SHADOWFOX CYBERSECURITY INTERNSHIP REPORT

By Priyanshu Nagar

Batch -> JULY B1

ShadowFox

LEARN • CREATE • LEAD

Program Coordinator: Akash

Assistant Mentor: Manasa.G.V

Report: Beginner Level

Acknowledgement

I'm truly grateful to ShadowFox Cybersecurity for giving me the chance to be part of such a meaningful and enriching internship.

A big thank you to Mr. Akash, the Program Coordinator, whose consistent guidance and support made a big difference throughout my journey. I'm also sincerely thankful to Ms. Manasa G.V, Assistant Mentor, for her insightful advice, timely feedback, and constant encouragement that helped me confidently complete the beginner-level tasks.

This experience has greatly deepened my understanding of cybersecurity and strengthened my problem-solving abilities, all while exposing me to practical, real-world challenges.

Table of Content

List of Figures

Introduction

This report follows a practical approach in identifying and researching possible weaknesses in web environments with three essential information security techniques: port scanning, brute force on directories, and analysis of packets. These tests were executed on a highly vulnerable website, http://testphp.vulnweb.com/, that served as the test bed of all the exercises.

Stage one involved the utilization of the Nmap scanner, in that we tried probing the intended system for active services and available ports. It helped in the identification of the server's surface area that was being put up in addition to the identification of potential attack points the attacker might use.

In the second stage, a directory brute-force attack was carried out to uncover hidden paths, resources, and sensitive files that are not normally visible through standard navigation. This enumeration process was achieved using wordlists and appropriate tools to simulate a real-world attacker's perspective.

Lastly, the fourth phase consisted of Wireshark-driven packet analysis, in which the login traffic had been observed and analyzed for data transmissions that might be accessible. By intercepting and scanning packets at the time of authentication, the exercise centered on determining how securely the target processed credential exchanges as well as if sensitive data was being passed in plain-text.

Together, they simulate a real-world cybersecurity audit of a web application and provide hands-on experience in commonly used techniques of vulnerability detection and penetration testing.

Important Tools and Software Required

1. **Nmap** – Used for port scanning to detect open ports and running services on the target system.
2. **Gobuster** – Used for directory brute forcing to enumerate hidden directories and files on the web server using a wordlist.
3. **Wireshark** – A network protocol analyzer used to capture and analyze packets, particularly to monitor login credentials during traffic interception.
4. **Kali Linux** – The operating system used to run the above tools, known for its built-in cybersecurity utilities and penetration testing environment.

TASK - 1

Aim: Find all the port that are open on the website: http://testphp.vulnweb.com/

Severity: Medium and its score is 6.5 - 7.5

Reason:

This task required the use of Nmap to scan all ports on a target website and analyze the results. While the command used was simple, understanding the output and interpreting open ports and services required a basic understanding of networking concepts. Since no advanced scanning techniques or evasions were used, the task falls under a medium difficulty level.

Networking :

Networking is the process of connecting computers and devices to share data and resources. It involves communication between systems using protocols like TCP/IP. Networking enables services like websites, emails, and file sharing over the internet or local networks. It forms the backbone of modern digital communication.

Ports :

Ports are virtual connection points used by a computer to distinguish between different services and applications. Each service runs on a specific port number (like HTTP on port 80 or SSH on port 22). Ports help systems identify what type of data is being received and where to send it. Nmap scans these ports to discover open and active services on a target.

Tool Used:

Nmap (Network Mapper), It is a powerful open-source tool used for network discovery and security auditing. It scans IP addresses and ports to detect live hosts, open ports, and running services on a target system. Nmap is commonly used by ethical hackers and cybersecurity professionals to gather information before performing deeper assessments

Procedure:

Step 1: Launch the terminal in the Linux environment and ensure that the internet connection is active.

Step 2: Target Website

The target web server provided for the task was:

`http://testphp.vulnweb.com`

This domain is intentionally vulnerable and is used for safe penetration testing.

Step 3: Command Executed

```
nmap -p- testphp.vulnweb.com
```

Step 4: Command Breakdown

- `nmap`: Invokes the Nmap tool.
- `-p-`: Scans **all 65535 TCP ports** (default scan checks only top 1000 ports).
- `testphp.vulnweb.com`: Target domain.

