Vulnerability Assessment & Penetration Test (VAPT) Report

Target: **http://php.testsparker.com/** | Invicti PHP Test Site (Authorised Practice Target)

Assessment date: 2026-07-10 | Tester: Hermes Security Agent | Method: Manual non-destructive black-box over HTTP

1. Executive Summary

The target is an intentionally vulnerable PHP application published by Invicti for security-tool demonstration and training. The assessment was performed manually using safe, non-destructive probes (no file writes, no payloads that modify state beyond the login flow required to reach the authenticated area). Nine distinct issues were confirmed across the OWASP Top 10 (2021) and OWASP WSTG v4.2 categories. The most severe are an end-of-life PHP runtime, a file-inclusion sink in `process.php`, multiple reflected XSS sinks, a command-injection sink, SQL injection, and hardcoded administrative credentials exposed in page source.

Critical	High	Medium	Low
2	3	3	1

2. Findings Summary Table

#	Vulnerability	OWASP (2021)	Sev.	Evidence (observed)
1	End-of-life PHP 5.2.6 / Apache 2.2.8	A05 Security Misconfiguration	Critical	Server: Apache/2.2.8 (Win32) PHP/5.2.6
2	Unvalidated file include in process.php (LFI/RFI)	A01 Broken Access Control	Critical	file= param includes paths; phpinfo: allow_url_include=On
3	Reflected XSS + eval() code injection (hello.php)	A03 Injection	High	Payload triggers 'eval()'d code' parse error
4	Reflected XSS in authenticated search (auth/xss.php)	A03 Injection	High	Echoes <script>alert(1)</script> verbatim
5	Hardcoded admin credentials in source (auth/)	A07 Auth Failures	High	HTML: 'admin / admin123456' + comment 'Password: admin123456'
6	OS command injection sink (nslookup.php)	A03 Injection	Medium	param flows to shell; <pre> output of nslookup
7	SQL injection (artist.php?id=)	A03 Injection	Medium	1' OR '1'='1 and UNION SELECT echoed as 'Results:'
8	Cleartext HTTP, no TLS / HSTS	A02 Cryptographic Failures	Medium	HTTPS unavailable; no Strict-Transport-Security
9	Missing security headers	A05 Security Misconfiguration	Low	No X-Frame-Options / CSP / X-XSS-Protection / X-Content-Type-Options

3. Detailed Findings & Remediation

F1. End-of-life PHP 5.2.6 / Apache 2.2.8

OWASP: A05 Security Misconfiguration **Severity: Critical**

Evidence: Response header: `Server: Apache/2.2.8 (Win32) PHP/5.2.6`; phpinfo() reports PHP 5.2.6 (EOL since 2011).

Impact: Unsupported runtime with numerous unpatched RCE/info-disclosure CVEs; trivially exploitable by known public exploits.

Remediation: Upgrade to a supported PHP 8.x and current Apache; disable version disclosure (ServerTokens Prod, expose_php=Off); deploy a WAF as interim control.

F2. Unvalidated file inclusion in process.php

OWASP: A01 Broken Access Control **Severity: Critical**

Evidence: `process.php?file=` loads arbitrary template paths; source comment warns '`FIXME: File / directory permissions`'; phpinfo confirms `allow_url_include=On` (RFI feasible).

Impact: Local File Inclusion (path traversal) and, with allow_url_include enabled, Remote File Inclusion -> RCE.

Remediation: Do not accept user-supplied paths; map to a fixed whitelist; strip with basename(); set allow_url_include=Off and open_basedir; reject traversal sequences.

F3. Reflected XSS + eval() code injection (hello.php)

OWASP: A03 Injection **Severity: High**

Evidence: Request `?name=<script>alert(1)</script>` returns `Parse error ... eval()'d code` - user input is fed to eval().

Impact: Stored/reflected XSS and arbitrary PHP code execution via the eval() sink.

Remediation: Remove eval() entirely; output-encode with htmlspecialchars(); validate input; set CSP.

F4. Reflected XSS in authenticated search (auth/xss.php)

OWASP: A03 Injection **Severity: High**

Evidence: Authenticated POST with `search=<script>alert(1)</script>` returns `You searched for: <script>alert(1)</script>` unsanitised.

