

Vulnerability Assessment and Penetration Testing Report

Client / Website Tested
testphp.vulnweb.com (<i>Public demo website</i>)
Assessment Type
Vulnerability Assessment and penetration Testing
Intern Name
Rushikesh Borse Cyber Security Intern – Future Interns (2026)
Assessment Date
February 2026

• The assessment was carried out using: Tools
• Browser Developer Tools
• Nmap (basic exposure analysis)
• OWASP ZAP (Passive Scan only)

1. No authentication bypass, brute-force attempts, or denial-of-service activities were performed.

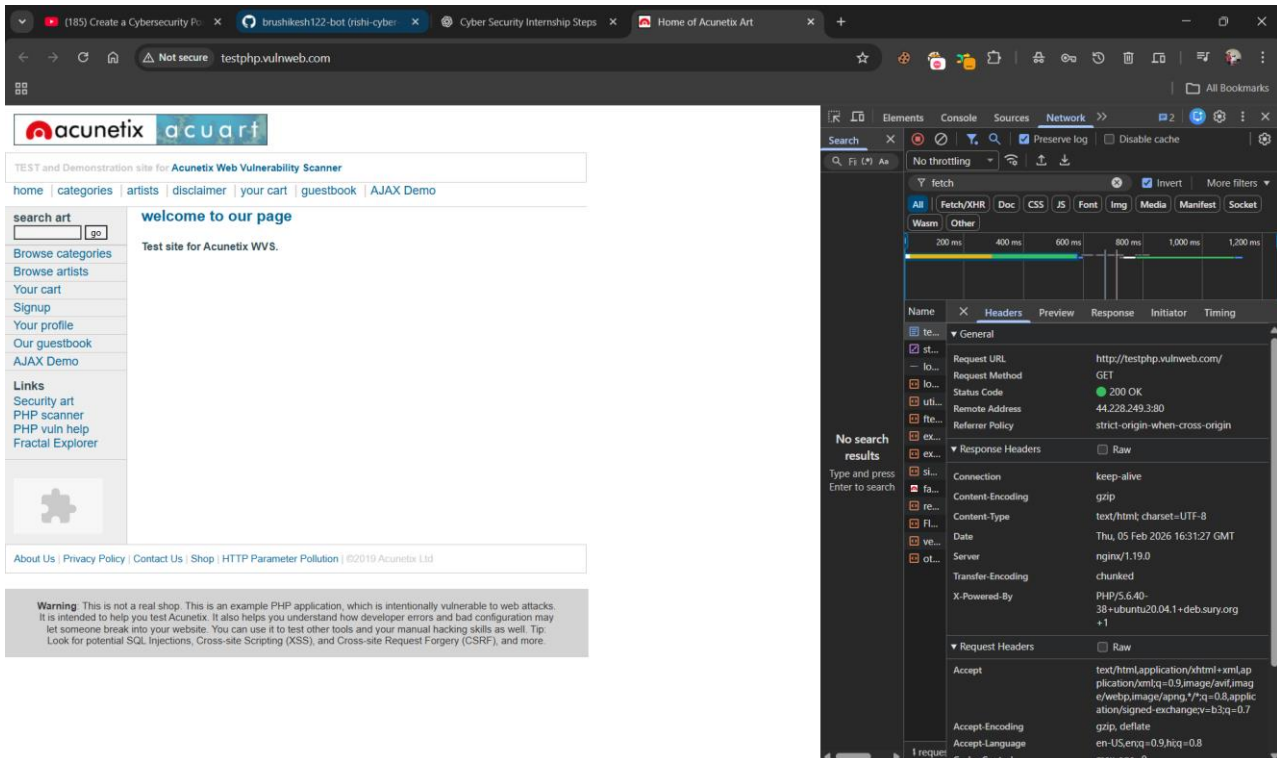
1. Executive Summary
This Vulnerability Assessment was conducted to identify common security weaknesses present on a publicly accessible website using non-intrusive and ethical methods.
The objective of this assessment is not to exploit or attack the website, but to provide clarity to a business owner about potential security risks and recommend practical remediation steps.

Finding 1: Missing Security Headers

Tool Used: Browser DevTools

What is the issue?

