duration: 3284ms]
| URL | http://testphp.vulnweb.com//# or send a letter to Creative Commons, 171 Second Street,
* FUZZ: # or send a letter to Creative Commons, 171 Second Street,
:: Progress: [42/220560] :: Job [1/1] :: 1 req/sec :: Duration: [0:00:04] ::
:: Progress: [45/220560] :: Job [1/1] :: 4 req/sec :: Duration: [0:00:04] ::
[Status: 200, Size: 4958, Words: 514, Lines: 110, Duration: 3288ms]
| URL | http://testphp.vulnweb.com//# on atleast 2 different hosts
* FUZZ: # on atleast 2 different hosts
:: Progress: [46/220560] :: Job [1/1] :: 5 req/sec :: Duration: [0:00:04] ::


Kali liux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
8:13
File Actions Edit View Help
:: Progress: [69/220560] :: Job [1/1] :: 6 req/sec :: Duration: [0:00:07] ::
:: Progress: [71/220560] :: Job [1/1] :: 7 req/sec :: Duration: [0:00:07] ::
:: Progress: [72/220560] :: Job [1/1] :: 7 req/sec :: Duration: [0:00:07] ::
[Status: 200, Size: 4958, Words: 514, Lines: 110, Duration: 6441ms]
| URL | http://testphp.vulnweb.com//# directory-list-2.3-medium.txt
* FUZZ: # directory-list-2.3-medium.txt
:: Progress: [72/220560] :: Job [1/1] :: 7 req/sec :: Duration: [0:00:07] ::
[Status: 200, Size: 4958, Words: 514, Lines: 110, Duration: 6430ms]
| URL | http://testphp.vulnweb.com//
* FUZZ:
:: Progress: [73/220560] :: Job [1/1] :: 7 req/sec :: Duration: [0:00:07] ::
:: Progress: [76/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:07] ::
[Status: 200, Size: 4958, Words: 514, Lines: 110, Duration: 6478ms]
| URL | http://testphp.vulnweb.com//# Suite 300, San Francisco, California, 94105, USA.
* FUZZ: # Suite 300, San Francisco, California, 94105, USA.
:: Progress: [76/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:07] ::
:: Progress: [78/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:07] ::
:: Progress: [78/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:07] ::
:: Progress: [78/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:08] ::
:: Progress: [79/220560] :: Job [1/1] :: 7 req/sec :: Duration: [0:00:08] ::
:: Progress: [83/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:08] ::
[Status: 200, Size: 4958, Words: 514, Lines: 110, Duration: 6290ms]
| URL | http://testphp.vulnweb.com//# license, visit http://creativecommons.org/licenses/by-sa/3.0/
* FUZZ: # license, visit http://creativecommons.org/licenses/by-sa/3.0/
:: Progress: [86/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:08] ::
[Status: 200, Size: 4958, Words: 514, Lines: 110, Duration: 6299ms]
| URL | http://testphp.vulnweb.com//# Attribution-Share Alike 3.0 License. To view a copy of this
Right Ctrl


Kali liux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
8:14
File Actions Edit View Help
view a copy of this
* FUZZ: # Attribution-Share Alike 3.0 License. To view a copy of this
:: Progress: [87/220560] :: Job [1/1] :: 9 req/sec :: Duration: [0:00:08] ::
[Status: 301, Size: 169, Words: 5, Lines: 8, Duration: 6393ms]
| URL | http://testphp.vulnweb.com//images
| --> | http://testphp.vulnweb.com/images/
* FUZZ: images
:: Progress: [89/220560] :: Job [1/1] :: 9 req/sec :: Duration: [0:00:08] ::
:: Progress: [90/220560] :: Job [1/1] :: 9 req/sec :: Duration: [0:00:08] ::
:: Progress: [90/220560] :: Job [1/1] :: 9 req/sec :: Duration: [0:00:08] ::
:: Progress: [96/220560] :: Job [1/1] :: 10 req/sec :: Duration: [0:00:08] ::
:: Progress: [96/220560] :: Job [1/1] :: 10 req/sec :: Duration: [0:00:08] ::
:: Progress: [96/220560] :: Job [1/1] :: 10 req/sec :: Duration: [0:00:08] ::
:: Progress: [97/220560] :: Job [1/1] :: 9 req/sec :: Duration: [0:00:09] ::
:: Progress: [97/220560] :: Job [1/1] :: 9 req/sec :: Duration: [0:00:09] ::
:: Progress: [97/220560] :: Job [1/1] :: 9 req/sec :: Duration: [0:00:09] ::
:: Progress: [97/220560] :: Job [1/1] :: 9 req/sec :: Duration: [0:00:09] ::
:: Progress: [97/220560] :: Job [1/1] :: 9 req/sec :: Duration: [0:00:09] ::
:: Progress: [97/220560] :: Job [1/1] :: 9 req/sec :: Duration: [0:00:09] ::
:: Progress: [97/220560] :: Job [1/1] :: 9 req/sec :: Duration: [0:00:09] ::
:: Progress: [98/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:10] ::
:: Progress: [98/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:10] ::
:: Progress: [98/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:10] ::
:: Progress: [98/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:10] ::
:: Progress: [98/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:10] ::
:: Progress: [98/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:10] ::
:: Progress: [98/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:10] ::
:: Progress: [104/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:10] ::
:: Progress: [106/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:11] ::
:: Progress: [106/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:11] ::
:: Progress: [106/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:11] ::
:: Progress: [107/220560] :: Job [1/1] :: 8 req/sec :: Duration: [0:00:11] ::
Right Ctrl

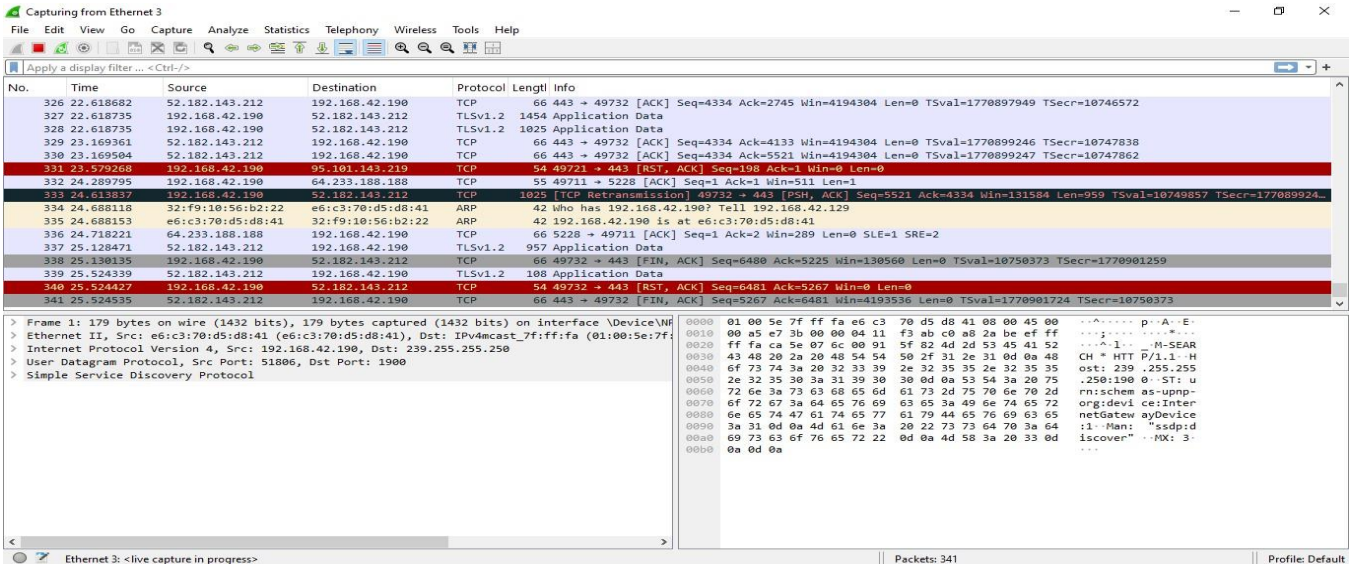


From this attack one major directory that was visible by the attack was the admin panel and this poses a great threat to the website because If the admin panel is not properly secured, attackers can use brute-force or credential stuffing attacks to gain access. Also we see the Images, Templates etc directories been made visible.

3. **Make a login in the website http://testphp.vulnweb.com/ and intercept the network traffic using wireshark and find the credentials that were transferred through the network.**

