



Cyber Security

Vulnerability Assessment Report

Security Assessment of a Web Application



TARGET WEBSITE

testphp.vulnweb.com



PREPARED BY

Thanvesh Reddy



PROJECT TITLE

Vulnerability Assessment on Web Application



DEGREE

B.Tech CSE – 2nd Year



DOMAIN

Cyber Security



cyber security



Confidential — Educational Use



Candidate Details

Two-column details master template

LABEL	VALUE
 NAME	Thanvesh Reddy
 DEGREE	B.Tech CSE – 2nd Year
 DOMAIN	Cyber Security
 PROJECT TITLE	Vulnerability Assessment on Web Application
 TOOLS USED	Nmap, OWASP ZAP, Browser DevTools, Canva

Structured two-column layout for specifications

Tip: Keep labels concise and use consistent capitalization for clarity.



Project Overview

Narrative overview master template

This project performs a Vulnerability Assessment (VA) of the publicly available test website testphp.vulnweb.com. The focus is to identify common web application weaknesses, assess their risk levels, and provide actionable remediation guidance.

The assessment combines passive scanning and manual analysis to avoid service disruption and data impact. Findings are documented clearly with risk classification and practical recommendations for mitigation.



TESTING TYPE

Black-box; passive and manual analysis



KEY TOOLS

Nmap, OWASP ZAP (passive),
Browser DevTools

Scope emphasizes discovery and documentation, not exploitation.



Focus

Assess the web application's exposed surface, including endpoints, HTTP behavior, and server responses, to uncover common misconfigurations and weaknesses.



Objectives

- Identify common web vulnerabilities
- Classify risks as Low or Medium
- Recommend practical mitigations
- Produce professional documentation



Ethical Approach

Conducted on an authorized educational target, using safe, non-intrusive methods. No exploitation, data modification, authentication bypass, or DoS activity was performed.



Objectives

Bulleted list master template

Key goals for the vulnerability assessment project:



Identify common web application vulnerabilities



Perform passive and manual security analysis



Classify risks as **Low** or **Medium**



Understand real-world web security issues



Prepare professional security documentation

Concise, scannable objectives for stakeholder alignment



Target Information

Two-column details template

LABEL

VALUE



TARGET WEBSITE

testphp.vulnweb.com



APPLICATION TYPE

Public test web application



AUTHORIZATION

The selected website is a publicly available test application provided specifically for security testing and educational purposes.



TESTING TYPE

Black-box testing



Authorized, non-intrusive assessment scope

Tip: Confirm authorization and scope before any testing.



Scope of Assessment

Clear separation of what is included vs excluded



In Scope

Activities permitted and assessed

- ✓ Web application endpoints
 - ✓ HTTP response headers
 - ✓ Open ports and services
 - ✓ Passive vulnerability detection
-



Out of Scope

Prohibited actions not performed

- ✗ Denial of Service (DoS) attacks
 - ✗ Data exploitation
 - ✗ Authentication bypass
 - ✗ Source code analysis
-

Defined boundaries ensure ethical, non-intrusive testing



Tools Used

Grid/list master template for security toolsets

Utilities applied for scanning, analysis, and presentation.



Nmap

Port and service discovery to identify exposed network services on the host.

Port/Service Scan



OWASP ZAP (Passive Scan)

Safe, non-intrusive analysis to detect common web issues from observed traffic only.

Passive App Scan



Browser Developer Tools

Manual inspection of HTTP headers, network requests, and DOM for security signals.

Manual Inspection



Canva

Design and presentation of findings in a clear, professional report format.

Report Design

Concise grid highlighting purpose and usage



Methodology

Process timeline master template — 6 steps

A structured, non-intrusive workflow for assessing the target application safely and consistently.

1 Website Selection



Choose an authorized educational target (testphp.vulnweb.com).

2 Nmap Scanning



Identify open ports and services to understand exposure.

3 OWASP ZAP (Passive)



Passively scan HTTP traffic for common web vulnerabilities.

4 Header Inspection

Review HTTP response headers with browser devtools.

5 Risk Classification



Categorize findings by likelihood and impact (Low/Medium).

6 Documentation



Compile a clear report with findings and remediation steps.

Sequential, safe, and repeatable assessment workflow



Nmap Scan Results

Findings summary master template



Open Port Identified

HTTP (unencrypted)

80 Port 80

● OPEN

🌐 HTTP



Web Server Detected

📦 nginx

1.19.0

</> Server banner exposed



Observation

The application is accessible over HTTP and reveals its web server version, which may aid targeted attacks.



Scan Context

TARGET

🌐 testphp.vulnweb.com

COMMAND

```
nmap -sV testphp.vulnweb.com
```

KEY RESULT

```
80/tcp open http nginx 1.19.0
```

Use this layout for concise technical findings with clear evidence and impact.



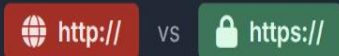
Vulnerability 1: HTTP Only Site (No HTTPS)

Vulnerability Risk Card — master template

 Risk: Medium



Unencrypted HTTP Traffic



The application is accessible over HTTP, meaning data is transmitted in clear text without TLS encryption.



DESCRIPTION

The web application does not use HTTPS to encrypt communication between clients and the server.



IMPACT

Sensitive information (such as session tokens or form data) may be intercepted or modified in transit by network attackers.



RECOMMENDATION

Implement HTTPS using SSL/TLS certificates (e.g., from a trusted CA), enable HSTS, and redirect all HTTP traffic to HTTPS.