The website is accessible over HTTP, which means data is transmitted without encryption.



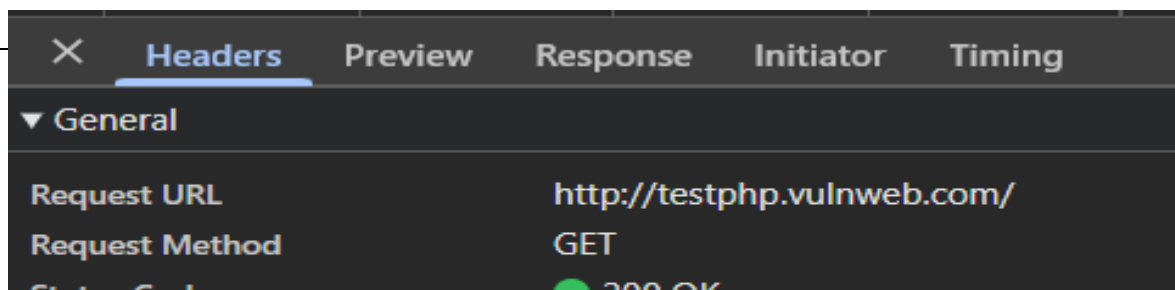
Why does it matter?

Without encryption, sensitive information such as login details or user data can be intercepted by attackers. This reduces user trust and may lead to data exposure.

Risk Level :- High

Suggested Remediation (Business-Friendly Fix)

- Enable HTTPS using an SSL/TLS certificate
- Redirect all HTTP traffic to HTTPS
- Ensure secure communication for all users



Observed Missing Headers:
<ul style="list-style-type: none">• Content-Security-Policy (CSP)
<ul style="list-style-type: none">• X-Frame-Options
<ul style="list-style-type: none">• X-Content-Type-Options
Why This Matters:
Missing headers increase the risk of:
<ul style="list-style-type: none">• Clickjacking
<ul style="list-style-type: none">• Cross-Site Scripting (XSS)
<ul style="list-style-type: none">• Content injection attacks
Risk Level: Medium
Recommended Remediation:
<ul style="list-style-type: none">• Implement standard security headers at the server level

- Content-Security-Policy (CSP)

- X-Frame-Options

- X-Content-Type-Options

Why This Matters:

Missing headers increase the risk of:

- Clickjacking
- Cross-Site Scripting (XSS)
- Content injection attacks

Risk Level: Medium

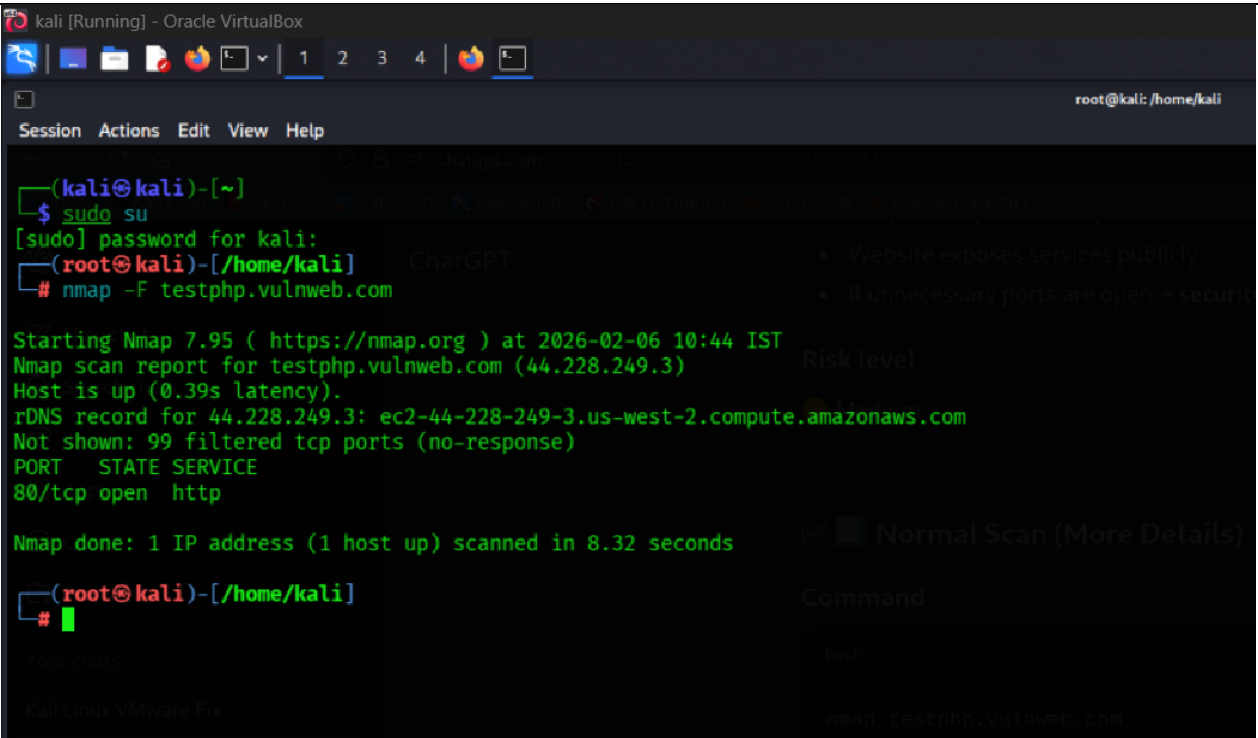
Recommended Remediation:

