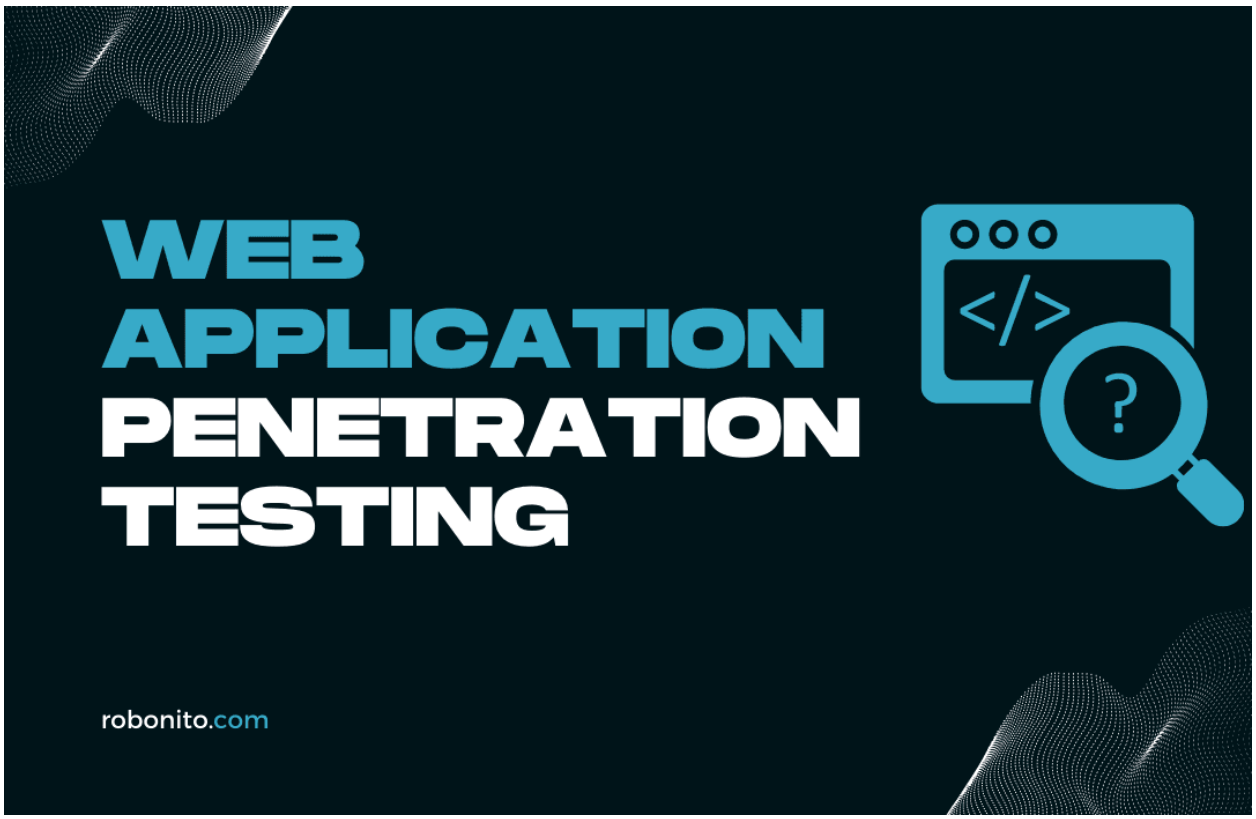
Penetration Testing Report



Tehreem Amna

Internship Task 1
Black Byt3

Table of Contents

List of Figures

Finding Severity Rating

Vulnerability	Severity	CVSS Score
WHOIS Information Disclosure	Low	3.7
Open Ports & Service Detection	Medium	5.3
Absence of Anti-CSRF Tokens	Medium	6.5
Missing Security Headers	Medium	6.1
Cross-Site Scripting (Guestbook)	High	8.2
DOM Manipulation (Stored XSS Variant)	High	8.2
Open Redirection	Medium	6.4
Information Disclosure via Headers	Medium	5.0
Stored XSS in Guestbook	High	6.1
Stored Cross-Site Scripting	High	5.4

Executive Summary

This penetration test targeted the testphp.vulnweb.com demo application. Reconnaissance, scanning, and exploitation were performed using Nmap, OWASP ZAP, Acunetix, and manual payload injection.

Key findings include:

- WHOIS lookup revealed the server's IP address (44.228.249.3).
- Nmap identified multiple open ports and service banners.
- Web application is missing critical HTTP security headers.
- XSS vulnerabilities were confirmed in the guestbook section.
- DOM-based XSS payloads allowed JavaScript execution.
- Open redirection payloads allowed attacker-controlled redirects.

The overall security posture is High Risk, primarily due to XSS vulnerabilities, which can be chained with other issues to perform session hijacking, phishing, or data theft.

Methodology

Testing was carried out in the following phases:

- **Reconnaissance:** WHOIS & DNS lookup.
- **Network Scanning:** Nmap scans for open ports, service versions, and aggressive OS fingerprinting.
- **Automated Scanning:** OWASP ZAP & Acunetix scans to detect missing headers, CSRF issues, and injection points.
- **Exploitation:** Manual payloads for **XSS, DOM manipulation, and open redirection.**
- **Evidence Collection:** Screenshots were captured for each confirmed vulnerability.