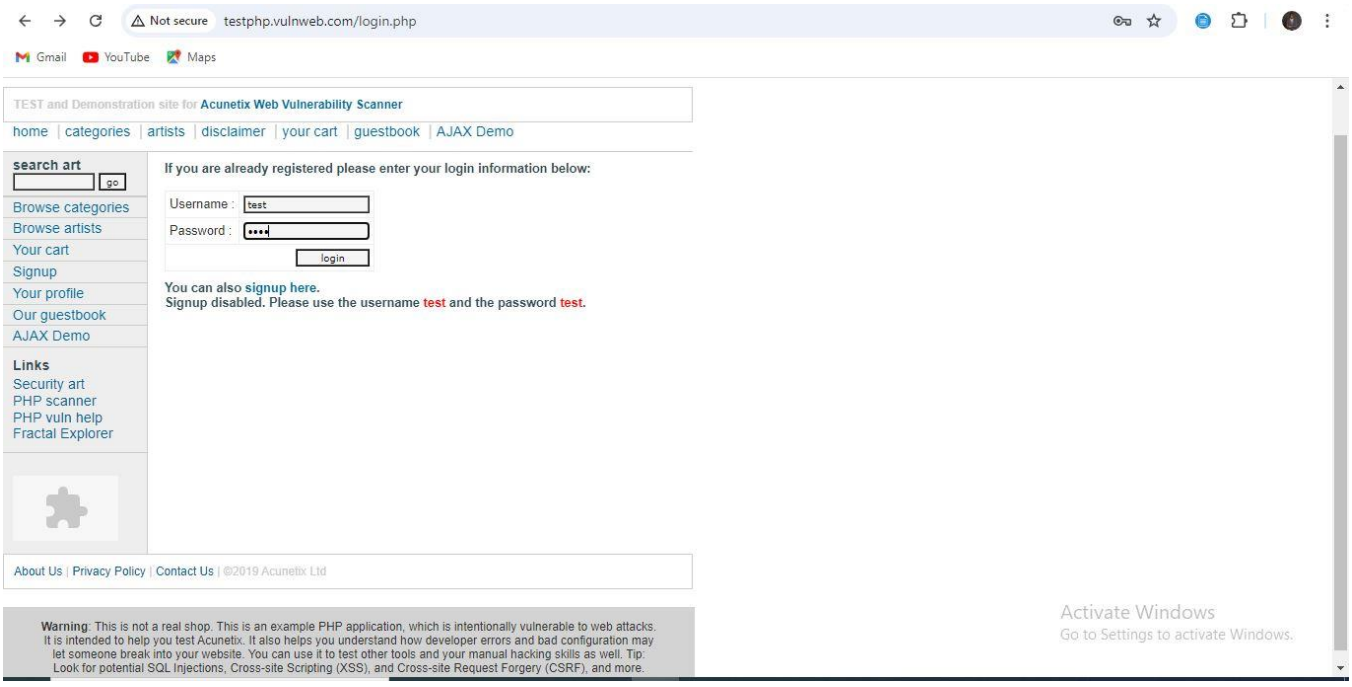
Step 1

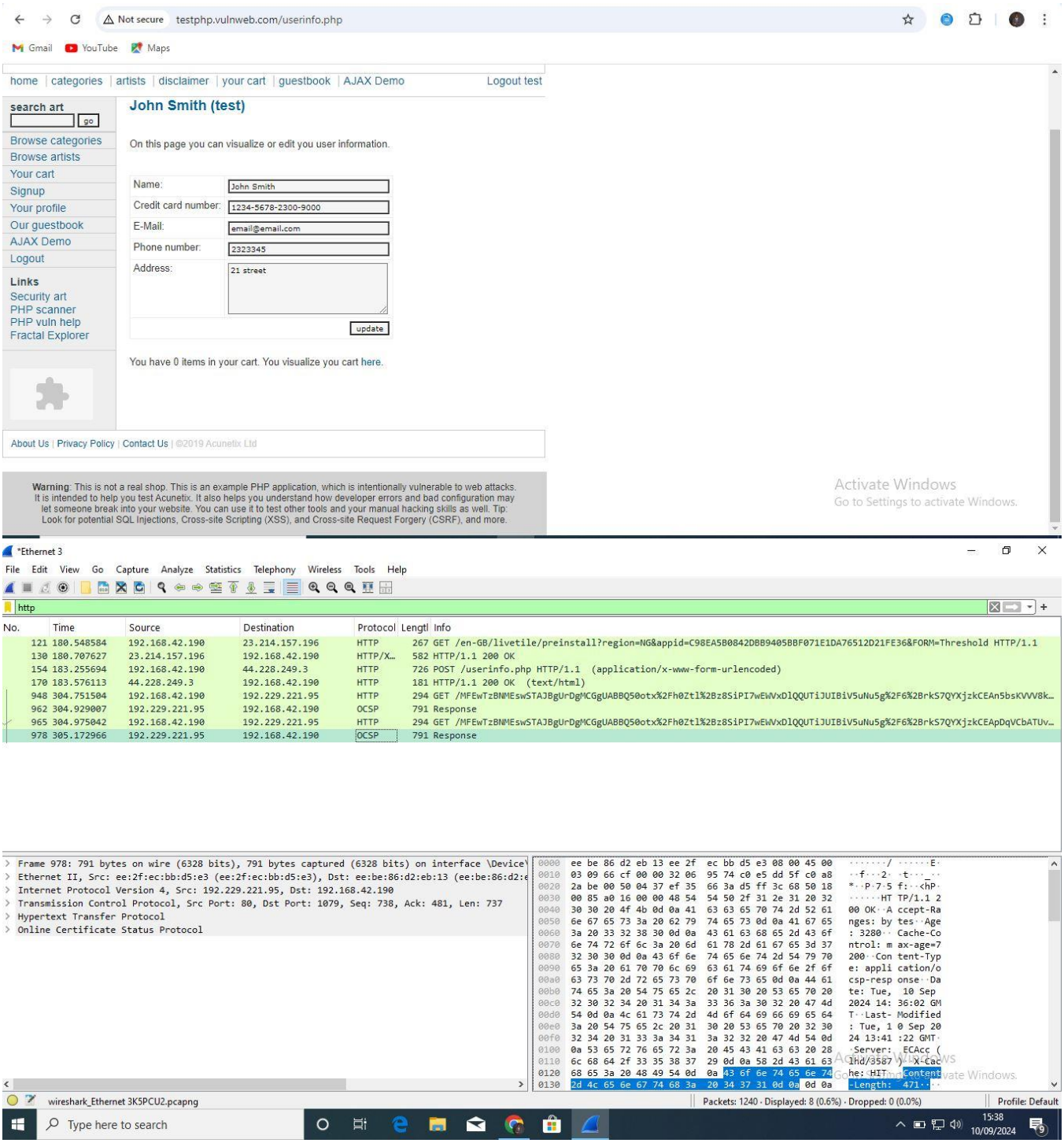
Open Wireshark and select the network interface that is actively connected to the network, for me i used Ethernet and start capturing traffic by clicking the shark fin icon.



Step 2

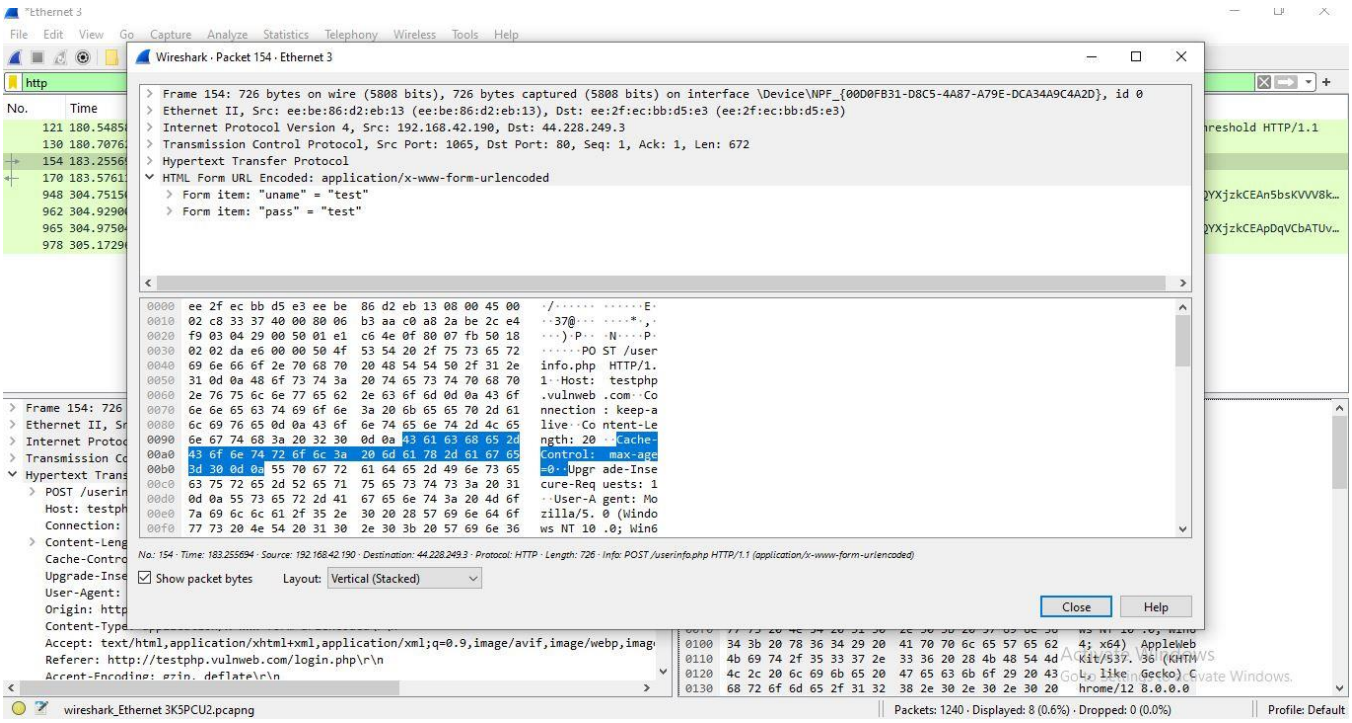
Open http://testphp.vulnweb.com/ and navigate to the login page. Login and analyze the network using the wireshark capture opened before. Then filter it to only analyze HTTP traffic.





Step 3

Analyze the HTTP packets: in analyzing the packets you will look for info that contains POST click on it.