- Implement standard security headers at the server level

Finding 2: Open HTTP Port (Port 80)

Tool Used: Nmap

Command Example: nmap -F testphp.vulnweb.com



```
kali [Running] - Oracle VirtualBox
root@kali: /home/kali

(kali@kali)-[~]
$ sudo su
[sudo] password for kali:
(root@kali)-[/home/kali]
# nmap -F testphp.vulnweb.com

Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-06 10:44 IST
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.39s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 99 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 8.32 seconds

(root@kali)-[/home/kali]
#
```

What was checked?

- Scanned the most commonly used ports to quickly identify exposed services.

Identified Issue :- Common web service ports are open and reachable

Why does it matter? :- Open ports increase the attack surface if not properly secured

Risk Level :- Medium

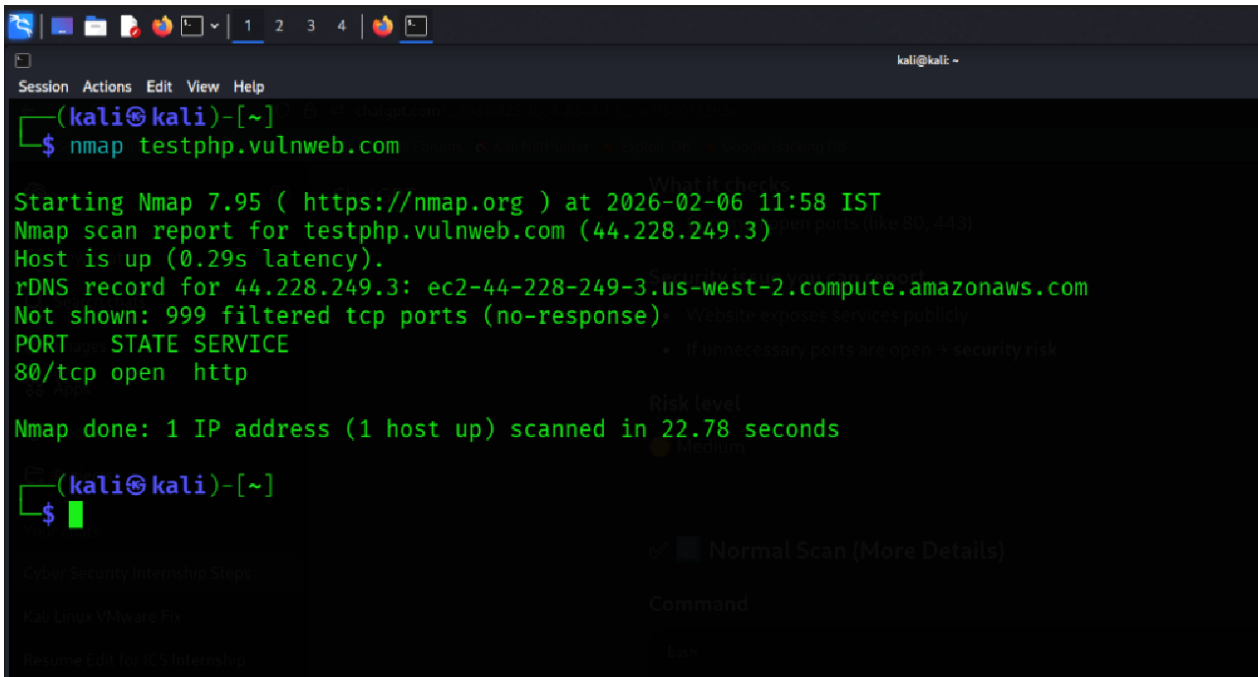
Clear Remediation Steps

- Close unused ports
- Restrict access using firewall rules

Nmap 2: Default Nmap Scan

Tool Used: Nmap

Command Example: nmap testphp.vulnweb.com



```
(kali㉿kali)-[~]
$ nmap testphp.vulnweb.com

Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-06 11:58 IST
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.29s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 22.78 seconds

(kali㉿kali)-[~]
$
```

What was checked?

- A standard scan to identify **open ports** and **basic service exposure** on the target server.

- **Why does it matter?**
- **Port 80 (HTTP) is open and publicly accessible**
Web service is exposed without enforced encryption

Identified Issue

- **Port 80 (HTTP) is open and publicly accessible**
- **Web service is exposed without enforced encryption**