Findings

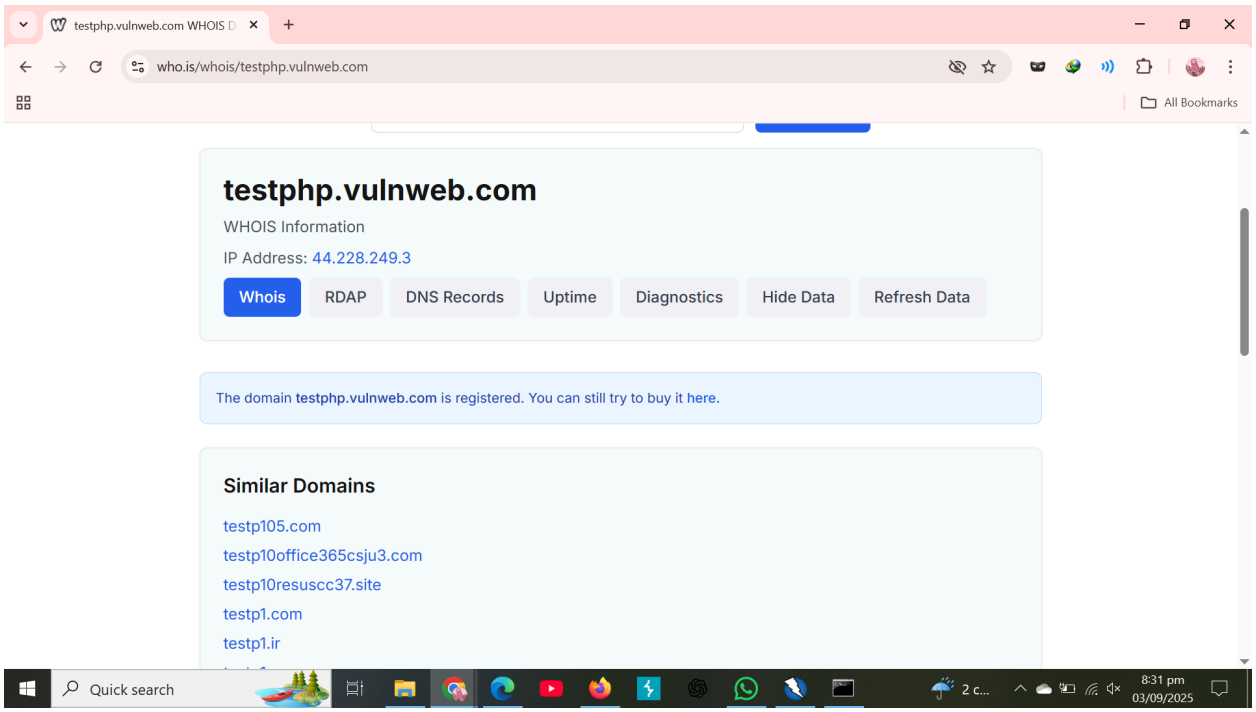
1. **WHOIS Information Disclosure**

Description:

A WHOIS lookup of `testphp.vulnweb.com` exposed the server's IP address `44.228.249.3` and domain details. This information can assist attackers in reconnaissance and infrastructure mapping.

Risk Level: Low

Proof of Concept (PoC):



Recommendation:

Enable WHOIS privacy protection or use a proxy service to hide sensitive domain registration details.

2. **Open Ports & Service Detection**

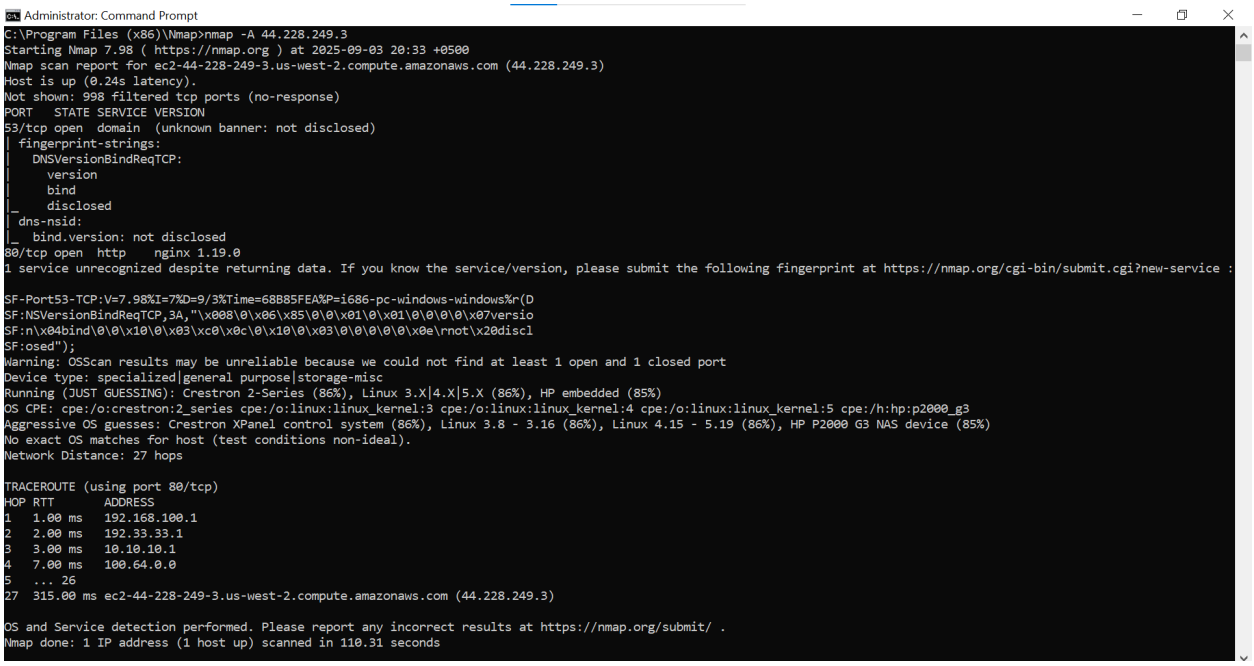
Description:

Nmap scans revealed multiple open ports (e.g., 80, 53) and service banners. Exposed services increase the attack surface and can be targeted for exploitation.