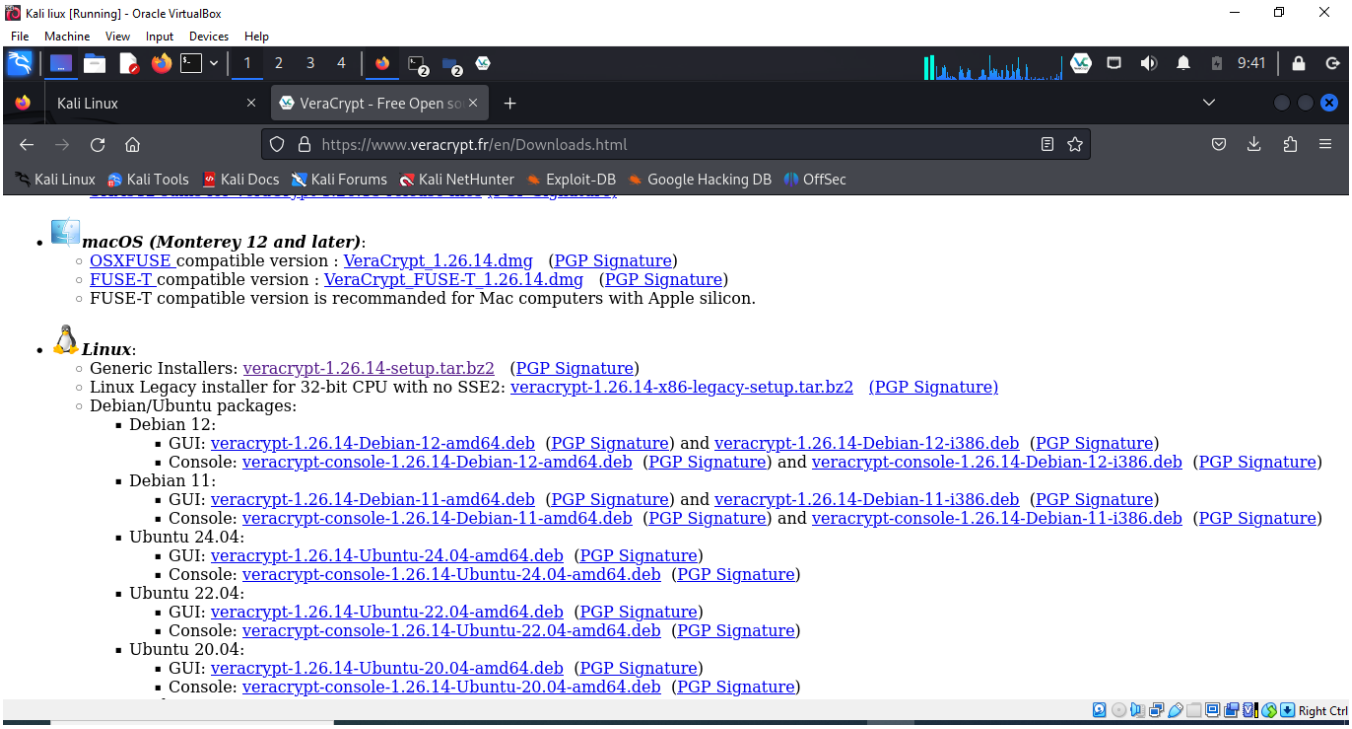
After clicking on it and you navigate to the ***HTML force URL Encoded*** here we see that the login details are not encrypted, very visible to be seen. This shows one of the threats of the HTTP. In analyzing network packets it displays the credentials in plain text.

Intermediate Level

1. **A file is encrypted using Veracrypt (A disk encryption tool). The password to access the file is encrypted in a hash format and provided to you in the drive with the name encoded.txt. Decode the password and enter in the vera crypt to unlock the file and find the secret code in it. The veracrypt setup file will be provided to you.**

Step 1

Installing Veracrypt: I was able to download it from the official veracrypt website



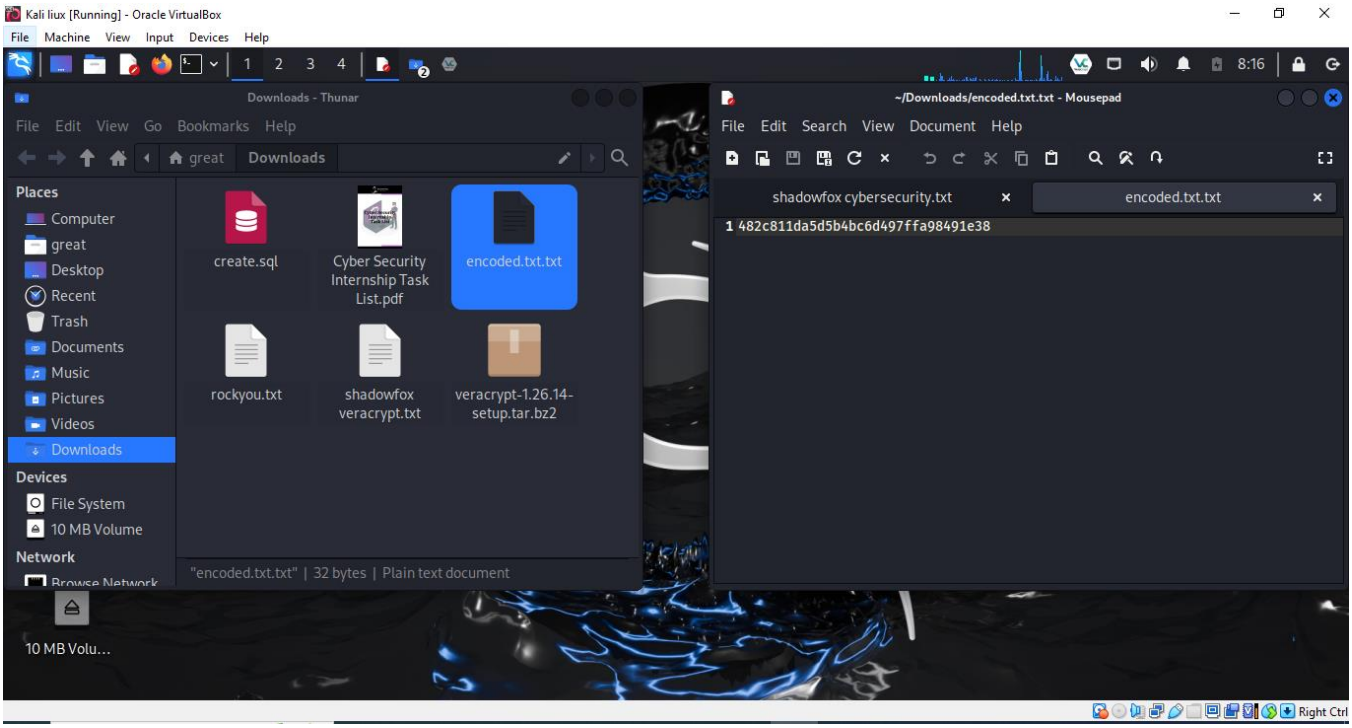
Step 2

Examine the encoded.txt File

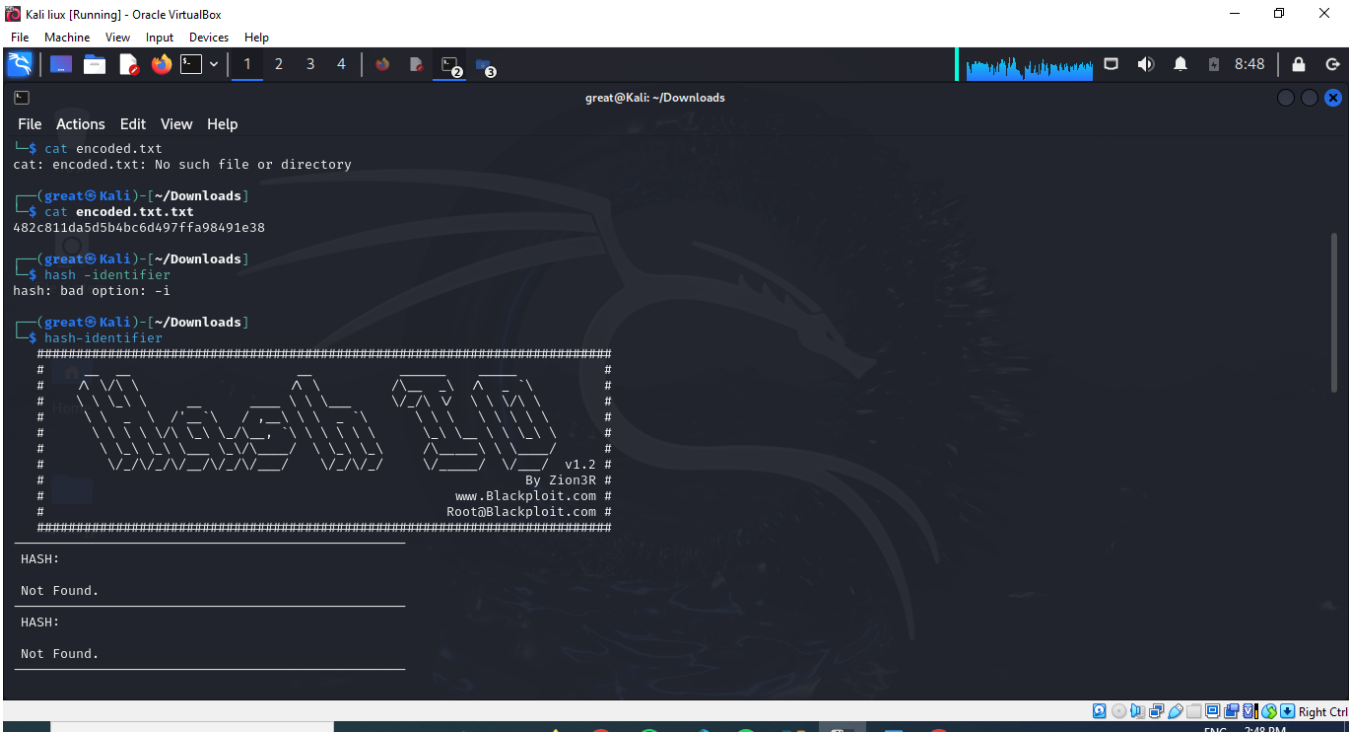
Open the encoded.txt file to check what kind of hash or encoding it contains.