This comprehensive scan ensures no open port is left undiscovered.

Step 5: Scan Output

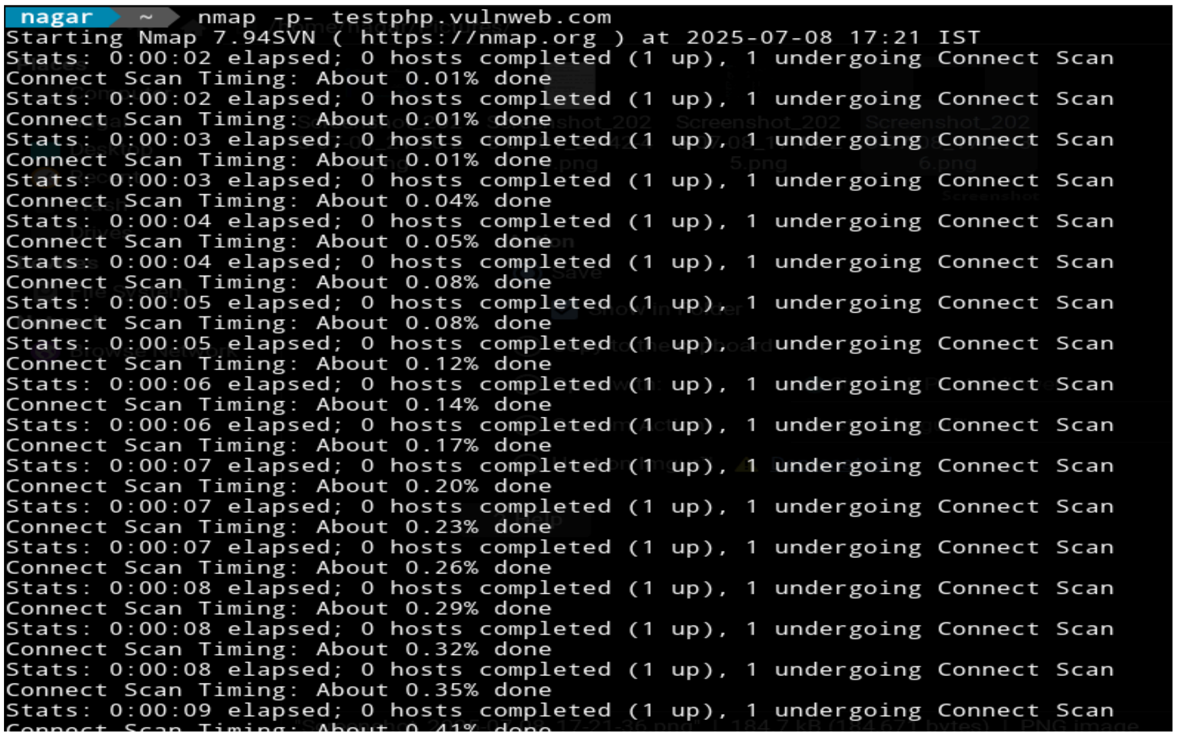
After a full TCP port scan, the result showed that only **one port** was open:

Port	**State**	**Service**
80	Open	HTTP

This means the server is hosting a web service on port 80 (usually used for websites).