Risk Level :- Medium

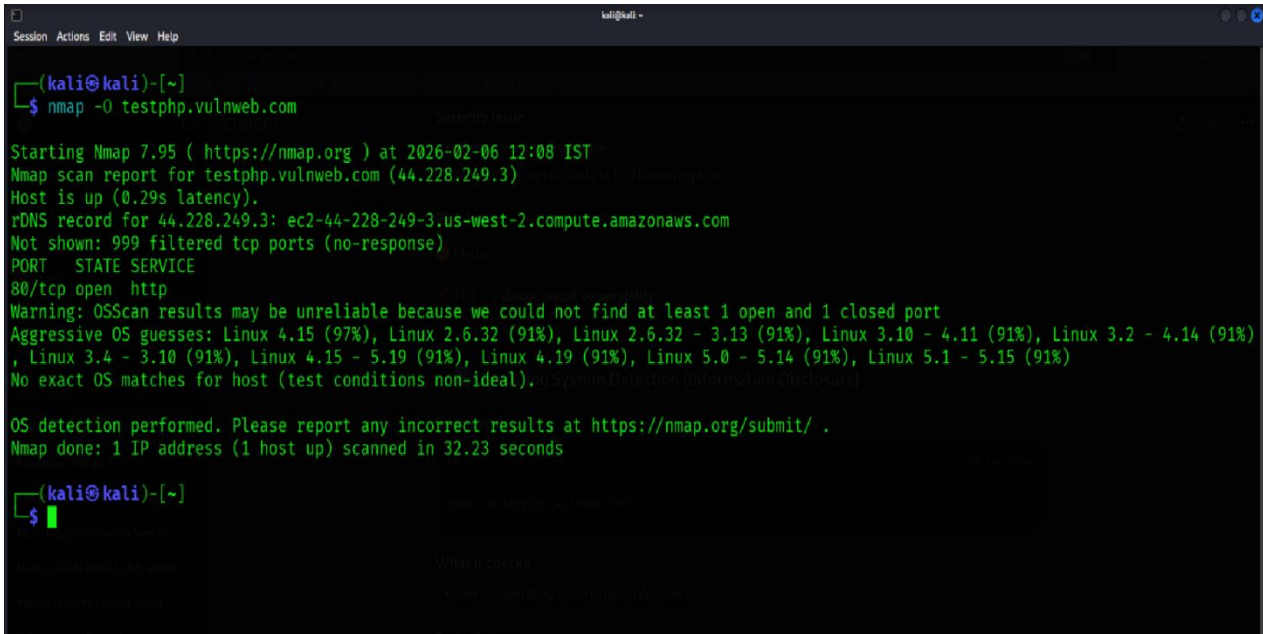
Clear Remediation Steps

- Redirect all HTTP traffic to HTTPS
- Allow only required ports to remain open

Nmap 2 : Operating System Detection

Tool Used: Nmap

Command Example: nmap -O testphp.vulnweb.com



```
(kali@kali)-[~]
$ nmap -O testphp.vulnweb.com

Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-06 12:08 IST
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.29s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 4.15 (97%), Linux 2.6.32 (91%), Linux 2.6.32 - 3.13 (91%), Linux 3.10 - 4.11 (91%), Linux 3.2 - 4.14 (91%), Linux 3.4 - 3.10 (91%), Linux 4.15 - 5.19 (91%), Linux 4.19 (91%), Linux 5.0 - 5.14 (91%), Linux 5.1 - 5.15 (91%)
No exact OS matches for host (test conditions non-ideal).
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 32.23 seconds

(kali@kali)-[~]
$
```

What was checked?

- Attempted to identify the operating system of the server.

Why does it matter?

OS fingerprinting information may be inferred

Identified Issue

- Knowing the operating system helps attackers craft targeted attacks.

Risk Level :- Low

Clear Remediation Steps

- Implement firewall and intrusion prevention rules
