Penetration Testing Report

Prepared by: Samuel Adeduntan

Email: adedunsam220@gmail.com

LinkedIn: https://www.linkedin.com/in/samueladeduntan

Date: April 16, 2025

# Objective

To assess whether the web application hosted at http://testphp.vulnweb.com/ is vulnerable to SQL Injection (SQLi), and if exploitable, to extract database information to simulate a real-world attack scenario for educational and ethical demonstration purposes.

# Tools Used

- Kali Linux  
- sqlmap  
- Burp Suite  
- Web Browser (Firefox)

# Scope of Assessment

Targeted pages:  
- login.php  
- search.php (specifically artists.php?artist= parameter)

# Testing Process & Findings

## Step 1: Manual SQLi on search.php

URL Tested: http://testphp.vulnweb.com/artists.php?artist=1  
Payloads: ' OR 1=1 --, ' OR '1'='1  
Observation: Indication of SQLi vulnerability confirmed.

## Step 2: Automated SQL Injection with sqlmap

Command Run:  
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --batch --dbs  
Result: Connection resets and possible WAF interference detected.

## Step 3: Targeted Database Enumeration

Commands Run:  
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" -D acuart --tables  
Followed by:  
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" -D acuart -T users --columns  
Outcome: Enumeration interrupted by connection resets.

## Step 4: Login Page Testing

Manual test using ' OR 1=1 -- failed to bypass login. Used Burp Suite and succeeded with brute force attack.

## Step 5: SQL Injection via Burp Intercept

Request saved: login2a.txt  
sqlmap command detected possible SQLi and XSS on parameter `uname`. Enumeration was interrupted.

# Risk Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Vulnerability | Severity | Impact | Likelihood |
| SQL Injection (artists.php) | Critical | Full database compromise | High |
| SQL Injection (login.php) | High | Bypass authentication | Medium |
| Brute-forceable login | Medium | Unauthorized access | High |
| XSS (possible) | Medium | Session hijacking | Medium |

# Business Impact

- Unauthorized access to sensitive data  
- Potential full database dump  
- Website defacement  
- Regulatory non-compliance  
- Loss of trust and revenue

# Recommendations & Mitigations

SQL Injection:  
- Use Prepared Statements  
- Input validation and sanitization  
- Use ORM frameworks  
  
WAF/IPS:  
- Tune WAF rules  
  
Brute-force:  
- Account lockout or rate limiting  
- CAPTCHA after failed attempts  
  
XSS:  
- Encode output  
- Apply CSP

# Conclusion

The assessment confirms SQLi vulnerabilities that could lead to full database compromise. Defensive mechanisms exist but are insufficient. Immediate patching is required.