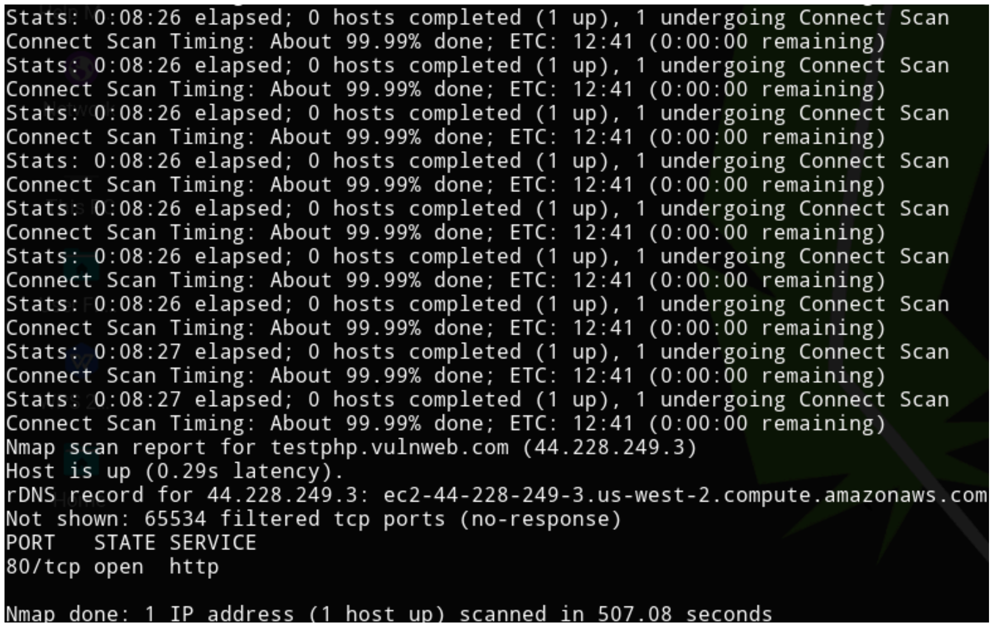
> 📌 *Note: No other ports (like FTP, SSH, HTTPS) were found open during this scan.*

Step 6: Screenshot Placeholder



```
 nagar  ~  nmap -p- testphp.vulnweb.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-08 17:21 IST
Stats: 0:00:02 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.01% done
Stats: 0:00:02 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.01% done
Stats: 0:00:03 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.01% done
Stats: 0:00:03 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.04% done
Stats: 0:00:04 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.05% done
Stats: 0:00:04 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.08% done
Stats: 0:00:05 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.08% done
Stats: 0:00:05 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.12% done
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.14% done
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.17% done
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.20% done
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.23% done
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.26% done
Stats: 0:00:08 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.29% done
Stats: 0:00:08 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.32% done
Stats: 0:00:08 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.35% done
Stats: 0:00:09 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.41% done
```

Part(i)



```
Stats: 0:08:26 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 12:41 (0:00:00 remaining)
Stats: 0:08:26 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 12:41 (0:00:00 remaining)
Stats: 0:08:26 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 12:41 (0:00:00 remaining)
Stats: 0:08:26 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 12:41 (0:00:00 remaining)
Stats: 0:08:26 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 12:41 (0:00:00 remaining)
Stats: 0:08:26 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 12:41 (0:00:00 remaining)
Stats: 0:08:26 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 12:41 (0:00:00 remaining)
Stats: 0:08:27 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 12:41 (0:00:00 remaining)
Stats: 0:08:27 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 99.99% done; ETC: 12:41 (0:00:00 remaining)
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.29s 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
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 507.08 seconds
```

Part(ii)

Limitations of Port Scanning

1. **Firewall or IDS can block scans**

   Some servers have firewalls or intrusion detection systems that hide or block port scan attempts, leading to false negatives.

2. **Only surface-level information**

   Port scanning only tells you what's open, not whether a vulnerability exists — for that, deeper vulnerability scanning is needed.

3. **UDP ports are not scanned by default**

   `nmap` by default scans **TCP ports**; for UDP, you must explicitly specify (`-sU`), which takes more time and can be unreliable.

4. **Scan results may vary**

   Due to network conditions or temporary blocking, the same scan can give different results at different times.

TASK - 2

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

Severity: Medium and it's score is 7 - 8.

Reason:

This task has **medium severity** because discovering sensitive directories can expose admin panels, configuration files, or backup folders. Though it's not directly harmful, it can lead to serious vulnerabilities if exploited further, especially on production websites.

Directories:

Web servers often have directories not linked on the website (like `/admin`, `/backup`, `/login`) which may contain sensitive data. These directories are sometimes left unprotected or forgotten by developers. Brute forcing helps reveal such hidden endpoints that attackers can potentially misuse.

Brute Forcing:

Brute forcing in this context means sending a large number of directory name requests to the server using a wordlist. The goal is to guess valid paths based on common names. If a request returns a status code like `200 OK` or `403 Forbidden`, it means the directory exists.

