

Vulnerability Assessment

Report For

testphp.vulnweb.com

Target Application: Acunetix Art

Target Ip: 44.228.249.3

Assessment Type: Passive Web Vulnerability Assessment

Prepared by: Iman Bedru

Program: Cyber Security Task 1(2026)-Future Interns

Date: February 2026

Table of Contents

1.Executive Summary.....	5
2.Scope of Assessment.....	6
3.Methodology.....	7
4.Summary of Findings.....	9
5.Detailed Findings.....	10
5.1 Sensitive Login Information Disclosure.....	10
5.2 Missing Content Security Policy (CSP).....	13
5.3 Missing Anti-CSRF Protection.....	15
5.4 Missing Anti-Clickjacking Protection.....	17
5.5 Missing X-Content-Type-Options Header.....	19
5.6 Server Version Disclosure.....	21
5.7 X-Powered-By Header Disclosure.....	22

5.8 Unencrypted HTTP Communication.....	23
6.Network Scan Summary.....	25
7.Overall Risk Assessment.....	26
8.Conclusion.....	27
9.Disclaimer.....	28

1. Executive Summary

This report presents the results of a passive vulnerability assessment conducted on a publicly accessible web application. The objective of this assessment was to identify security misconfigurations, information disclosure risks, and missing security controls using ethical, non-intrusive techniques.

The assessment was performed using:

- Nmap – Network and port analysis
- OWASP ZAP – Passive vulnerability scanning
- Browser Developer Tools – HTTP header and configuration review

No exploitation, credential attacks, or service disruption were performed.

The assessment identified multiple security weaknesses, including exposed login credentials, missing HTTP security headers, and server information disclosure. While no critical exploitation was attempted, these weaknesses increase the application's exposure to attacks such as cross-site scripting (XSS), clickjacking, credential abuse, and reconnaissance-based targeting.

Overall Risk Rating: MEDIUM

Addressing the identified issues will significantly improve the security posture of the application and reduce its attack surface.

2. Scope of Assessment

The assessment was conducted under the following conditions:

- Testing Type: Passive (Read-Only)
- Scope: Publicly accessible pages only
- Exploitation: Not performed
- Denial-of-Service Testing: Not performed
- Credential Brute Force: Not performed

All activities complied with ethical security assessment standards and internship guidelines provided by Future Interns.

3. Methodology

The assessment followed a structured and controlled approach:

1. Network reconnaissance to identify exposed services
2. Port scanning and service detection
3. Passive vulnerability scanning
4. Authentication page inspection
5. HTTP header and configuration analysis
6. Risk classification based on impact and likelihood

Findings were categorized as Low, Medium, or High risk depending on potential business impact

FileEditViewAnalyseReportToolsImportExportOnlineHelp

Safe Mode

Sites+ContextsDefault ContextSiteshttp://testphp.vulnweb.comGET:/GET:index.php

HistorySearchAlertsOutput

Alerts (7)Absence of Anti-CSRF Tokens (2)Content Security Policy (CSP) Header Not Set (2)Missing Anti-clickjacking Header (2)Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (2)Server Leaks Version Information via "Server" HTTP Response Header Field (2)X-Content-Type-Options Header Missing (2)Charset Mismatch (Header Versus Meta Content-Type Charset) (2)

Alerts 0 3 3 1 Main Proxy: localhost:8080

Quick StartRequestResponseRequester

<⚡

Manual Explore

This screen allows you to launch the browser of your choice so that you can explore your application while proxying through ZAP.

The ZAP Heads Up Display (HUD) brings all of the essential ZAP functionality into your browser.

URL to explore:

http://testphp.vulnweb.com/

Enable HUD:

☐

Explore your application:

Launch Browser

Firefox

You can also use browsers that you don't launch from ZAP, but will need to configure them to proxy through ZAP and to import the ZAP root

Input Vector:

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 consent. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The

Other Info:

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 1: "goButton" "search"]

Solution:

Phase: Architecture and Design

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard.

Reference:

https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

<https://cwe.mitre.org/data/definitions/352.html>

4. Summary of Findings

ID	Vulnerability	Risk Level
V1	Sensitive Login Information Disclosure	High
V2	Missing Content Security Policy (CSP)	High
V3	Missing Anti-CSRF Protection	Medium
V4	Missing Anti-Clickjacking Protection	Medium
V5	Missing X-Content-Type-Options Header	Medium
V6	Server Version Information Disclosure	Low
V7	X-Powered-By Header Disclosure	Low

Risk Distribution:

- 1. High: 2
- 2. Medium: 4
- 3. Low: 2

5. Detailed Findings

5.1 Sensitive Login Information Disclosure

Risk Level: HIGH

Description:

The login page publicly displays valid authentication credentials (e.g., suggested username and password). Authentication interfaces are designed to protect access to restricted areas of an application. Displaying usable credentials directly on the login page weakens this control and reduces the effectiveness of access restrictions.

