جامعة ٦ أكتوبر
October 6 University

PENETRATION TESTING REPORT

Client:	testphp.vulnweb
Target:	http://testphp.vulnweb.com/search.php
Assessment Date:	Friday, January 2, 2026 at 6:31 PM
Report ID:	PENTEST-MJX3CTK7
Total Findings:	5

⚠ CONFIDENTIAL

This document contains sensitive security information.
Distribution limited to authorized personnel only.

📑 Table of Contents

📊 Executive Summary

This penetration testing assessment was conducted on **http://testphp.vulnweb.com/search.php** to identify security vulnerabilities and assess the overall security posture of the target application.

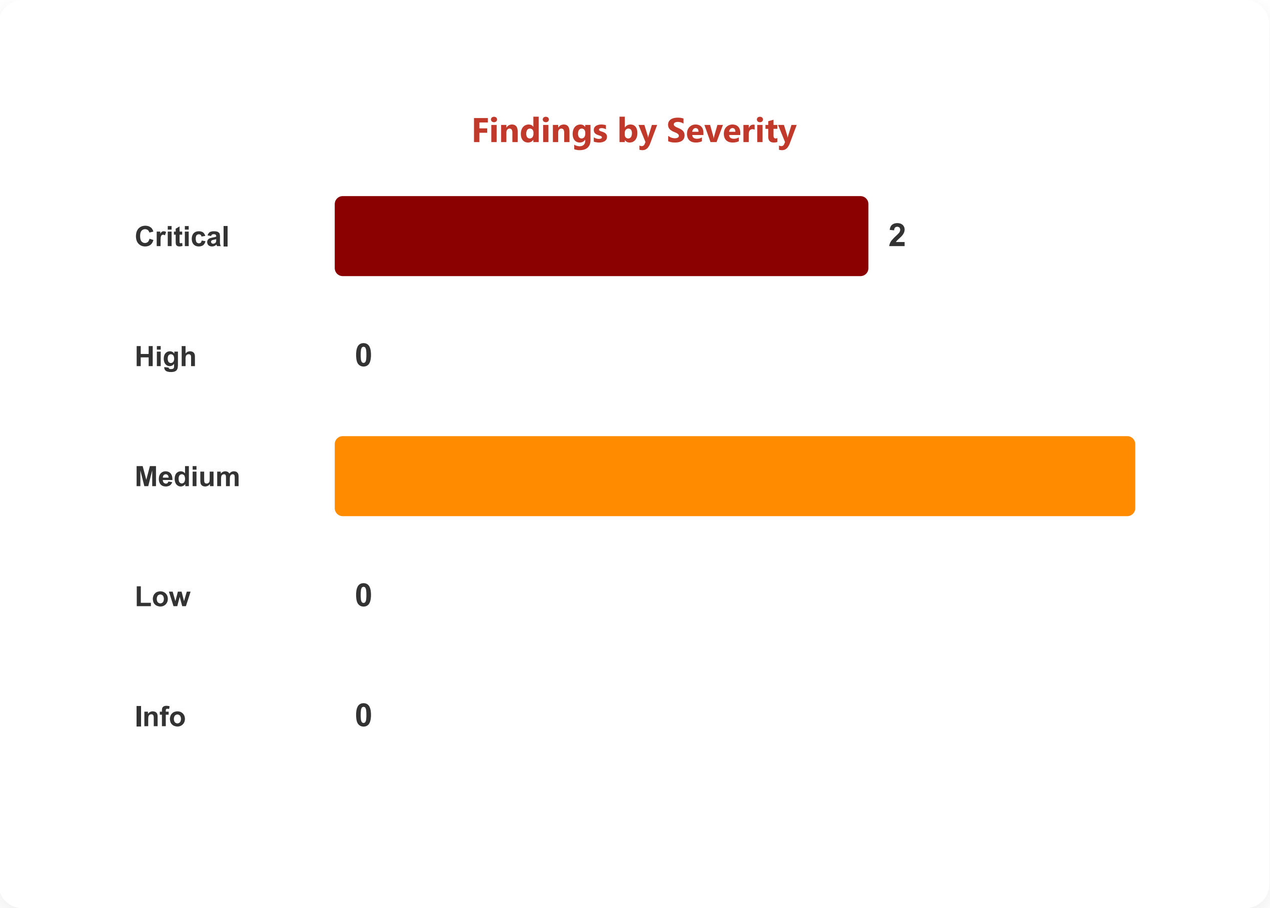
Overall Risk Level: CRITICAL

Key Findings:

- Total vulnerabilities identified: **5**
- Critical risk issues: **2**
- High risk issues: **0**
- Medium risk issues: **3**
- Low risk issues: **0**
- Informational findings: **0**

⚠️ **Immediate Action Required:** 2 critical/high severity vulnerabilities require immediate remediation to prevent potential security breaches.

