

REPORT

VULNERABILITY ASSESSMENT REPORT

Title: Vulnerability Assessment of Web Application

Submitted By: P RENUGA

Course: Cyber Security Lab

EXECUTIVE SUMMARY

This report presents the results of a vulnerability assessment conducted on a publicly available test website. The objective of this assessment was to identify potential security vulnerabilities using ethical and non-intrusive testing techniques. Various security testing tools and manual inspection methods were used to analyze the target system. The assessment identified multiple security weaknesses that may expose the system to cyber threats. Recommendations have been provided to mitigate these risks and improve overall security.

SCOPE OF ASSESSMENT

“

The scope of this assessment includes the analysis of a publicly accessible web application. The testing was limited to passive and non-intrusive methods.

Target Website:<http://testphp.vulnweb.com>

Assessment Type: Vulnerability Assessment

Testing Method: Ethical and non-intrusive

”

METHODOLOGY

The vulnerability assessment was conducted using the following approach:

Network reconnaissance was performed to identify open ports and services running on the target system.

Automated vulnerability scanning was conducted to detect security misconfigurations and common web vulnerabilities.

Manual inspection was performed using browser developer tools to analyze cookies, HTTPS configuration, and network requests

Tools Used

Network scanning tool for port and service detection

Automated vulnerability scanner for identifying security risks

Browser developer tools for manual security analysis

FINDING 1: MISSING SECURE COOKIE FLAG

DESCRIPTION:

COOKIES WERE FOUND WITHOUT THE SECURE ATTRIBUTE.

RISK LEVEL: HIGH

IMPACT:

ATTACKERS MAY INTERCEPT SESSION COOKIES OVER UNSECURED NETWORKS, LEADING TO SESSION HIJACKING.

RECOMMENDATION:

ENABLE SECURE AND HTTPONLY FLAGS FOR ALL COOKIES.

FINDING 2: MIXED CONTENT VULNERABILITY

DESCRIPTION:

THE WEBSITE LOADS SOME RESOURCES USING HTTP INSTEAD OF HTTPS.

RISK LEVEL: MEDIUM

IMPACT:

DATA TRANSMITTED MAY BE INTERCEPTED OR MODIFIED BY ATTACKERS.

RECOMMENDATION:

ENSURE ALL WEBSITE RESOURCES ARE LOADED THROUGH HTTPS.

FINDING 3: MISSING SECURITY HEADERS

DESCRIPTION:

THE APPLICATION LACKS ESSENTIAL SECURITY HEADERS.

RISK LEVEL: MEDIUM

IMPACT:

THIS MAY EXPOSE THE SYSTEM TO CLICKJACKING AND CROSS-SITE SCRIPTING ATTACKS.

RECOMMENDATION:

IMPLEMENT NECESSARY SECURITY HEADERS SUCH AS X-FRAME-OPTIONS AND CONTENT SECURITY POLICY.

TOOL: NMAP

OPEN PORT(80,ETC)
SERVICE VERSION
OUTDATED VERSION

The screenshot shows the Nmap interface with the following details:

- Target:** testphp.vulnweb.com
- Profile:** Regular scan
- Command:** nmap testphp.vulnweb.com
- Services Tab (Selected):** Shows a list of services, with "http" selected.
- Nmap Output Tab:** Displays the scan results:

```
nmap testphp.vulnweb.com
Starting Nmap 7.96 ( https://nmap.org ) at 2026-02-21 18:40 India
Standard Time
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.26s 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 20.58 seconds
```
- Filter Hosts:** A button at the bottom left.

Scan Tools Profile Help

Target: testphp.vulnweb.com Profile: Regular scan Scan Cancel

Command: nmap testphp.vulnweb.com

Hosts Services

OS Host

Scans

Status Command

Unsaved nmap testphp.vulnweb.com

Filter Hosts Append Scan Remove Scan Cancel Scan

This screenshot shows a network scanning interface with the 'Scans' tab selected. The target is set to 'testphp.vulnweb.com' and the profile is 'Regular scan'. A single scan entry, 'nmap testphp.vulnweb.com', is listed under the 'Scans' tab. The 'Ports / Hosts' tab is also visible.

Scan Tools Profile Help

Target: testphp.vulnweb.com Profile: Regular scan Scan Cancel

Command: nmap testphp.vulnweb.com

Hosts Services

OS Host

Ports / Hosts

Topology Host Details Scans

Port Protocol State Service Version

80 tcp open http

Filter Hosts

This screenshot shows a network scanning interface with the 'Ports / Hosts' tab selected. The target is set to 'testphp.vulnweb.com' and the profile is 'Regular scan'. A single port entry, '80/tcp [open] http', is listed under the 'Ports / Hosts' tab. The 'Hosts' tab is also visible.

