# **Penetration Testing Report**

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

**Test Date:** April 13, 2025  
 **Tester:** [Your Name]  
 **Purpose:** Directory enumeration, brute force login testing, and administrative access verification.

## **🔍 Executive Summary**

A penetration test was performed on the website http://testphp.vulnweb.com to assess its resilience to directory enumeration and brute-force attacks. The test revealed several publicly accessible directories and an **exposed administrative interface** vulnerable to default credentials.

**Critical findings** include:

* Successful access to the /admin/ panel with **default credentials**.
* Exposure of **sensitive user data**, including credit card information.
* Accessible source control and configuration directories such as /CVS/, /vendor/, and /secured/.

These vulnerabilities pose a **severe security risk** and require immediate remediation.

## **⚙️ Methodology**

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1. | Verified website availability using curl. |
| 2. | Conducted directory enumeration using dirb and gobuster. |
| 3. | Bypassed potential blocks via User-Agent spoofing. |
| 4. | Identified accessible directories and manually explored high-risk paths. |
| 5. | Attempted login to the /admin/ interface using common/default credentials. |
| 6. | Validated data exposure on authenticated admin interface. |

## **📂 Technical Findings**

### **✅ Step 1 – Website Reachability Check**

**Command:**

bash

CopyEdit

curl -I http://testphp.vulnweb.com/

**Result:** HTTP/1.1 200 OK – Site is reachable. Server identified as nginx/1.19.0. PHP version: 5.6.40.

### **❌ Step 2 – Initial Directory Brute Force (DIRB)**

**Tool:** dirb  
 **Result:** Scan failed – multiple connection issues.

### **🔁 Step 3 – Directory Enumeration (User-Agent Spoofing)**

**Tool:** gobuster  
 **Wordlist:** /usr/share/wordlists/dirb/common.txt  
 **Notable Directories Identified:**

|  |  |  |
| --- | --- | --- |
| **Path** | **Status** | **Notes** |
| /admin | 301 | Redirects to /admin/ |
| /secured | 301 | Sensitive-sounding directory |
| /vendor | 301 | Could expose libraries/configs |
| /cgi-bin/ | 403 | Access restricted |
| /CVS/ | 301 | Version control data |
| /CVS/Entries | 200 | Access granted |
| /index.php | 200 | Main index |
| /crossdomain.xml | 200 | Public XML policy |
| /favicon.ico | 200 | Common file |

Multiple attempts triggered **connection resets**, indicating possible IDS/IPS involvement.

### **🧠 Step 4 – Admin Login Attempt**

**Target URL:** http://testphp.vulnweb.com/admin/  
 **Credentials Used:** Username: test Password: test  
 **Status:** **Login successful**

**Exposed Data Inside Admin Panel:**

* **Name:** John Smith
* **Credit Card Number:** 1234-5678-2300-9000
* **Email:** email@email.com
* **Phone Number:** 2323345
* **Address:** 21 street

This is a **severe data exposure** that violates data protection standards and indicates insecure access control.

### **🚨 Step 5 – Aggressive Wordlist Scan**

**Tool:** gobuster  
 **Wordlist:** /usr/share/wordlists/dirb/big.txt  
 **Findings:** Confirmed previous results and exposed additional attack surface with various directory names triggering 403s and timeouts.

## **⚠️ Risk Assessment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Impact** | **Likelihood** | **Severity** |
| Default Admin Credentials | High | High | **Critical** |
| Sensitive Data Exposure | High | High | **Critical** |
| Public Admin Access | High | High | **Critical** |
| Unrestricted Directory Listing | Medium | Medium | **High** |
| Server Misconfigurations | Medium | High | **High** |
| Source Control Access (/CVS/) | Medium | Medium | **High** |

## **🛠️ Recommendations**

1. **Immediately change all default credentials.**
2. **Restrict access** to /admin/ via IP whitelisting or secure login mechanisms.
3. Implement **multi-factor authentication** (MFA) for administrative accounts.
4. Remove or restrict access to **source control directories** like /CVS/.
5. Configure the web server to **suppress verbose error messages** and block directory listings.
6. Deploy a **Web Application Firewall (WAF)** to detect and block brute force/directory scanning attempts.
7. Perform regular **vulnerability assessments** and update server software (PHP 5.6 is outdated and unsupported).
8. Ensure **compliance** with data protection regulations (e.g., GDPR, PCI-DSS).

## **📝 Conclusion**

The test confirms that http://testphp.vulnweb.com contains serious security flaws, including unrestricted administrative access and critical data exposure. Without remediation, the site remains highly vulnerable to exploitation.