Risk Level: Medium

Proof of Concept (PoC):

- **Aggressive Scan**



- **Open Ports**

```
C:\Windows\system32>cd "C:\Program Files (x86)\Nmap"

C:\Program Files (x86)\Nmap>nmap 44.228.249.3
Starting Nmap 7.98 ( https://nmap.org ) at 2025-09-03 20:32 +0500
Nmap scan report for ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)
Host is up (0.41s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE
53/tcp open  domain
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 34.14 seconds
```

- **Service Version Detection**

- **Specific Port Scan**

```
C:\Program Files (x86)\Nmap>nmap -p 21,22,80,443 44.228.249.3
Starting Nmap 7.98 ( https://nmap.org ) at 2025-09-03 20:36 +0500
Nmap scan report for ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)
Host is up (0.30s latency).

PORT    STATE    SERVICE
21/tcp  filtered ftp
22/tcp  filtered ssh
80/tcp  open     http
443/tcp filtered https

Nmap done: 1 IP address (1 host up) scanned in 4.24 seconds
```

Recommendation:

Close unused ports, update running services, and restrict access through a properly configured firewall.

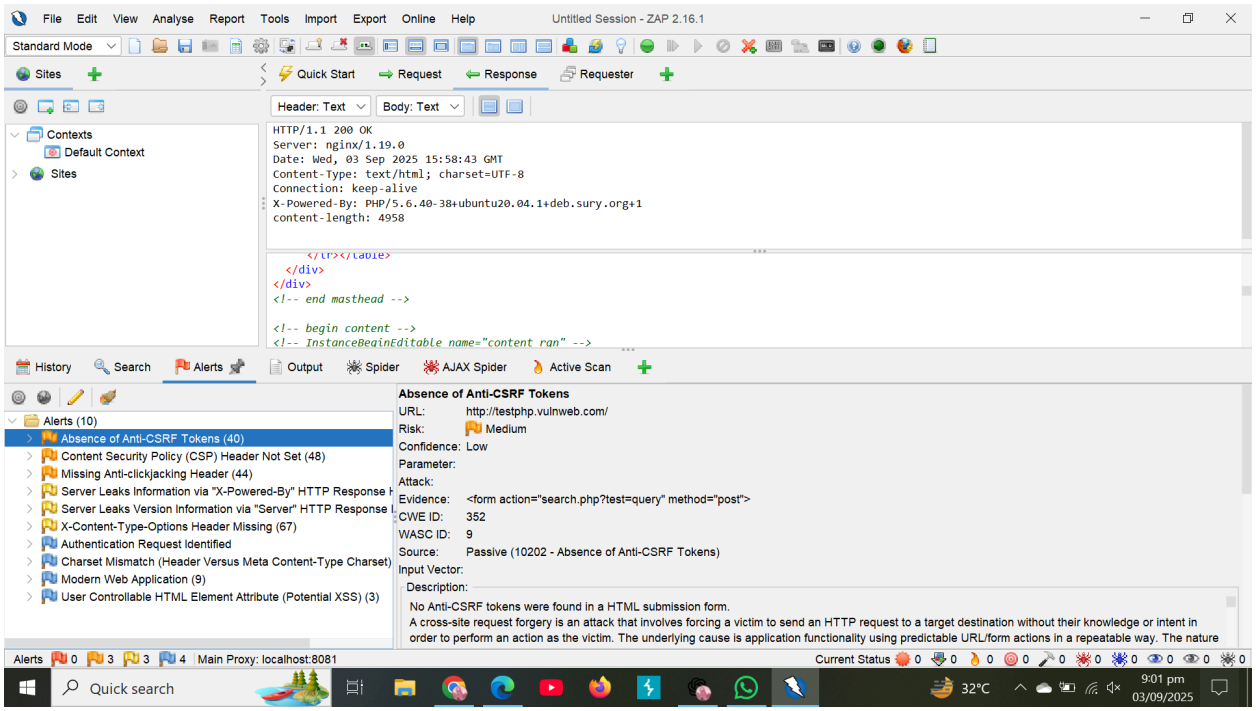
3. **Absence of Anti-CSRF Tokens**

Description:

Forms on the application lack CSRF protection. Attackers could craft malicious links or forms that trick authenticated users into performing unintended actions.

Risk Level: Medium

Proof of Concept (PoC):



Recommendation:

Implement unique anti-CSRF tokens in all forms and validate them on the server side to prevent unauthorized requests.