- Apply OS hardening measures
- Keep OS patches up to date

Nmap 3 : robots.txt Information Disclosure

Tool Used: Nmap

Command Example: nmap --script=http-robots.txt testphp.vulnweb.com

```
(kali㉿kali)-[~]
$ nmap --script=http-robots.txt testphp.vulnweb.com

Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-06 12:05 IST
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.29s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 25.86 seconds
```

What was checked?

- Reviewed the robots.txt file for exposed paths.

Why does it matter?

robots.txt file is publicly accessible and reveals internal paths

Identified Issue

- Attackers can use these paths to locate hidden or sensitive sections of the website.

Risk Level :- Low

Clear Remediation Steps

- ☐ Avoid listing sensitive directories in robots.txt
- ☐ Use authentication and access control instead
- ☐ Review robots.txt regularly
-

Finding: - 3

OWASP ZAP Vulnerability Assessment (Passive Scan)

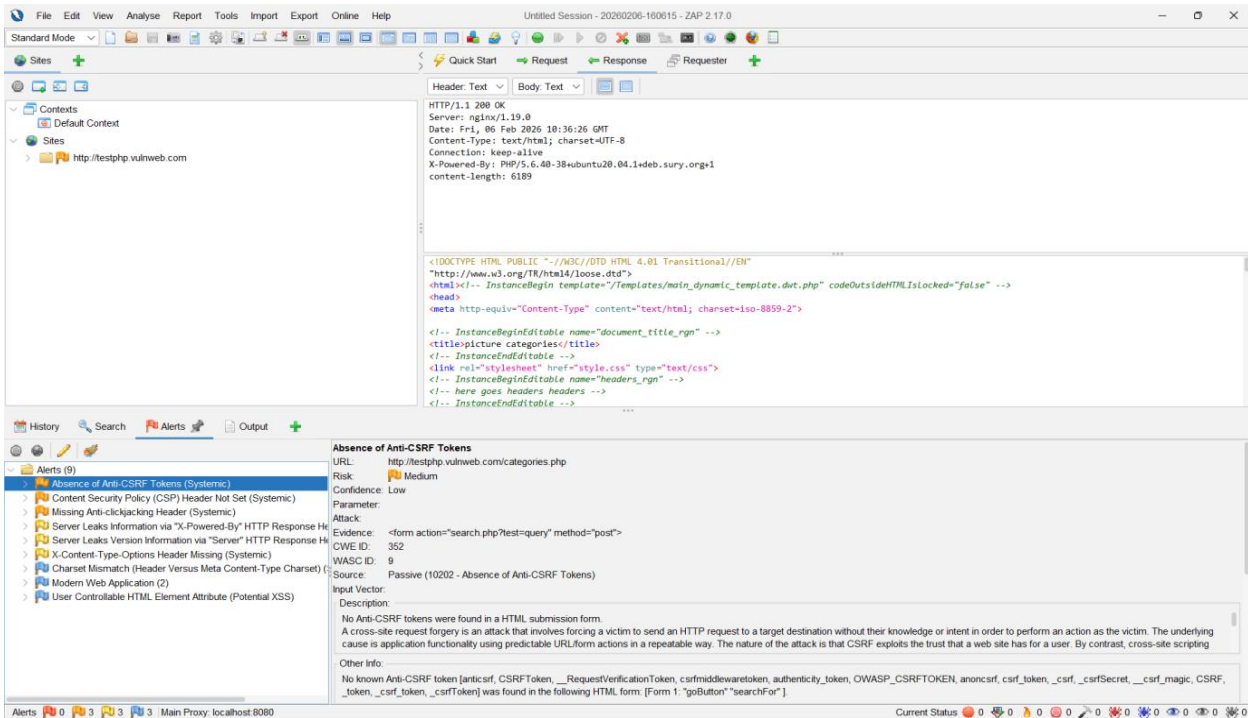
- | |
|---|
| • Tool Used: OWASP ZAP (Zed Attack Proxy) |
| • Scan Type: Passive Scan (Read-Only) |
| • Target Website: testphp.vulnweb.com |
| • Testing Approach: Non-intrusive, ethical security analysis |