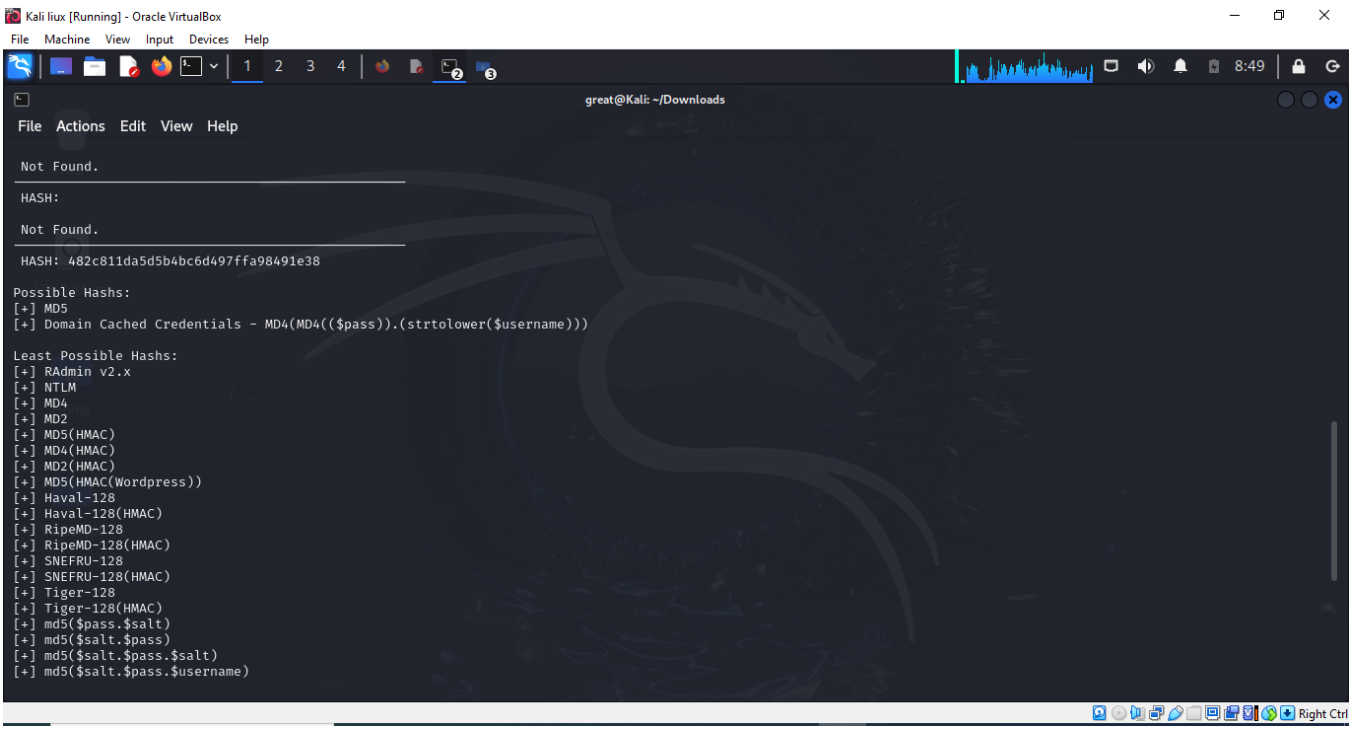
If it looks like a base64 string (a common encoding), it might be decoded easily. If it is a hash like MD5, SHA-256, or another type, you'll need to crack it using a tool.

When we opened the file it was with this:



Looking at this the code is hashed, but we don't know what kind of hashed code it is. So we will have to identify it. We will identify it using the following commands:

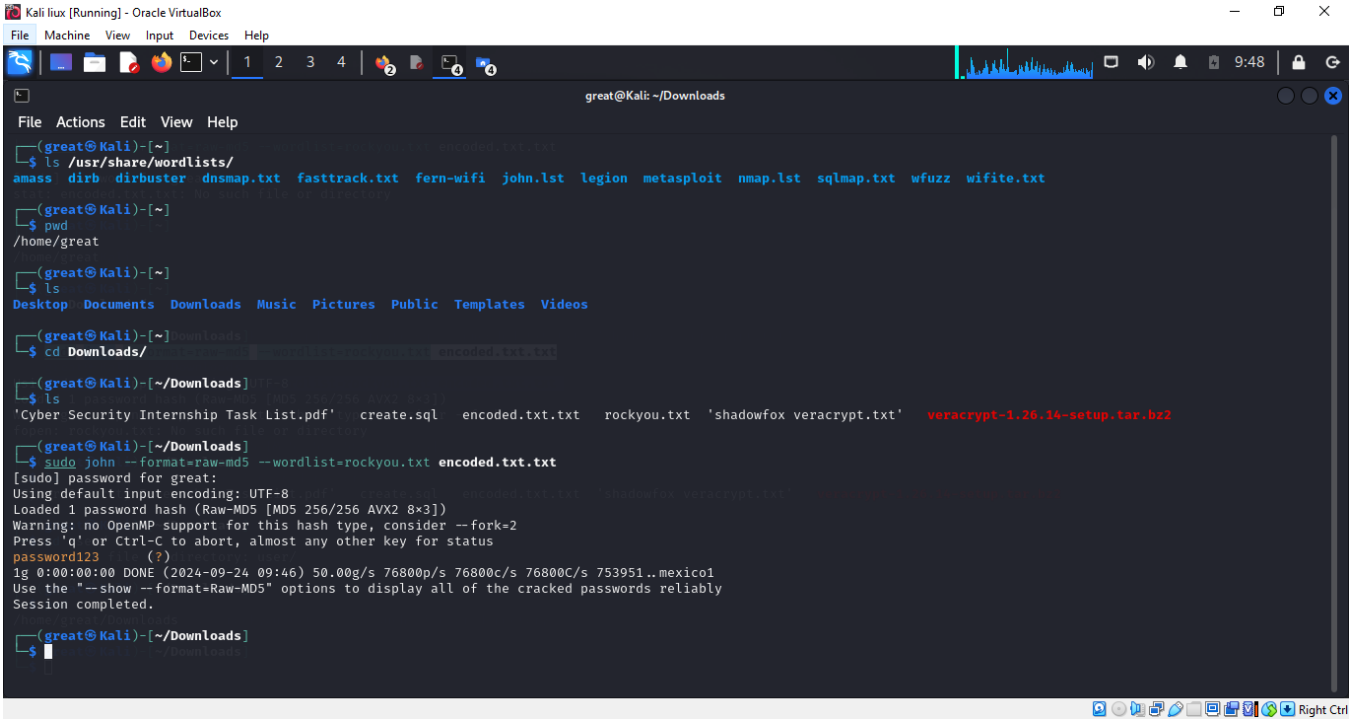




From this we can see that the code is an MD5 code. Then now we will be cracking it to crack it we will be using a tool in Linux called John the Ripper

