Cyber Security Internship - Beginner Level Report

Topic:
Basic Web Security Assessment - testphp.vulnweb.com

Name: **N. Jashwanth Reddy**

Batch: **October B1**

Table of Contents

List of Figures

Introduction

This report documents three beginner-level tasks performed against the target web application provided in the internship task list: http://testphp.vulnweb.com/. The purpose of these exercises is to practice basic web security assessment techniques including port scanning, directory enumeration, and network traffic interception. All activities described here are intended to be performed in a controlled lab environment or on systems where the user has explicit permission.

Information about the Target / Machine

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

Description: Public test web application provided for security testing practice. The target is used to demonstrate common web vulnerabilities and assessment techniques. The following sections provide the attack vector plans, commands executed, expected outputs, and remediation suggestions.

Task 1- Port Scanning

Attack name: Port Scanning (Service Discovery)

Severity: Information Gathering — Score: 3.3
(Low)

Impact:

Port scanning itself is an information-gathering technique. It reveals open ports and services which an attacker could use as an attack surface. If sensitive services (e.g., database or remote management) are exposed to the internet, the risk increases.

Steps to reproduce:

1. Open a terminal on your attacker VM.
2. Run an Nmap TCP SYN scan with service/version detection.
   Command: nmap -sS -sV testphp.vulnweb.com
3. Review the output for open ports and service versions.

Example Command and Expected Output:

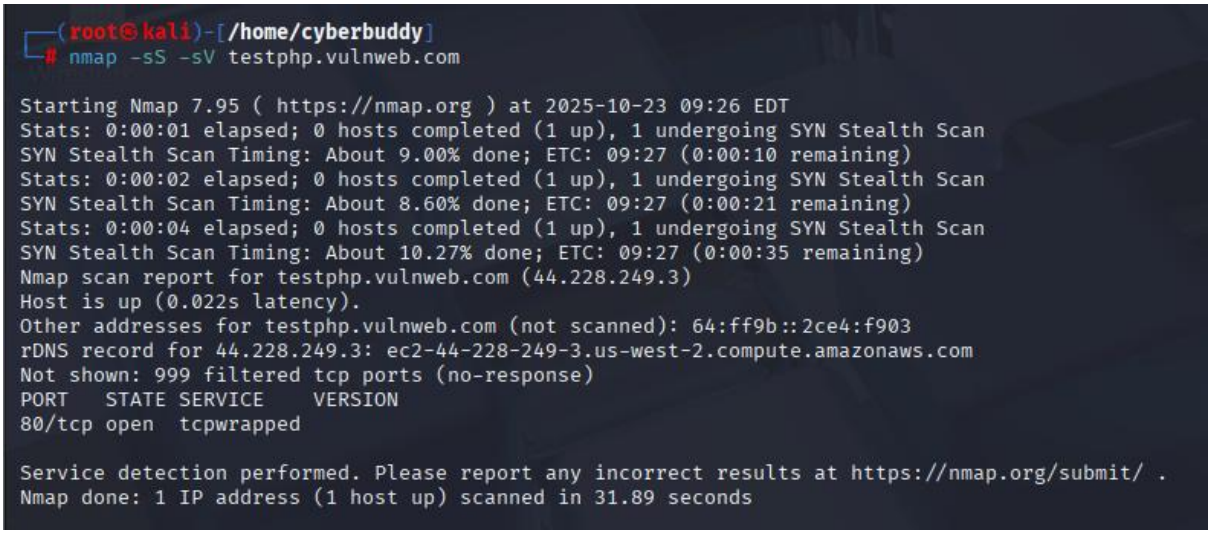
Command: `nmap -sS -sV testphp.vulnweb.com`

Example output (sample):

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-23 09:26 EDT

Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan

SYN Stealth Scan Timing: About 9.00% done; ETC: 09:27 (0:00:10 remaining)

Stats: 0:00:02 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan

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

Host is up (0.022s latency).