Methodology
OWASP ZAP was used as a passive security analysis tool to monitor traffic between the browser and the website.
The tool identified common web security misconfigurations and weaknesses without sending malicious requests.

Owasp zap 1 : Missing Content Security Policy (CSP)

Tool Used: Owasp zap

**Command Example: nmap --script=http-robots.txt
testphp.vulnweb.com**



What was checked?

- The website does not implement a Content Security Policy (CSP) header.

Why does it matter?

Without CSP, attackers may inject malicious scripts into the website, potentially leading to data theft or page manipulation.

Identified Issue

- Attackers can use these paths to locate hidden or sensitive sections of the website.

Risk Level :- medium

Clear Remediation Steps

- ☐ Implement a Content-Security-Policy header
- ☐ Review and update CSP rules during application changes

Finding 2: Missing X-Frame-Options Header

What is the issue?

The website does not include the X-Frame-Options header.

Why does it matter?

This can allow attackers to embed the website inside a malicious page, leading to **clickjacking attacks**.

Risk Level :- Medium

Suggested Remediation

- | |
|---|
| <ul style="list-style-type: none">• Set X-Frame-Options to DENY or SAMEORIGIN• Prevent unauthorized framing of the website |
|---|

The screenshot displays the ZAP (Zed Attack Proxy) interface, a tool used for web application security testing. The interface is divided into several panes:

- Top Pane:** Contains the 'Standard Mode' toolbar and the 'Sites' list. The 'Sites' list shows a single site: 'http://testphp.vulnweb.com'.
- Middle Pane:** Displays the 'Quick Start' tab with buttons for 'Request', 'Response', and 'Requester'. Below these buttons are tabs for 'Header Text' and 'Body Text'.
- Right Pane:** Shows the 'Response' details for the selected site. It displays the HTTP status '200 OK', the server 'nginx/1.19.0', the date 'Fri, 06 Feb 2026 10:36:26 GMT', the content type 'text/html; charset=UTF-8', the connection 'keep-alive', the X-Powered-By 'PHP/5.6.40-38ubuntu20.04.1deb.sury.org', and the content length '6189'. Below this, the HTML content is displayed, showing a document structure with a title 'categories.php' and a link to 'style.css'.
- Bottom Pane:** Displays the 'Alerts' list. The 'Absence of Anti-CSRF Tokens (Systemic)' alert is highlighted. The alert details pane shows the URL 'http://testphp.vulnweb.com/categories.php', the risk level 'Medium', the parameter 'csrf_token', the attack type 'Cross-Site Request Forgery', the evidence 'csrf_token=csrf_token', the CWE ID '352', the WASC ID '9', the source 'Passive (10202 - Absence of Anti-CSRF Tokens)', the input vector 'Request', and the description: 'No Anti-CSRF tokens were found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting'.