Step 3

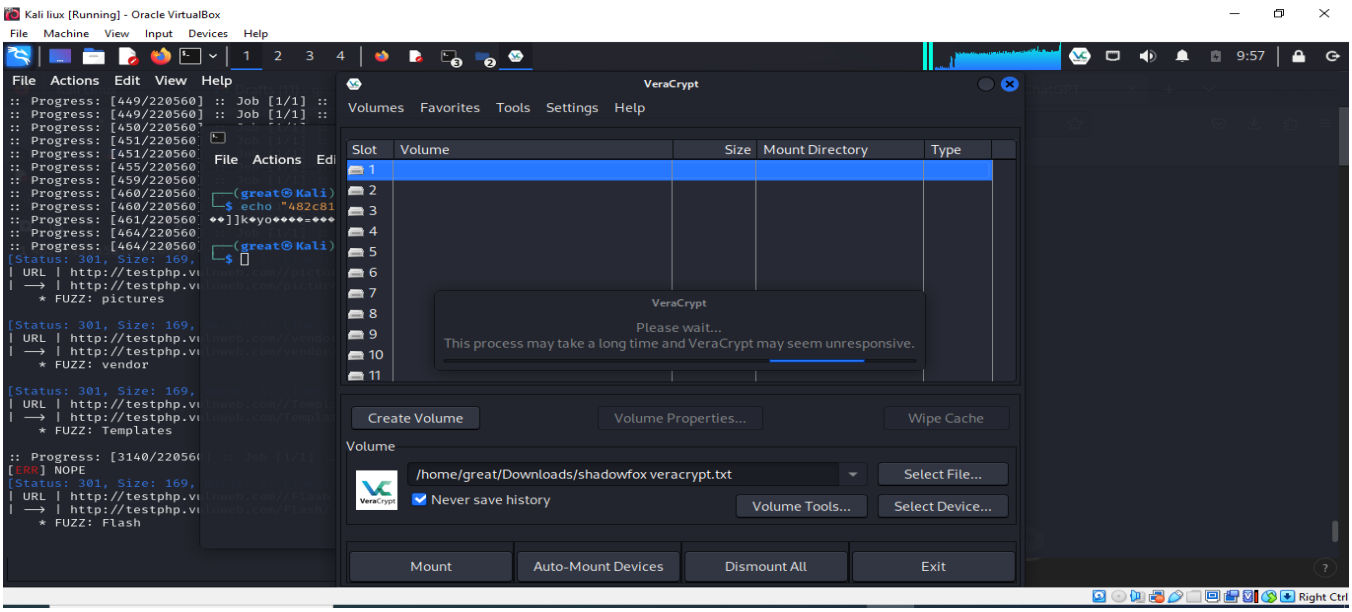
Cracking the MD5 code

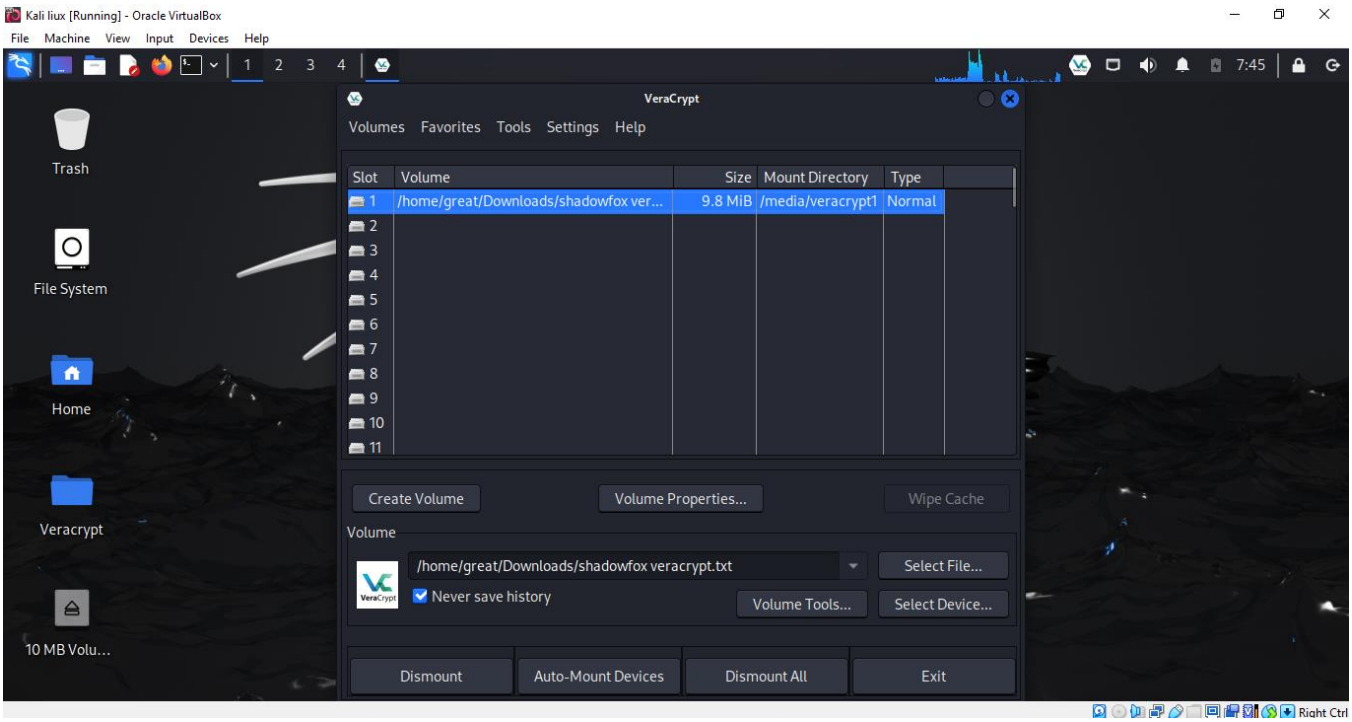


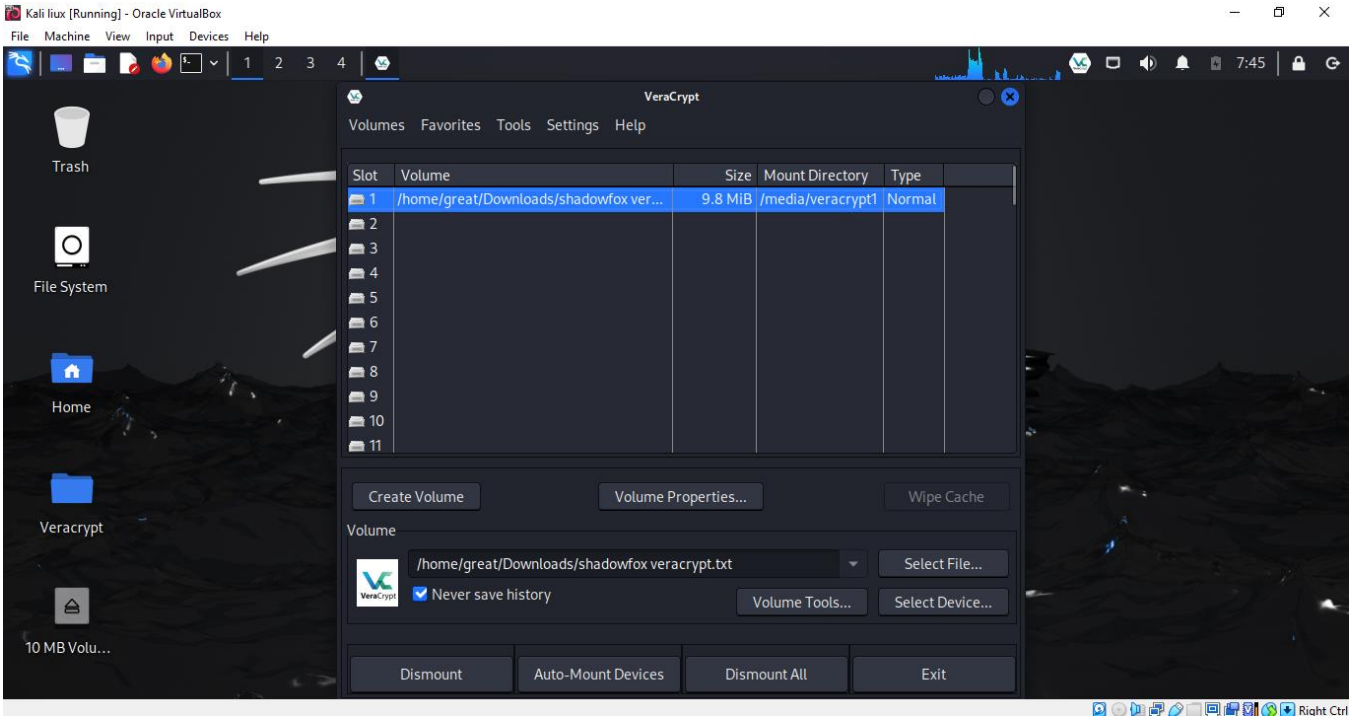
From this we can see that the code is ***passowrd123***

Step 4

Use the Cracked Password to Unlock VeraCrypt: Once I cracked the password, I opened VeraCrypt. Selected the encrypted file, entered the cracked and mounted the file, so I'll have access to the encrypted file.