Other addresses for testphp.vulnweb.com (not scanned): 64:ff9b::2ce4:f903

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

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

PORT   STATE SERVICE    VERSION

80/tcp open  tcpwrapped
```

```
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 31.89 seconds
```

Figure 1: Nmap scan output



Mitigation Steps:

- Restrict access to management services (SSH, FTP) using firewalls and IP whitelisting.
- Disable unnecessary services or bind them to localhost where possible.
- Keep services updated and patch known vulnerabilities.
- Use strong authentication methods (key-based SSH, disable password root login).

Task 2- Directory Brute-force

Attack name: Directory/Endpoint Enumeration (Brute-force)
Severity: Medium — Score:
6.5

Impact:

Exposed directories (e.g., /admin, /uploads) can host administrative panels, allow file uploads, or expose sensitive files. Attackers can use these to find vulnerabilities, upload backdoors, or access restricted functionality.

Steps to reproduce:

1. Use a directory brute-force tool such as dirb or gobuster.
2. Select a common wordlist (e.g., /usr/share/wordlists/dirb/common.txt or rockyou variants).
3. Run the scan against the target base URL.

Example (dirb):

dirb http://testphp.vulnweb.com/ /usr/share/wordlists/dirb/common.txt

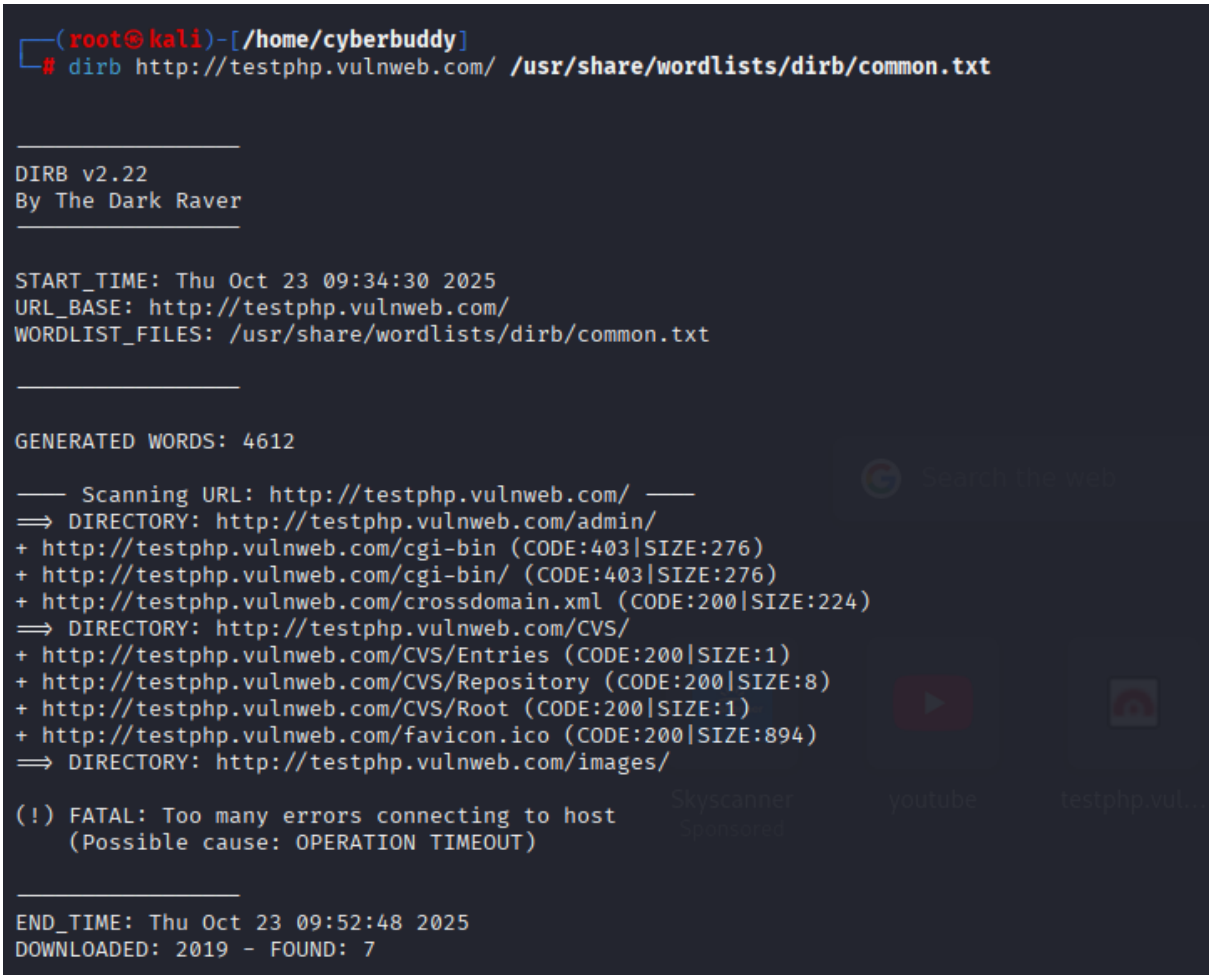
Example output :