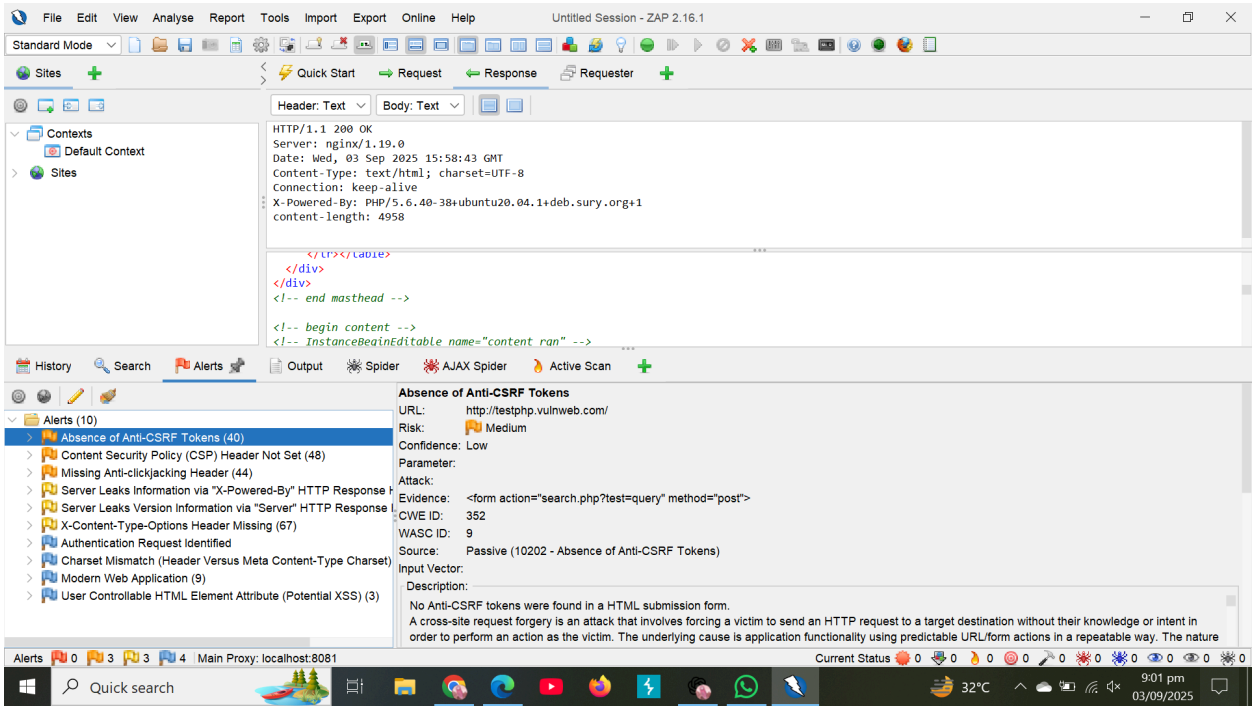
4. **Missing Security Headers**

Description:

The web application does not implement critical HTTP security headers (CSP, HSTS, X-Frame-Options, X-Content-Type-Options). Without these, the app is vulnerable to XSS, clickjacking, and other client-side attacks.

Risk Level: Medium

Proof of Concept (PoC):



Recommendation:

Add essential headers like CSP, HSTS, X-Frame-Options, and X-Content-Type-Options to enhance client-side security.

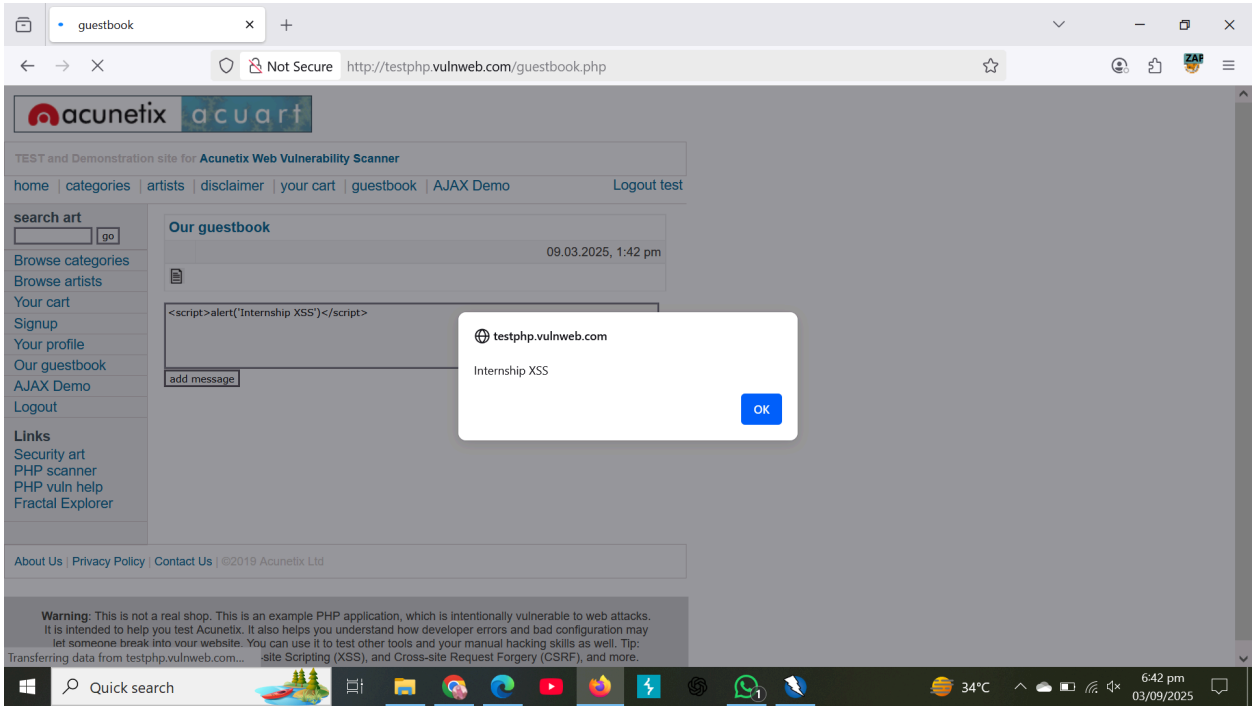
5. **Cross-Site Scripting**

Description:

The Guestbook feature is vulnerable to stored XSS. Injecting the payload `<script>alert('Internship XSS')</script>` resulted in successful JavaScript execution.

Risk Level: High

Proof of Concept (PoC):



Recommendation:

Validate and sanitize all user inputs, encode outputs properly, and implement a strong Content Security Policy (CSP).