TOOL:OWASP ZAP

DIRECTORY BROWER MISSING HEADER ZSS VULNARABILITY COOKIES ISSUES MIXED CONTENT

The screenshot shows the OWASP ZAP interface in standard mode. The top navigation bar includes 'Sites', 'Header.Text', 'Body.Text', 'Request', 'Response', and 'Requester'. The left sidebar shows 'Contexts' (Default Context, Sites) and 'Alerts (18)' which is expanded to show 'Cross Site Scripting (DOM Based)' and 'Cross Site Scripting (Reflected) (19)'. The 'Cross Site Scripting (Reflected)' item is selected. The main pane displays the HTTP response header and the HTML source code. The source code contains a reflected XSS payload: <script>alert(1)</scRipt>. Below the code, a detailed alert summary is provided:

Cross Site Scripting (Reflected)
URL: http://testphp.vulnweb.com/guestbook.php
Risk: High
Confidence: Medium
Parameter: name
Attack: <script>alert(1)</scRipt>
Evidence: <script>alert(1)</scRipt>
CWE ID: 79
WASC ID: 8
Source: Active (40012 - Cross Site Scripting (Reflected))
Input Vector: Form Query
Description: Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser or a client-side application embedded in another application such as a mobile app or a game. The attack allows an attacker to inject malicious code into the victim's browser, which can then be executed.

At the bottom, there are buttons for History, Search, Alerts, AJAX Spider, Output, Spider, and Active Scan. The status bar shows 'Current Status' with various icons.

This screenshot shows the OWASP ZAP interface in standard mode, similar to the first one but with a different alert selected. The 'Alerts' section now highlights 'Missing Anti-clickjacking Header (Systemic)'. The alert details are as follows:

Missing Anti-clickjacking Header
URL: http://testphp.vulnweb.com
Risk: Medium
Confidence: Medium
Parameter: x-frame-options
Attack:
Evidence:
CWE ID: 1021
WASC ID: 15
Source: Passive (10020 - Anti-clickjacking Header)
Alert Reference: 10020-1
Input Vector:
Description: This alert indicates that the 'x-frame-options' header is missing from the response, which is crucial for preventing clickjacking attacks.

The rest of the interface is identical to the first screenshot, including the top bar, sidebar, and status bar.

Header.Text Body.Text

```
POST http://testphp.vulnweb.com/secured/newuser.php HTTP/1.1
host: testphp.vulnweb.com
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/5
pragma: no-cache
cache-control: no-cache
content-type: application/x-www-form-urlencoded
referer: http://testphp.vulnweb.com/signup.php
content-length: 96
```

uuname=ZAP&upass=ZAP&upass2=ZAP&username=ZAP&ucc=ZAP&uemail=ZAP&uphone=ZAP&uaddress=&signup=signup

Authentication Request Identified

URL: http://testphp.vulnweb.com/secured/newuser.php
Risk: Informational
Confidence: Low
Parameter: uemail
Attack:
Evidence: upass
CWE ID:
WASC ID:
Source: Passive (10111 - Authentication Request Identified)
Input Vector:
Description:

The given request has been identified as an authentication request. The 'Other Info' field contains a set of key-value lines which identify any relevant fields. If the request is in a context which has an 'Authentication Method' set to "Auto-Detect" then this rule will change the authentication to match the request identified.

TOOL:BROWSER DEVTOOLS CHECK

SECURITY TAB
SECURE OR NOT
NETWORK TAB
REQUEST URL
HEADER
APPLICATION
COOKIE
HTTP ONLY ENABLE

Not secure testphp.vulnweb.com/login.php

Dimensions: Responsive 658 x 557 100% No throttling 'Save-Data': default

Privacy Controls Third-party cookies

Security Overview Main origin Reload to view details

This page is not secure.

Console Issues Network conditions

Caching Disable cache

Network No throttling Save-Data: default

User agent Use browser default

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art

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

Username:

Password:

login

You can also [signup here](#).

Signup disabled. Please use the username **test** and the password **test**.

About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd.

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL injection, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

Name: login.php

Request URL: http://testphp.vulnweb.com/login.php

Request Method: GET

Status Code: 200 OK

Remote Address: 44.228.249.3:80

Referer Policy: origin-when-cross-origin

Response headers:

Connection: keep-alive

Dimensions: Responsive ▾ 658 x 557 100% ▾ No throttling ▾ 'Save-Data': default ▾

Not secure testphp.vulnweb.com/login.php

Acunetix acuart

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art

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

Username:

Password:

login

You can also [signup here](#).

Signup disabled. Please use the username **test** and the password **test**.

About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd.

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL injection, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

Primary static storage

- Interest groups
- Shared storage
- Cache storage
- Storage buckets

Background services

- Backforward cache
- Background fetch
- Background sync
- Bounce tracking mitigation
- Notifications
- Payment handler
- Periodic background sync
- Speculative loads
- Push messaging
- Reporting API

No report or endpoint

On this page you will be able to inspect Reporting API reports and endpoints. [learn more](#)

Frames

- top

Console Issues Network conditions

Caching Disable cache

Network No throttling 'Save-Data': default

User agent Use browser default

RISK CLASSIFICATION

Risk Level

Description

High

Can lead to major security breaches

Medium

May allow limited system exploitation

Low

Minor security weaknesses

Remediation Plan

To improve the overall security posture, the following measures are recommended:

Implement secure cookie configurations

Enforce HTTPS across all web resources

Enable essential security headers

Regularly update software and server component

Avoid storing sensitive data in client-side storage

Conclusion

The vulnerability assessment identified several security weaknesses in the target web application. Although no critical system compromise was observed, the presence of misconfigurations increases the risk of cyber attacks. Implementing the recommended mitigation strategies will significantly enhance the security of the system.

References

Security testing best practices

Web application security guidelines

Vulnerability assessment methodologies