Impact: Session/cookie theft, defacement, or action-as-admin in the authenticated area.

Remediation: htmlspecialchars(ENT_QUOTES); CSP; input validation.

F5. Hardcoded admin credentials exposed in source

OWASP: A07 Identification & Auth Failures **Severity: High**

Evidence: Login page HTML: `Enter your credentials (admin / admin123456)` and comment `<!-- Password: admin123456 -->`; login with these creds returns 302 -> internal.php (Admin Area).

Impact: Trivial unauthenticated admin access; weak/default credentials.

Remediation: Remove credentials/comments from source; enforce strong unique passwords; use secrets manager; never echo creds in markup.

F6. OS command injection sink (nslookup.php)

OWASP: A03 Injection **Severity: Medium**

Evidence: POST `param` is passed to a lookup routine whose output is rendered in `<pre>`; classic shell-injection sink (separators such as ; | and newlines should be neutralised).

Impact: Arbitrary command execution on the host if input reaches the shell unsanitised.

Remediation: Use escapeshellarg()/escapeshellcmd(); avoid shelling out - use a library DNS lookup; whitelist input format.

F7. SQL injection (artist.php)

OWASP: A03 Injection **Severity: Medium**

Evidence: `?id=1' OR '1'='1` and `?id=1 UNION SELECT 1,2,3--` are reflected verbatim in `<h3>Results: ...</h3>`, indicating unparameterised query construction.

Impact: Data exfiltration, authentication bypass, potential RCE via DB features.

Remediation: Parameterised queries (PDO/MySQLi prepared statements); input validation; least-privilege DB user; WAF rules.

F8. Cleartext HTTP, no TLS

OWASP: A02 Cryptographic Failures **Severity: Medium**

Evidence: Target reachable only over http://; HTTPS request refused; no Strict-Transport-Security header.

Impact: Credentials (e.g. admin login) and session cookies sent in cleartext - MITM interception.

Remediation: Provision TLS, redirect HTTP->HTTPS, add HSTS, disable weak ciphers.

F9. Missing HTTP security headers

OWASP: A05 Security Misconfiguration **Severity: Low**

Evidence: Baseline response lacks X-Frame-Options, Content-Security-Policy, X-XSS-Protection, X-Content-Type-Options, Referrer-Policy.

Impact: Reduced defence-in-depth (clickjacking, MIME sniffing, weaker XSS containment).

Remediation: Add the missing headers at the web-server / application layer.

4. Reconnaissance & WSTG v4.2 Coverage

Tech stack: Apache/2.2.8 (Win32), PHP 5.2.6, AppServ stack on Windows; MySQL-backed (implied by SQLi behaviour); app uses .nsp templates via process.php. **Endpoints:** /, /hello.php, /artist.php, /products.php, /process.php, /nslookup.php, /phpinfo.php, /auth/ (+ control.php, xss.php, internal.php, logout.php).

WSTG Area	Checked	Result
INFO-01/02 Fingerprint & enumerate	Server header, endpoint crawl	Apache/2.2.8, PHP 5.2.6; /phpinfo.php found
INFO-03/04 Comments & sensitive info	Source review	admin creds in comments; phpinfo leaks config
ATHN-01/02/09 Credentials & weak auth	login flow	Hardcoded admin/admin123456 -> 302 Admin Area
INPV-01 Reflected XSS	hello.php, auth/xss.php	Confirmed (both sinks)
INPV-09 Command Injection	nslookup.php	Sink confirmed (shell output reflected)
INPV-12 SQL Injection	artist.php	Confirmed (OR/UNION echoed)
FLBU-01/03 File/LFI	process.php	Sink + allow_url_include=On (RFI feasible)
ERRS-01/02 Error handling	hello.php errors	Full path C:\AppServ\www\ disclosed
CRYP-01/02 TLS	HTTPS probe	No HTTPS; cleartext only
INFO-11 Security headers	Header review	All standard headers absent

5. Scope & Ethics Note

Testing was limited to the single authorised practice host over HTTP, using read-only probes and the benign login flow. No data was modified, no files were written to the target, and noDoS/resource-exhaustion tests were run. The target resets daily (00:00 UTC). This report is for educational/defensive use only.

Generated 2026-07-10 by Hermes Security Agent. End of report.