

FUTURE INTERNS

Cyber Security Internship | Task 1

VULNERABILITY ASSESSMENT REPORT

Scanned with: OWASP ZAP 2.17.0 | Nmap 7.94 | Browser DevTools

TARGET

<http://testphp.vulnweb.com>

1

HIGH
FINDINGS

4

MEDIUM
FINDINGS

4

LOW
FINDINGS

4

INFO
FINDINGS

✓ Real scan performed | 13 alerts detected | OWASP ZAP 2.17.0 | 19 Feb 2026

Prepared by:

BATHO PILIOUZIWE ABRAHAM

Cyber Security Intern | Future Interns Fellowship Program | IPNET Institute of Technology

February 2026

GitHub: FUTURE_CS_01

1. Executive Summary

A full vulnerability assessment was conducted on testphp.vulnweb.com using OWASP ZAP 2.17.0 (active + passive scan), Nmap, and Browser DevTools. The scan detected 13 real alerts across 4 severity levels. Critical findings include confirmed SQL Injection and XSLT Injection. All results are based on live scan data obtained on 19 Feb 2026.

2. Target & Scan Information

Target URL	http://testphp.vulnweb.com
IP / Server	nginx/1.19.0 (confirmed via ZAP response header)
PHP Version	PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1 (X-Powered-By header)
Protocol	HTTP only — port 443 (HTTPS) closed
Scan Tool	OWASP ZAP 2.17.0 — Active + Passive scan
Total Alerts	13 alerts 1 High 4 Medium 4 Low 4 Info
Assessment Date	19 February 2026
Assessment Type	Black-box — Active & Passive automated scan + manual verification

3. Methodology

01

RECON
Nmap scan
ports & services

02

ZAP SPIDER
Automated crawl
of all pages

03

ACTIVE SCAN
ZAP active scan
13 alerts found

04

MANUAL
DevTools header
& cookie check

05

REPORT
CVSS scoring
& documentation

4. Real HTTP Response Headers (captured by ZAP)

HEADER / VALUE	SECURITY NOTE
HTTP/1.1 200 OK	Status — No HTTPS redirect
Server: nginx/1.19.0	■ Reveals exact server version — VF-07
X-Powered-By: PHP/5.6.40...	■ Reveals PHP version (outdated) — VF-07
Content-Type: text/html; UTF-8	OK
Connection: keep-alive	OK
No X-Frame-Options	■ Missing — clickjacking possible — VF-04
No Content-Security-Policy	■ Missing — XSS amplified — VF-04
No X-Content-Type-Options	■ Missing — MIME sniffing — VF-05

5. Vulnerability Findings (Real ZAP Results)

VF-01 SQL Injection — MySQL

CVSS 7.8 / HIGH Critical

URL: /search.php?test=%27 | Parameter: test | Attack: '

DESCRIPTION

ZAP confirmed SQL Injection via error-based detection. The parameter 'test' is passed directly into a MySQL query. Error message exposed: 'You have an error in your SQL syntax'.

BUSINESS IMPACT

Full database compromise, unauthorized data extraction, data manipulation or deletion, potential remote code execution.

EVIDENCE (ZAP)

ZAP Proof: 'You have an error in your SQL syntax' — CWE-89, WASC-19, Scan #40018

REMEDIATION

Use parameterized queries (prepared statements). Implement input validation. Deploy WAF. Restrict database user privileges to minimum required.

VF-02 XSLT Injection

CVSS 7.8 / MEDIUM High

URL: Multiple pages | Source: OWASP ZAP Active Scan

DESCRIPTION

XSLT (Extensible Stylesheet Language Transformation) injection was detected. Unsanitized user input is processed by an XSLT parser, allowing transformation logic manipulation.

BUSINESS IMPACT

Server-side file read, arbitrary code execution via XSLT extensions, data exfiltration, application logic bypass.

EVIDENCE (ZAP)

ZAP Alert: XSLT Injection (2 instances) — CWE-91

REMEDIATION

Sanitize and validate all user input before XSLT processing. Disable dangerous XSLT functions. Use a secure XML parser with restricted capabilities.