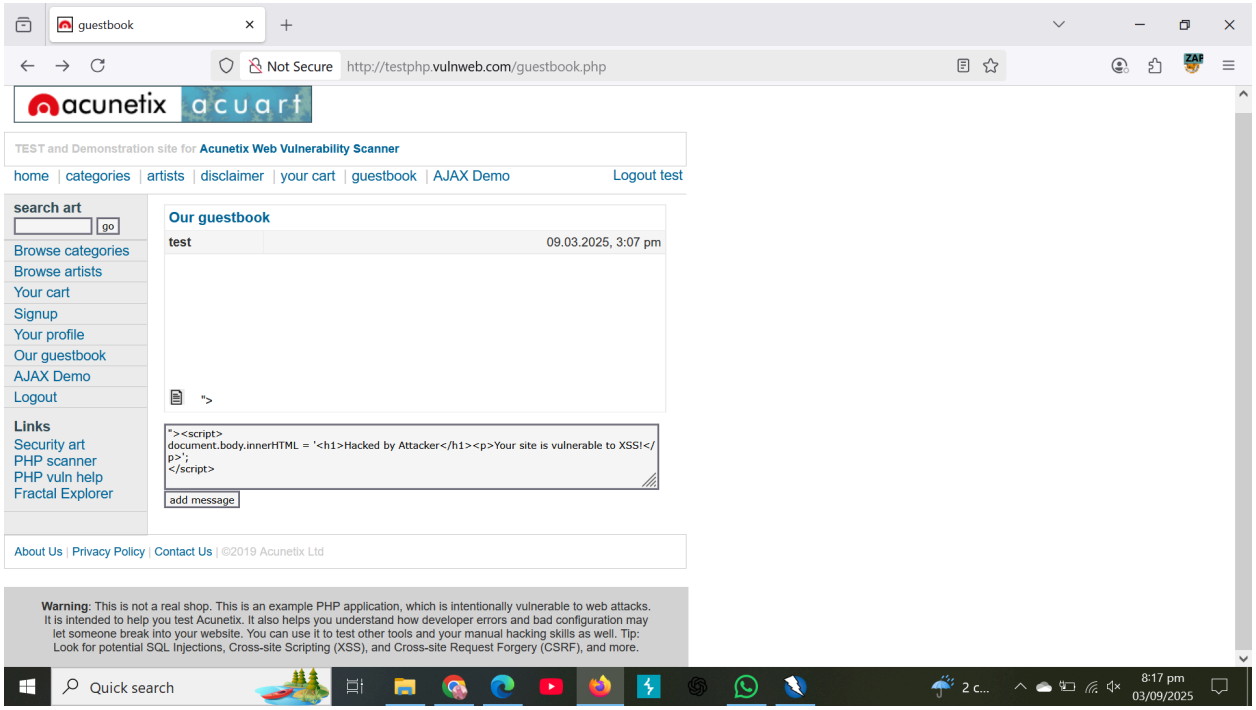
6. **DOM Manipulation Payload**

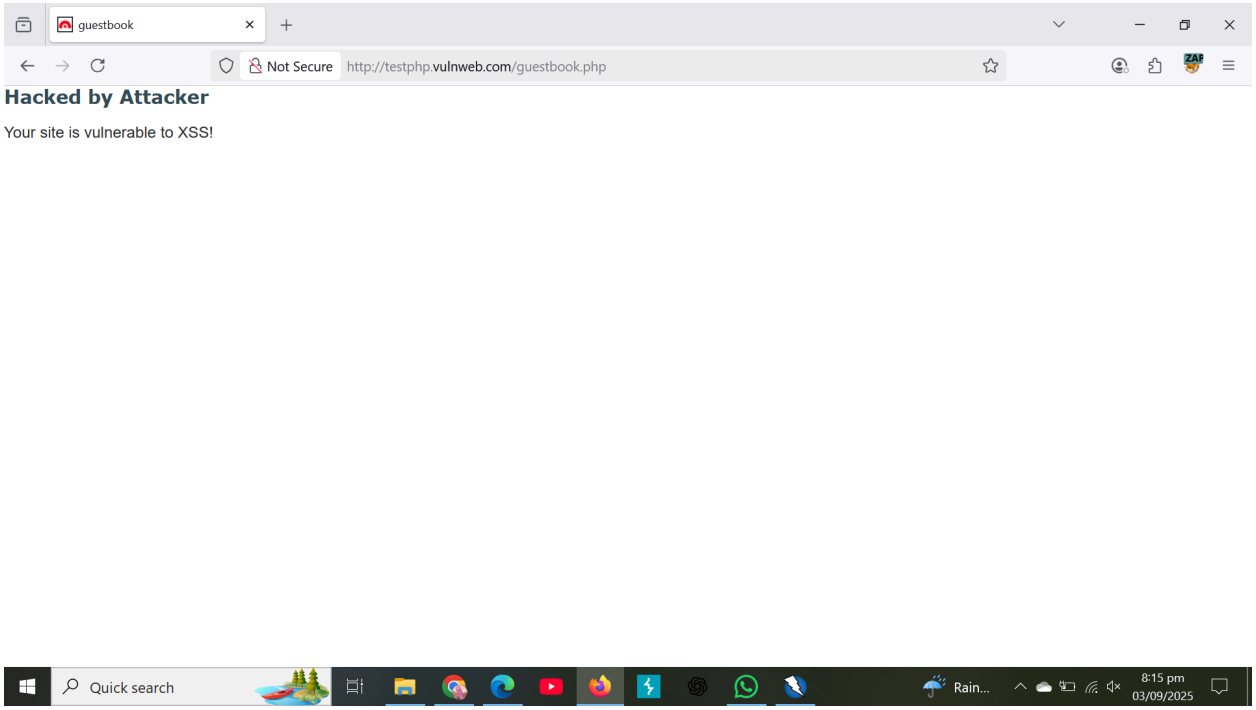
Description:

DOM-based injection allowed execution of arbitrary JavaScript code in the browser, confirming a DOM XSS vulnerability.