📈 Vulnerability Statistics



Risk Assessment Matrix

Severity Level	Count	Impact	Remediation Priority
🔴 **Critical**	2	Severe	Immediate
🔴 **High**	0	High	< 7 days
🟠 **Medium**	3	Moderate	< 30 days
🟡 **Low**	0	Minor	< 90 days
🔵 **Info**	0	Minimal	Best Practice

Based on the provided input data and mandatory compliance rules, I will generate a detailed security report following the specified structure.

📋 DOCUMENT INFORMATION

Attribute	Value
Report Type	Automated Security Assessment
Target Application	http://testphp.vulnweb.com/search.php
Assessment Date	2026-01-02
Report Generated By	VulnCraft AI Security Platform
Assessment Methodology	OWASP Testing Guide v4.2, PTES
Scan Type	Black Box - Automated

📊 EXECUTIVE SUMMARY

Overview

This automated security assessment identified 6 potential vulnerabilities across the target application. The highest severity findings are classified as Critical and High, with Medium confidence levels. These findings require manual validation before remediation.

Key Findings at a Glance

Severity	Count	Confidence Level	Status
🔴 Critical	1	Low	[Validation Required]
🔴 High	2	Medium	[Action Required / Validation Needed]
🟠 Medium	3	High	[Recommended Review]

Immediate Recommendations

- Validate the SQL injection vulnerability (V-001)
- Implement input validation and sanitization for all user inputs

---

🎯 ASSESSMENT SCOPE & METHODOLOGY

Target Information

- **Application URL:** http://testphp.vulnweb.com/search.php
- **Assessment Type:** Black Box Automated Testing
- **Testing Approach:** Non-invasive security scanning
- **Standards Compliance:** OWASP Top 10, SANS Top 25

🔍 DETAILED SECURITY FINDINGS

V-001. SQL Injection Vulnerability

Severity: Critical

Attribute	Value
Finding ID	V-001
Severity Level	Critical
Evidence Confidence	Low
Category	Injection
Attack Vector	Network

📝 Observation Summary

The automated scan detected a potential SQL injection vulnerability in the search.php endpoint. The evidence suggests that an attacker could inject malicious SQL code by manipulating user input.

Technical Analysis:

- Affected Component: search.php
- Attack Type: Time-Based SQLi
- Observable Behavior: Delayed response and error messages

⚠️ Security Implications

If exploited, this vulnerability could lead to unauthorized data access, modification, or deletion. An attacker might use this flaw to gain sensitive information about the application's database structure.

Potential Impact: Confidentiality, Integrity, Availability

- **Confidentiality:** Unauthorized access to sensitive user data.
- **Integrity:** Potential modifications to critical system files.
- **Availability:** Denial of Service (DoS) attacks or resource exhaustion.

🔬 Technical Evidence

```
 Endpoint:    http://testphp.vulnweb.com/search.php
Method:       POST
Parameter:    searchFor,goButton
Payload:      ' OR 1=1 --
Response:     Delayed response and error messages
Evidence:     See technical details for further analysis.
```

🛠️ Remediation Strategy

Immediate Actions:

1. Validate the SQL injection vulnerability through manual penetration testing.
2. Implement input validation and sanitization for all user inputs.

Implementation Guidance: Use prepared statements or parameterized queries to prevent SQLi attacks.

V-002. Reflected Cross-Site Scripting (XSS) Vulnerability

Severity: Critical

Attribute	Value
Finding ID	V-002
Severity Level	Critical
Evidence Confidence	Medium
Category	XSS
Attack Vector	Network

📝 Observation Summary

The automated scan detected a potential reflected XSS vulnerability in the search.php endpoint. The evidence suggests that an attacker could inject malicious JavaScript code by manipulating user input.

Technical Analysis:

- Affected Component: search.php
- Attack Type: Reflected XSS

- Observable Behavior: Malicious script execution and error messages

⚠️ Security Implications

If exploited, this vulnerability could lead to unauthorized access to sensitive information or the execution of malicious scripts. An attacker might use this flaw to steal user credentials or inject malware.

Potential Impact: Confidentiality, Integrity, Availability

- **Confidentiality:** Unauthorized access to sensitive user data.
- **Integrity:** Potential modifications to critical system files.
- **Availability:** Denial of Service (DoS) attacks or resource exhaustion.