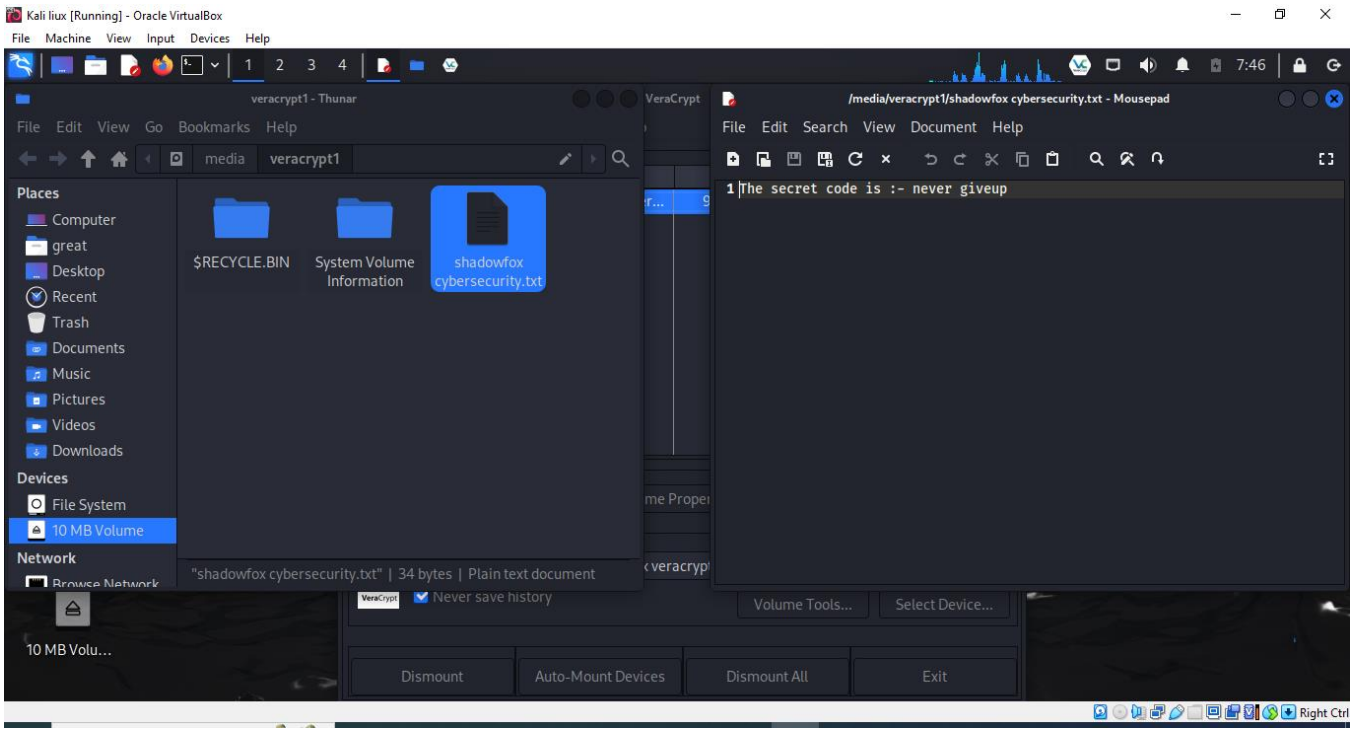






Step 5

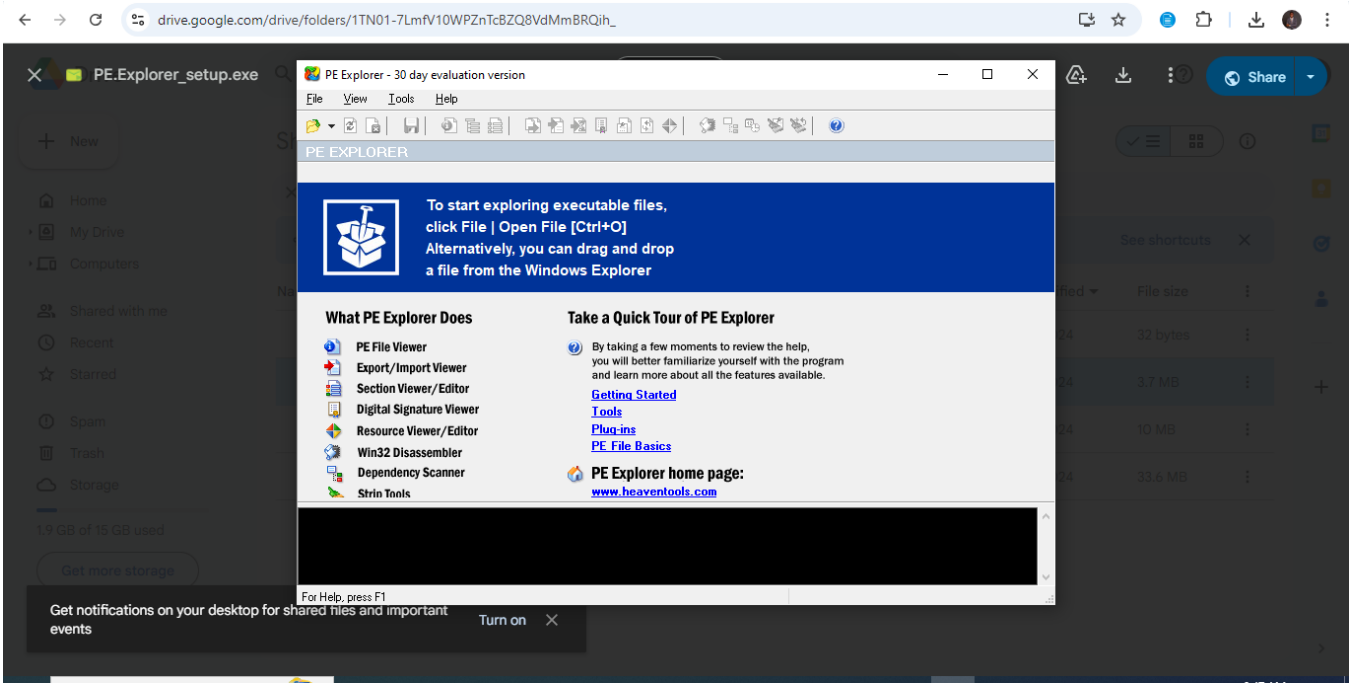
Open the Mounted file on Veracrypt to view the secret code: I did this by double clicking in the mounted file.



2. An executable file of veracrypt will be provided to you. Find the address of the entry point of the executable using PE explorer tool and provide the value as the answer as a screenshot.

Step 1

Download, Install and launch PE Explorer:

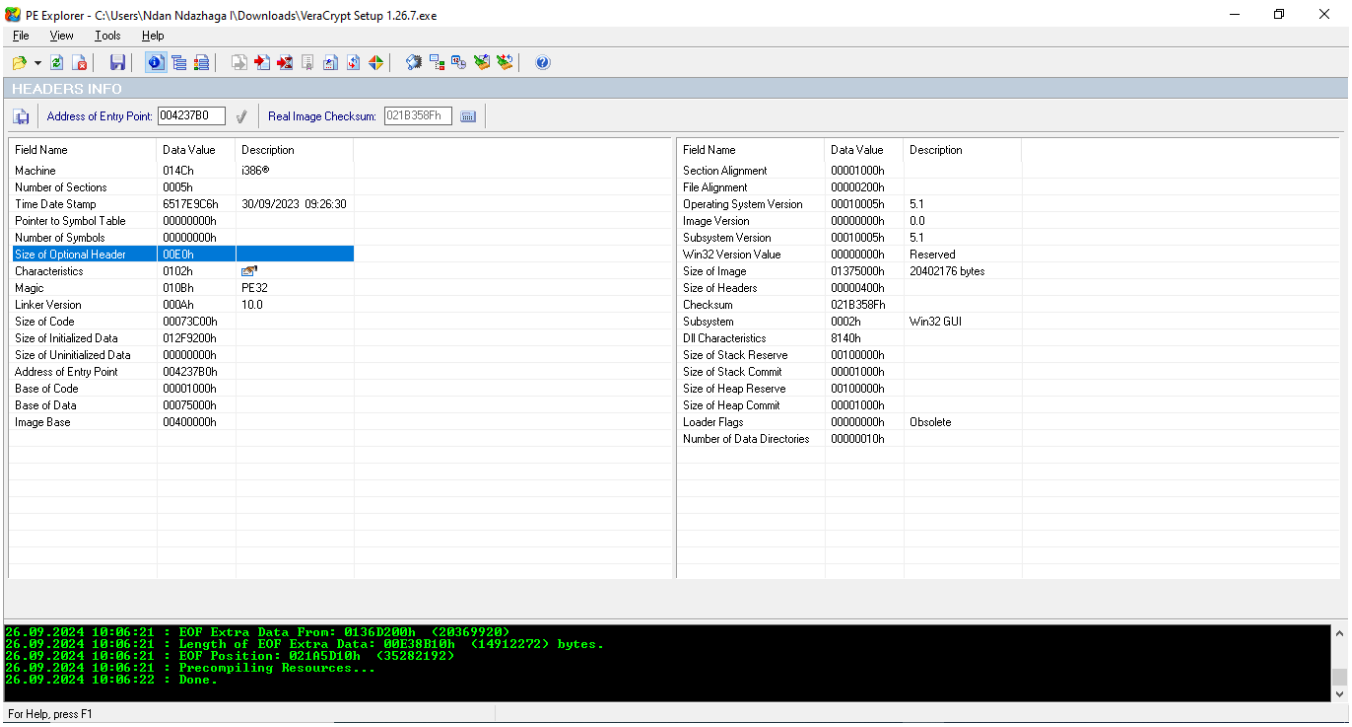


Step 2

Open the Executable File and Locate the address: After loading the executable file, PE Explorer will display information about its structure. In the left-hand pane, you will see sections such as "Headers Info".

I Looked for the field labeled "AddressOfEntryPoint". This value represents the entry point of the executable, where the code execution starts.

The entry point is typically displayed as a hexadecimal value.



Looking at the picture the Address of entry point is ***004237B0***

3. Create a payload using Metasploit and make a reverse shell connection from a Windows 10 machine in your virtual machine setup.

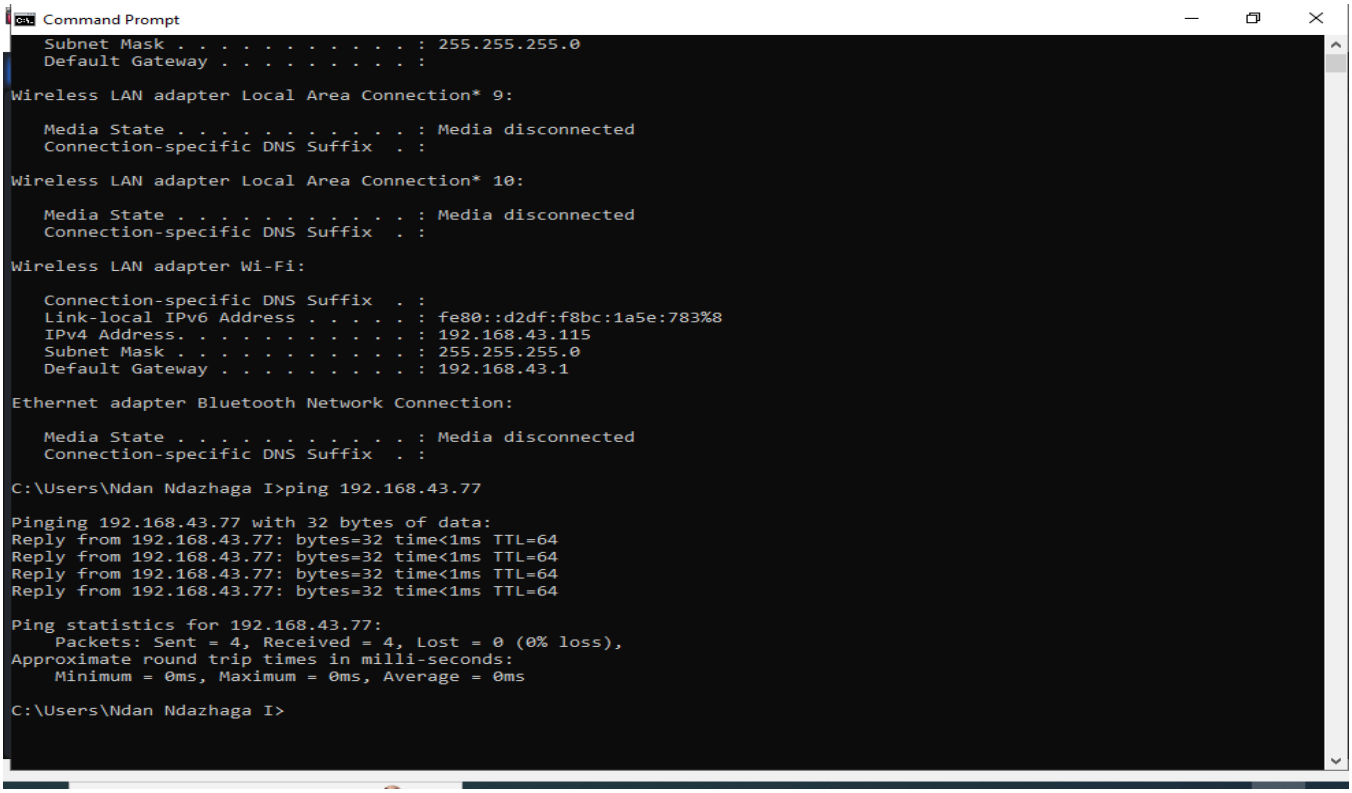
Step 1

Configure Your VM Network:

Ensure both the Kali Linux and Windows 10 VMs are on the same network, either by using NAT or Host-Only Adapter in your virtual machine settings.