Tool Used:

Gobuster v3.6 — a fast and flexible directory brute forcing tool commonly used during web reconnaissance. It sends multiple HTTP requests to the target URL using a dictionary file and checks for valid directories based on the response status. It's lightweight, fast, and works well in discovering hidden paths on web servers.

Procedure:

Step 1: Launch the terminal in the Linux environment and ensure that the internet connection is active.

Step 2: Target Selection

The target web application for directory brute forcing was:

http://testphp.vulnweb.com/

Step 3: Wordlist Selection

For the brute-force attack, the commonly used wordlist:

common.txt

was selected from the local wordlists directory. It contains frequently used directory and file names to maximize discovery chances.

Step 4: Command Executed

```
gobuster dir -u http://testphp.vulnweb.com/ -w ~/wordlists/common.txt
```

Explanation of command parameters:

- dir → Tells Gobuster to perform directory brute-forcing.
- -u → Specifies the target URL.
- -w → Specifies the wordlist used.

Step 5: Scan Configuration

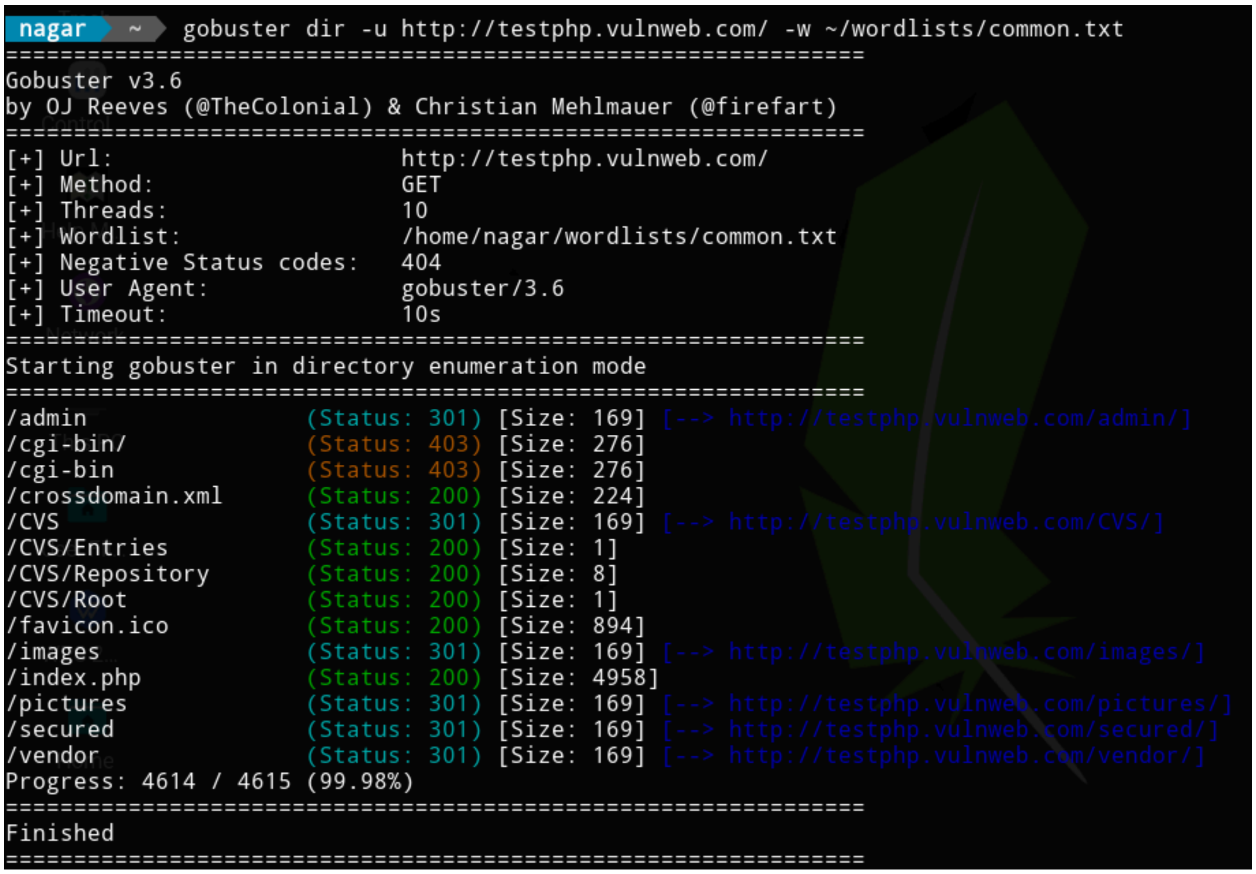
- Threads Used: 10
- Request Method: GET
- Timeout: 10 seconds
- Excluded Status Codes: 404 (to ignore 'Not Found' responses)