Risk Level: High

Proof of Concept (PoC):





Recommendation:

Avoid unsafe DOM methods like `innerHTML`, sanitize client-side inputs, and enforce a strict Content Security Policy (CSP).

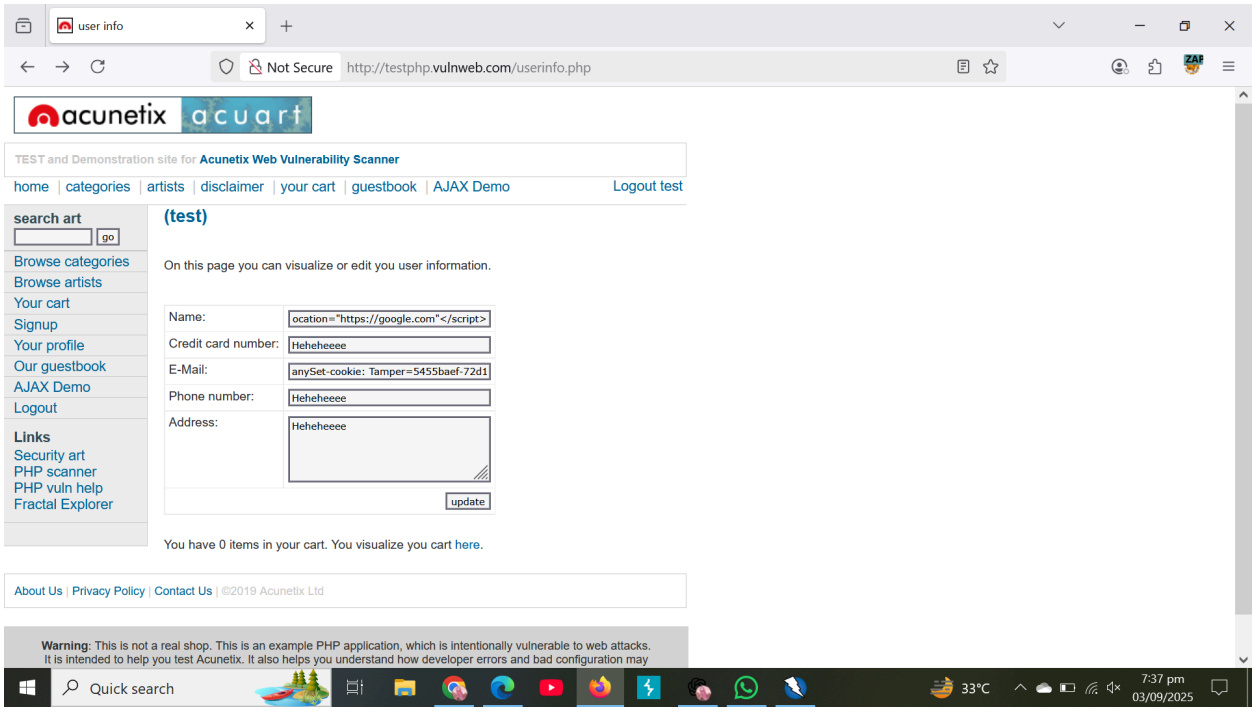
7. **Open Redirection Vulnerability**

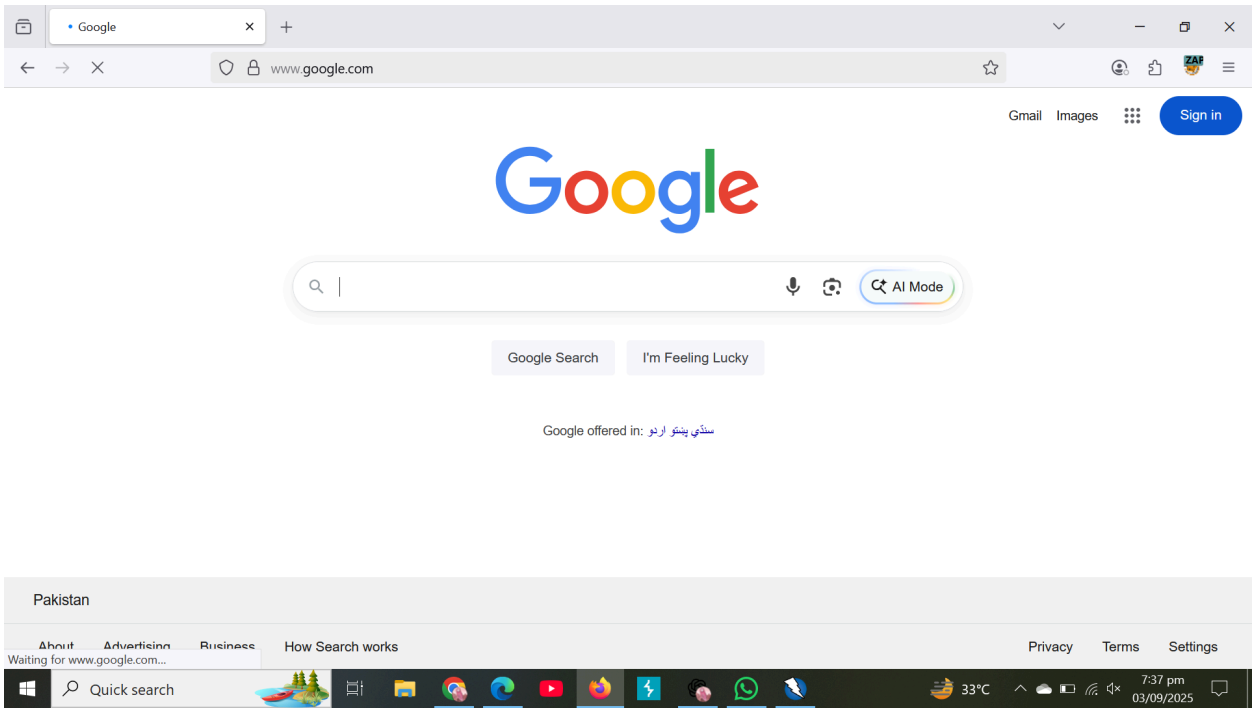
Description:

The application accepted crafted URLs that redirected users to attacker-controlled domains, enabling phishing or malware delivery.