VF-03 Absence of Anti-CSRF Tokens

CVSS 7.8 / MEDIUM Medium

URL: All forms (login, search, comment) | Type: Systemic

DESCRIPTION

No CSRF tokens are present in any form. An attacker can trick authenticated users into unknowingly submitting malicious requests by embedding forms on external sites.

BUSINESS IMPACT

Unauthorized actions performed on behalf of authenticated users — password changes, purchases, data deletion without user consent.

EVIDENCE (ZAP)

ZAP Alert: Absence de Jetons Anti-CSRF (Systemic) — CWE-352

REMEDIATION

Implement CSRF tokens (synchronizer token pattern) on all state-changing forms. Use SameSite=Strict cookie attribute. Validate Origin/Referer headers.

5. Vulnerability Findings (continued)

VF-04

Content-Security-Policy (CSP) Header Not Set

CVSS MEDIUM / Medium

URL: All pages | Type: Systemic

DESCRIPTION

The Content-Security-Policy header is absent on all responses. CSP is a critical browser security mechanism that restricts which resources can be loaded and executed.

BUSINESS IMPACT

Amplifies XSS attack surface — malicious scripts can be injected and executed without browser-level restriction. Enables data injection attacks.

EVIDENCE (ZAP)

ZAP Alert: Content Security Policy (CSP) Header Not Set (Systemic) — CWE-693

REMEDIATION

Configure nginx to send: Content-Security-Policy: default-src 'self'. Validate with Mozilla Observatory. Use report-only mode first to test.

VF-05

Missing Anti-Clickjacking Header

CVSS MEDIUM / Medium

URL: All pages | Type: Systemic

DESCRIPTION

X-Frame-Options header is missing from all HTTP responses. Without this header, the site can be embedded in iframes on attacker-controlled websites.

BUSINESS IMPACT

Clickjacking attacks — users can be tricked into clicking buttons or links they cannot see, leading to account takeover or unauthorized actions.

EVIDENCE (ZAP)

ZAP Alert: Missing Anti-clickjacking Header (Systemic) — CWE-1021

REMEDIATION

Add to nginx config: add_header X-Frame-Options 'SAMEORIGIN'; Or use CSP frame-ancestors directive.

VF-06

X-Content-Type-Options Header Missing

CVSS MEDIUM / Low

URL: All pages | Type: Systemic

DESCRIPTION

The X-Content-Type-Options: nosniff header is absent. Browsers may attempt to determine content type by sniffing the response body rather than trusting the declared Content-Type.

BUSINESS IMPACT

MIME-type confusion attacks — an attacker serving a malicious file could have it executed as a different type (e.g., text treated as script).

EVIDENCE (ZAP)

ZAP Alert: X-Content-Type-Options Header Missing (Systemic) — CWE-693

REMEDIATION

Add to nginx config: add_header X-Content-Type-Options 'nosniff'; — simple one-line fix.

VF-07

Server & Technology Version Disclosure

CVSS MEDIUM / Low

Headers: Server: nginx/1.19.0 | X-Powered-By: PHP/5.6.40

DESCRIPTION

HTTP response headers expose exact server software versions. PHP 5.6 is end-of-life since December 2018 and has known unpatched CVEs.

BUSINESS IMPACT

Reduces attacker reconnaissance effort. PHP 5.6 EOL status means known CVEs exist with no official patches available.

EVIDENCE (ZAP)

ZAP Alerts: Server Leaks via X-Powered-By + Server headers — CWE-200

REMEDIATION

Set 'server_tokens off;' in nginx.conf. Set 'expose_php = Off' in php.ini. Upgrade PHP to a supported version (8.x).

VF-08 User Controllable HTML Element Attribute (Potential XSS)

CVSS 2.5 / Low

URL: Multiple pages | 3 instances detected by ZAP

DESCRIPTION

User-supplied data is reflected in HTML element attributes without proper encoding. While not confirmed as exploitable XSS, these represent injection points requiring investigation.

BUSINESS IMPACT

If exploited: session hijacking, defacement, phishing via site content manipulation.

EVIDENCE (ZAP)

ZAP Alert: User Controllable HTML Element Attribute (Potential XSS) x3 — CWE-80

REMEDIATION

Apply context-aware output encoding to all user-supplied data rendered in HTML. Use templating engines with auto-escaping enabled.