Clear structure: Description • Impact • Recommendation

 Medium risk indicates meaningful exposure without immediate critical loss



Vulnerability 2: Missing Security Headers

Vulnerability Risk Card — master template

✓ Risk: Low

H Missing Protective Headers

✗ Content-Security-Policy

✗ X-Frame-Options

⚠ X-Content-Type-Options

Several recommended HTTP response headers are absent, reducing browser-side defenses.



DESCRIPTION

The application responses lack important security headers (e.g., Content-Security-Policy, X-Frame-Options, X-Content-Type-Options).



IMPACT

Increases exposure to clickjacking and client-side attacks (e.g., framing-based UI redress and script injection).



RECOMMENDATION

Configure headers such as Content-Security-Policy, X-Frame-Options (SAMEORIGIN or DENY), X-Content-Type-Options (nosniff), Referrer-Policy, and Permissions-Policy. Verify via security scanners and browser DevTools.

Clear structure: Description • Impact • Recommendation

● Low risk: limited exposure; address during security hardening



Vulnerability 3: Server Information Disclosure

Vulnerability Risk Card — consistent template

✓ Risk: Low



Exposed Version Details

</> HTTP Response Headers

Server: nginx/1.19.0

X-Powered-By: PHP

Note: Revealing specific versions aids fingerprinting.



DESCRIPTION

The application discloses server/framework details (e.g., web server and scripting platform versions) via HTTP headers or error messages.



IMPACT

Attackers can correlate exposed versions with known CVEs to craft targeted attacks and improve reconnaissance accuracy.



RECOMMENDATION

Suppress version banners and headers: disable or minimize server tokens (e.g., nginx server_tokens off; Apache ServerSignature Off/ServerTokens Prod), remove X-Powered-By, standardize custom error pages, and mask framework/version identifiers.


Clear structure: Description • Impact • Recommendation

● Low risk: reduces attacker reconnaissance effectiveness when remediated



Vulnerability 4: Potential Cross-Site Scripting (XSS)

Vulnerability Risk Card — master style

 Risk: Medium

User-Controlled Input Reflected



Untrusted Input



Sanitized Output

`<script>alert('XSS')</script>` should be neutralized as text, not executed.



DESCRIPTION

User-controllable fields (e.g., query parameters or form inputs) appear to be reflected or used in the DOM without proper encoding. This may enable script injection under certain conditions.



IMPACT


Malicious scripts may execute in a user's browser, leading to session theft, credential harvesting, defacement, or unauthorized actions on behalf of the user.



RECOMMENDATION

Validate and sanitize all user inputs server-side; apply context-appropriate output encoding (HTML, attribute, JS). Implement a strict Content Security Policy (CSP) and avoid unsafe inline JavaScript.

Clear structure: Description • Impact • Recommendation

 Medium risk indicates meaningful exposure without immediate critical loss



Risk Summary

Counts by risk level and overall rating

Overall Risk: Medium



Risk Level Counts

RISK LEVEL	COUNT
● Medium	2
● Low	3

Summary reflects identified Medium (2) and Low (3) vulnerabilities from the assessment.



Distribution



● Medium ● Low



Overall Risk Rating

Medium

Prioritize HTTPS enforcement and XSS mitigation to lower overall risk.

Concise, color-coded summary for executive visibility



Recommendations

Bulleted list master template

Recommended actions to strengthen the web application's security posture:



Enforce HTTPS across the application



Add security headers (e.g., CSP, X-Frame-Options)



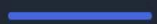
Hide server and framework version details



Validate and sanitize user inputs



Conduct regular vulnerability assessments



Prioritized, actionable hardening steps



Learning Outcomes

Bulleted list master template

Key takeaways from the vulnerability assessment:



Practical tool experience

Hands-on with Nmap, OWASP ZAP, and DevTools



Understanding web security risks

Recognized common weaknesses and their impacts



Risk analysis skills

Classified findings and prioritized remediation



Ethical cybersecurity practices

Followed authorized, non-intrusive assessment methods

Concise, scannable outcomes for stakeholders



Conclusion

Narrative overview template

The vulnerability assessment identified multiple security weaknesses in the web application, including the use of HTTP only, missing security headers, server information disclosure, and potential XSS exposure.

Addressing these issues will enhance the application's security posture by improving confidentiality, integrity, and resilience. The overall assessed risk is **Medium**, with clear remediation paths available.

Remediation and verification are recommended to reduce exposure and validate improvements.



Overall Risk

Medium

Moderate likelihood and impact; prioritized fixes will measurably reduce risk.



Top Priorities

- Enforce HTTPS across the application
- Add recommended security headers
- Validate/sanitize inputs and apply CSP



Next Steps

Implement fixes, conduct a follow-up assessment, and monitor continuously to sustain improvements.

Clear narrative layout for overviews and conclusions



Disclaimer

Narrative overview master template

This assessment was conducted strictly for educational and internship purposes on an authorized public test website provided for security learning and practice.

Activities were limited to safe, non-intrusive techniques (passive scanning and manual inspection). No exploitation, unauthorized testing, data modification, authentication bypass, or denial-of-service actions were performed.

AUTHORIZED TARGET

Public test web application designated for training.



EDUCATIONAL USE

Performed solely for learning and documentation.

Scope emphasizes discovery and documentation, not exploitation.



Non-Intrusive

No active exploitation, fuzzing beyond safe limits, or service disruption.



Ethical Conduct

Actions aligned with responsible security testing practices.



No Unauthorized Testing

No data alteration, credential access, or authentication bypass attempted.