Even if intended for demonstration purposes, publicly exposing credentials represents a serious security misconfiguration in real-world environments.



TEST and Demonstration site for Acunetix Web Vulnerability Scanner

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

search art

[Browse categories](#)

[Browse artists](#)

[Your cart](#)

[Signup](#)

[Your profile](#)

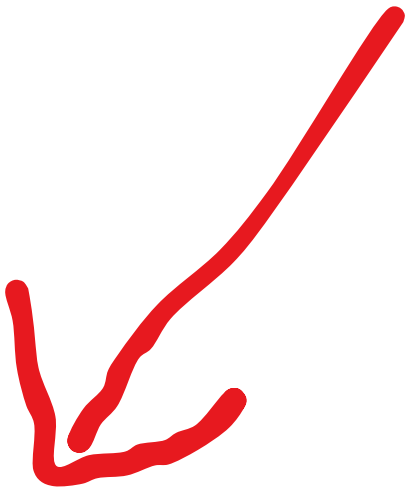
[Our guestbook](#)

[AJAX Demo](#)

Links
[Security art](#)
[PHP scanner](#)

If you are already registered please enter your login information below:

Username :	<input type="text"/>
Password :	<input type="password"/>
<input type="button" value="login"/>	



You can also signup here.
Signup disabled. Please use the username **test** and the password **test**.

Impact

This issue may result in:

- Unauthorized access to user accounts
- Abuse of authenticated features
- Credential reuse testing on other platforms
- Reduced confidence in authentication security
- Increased exposure to automated attacks

*In a production environment, this could lead to account compromise and data exposure.

Recommendation

- Immediately remove any displayed credentials from the login interface.
- Provide demonstration credentials only in restricted documentation if required.
- Enforce strong password complexity requirements.
- Implement account lockout mechanisms after repeated failed attempts.
- Enable authentication logging and monitoring.
- Ensure development environments are isolated from production systems.

5.2 Missing Content Security Policy (CSP)

Risk Level: HIGH

Description

The application does not implement a Content Security Policy header. CSP restricts which external resources the browser is allowed to load and execute. Without this control, browsers may execute scripts from untrusted sources.

Impact

- Increased risk of cross-site scripting (XSS) attacks
- Possible session hijacking
- Data theft through injected scripts
- Manipulation of client-side content

Client-side attacks can damage user trust and may result in reputational harm.

Recommendation

- Implement a restrictive Content Security Policy at the server level.
- Begin with a default self-only policy.
- Gradually whitelist required trusted sources.
- Combine CSP with proper input validation and output encoding.

Example:

Content-Security-Policy: default-src 'self';

⌵

Headers

Cookies

Request

Response

Cache

Timings

Stack Trace

7 Filter Headers

Block

Res

GET http://testphp.vulnweb.com/favicon.ico

Status

200 OK ⓘ

Version

HTTP/1.1

Transferred

894 B (894 B size)

Referrer Policy

strict-origin-when-cross-origin

DNS Resolution

System

Response Headers (241 B)

Raw ⓘ

ⓘ Accept-Ranges: bytes

ⓘ Connection: keep-alive

ⓘ Content-Length: 894

ⓘ Content-Type: image/x-icon

ⓘ Date: Sun, 22 Feb 2026 04:41:56 GMT

ⓘ ETag: "4dca64a4-37e"

ⓘ Last-Modified: Wed, 11 May 2011 10:27:48 GMT

ⓘ Server: nginx/1.19.0

Request Headers (360 B)

Raw ⓘ

ⓘ Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5

ⓘ Accept-Encoding: gzip, deflate

ⓘ Accept-Language: en-US,en;q=0.5

5.3 Missing Anti-CSRF Protection)

Risk Level: MEDIUM

Description:

The application does not appear to implement Anti-CSRF tokens for state-changing requests. CSRF protection ensures that actions performed within authenticated sessions are legitimate and intentionally initiated by users

Impact

Attackers may trick authenticated users into submitting unauthorized requests, potentially leading to:

- Account modifications
- Unauthorized transactions
- Privilege misuse

Recommendation

- Implement Anti-CSRF tokens for all sensitive forms and requests.
- Validate tokens server-side.
- Use SameSite cookie attributes for additional protection.