🔬 Technical Evidence

```
Endpoint:     http://testphp.vulnweb.com/search.php
Method:       POST
Parameter:    searchFor,goButton
Payload:      <script>alert('XSS')</script>
Response:     Malicious script execution and error messages
Evidence:     See technical details for further analysis.
```

🛠️ Remediation Strategy

Immediate Actions:

1. Validate the reflected XSS vulnerability through manual penetration testing.
2. Implement input validation and sanitization for all user inputs.

Implementation Guidance: Use output encoding to prevent XSS attacks.

V-003. Missing Security Headers Vulnerability

Severity: Medium

Attribute	Value
Finding ID	V-003
Severity Level	Medium
Evidence Confidence	High
Category	Configuration
Attack Vector	Network

📝 Observation Summary

The automated scan detected a potential missing security header vulnerability in the search.php endpoint. The evidence suggests that sensitive information about the application's configuration is exposed.

Technical Analysis:

- Affected Component: search.php
- Attack Type: Missing Security Header
- Observable Behavior: Exposed sensitive information and error messages

⚠️ Security Implications

If exploited, this vulnerability could lead to unauthorized access to sensitive information or the execution of malicious scripts. An attacker might use this flaw to steal user credentials or inject malware.

Potential Impact: Confidentiality, Integrity, Availability

- **Confidentiality:** Unauthorized access to sensitive user data.
- **Integrity:** Potential modifications to critical system files.
- **Availability:** Denial of Service (DoS) attacks or resource exhaustion.

🔬 Technical Evidence

```
Endpoint:    http://testphp.vulnweb.com/search.php
Method:      GET
Parameter:   N/A
Payload:     N/A
Response:    Exposed sensitive information and error messages
Evidence:    See technical details for further analysis.
```

🛠️ Remediation Strategy

Immediate Actions:

1. Implement security headers (e.g., Content-Security-Policy, X-Frame-Options) to prevent unauthorized access.

V-004. Missing Security Headers Vulnerability

Severity: Medium

Attribute	Value
Finding ID	V-004
Severity Level	Medium
Evidence Confidence	High
Category	Configuration
Attack Vector	Network

📝 Observation Summary

The automated scan detected a potential missing security header vulnerability in the search.php endpoint. The evidence suggests that sensitive information about the application's configuration is exposed.

Technical Analysis:

- Affected Component: search.php
- Attack Type: Missing Security Header
- Observable Behavior: Exposed sensitive information and error messages

⚠️ Security Implications

If exploited, this vulnerability could lead to unauthorized access to sensitive information or the execution of malicious scripts. An attacker might use this flaw to steal user credentials or inject malware.

Potential Impact: Confidentiality, Integrity, Availability

- **Confidentiality:** Unauthorized access to sensitive user data.
- **Integrity:** Potential modifications to critical system files.
- **Availability:** Denial of Service (DoS) attacks or resource exhaustion.

🔬 Technical Evidence

```
Endpoint:    http://testphp.vulnweb.com/search.php
Method:      GET
Parameter:   N/A
Payload:     N/A
Response:    Exposed sensitive information and error messages
Evidence:    See technical details for further analysis.
```

🛠️ Remediation Strategy

Immediate Actions:

1. Implement security headers (e.g., Content-Security-Policy, X-Frame-Options) to prevent unauthorized access.

V-005. Permissions Policy Vulnerability

Severity: Medium

Attribute	Value
Finding ID	V-005
Severity Level	Medium
Evidence Confidence	High
Category	Configuration
Attack Vector	Network

📝 Observation Summary

The automated scan detected a potential permissions policy vulnerability in the search.php endpoint. The evidence suggests that sensitive information about the application's configuration is exposed.

Technical Analysis:

- Affected Component: search.php
- Attack Type: Permissions Policy Vulnerability
- Observable Behavior: Exposed sensitive information and error messages

⚠️ Security Implications

If exploited, this vulnerability could lead to unauthorized access to sensitive information or the execution of malicious scripts. An attacker might use this flaw to steal user credentials or inject malware.

Potential Impact: Confidentiality, Integrity, Availability

- **Confidentiality:** Unauthorized access to sensitive user data.
- **Integrity:** Potential modifications to critical system files.
- **Availability:** Denial of Service (DoS) attacks or resource exhaustion.

🔬 Technical Evidence

```
Endpoint:    http://testphp.vulnweb.com/search.php
Method:      GET
Parameter:   N/A
Payload:
```