CYBER SECURITY DIVISION

Vulnerability Assessment Report

Comprehensive security analysis and risk evaluation for live web infrastructure.

PREPARED BY
Shubhrat Chaursiya
Cyber Security Intern

ORGANIZATION
Future Interns
Task 1 : Internship Program

CID
FIT/JAN26/CS5559

INTERNSHIP DATE
5 Jan, 2026 - 5 Feb, 2026

Objective & Scope

02

Core Objective

The primary objective is to identify security vulnerabilities in the target live website using industry-standard passive testing methodologies. This assessment aims to evaluate risk levels and provide actionable remediation steps to improve the overall security posture.

Scope of Assessment

PRIMARY TARGET
`https://testphp.vulnweb.com`

ASSESSMENT TYPE
Passive Security Audit

- Web application vulnerabilities
- Security misconfigurations
- HTTP Header & Information Disclosure

⚠ **Constraint Note:** No intrusive or harmful exploits were performed. The testing remained strictly within the boundaries of passive observation and ethical scanning.

Tools & Methodology

03

Nmap
Network discovery & port scanning.

OWASP ZAP
Automated passive web scanning.

DevTools
Header analysis and manual inspection.

Canva
Professional report design & layout.

Workflow Process

1. Reconnaissance & Asset Identification
2. Passive Vulnerability Identification
3. Risk Classification & Threat Modeling
4. Remediation Strategy Development

Findings: High Risk

04

Vulnerability 1: SQL Injection — CRITICAL

STATUS
Unpatched

IMPACT
Data Extraction

VECTOR
Web Forms/URL

Description

The application accepts unsanitized user input in query parameters, which allows an attacker to inject malicious SQL commands directly into the backend database.

Technical Impact

- Unauthorized access to sensitive user records.
- Modification or deletion of database contents.
- Potential compromise of the entire web server.

Recommended Remediation

Implement **Parameterized Queries** (Prepared Statements) for all database interactions. This ensures the database treats input as data, not as executable code.

```
// SECURE PATTERN EXAMPLE
$stmt = $pdo->prepare('SELECT * FROM users WHERE id = :id');
$stmt->execute(['id' => $user_id]);
```

Findings: Medium & Low

05

V2: Missing Security Headers — MEDIUM

Description: Vital HTTP security headers such as `Content-Security-Policy` (CSP) and `X-Frame-Options` are not present in server responses.

Impact: High susceptibility to Clickjacking attacks and Cross-Site Scripting (XSS).

```
# Recommendation: Add to .htaccess or Nginx Config
Header set X-Frame-Options "SAMEORIGIN"
Header set Content-Security-Policy "default-src 'self';"
```

V3: Information Disclosure — LOW

Description: The server header reveals internal software versions (e.g., PHP/5.3.10, Apache/2.2.22).

Impact: Attackers can pinpoint known exploits for specific legacy software versions.

Fix: Disable `ServerTokens` and `ServerSignature` in server configuration.

Risk Summary Table

Vulnerability	Risk	Status
SQL Injection	High	🔴 Critical
Missing Security Headers	Medium	🟠 Pending
Information Disclosure	Low	🟡 Information

Final Summary

06

✅ Conclusion

The assessment of `testphp.vulnweb.com` highlights significant security weaknesses, primarily the Critical-level SQL Injection vulnerability. While the application is a testing platform, these findings represent real-world threats that could lead to complete system compromise if found in a production environment. Prompt implementation of the recommended fixes—specifically input sanitization and server hardening—is essential.

🎓 Learning Outcomes

TECHNICAL
Practical usage of industry-standard tools like Nmap and OWASP ZAP.

ANALYTICAL
Understanding vulnerability classification and severity mapping.

STRATEGIC
Translating complex code flaws into business-friendly remediation.

REPORTING
Mastering professional cyber security documentation standards.