File Edit View Analyse Report Tools Import Export Online Help

Mode [dropdown] [icons]

Quick Start Request Response Requester [plus]

Header: Text Body: Text [icons]

Contexts
Default Context

Sites
http://testphp.vulnweb.com
GET: /
GET: index.php

GET http://testphp.vulnweb.com/index.php HTTP/1.1
host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Referer: http://testphp.vulnweb.com/
Upgrade-Insecure-Requests: 1
Priority: u=0, i

History Search Alerts Output [plus]

Alerts (7)

- Absence of Anti-CSRF Tokens (2)
- Content Security Policy (CSP) Header Not Set (2)
- Missing Anti-clickjacking Header (2)
- Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (2)
- Server Leaks Version Information via "Server" HTTP Response Header Field (2)
- X-Content-Type-Options Header Missing (2)
- Charset Mismatch (Header Versus Meta Content-Type Charset) (2)

Input Vector:

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

Other Info:

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 1: "goButton" "searchFor"].

Solution:

Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
For example, use anti-CSRF packages such as the OWASP CSRFGuard.

5.4 Missing Anti-Clickjacking Protection)

Risk Level: MEDIUM

Description

The application does not define the X-Frame-Options header or equivalent frame restrictions.

Impact

- Attackers may embed the application within malicious pages and trick users into clicking hidden elements.

Recommendation

- Add:
X-Frame-Options: SAMEORIGIN
- Or configure:
Content-Security-Policy: frame-ancestors 'self';

File Edit View Analyse Report Tools Import Export Online Help

Safe Mode

Sites +

Contexts

- Default Context

Sites

- http://testphp.vulnweb.com
 - GET: /
 - GET: index.php

Quick Start Request Response Requester +

Header: Text Body: Text

```
GET http://testphp.vulnweb.com/index.php HTTP/1.1
host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Referer: http://testphp.vulnweb.com/
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

History Search Alerts Output +

Alerts (7)

- Absence of Anti-CSRF Tokens (2)
- Content Security Policy (CSP) Header Missing (2)
- Missing Anti-clickjacking Header (2)
- Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (2)
- Server Leaks Version Information via "Server" HTTP Response Header Field (2)
- X-Content-Type-Options Header Missing (2)
- Charset Mismatch (Header Versus Meta Content-Type Charset) (2)

Input Vector:

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, or 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 this attack is that it can be performed by an attacker who is not the user, but who is able to impersonate the user.

Other Info:

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 1: "goButton" "searchFor"].

Solution:

Phase: Architecture and Design

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard.

5.5 Missing X-Content-Type-Options Header

Risk Level: MEDIUM

Description

The X-Content-Type-Options header is not configured. This header prevents browsers from interpreting files as a different MIME type than declared.

Impact

- Without this control, browsers may incorrectly interpret malicious files, increasing risk of code execution.

Recommendation

Add:

- X-Content-Type-Options: nosniff

FileEditViewAnalyseReportToolsImportExportOnlineHelp

Safe Mode

Sites+ContextsDefault ContextSiteshttp://testphp.vulnweb.comGET:/GET:index.php

Quick StartRequestResponseRequester+Header: TextBody: TextGET http://testphp.vulnweb.com/index.php HTTP/1.1host: testphp.vulnweb.comUser-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8Accept-Language: en-US,en;q=0.5Connection: keep-aliveReferer: http://testphp.vulnweb.com/Upgrade-Insecure-Requests: 1Priority: u=0, i

HistorySearchAlertsOutput+Alerts (7)> Absence of Anti-CSRF Tokens (2)> Content Security Policy (CSP) Header Not Set (2)> Missing Anti-clickjacking Header (2)> Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (2)> Server Leaks Version Information via "Server" HTTP Response Header Field (2)> X-Content-Type-Options Header Missing (2)> Charset Mismatch (Header Versus Meta Content-Type Charset) (2)

Input Vector:
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
Other Info:
No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 1: "goButton" "searchFor"].
Solution:
Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
For example, use anti-CSRF packages such as the OWASP CSRFGuard.

5.6 Server Version Information Disclosure

Risk Level: LOW

Description

The server discloses software and version details in HTTP responses.

Impact

- Technology disclosure assists attackers in identifying known vulnerabilities and planning targeted attacks.

Recommendation

- Disable version disclosure within the web server configuration.



```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sun, 22 Feb 2026 06:18:52 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
content-length: 4958
```

5.7 X-Powered-By Header Disclosure

Risk Level: LOW

Description



The application exposes underlying technology information via the X-Powered-By header.

Impact

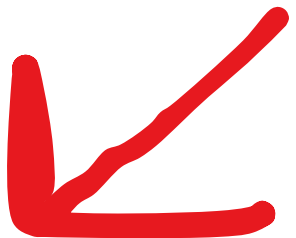
- Information disclosure supports reconnaissance efforts and reduces uncertainty for attackers.

Recommendation

- Remove or disable the X-Powered-By header in server configuration settings.

Header: Text ▾ Body: Text ▾  

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sun, 22 Feb 2026 06:18:52 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
content-length: 4958
```



5.8 Unencrypted HTTP Communication

Risk Level: MEDIUM

Description

The application is accessible over HTTP (Port 80) without automatic redirection to HTTPS. When accessed through HTTP, the browser displays a “Not Secure” warning, indicating that the connection is not encrypted.