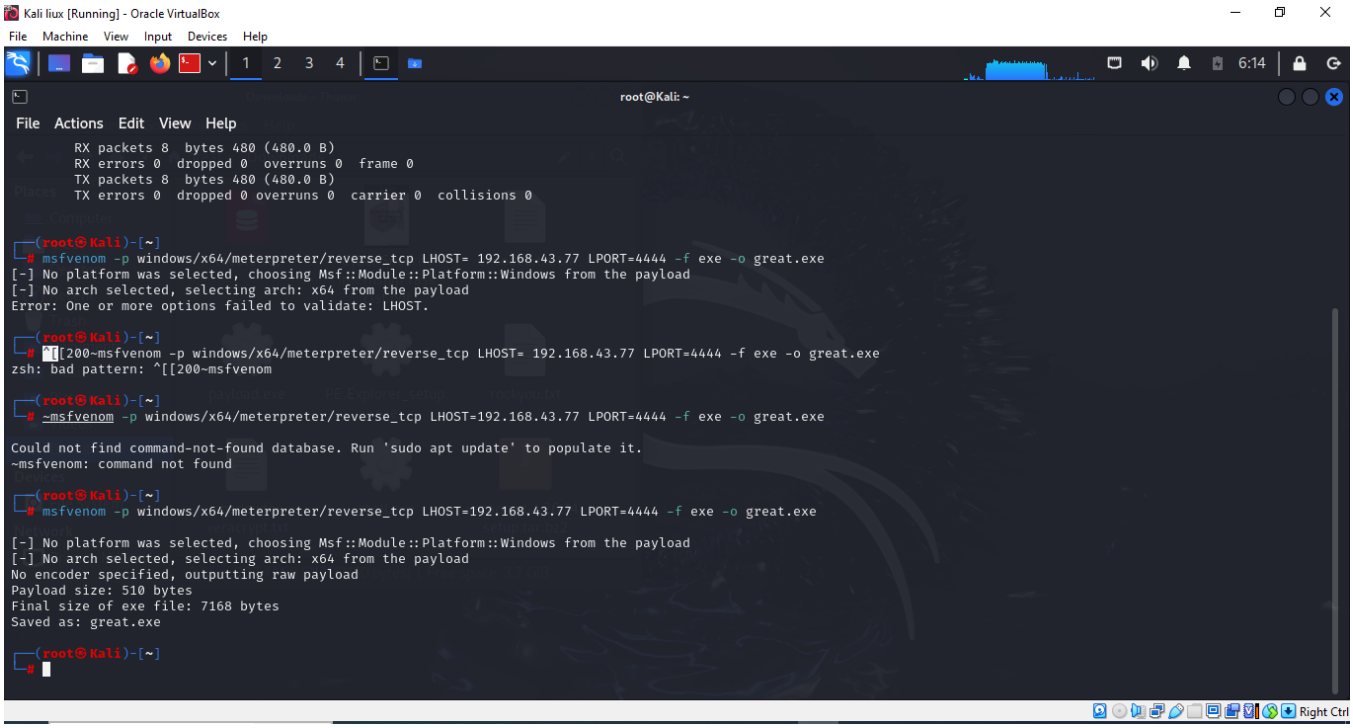
You can verify communication by pinging the Linux machine from the Windows 10 machine:

The Ip address of our Kali Linux machine is 162.198.43.77

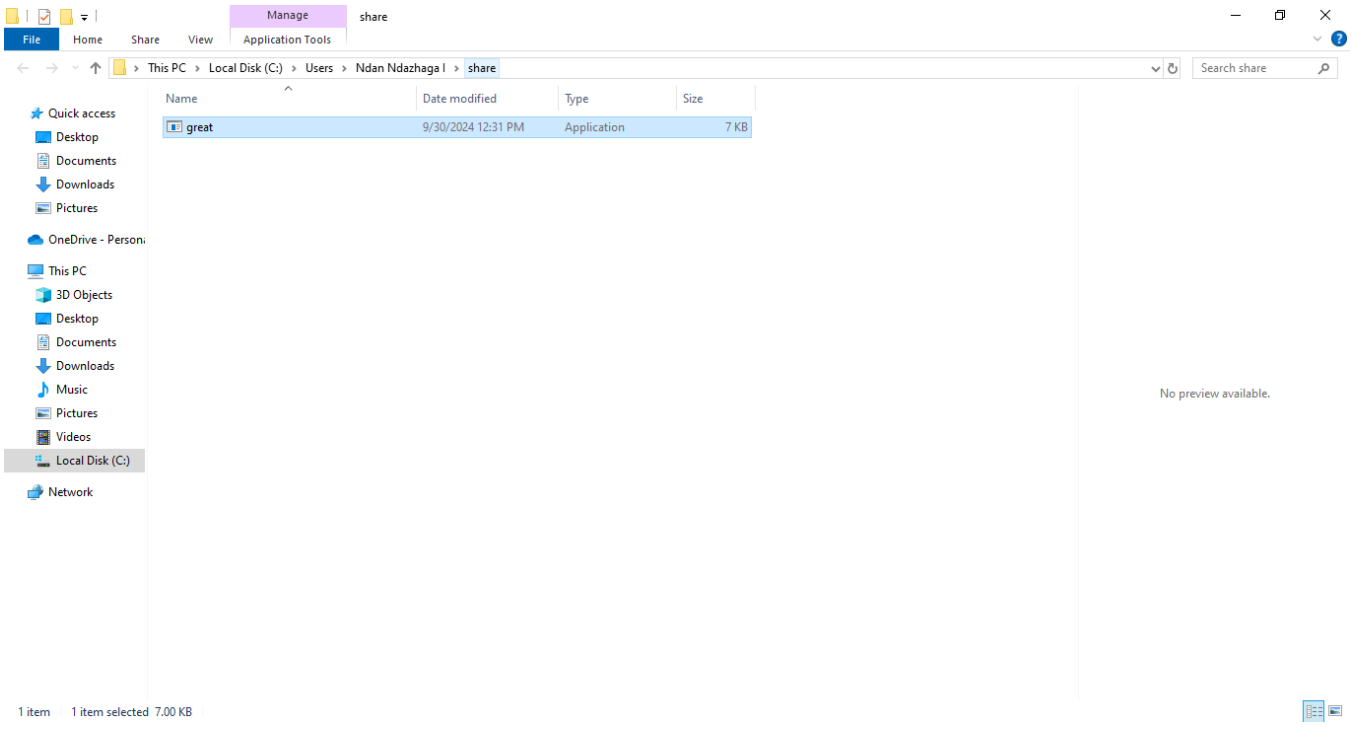


Step 2

Open Kali Linux Terminal: Launch Metasploit by typing msfconsole and generating a payload using msfvenom to create the reverse shell payload. The payload will be an executable that the Windows 10 machine runs to connect back to the Kali Linux machine.



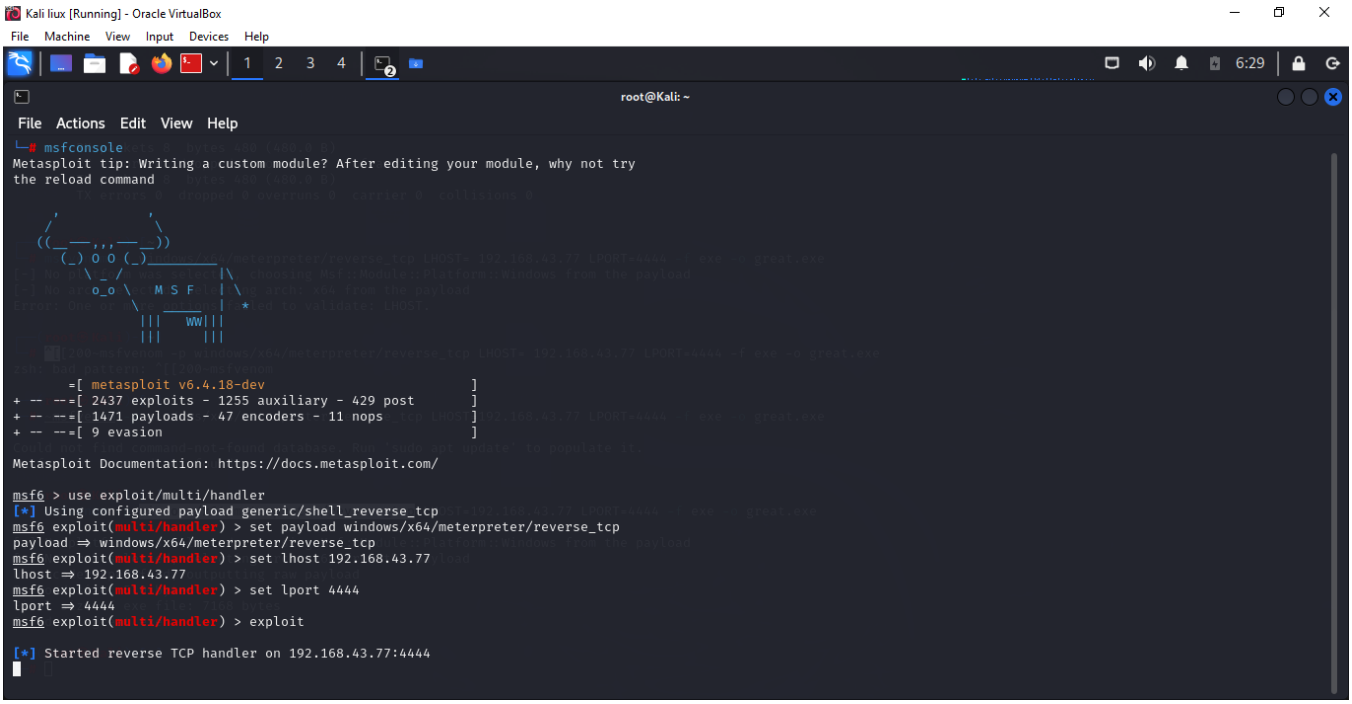
Transfer the Payload to the Windows 10 Machine: I used a method called USB, shared folder in Kali Linux to transfer the payload to the Windows machine.