VF-09 In Page Banner Information Leak

CVSS 2.6 / Low

URL: Multiple pages | 3 instances detected by ZAP

DESCRIPTION

Application banners or error messages visible in page content reveal internal technology stack details including framework and version information.

BUSINESS IMPACT

Assists attacker reconnaissance — version-specific exploits can be identified and targeted more precisely.

EVIDENCE (ZAP)

ZAP Alert: In Page Banner Information Leak (3) — CWE-200

REMEDIATION

Suppress detailed error messages in production. Use generic error pages. Disable debug mode.

6. Consolidated Findings — ZAP Scan Summary

#	ID	VULNERABILITY	RISK	CVSS	ZAP ALERT
1	VF-01	SQL Injection — MySQL	HIGH	9.8	SQL Injection - MySQL (12)
2	VF-02	XSLT Injection	MEDIUM	7.2	XSLT Injection (2)
3	VF-03	Absence of Anti-CSRF Tokens	MEDIUM	6.5	Anti-CSRF Tokens (Systemic)
4	VF-04	CSP Header Not Set	MEDIUM	5.4	CSP Header Not Set (Systemic)
5	VF-05	Missing Anti-Clickjacking Header	MEDIUM	4.7	Anti-clickjacking (Systemic)
6	VF-06	X-Content-Type-Options Missing	LOW	3.7	X-Content-Type-Options (Sys.)
7	VF-07	Server Version Disclosure (nginx + PHP)	LOW	3.7	Server Leaks Version Info
8	VF-08	User Controllable HTML Attr. (Potential XSS)	LOW	3.5	HTML Element Attribute (3)
9	VF-09	In Page Banner Information Leak	LOW	2.6	In Page Banner Leak (3)

7. Remediation Priority Plan

#	ID	Action	Effort	Priority
1	VF-01	Use parameterized queries — fix SQL Injection	1-2 days / Code refactor	HIGH
2	VF-03	Implement CSRF tokens on all forms	1 day / Framework feature	HIGH
3	VF-02	Sanitize XSLT input / restrict XSLT functions	1 day / Code review	MEDIUM
4	VF-04	Add Content-Security-Policy header in nginx	30 min / Config change	MEDIUM
5	VF-05	Add X-Frame-Options: SAMEORIGIN header	10 min / Config change	MEDIUM
6	VF-06	Add X-Content-Type-Options: nosniff header	10 min / Config change	LOW
7	VF-07	Set server_tokens off + expose_php=Off + upgrade PHP	1 hour / Config + upgrade	LOW
8	VF-08	Apply output encoding to HTML attributes	Few hours / Code review	LOW
9	VF-09	Disable debug/error messages in production	30 min / Config change	LOW

8. Conclusion

The active scan with OWASP ZAP 2.17.0 on testphp.vulnweb.com confirmed 13 real vulnerabilities across 4 severity levels. The most critical finding — SQL Injection — was proven with live error-based evidence from the MySQL server. Combined with XSLT Injection, missing CSRF protection, and absent security headers, this application represents a highly insecure environment that should not be exposed to real users.

All findings have clear, well-documented remediation paths. Quick wins (security headers) can be implemented in under 30 minutes. The SQL Injection fix requires code refactoring but is the highest priority action.

KEY TAKEAWAY FROM REAL SCAN

SQL Injection (VF-01) is 100% confirmed with live proof. Fix it first.

Security Headers (VF-04/05/06) can be fixed in under 1 hour — do them immediately!

9. Disclaimer

This assessment was conducted on testphp.vulnweb.com — a deliberately vulnerable application by Acunetix provided specifically for educational and authorized security testing. No unauthorized systems were accessed. Prepared for Future Interns Cyber Security Internship Task 1 (FUTURE_CS_01) — February 2026.

PREPARED BY

BATHO PILIOUZIWE ABRAHAM

Cyber Security Intern — Future Interns | IPNET Institute of Technology
CIN: FIT/FEB26/CS6250 | GitHub: FUTURE_CS_01

February 2026

Scan: OWASP ZAP 2.17.0
19 findings confirmed