```
┌──(root㉿kali)-[/home/cyberbuddy]
└─# dirb http://testphp.vulnweb.com/ /usr/share/wordlists/dirb/common.txt


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

START_TIME: Thu Oct 23 09:34:30 2025
URL_BASE: http://testphp.vulnweb.com/
WORDLIST_FILES: /usr/share/wordlists/dirb/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/

(!) FATAL: Too many errors connecting to host
```

(Possible cause: OPERATION TIMEOUT)

END_TIME: Thu Oct 23 09:52:48 2025
DOWNLOADED: 2019 - FOUND: 7

Figure 2: Dirb/Gobuster results.



Mitigation Steps:

- Remove or restrict access to unused directories; apply authentication to admin panels.
- Use proper access controls on upload directories and validate/scan uploaded files.
- Implement web application firewall (WAF) rules to block suspicious enumeration traffic.
- Use robots.txt only for crawling hints (do not rely on it for security).

Task 3- Intercept Login Traffic and Find Credentials

Attack name: Network Sniffing (Credentials Interception)

Severity: High — Score: 9.8

Impact:

If login credentials are transmitted in plaintext over HTTP, an attacker on the same network or any man-in-the-middle position can capture user credentials. This can lead to account takeover, data breach, and unauthorized access to sensitive resources.

Steps to reproduce (lab only):

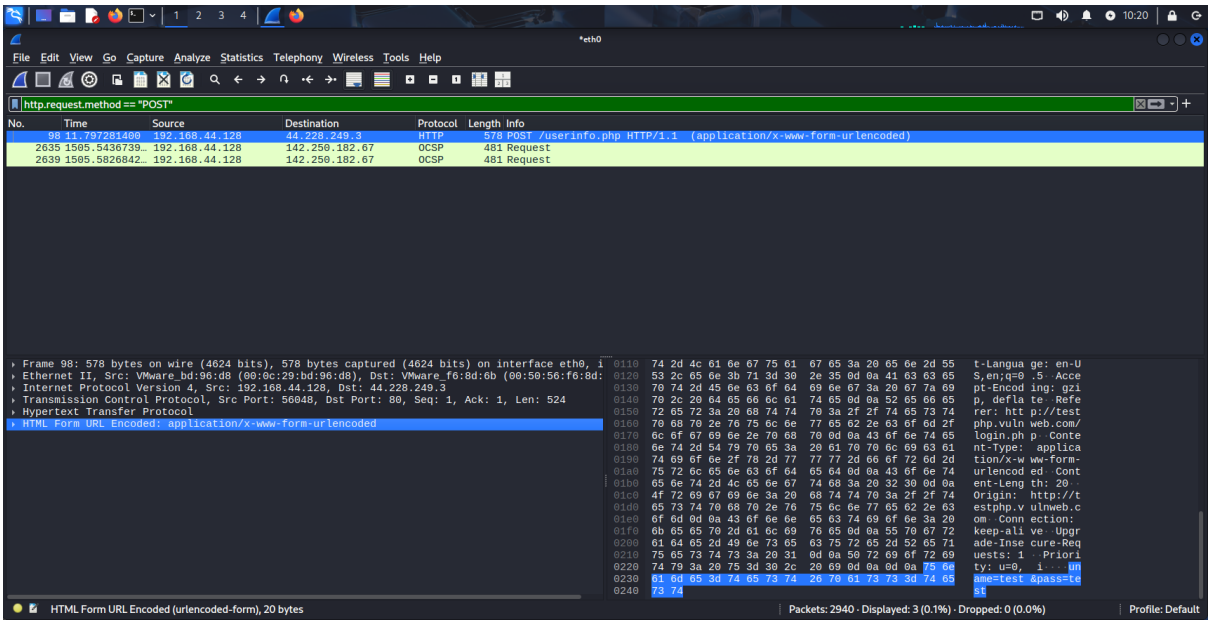
1. Start Wireshark on the attacker machine and capture on the active interface.
2. Filter for HTTP POST requests: http.request.method == "POST"
3. Perform a login on the target (use test/dummy credentials in a lab).
4. Inspect the POST request body to find fields like username and password in plaintext.

Example captured POST body :