Risk Level: Medium

Proof of Concept (PoC):





Recommendation:

Validate and restrict redirect URLs using an allow-list, and block user-supplied values from directly controlling redirects.

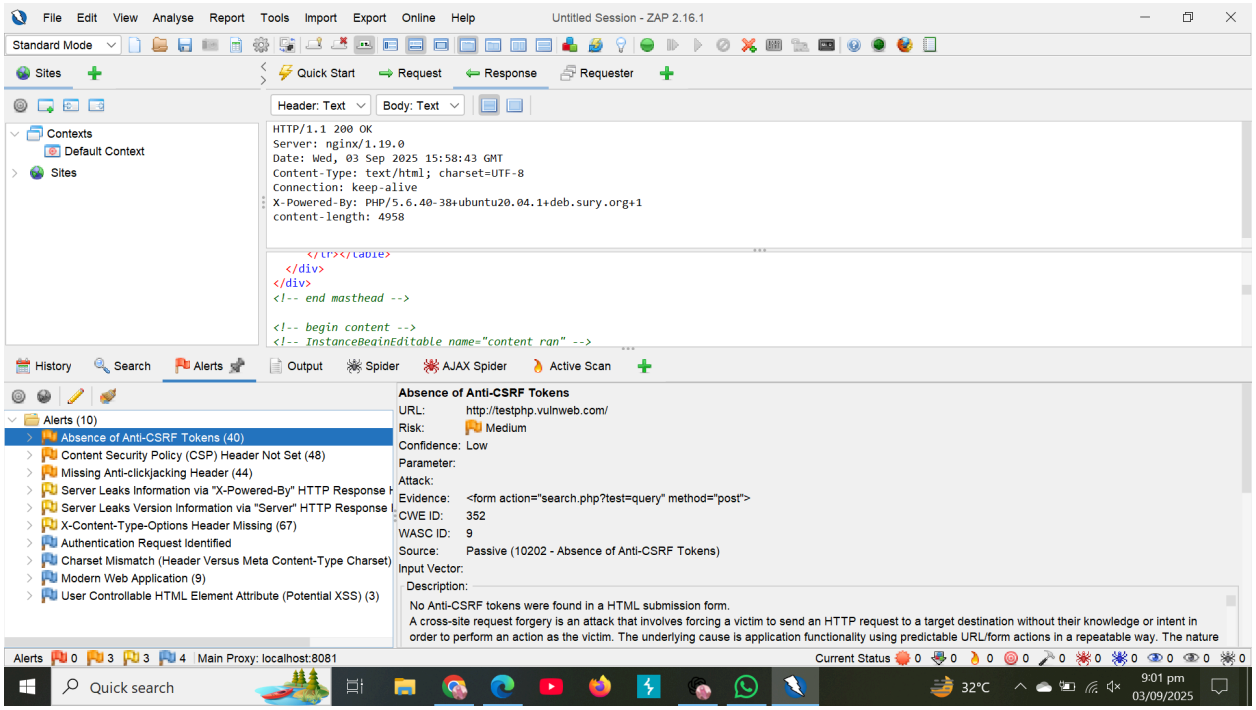
8. **Information Disclosure via Headers**

Description:

Server response headers revealed sensitive details about Apache and PHP versions. Attackers can use this information to launch version-specific exploits.

Risk Level: Medium

Proof of Concept (PoC):



Recommendation:

Disable or obfuscate server version details in HTTP headers to prevent attackers from gathering sensitive system information.