Step 6: Observed Results

The scan successfully discovered several directories and files with various HTTP status codes. Key findings include:

Path	Status Code	Description
/admin	301	Redirected, likely admin panel
/cgi-bin/	403	Forbidden, access restricted
/images/	301	Contains static media
/secured	301	Potentially protected content
/index.php	200	Main landing page of the site
/vendor/	301	Possible libraries or frameworks

Step 7: Screenshot Placeholder



```
nagar  ~  gobuster dir -u http://testphp.vulnweb.com/ -w ~/wordlists/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://testphp.vulnweb.com/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /home/nagar/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/admin                (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/admin/]
/cgi-bin/             (Status: 403) [Size: 276]
/cgi-bin              (Status: 403) [Size: 276]
/crossdomain.xml      (Status: 200) [Size: 224]
/CVS                  (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/CVS/]
/CVS/Entries          (Status: 200) [Size: 1]
/CVS/Repository       (Status: 200) [Size: 8]
/CVS/Root             (Status: 200) [Size: 1]
/favicon.ico          (Status: 200) [Size: 894]
/images               (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/images/]
/index.php            (Status: 200) [Size: 4958]
/pictures             (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/pictures/]
/secured              (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/secured/]
/vendor               (Status: 301) [Size: 169] [--> http://testphp.vulnweb.com/vendor/]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

Limitations / Precautions:

1. Brute forcing can generate heavy traffic and may be detected by intrusion detection systems.
2. Some directories may respond with misleading status codes (e.g., custom 403/404).
3. If rate-limiting is enabled on the server, it may block your IP.
4. Ethical boundaries must be respected: always perform such tasks only on authorized systems.

TASK - 3

Aim: 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.

Severity: Hard and its scoring is 8 - 9

Reason:

This task is marked **hard** because it involves real-time packet interception and analyzing login credentials over a network. Identifying sensitive data from raw packets requires strong understanding of HTTP protocols and packet structure.

Packets:

Packets are small units of data transmitted over a network. Each packet contains headers and payload data used for communication between devices.

IP Address, Source IP, Destination IP:

An IP address identifies a device on a network.

- **Source IP** is the address from which the data is sent.
- **Destination IP** is where the data is being delivered.

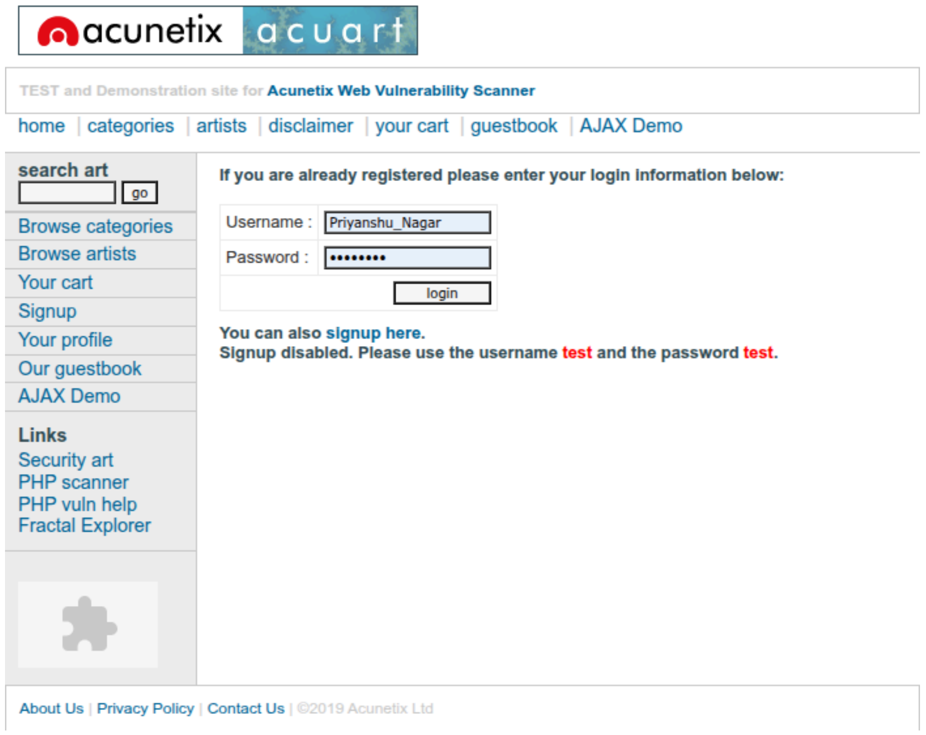
Tool Used – Wireshark:

Wireshark is an open-source packet analyzer tool that captures, filters, and inspects data flowing across a network in real time.

Procedure (Step-by-Step):