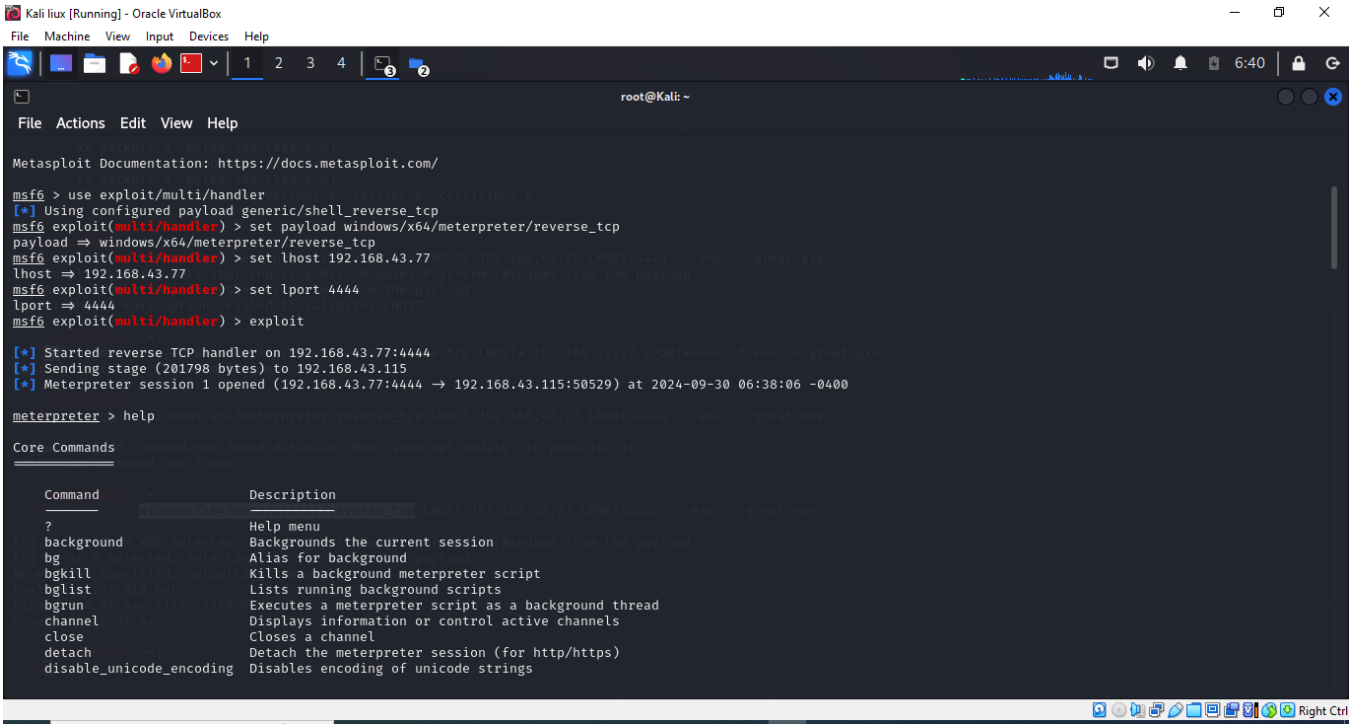


Step 3

Set Up a Listener in Metasploit:

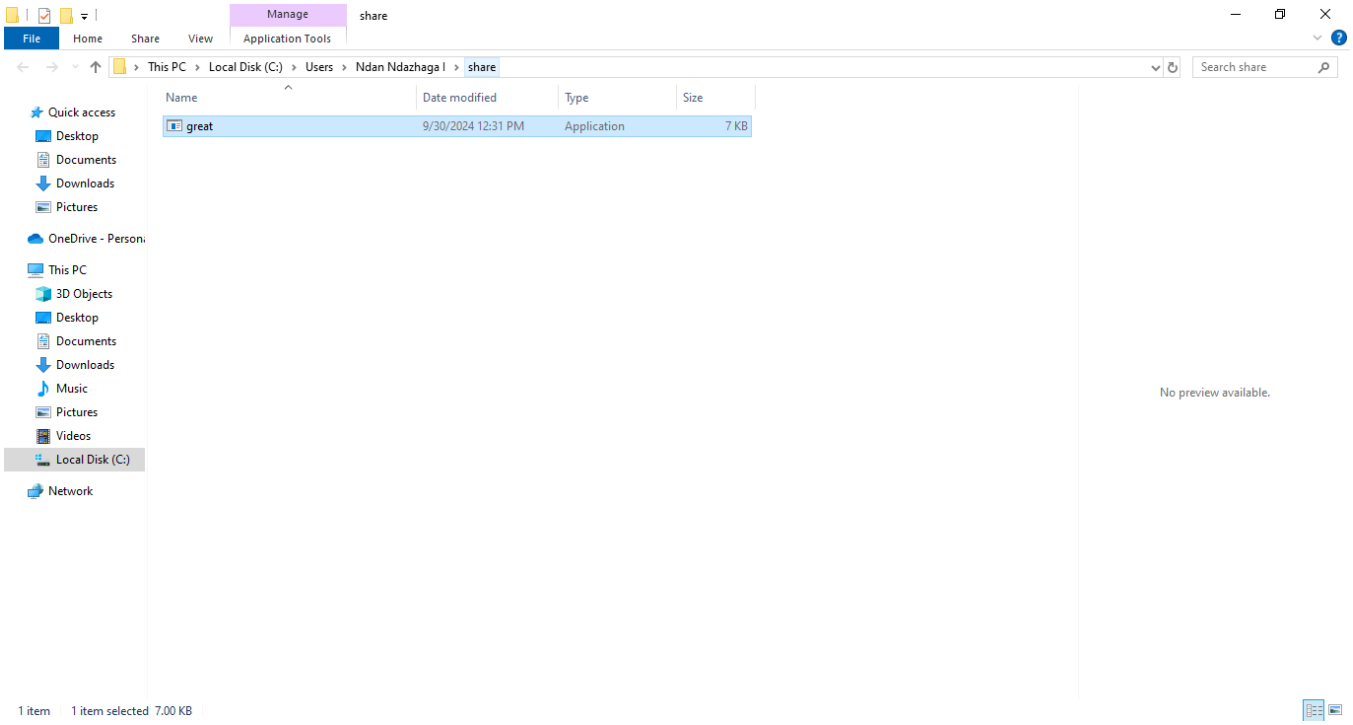
Start the Metasploit Handler: In msfconsole, start a listener for the payload then Configure the Payload Options: Set the payload to match the one you created earlier and starting the Exploit.





Step 4

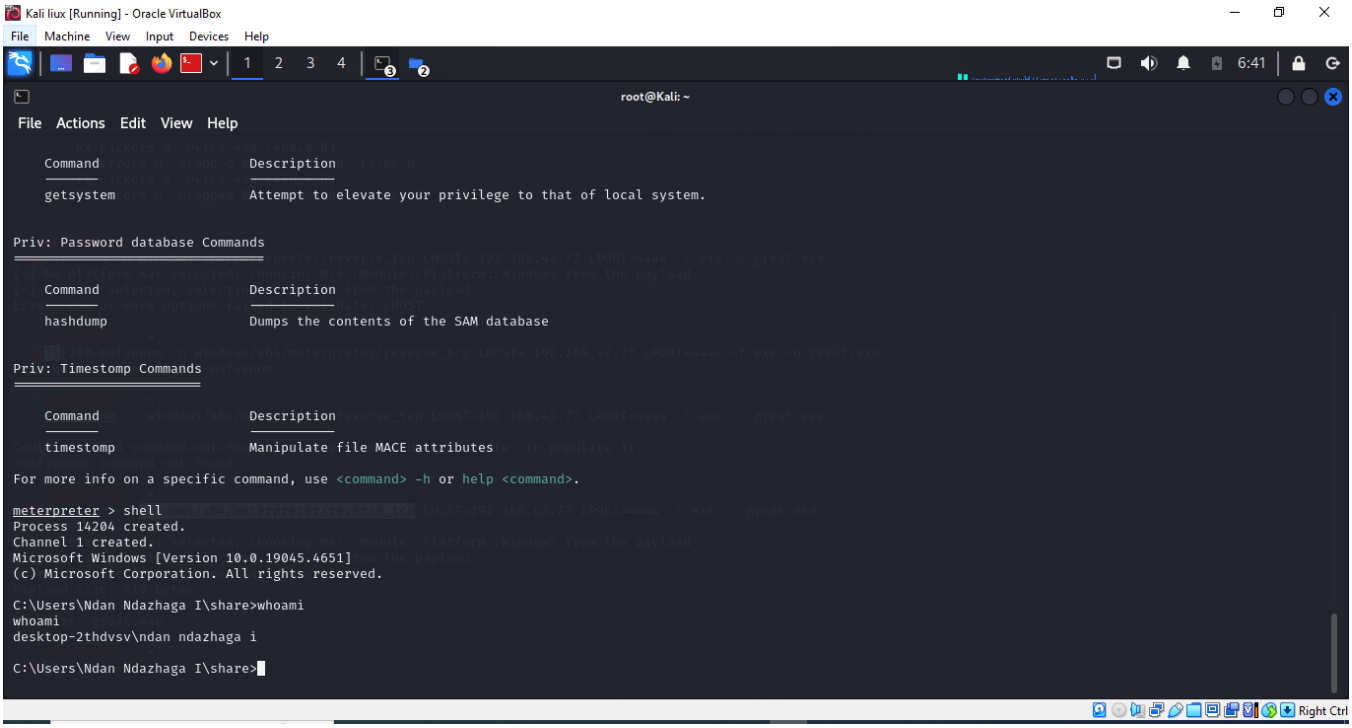
Run the Payload: On the Windows 10 machine, double-click the shell.exe payload that was transferred earlier. This will execute the reverse shell and connect back to your Kali Linux machine.



Step 5

Gain Access to the Windows Machine

Meterpreter Session: Once the payload is executed, the reverse shell connection will be established, and you'll get a Meterpreter session on Kali Linux.


Kali liux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@Kali: ~
File Actions Edit View Help
Command Description
getsystem Attempt to elevate your privilege to that of local system.
Priv: Password database Commands
Command Description
hashdump Dumps the contents of the SAM database
Priv: Timestomp Commands
Command Description
timestomp Manipulate file MACE attributes
For more info on a specific command, use <command> -h or help <command>.
meterpreter > shell
Process 14204 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Ndan Ndazhaga I\share>whoami
whoami
desktop-2thdvsv\ndan ndazhaga i
C:\Users\Ndan Ndazhaga I\share>
Right Ctrl