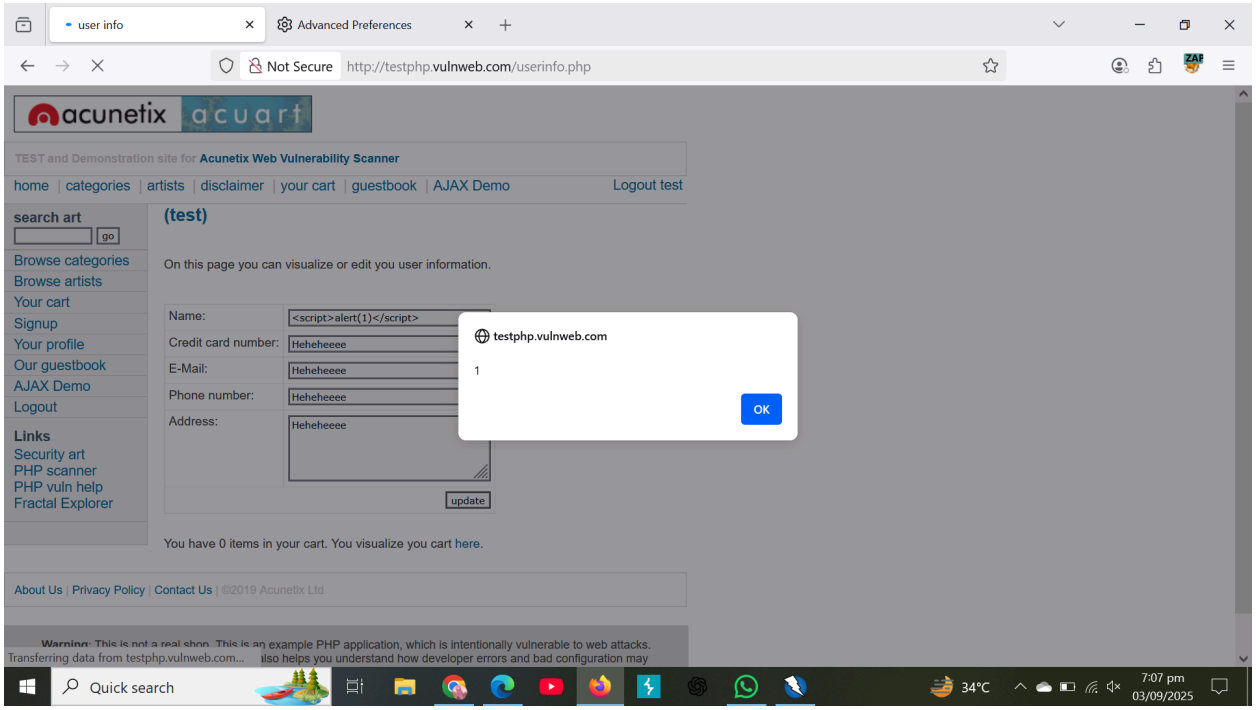
9. **Stored Cross-Site Scripting**

Description

The `userinfo.php` page is vulnerable to Stored XSS. By injecting the payload `<script>alert(1)</script>` into the **Name** field, the malicious script was stored and executed when the page was loaded.

Risk Level: High

Proof of Concept (PoC):



Recommendation:

Sanitize and encode all stored user inputs before displaying them, and apply a strict Content Security Policy (CSP).

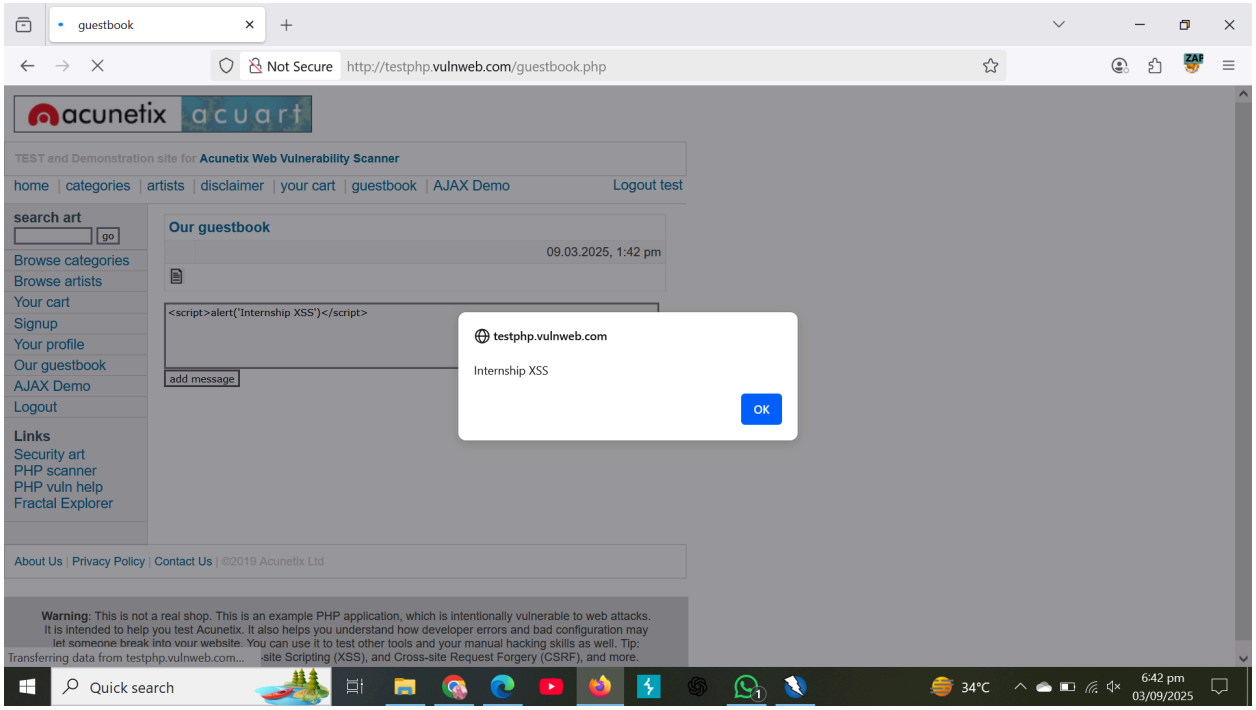
10. **Stored XSS in Guestbook (`guestbook.php`)**

Description:

The guestbook page fails to properly sanitize user-supplied input before rendering it back to other users. An attacker can inject malicious JavaScript (e.g., `<script>alert('Internship XSS')</script>`), which executes in the browser of any visitor who views the guestbook. This could lead to cookie theft, session hijacking, defacement, or phishing attacks.

Risk Level: High

Proof of Concept (PoC):



Recommendation:

Properly validate and encode guestbook inputs before saving or displaying, and enforce a Content Security Policy (CSP) to block script execution.

CONCLUSION

The overall security posture of the application is weak, as multiple vulnerabilities such as Stored XSS, Reflected XSS, DOM-based XSS, Open Redirects, Missing Security Headers, and Open Ports were identified. These issues demonstrate a lack of proper input validation, insufficient client-side protections, and inadequate server hardening, which collectively increase the application's risk exposure.

The most critical vulnerabilities are the Stored XSS flaws in both the guestbook and name field, as they allow persistent malicious script execution that can lead to session hijacking, data theft, and phishing attacks. Open ports and service disclosure further expand the attack surface, while missing headers and absence of CSRF tokens leave users vulnerable to common web attacks.