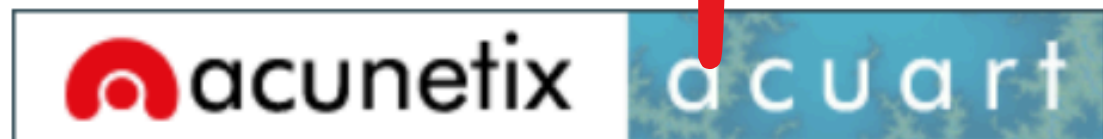
Data transmitted over HTTP is sent in plaintext and can be intercepted by third parties.

Impact

- Possible interception of login credentials
- Exposure of session information
- Increased risk of Man-in-the-Middle attacks
- Reduced user trust

Recommendation

- Enforce HTTPS across the application.
- Redirect all HTTP traffic to HTTPS.
- Implement a valid SSL/TLS certificate.
- Configure HTTP Strict Transport Security (HSTS).



TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

search art

[Browse categories](#)

[Browse artists](#)

[Your cart](#)

[Signup](#)

[Your profile](#)

[Our guestbook](#)

[AJAX Demo](#)

Links

[Security art](#)

[PHP scanner](#)

[PHP vuln help](#)

[Fractal Explorer](#)



welcome to our page

Test site for Acunetix WVS.

6. Network Scan Summary

Port scanning identified the following open ports:

- Port 80 (HTTP)
- Port 443 (HTTPS)

Open ports increase the attack surface and should be continuously monitored and securely configured.

```
$ nmap -A -Pn testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-22 00:32 EST
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.17s 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 VERSION
80/tcp    open  http    nginx 1.19.0
|_http-title: Home of Acunetix Art
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: bridge|VoIP adapter|general purpose
Running (JUST GUESSING): Oracle Virtualbox (98%), Slirp (98%), AT&T embedded (95%), QEMU (94%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:danny_gasparovski:slirp cpe:/a:qemu:qemu
Aggressive OS guesses: Oracle Virtualbox Slirp NAT bridge (98%), AT&T BGW210 voice gateway (95%), QEMU user mode ne
twork gateway (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   246.53 ms ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 60.91 seconds
```

7. Overall Risk Assessment

- The passive vulnerability assessment identified multiple security weaknesses primarily related to authentication misconfiguration, missing browser security headers, and information disclosure through HTTP responses.
- Although no active exploitation was conducted, the presence of exposed login credentials and missing security controls significantly increases the attack surface of the application. These weaknesses could allow attackers to perform reconnaissance, manipulate client-side behavior, or potentially gain unauthorized access under certain conditions.
- The most critical concern identified during the assessment is the exposure of login credentials and the absence of Content Security Policy protections. These issues directly affect authentication integrity and client-side security.
- While the application appears functional, the accumulation of medium- and high-risk misconfigurations elevates the overall security exposure.
- **Final Overall Risk Rating: MEDIUM**
- Remediation of the high-risk findings should be prioritized immediately, followed by implementation of the recommended security headers and hardening measures.

8. Conclusion

This vulnerability assessment was conducted using passive, non-intrusive testing techniques to evaluate the security posture of the target web application.

The assessment identified several configuration weaknesses that increase exposure to:

- Client-side attacks such as Cross-Site Scripting (XSS)
- Authentication misuse and credential abuse
- Clickjacking and request forgery attacks
- Information disclosure aiding attacker reconnaissance

Although no critical system compromise was attempted, the identified vulnerabilities demonstrate gaps in secure configuration practices and defensive implementation.

Addressing the identified issues will:

- Strengthen authentication controls
- Reduce client-side exploitation risk
- Minimize information leakage
- Improve overall security maturity

Regular vulnerability assessments, continuous monitoring, and secure configuration management are strongly recommended to maintain long-term resilience.

Proactive remediation will significantly reduce the likelihood of future security incidents.

9. Disclaimer

- This assessment was conducted strictly using passive, read-only security testing techniques.
- No exploitation, denial-of-service testing, brute-force attacks, or unauthorized access attempts were performed during the assessment process.
- The findings presented in this report are based solely on observable configurations and publicly accessible responses at the time of testing.
- This report has been prepared for educational and internship evaluation purposes under the Cyber Security Internship Program at Future Interns.
- The vulnerabilities identified may not represent the full security posture of the application and should be validated in a controlled remediation process.