Step 1: Open browser and visit http://testphp.vulnweb.com/

Step 2: Log in with your internship-assigned **username** (e.g., `priyanshunagar`)


acunetix acuart
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
Fractal Explorer
login
About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd

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

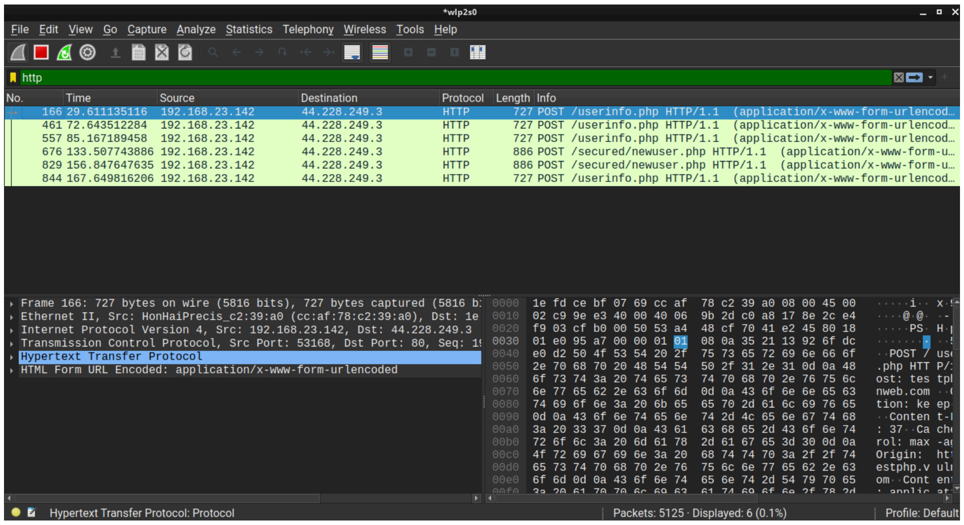
Username : Priyanshu_Nagar
Password : ••••••••

You can also signup here.
Signup disabled. Please use the username test and the password test.

Step 3: Open **Wireshark** and start a live capture on your active network interface.

Step 4: Filter packets using `http` in the Wireshark filter bar to reduce noise.

Step 5: Screenshot Placeholder


*wlp2s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.  Time           Source          Destination   Protocol Length Info
 166 29.611135116   192.168.23.142  44.228.249.3  HTTP     727 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencod…
 461 72.643512284   192.168.23.142  44.228.249.3  HTTP     727 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencod…
 557 85.167189458   192.168.23.142  44.228.249.3  HTTP     727 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencod…
 676 133.507743886  192.168.23.142  44.228.249.3  HTTP     886 POST /secured/newuser.php HTTP/1.1  (application/x-www-form-u…
 829 156.847647635  192.168.23.142  44.228.249.3  HTTP     886 POST /secured/newuser.php HTTP/1.1  (application/x-www-form-u…
 844 167.649816206  192.168.23.142  44.228.249.3  HTTP     727 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencod…
```

```
Frame 166: 727 bytes on wire (5816 bits), 727 bytes captured (5816 b
Ethernet II, Src: HonHaiPrecis_c2:39:a0 (cc:af:78:c2:39:a0), Dst: 1e
Internet Protocol Version 4, Src: 192.168.23.142, Dst: 44.228.249.3
Transmission Control Protocol, Src Port: 53168, Dst Port: 80, Seq: 1
Hypertext Transfer Protocol
HTML Form URL Encoded: application/x-www-form-urlencoded
```

```
0000  1e fd ce bf 07 69 cc af  78 c2 39 a0 08 00 45 00   ·····i·· x·
0010  02 c9 9e e3 40 00 40 06  9b 2d c0 a8 17 8e 2c e4   ····@·@· ·-
0020  f9 03 cf b0 00 50 53 a4  48 cf 70 41 e2 45 80 18   ·····PS· H·
0030  01 e0 95 a7 00 00 01 01  08 0a 35 21 13 92 6f dc   ········ ··
0040  e0 d2 50 4f 53 54 20 2f  75 73 65 72 69 6e 66 6f   ··POST / use
0050  2e 70 68 70 20 48 54 54  50 2f 31 2e 31 0d 0a 48   .php HTT P/
0060  6f 73 74 3a 20 74 65 73  74 70 68 70 2e 76 75 6c   ost: tes tph
0070  6e 77 65 62 2e 63 6f 6d  0d 0a 43 6f 6e 6e 65 63   nweb.com ··
0080  74 69 6f 6e 3a 20 6b 65  65 70 2d 61 6c 69 76 65   tion: ke ep
0090  0d 0a 43 6f 6e 74 65 6e  74 2d 4c 65 6e 67 74 68   ··Conten t-
00a0  3a 20 33 37 0d 0a 43 61  63 68 65 2d 43 6f 6e 74   : 37··Ca che
00b0  72 6f 6c 3a 20 6d 61 78  2d 61 67 65 3d 30 0d 0a   rol: max -a
00c0  4f 72 69 67 69 6e 3a 20  68 74 74 70 3a 2f 2f 74   Origin:  ht
00d0  65 73 74 70 68 70 2e 76  75 6c 6e 77 65 62 2e 63   estphp.v ul
00e0  6f 6d 0d 0a 43 6f 6e 74  65 6e 74 2d 54 79 70 65   om··Cont ent
```