```
0000   00 50 56 f6 8d 6b 00 0c 29 bd 96 d8 08 00 45 00   .PV..k..).....E.
0010   02 34 36 60 40 00 40 06 ef 53 c0 a8 2c 80 2c e4   .46`@.@..S..,.,.
0020   f9 03 da f0 00 50 66 e0 96 61 2a cc 09 d4 50 18   .....Pf..a*...P.
0030   fa f0 15 37 00 00 50 4f 53 54 20 2f 75 73 65 72   ...7..POST /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 (X11; 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) Gecko/201001
00b0   30 31 20 46 69 72 65 66 6f 78 2f 31 32 38 2e 30   01 Firefox/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,application/
00e0   78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 70 6c 69 63   xhtml+xml,applic
00f0   61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d 30 2e 39 2c   ation/xml;q=0.9,
0100   2a 2f 2a 3b 71 3d 30 2e 38 0d 0a 41 63 63 65 70   */*;q=0.8..Accep
0110   74 2d 4c 61 6e 67 75 61 67 65 3a 20 65 6e 2d 55   t-Language: en-U
0120   53 2c 65 6e 3b 71 3d 30 2e 35 0d 0a 41 63 63 65   S,en;q=0.5..Acce
0130   70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 67 7a 69   pt-Encoding: gzi
0140   70 2c 20 64 65 66 6c 61 74 65 0d 0a 52 65 66 65   p, deflate..Refe
0150   72 65 72 3a 20 68 74 74 70 3a 2f 2f 74 65 73 74   rer: http://test
0160   70 68 70 2e 76 75 6c 6e 77 65 62 2e 63 6f 6d 2f   php.vulnweb.com/
0170   6c 6f 67 69 6e 2e 70 68 70 0d 0a 43 6f 6e 74 65   login.php..Conte
0180   6e 74 2d 54 79 70 65 3a 20 61 70 70 6c 69 63 61   nt-Type: applica
0190   74 69 6f 6e 2f 78 2d 77 77 77 2d 66 6f 72 6d 2d   tion/x-www-form-
01a0   75 72 6c 65 6e 63 6f 64 65 64 0d 0a 43 6f 6e 74   urlencoded..Cont
01b0   65 6e 74 2d 4c 65 6e 67 74 68 3a 20 32 30 0d 0a   ent-Length: 20..
01c0   4f 72 69 67 69 6e 3a 20 68 74 74 70 3a 2f 2f 74   Origin: http://t
01d0   65 73 74 70 68 70 2e 76 75 6c 6e 77 65 62 2e 63   estphp.vulnweb.c
01e0   6f 6d 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20   om..Connection:
01f0   6b 65 65 70 2d 61 6c 69 76 65 0d 0a 55 70 67 72   keep-alive..Upgr
0200   61 64 65 2d 49 6e 73 65 63 75 72 65 2d 52 65 71   ade-Insecure-Req
```

```
0210   75 65 73 74 73 3a 20 31 0d 0a 50 72 69 6f 72 69   uests: 1..Priori
0220   74 79 3a 20 75 3d 30 2c 20 69 0d 0a 0d 0a 75 6e   ty: u=0, i....un
0230   61 6d 65 3d 74 65 73 74 26 70 61 73 73 3d 74 65   ame=test&pass=te
0240   73 74                                             st
```

Figure 3: Wireshark captured POST request.



Mitigation Steps:

- Enforce HTTPS for the entire website (redirect HTTP to HTTPS) using a valid TLS certificate.
- Use HSTS (HTTP Strict Transport Security) to prevent protocol downgrade attacks.
- Implement secure cookie flags (Secure, HttpOnly) and consider multi-factor authentication.
- Avoid sending credentials in query parameters or in cleartext; always use encrypted channels.

References

Nmap Documentation - https://nmap.org/book/ (useful for port scanning options)
Dirb / Gobuster manuals and wordlists (installed on Kali Linux)
Wireshark User Guide - https://www.wireshark.org/docs/wsug_html_chunked/
OWASP Guidance - https://owasp.org/ (for mitigation and best practices)

Resources Used

Kali Linux tools: nmap, dirb, gobuster, wireshark
/usr/share/wordlists/dirb/common.txt and custom wordlists
Test target: http://testphp.vulnweb.com/ (as provided in internship task list)
Vulnerable webapp documentation and OWASP resources