Hypertext Transfer Protocol: Protocol   Packets: 5125 · Displayed: 6 (0.1%)   Profile: Default

Limitations:

1. Wireshark only captures unencrypted data. It is ineffective against websites using HTTPS (TLS encryption).
2. Requires the user to be on the same network as the victim to sniff their traffic (e.g., public Wi-Fi).
3. Packet capture can generate huge logs; filtering and analysis require practice and expertise.
4. Many modern sites use secure authentication and techniques like tokenization, rendering this method ineffective.

Conclusion

During the cybersecurity internship at ShadowFox, I successfully completed three foundational yet impactful tasks using industry-standard tools: Nmap for port scanning, Gobuster for directory brute forcing, and Wireshark for credential sniffing. Each task deepened my understanding of the various stages of penetration testing — reconnaissance, enumeration, and exploitation.

Through these practical exercises, I learned how exposed ports can reveal vital services, how hidden web directories might leak sensitive data, and how unsecured communication channels can lead to credential theft.This hands-on experience not only enhanced my technical skills but also sharpened my mindset as a security professional — always questioning, analyzing, and testing every layer of a digital system.

References

1. Nmap Official Documentation

   https://nmap.org/book/

   (Covers usage, scan types, and flags used in task 1)
2. OWASP DirBuster Project

   https://owasp.org/www-project-dirbuster/

   (Dirb and DirBuster work similarly; useful for understanding brute force directory discovery)
3. Wireshark User Guide

   https://www.wireshark.org/docs/wsug_html_chunked/

   (Covers packet analysis, filtering techniques, and best practices)
4. OWASP Testing Guide v4

   https://owasp.org/www-project-web-security-testing-guide/

   (Explains how to test for common web app vulnerabilities like insecure login forms)
5. Mozilla Developer Network – HTTP vs HTTPS

   https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

   (Helpful for understanding how unencrypted HTTP traffic is vulnerable to sniffing)
6. MITRE ATT&CK Framework – Initial Access & Credential Access

   https://attack.mitre.org/

   (A trusted framework describing attacker behavior, matching techniques used in all 3 tasks)